iOrthoPredictor: Model-guided Deep Prediction of Teeth Alignment

LINGCHEN YANG, ZEFENG SHI, YIQIAN WU, and XIANG LI, State Key Lab of CAD&CG, Zhejiang University
KUN ZHOU, Zhejiang University and ZJU-FaceUnity Joint Lab of Intelligent Graphics, China
HONGBO FU, School of Creative Media, City University of Hong Kong
YOUYI ZHENG*, State Key Lab of CAD&CG, Zhejiang University

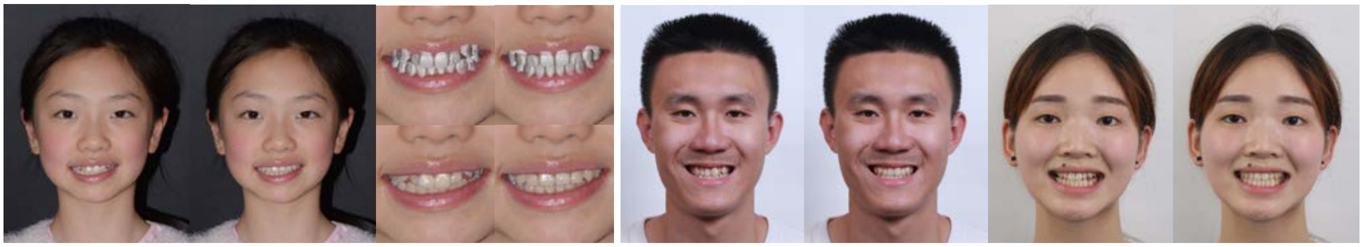

Fig. 1. Given a face photograph of a patient with malpositioned teeth and a corresponding 3D teeth model (obtained by dental scanning), our method is able to produce a face image with the teeth aligned, mimicking an orthodontic treatment effect. The input teeth model and the automatically aligned teeth for the first patient are overlaid with the mouth area shown aside. All the results are obtained fully automatically.

In this paper, we present iOrthoPredictor, a novel system to visually predict teeth alignment in photographs. Our system takes a frontal face image of a patient with visible malpositioned teeth along with a corresponding 3D teeth model as input, and generates a facial image with aligned teeth, simulating a real orthodontic treatment effect. The key enabler of our method is an effective disentanglement of an explicit representation of the teeth geometry from the in-mouth appearance, where the accuracy of teeth geometry transformation is ensured by the 3D teeth model while the in-mouth appearance is modeled as a latent variable. The disentanglement enables us to achieve fine-scale geometry control over the alignment while retaining the original teeth appearance attributes and lighting conditions. The whole pipeline consists of three deep neural networks: a U-Net architecture to explicitly extract the 2D teeth silhouette maps representing the teeth geometry in the input photo, a novel multilayer perceptron (MLP) based network to predict the aligned 3D teeth model, and an encoder-decoder based generative model to synthesize the in-mouth appearance conditional on the original teeth appearance and the aligned teeth geometry. Extensive experimental results and a user study demonstrate that iOrthoPredictor is effective in qualitatively predicting teeth alignment, and applicable to the orthodontic industry.

CCS Concepts: • **Computing methodologies → Image manipulation**; *Artificial intelligence.*

Additional Key Words and Phrases: Orthodontics, teeth alignment, image synthesis, generative networks

ACM Reference Format:
Lingchen Yang, Zefeng Shi, Yiqian Wu, Xiang Li, Kun Zhou, Hongbo Fu, and Youyi Zheng. 2020. iOrthoPredictor: Model-guided Deep Prediction of Teeth Alignment. *ACM Trans. Graph.* 39, 6, Article 216 (December 2020), 15 pages. https://doi.org/10.1145/3414685.3417771

1 INTRODUCTION

I can sing and dance. I can smile – a lot. —Chris Colfer

A warm, friendly smile is the signal of being secure and confident. It often goes beyond a gesture and can genuinely foster one's confidence in social interactions. One of the keys to a confident smile is probably a set of beautiful and regularly aligned teeth. However, according to a study conducted in the United States, over 90% of the population there could suffer from malocclusion of teeth [Graber et al. 2016]. Orthodontics thus has taken place as a specialty of dentistry to deal with malpositioned teeth and jaws, or even to modify the facial growth.

In orthodontics, visualizing the outcome of orthodontic treatment is essential to the attainment of confidence in the treatment by helping patients foresee their future teeth and facial appearance. It also facilitates the communication between orthodontists and patients. The development of commercial orthodontic imaging software such as Dolphin Imaging [Power et al. 2005] and Quick Ceph Image has paved the way for predicting surgical outcomes in photographs [Peterman et al. 2016]. However, these programs are only capable of altering facial profiles by simulating hard and soft tissue changes based on cephalometric predictions.

In this paper, we introduce a novel system named iOrthoPredictor to visually predict teeth alignment effects in images. Given a front face photo of a patient with visible malpositioned teeth, our goal is to synthesize an image of the mouth region with aligned teeth. This problem is essentially ill-posed. First, synthesizing the aligned effect requires accurate estimates of not only the geometry transformations of each individual tooth but also the transformations of gums, where estimating the transformed shape of a single

*Corresponding author.

Authors' addresses: Lingchen Yang; Zefeng Shi; Yiqian Wu; Xiang Li, State Key Lab of CAD&CG, Zhejiang University; Kun Zhou, Zhejiang University and ZJU-FaceUnity Joint Lab of Intelligent Graphics, China; Hongbo Fu, School of Creative Media, City University of Hong Kong; Youyi Zheng, State Key Lab of CAD&CG, Zhejiang University.

0730-0301/2020/12-ART216 $15.00
https://doi.org/10.1145/3414685.3417771

tooth alone in 2D image is difficult. Second, the synthesis also needs to accommodate the in-mouth appearance changes caused by the teeth movements, and is conditional on the teeth and gums' materials, and the in-mouth lighting conditions. Third, the alignment of teeth could bring in holes and unseen parts of the mouth region, which are not easy to recover from a single image.

To address the aforementioned challenges, our system exploits a novel deep learning based paradigm. The central idea is to disentangle in-mouth appearance synthesis from teeth geometry transformation. To accurately estimate the aligned teeth shape, our system acquires an additional input of a 3D teeth model of the patient. Then, as an essential ingredient of our system, we introduce 2D teeth silhouette maps to represent the teeth geometry in the photo. Such a representation enables us to accurately compute the transformed teeth shape (in 2D) using the 3D teeth model. This also bridges the domain gap since silhouette maps can be computed from both the input photograph and the 3D teeth model. For the in-mouth appearance, we model it as a latent code that can be effectively extracted from the input photo.

iOrthoPredictor operates as follows. We first introduce a deep convolutional network called *TGeoNet*, to extract the teeth geometry maps (i.e., silhouettes) and a mouth cavity mask from the facial image. The global pose of the 3D teeth model is then optimized according to the extracted silhouette maps. We then introduce a novel MLP-based neural network termed *TAligNet* to learn the target teeth arrangement after alignment, trained over massive orthodontic cases. Afterwards, the silhouette of the aligned teeth is projected back to the 2D mouth region using the optimized global pose to generate the target teeth geometry maps. In an essential step, the target teeth silhouette maps together with the original mouth image are fed into a generative neural network called *TSynNet*, to synthesize the final mouth image. Our *TSynNet* contains two encoders which respectively encode the received geometry maps into a geometry code and the original mouth image into an appearance code, and a decoder which learns to generate the final mouth image from the combined latent codes.

Our networks are trained with thousands of face images and 3D teeth models obtained from a dental company. We validate our approach via extensive experimental results and a user study. We also show the superiority of our networks by comparing with the prior art. In sum, our work makes the following contributions:

- We present the first model-guided deep learning based system to visually predict teeth alignment in photos.
- We introduce three neural networks which are seamlessly integrated in our pipeline to estimate the teeth geometry transformations and the in-mouth appearance changes;
- Our system allows orthodontists to have accurate and fine-grained geometry control over the alignment process while retaining the original attributes of a patient's teeth, and is beneficial to the orthodontic industry.

2 RELATED WORK

Digital Orthodontics. Crowded, irregular, and protruding teeth have been a problem for many individuals since antiquity [Proffit et al. 2006]. Orthodontics, as a modern science, which deals with the diagnosis, prevention, and correction of malpositioned teeth and jaws, dates back to the mid of 1800s [Kingsley 1880]. However, it was not until the mid of the 1970s when braces were introduced for orthodontic treatment [Asbell 1990]. With the recent advances in 3D printing, invisible braces have been introduced in orthodontics and the market grows dramatically due to their convenience, leading to a fresh research area of digital orthodontics. Meanwhile, a set of emerging techniques have been introduced in 3D teeth modeling [Wu et al. 2016], teeth segmentation [Cui et al. 2019; Xu et al. 2018], and teeth appearance capture [Velinov et al. 2019]. In the same line of research, our work, to the best of our knowledge, is the first approach to predict the orthodontic treatment outcome in an image.

Image Inpainting. Our work is related to image inpainting methods. Early image inpainting techniques exploit diffusion-based techniques [Ballester et al. 2001; Bertalmio et al. 2000; Levin et al. 2003] to propagate local image appearance within small holes, or use similar patches to synthesize holes [Barnes et al. 2009, 2010; Efros and Freeman 2001; Efros and Leung 1999; Kwatra et al. 2005; Simakov et al. 2008; Wexler et al. 2007]. The structure-guided approaches [Criminisi et al. 2004; Drori et al. 2003; Huang et al. 2014; Kopf et al. 2012; Pavić et al. 2006; Sun et al. 2005] integrate the guidance of high-level structure priors for inpainting, and data-driven approaches [Hays and Efros 2007; Whyte et al. 2009] use similar patches from internet images. Lately, methods based on deep learning are introduced to synthesize the missing regions by generative models [Iizuka et al. 2017; Köhler et al. 2014; Ren et al. 2015, 2019; Xie et al. 2019, 2012; Xiong et al. 2019; Yang et al. 2017; Yu et al. 2018; Zheng et al. 2019]. Unfortunately, all these methods are not applicable in our case, since our goal is to synthesize the mouth region with aligned teeth shape while retaining the appearance of a patient's original teeth where the aligned teeth shape is not directly obtainable from the original image or learnable from exemplars.

Facial Image Manipulation. Facial image manipulation has drawn intensive research interests in recent years, especially with the development of deep learning techniques. Among them, a series of research studies have focused on face reenactment [Dale et al. 2011; Thies et al. 2015], face swap [Korshunova et al. 2017], face completion [Deng et al. 2011; Mohammed et al. 2009], expression editing and synthesis [Yeh et al. 2016], or makeups [Chang et al. 2018; Gu et al. 2019; Liu et al. 2016; Wang and Fu 2016], while other works are interested in manipulating a particular facial region, for example, eyes [Ganin et al. 2016; Kuster et al. 2012], lip motions [Garrido et al. 2015], or a mouth region [Blanz et al. 2003; Kawai et al. 2013, 2014; Suwajanakorn et al. 2017]. Geng et al. [2018] introduce a face reenactment method which also synthesizes an in-mouth region using a deep inpainting method [Iizuka et al. 2017]. These methods can generate realistic facial images, including the teeth, however, the generated teeth are often synthetic and lose the characteristics of the original teeth. In addition, it is unclear how to incorporate explicit geometry control into their pipelines to predict the effect of aligned teeth, which is our focus.

There exist a series of studies which employ generative adversarial networks (GANs) [Goodfellow et al. 2014] or conditional GANs [Mirza and Osindero 2014] to synthesize facial details for facial image manipulation [Ding et al. 2018; Kim et al. 2018; Olszewski

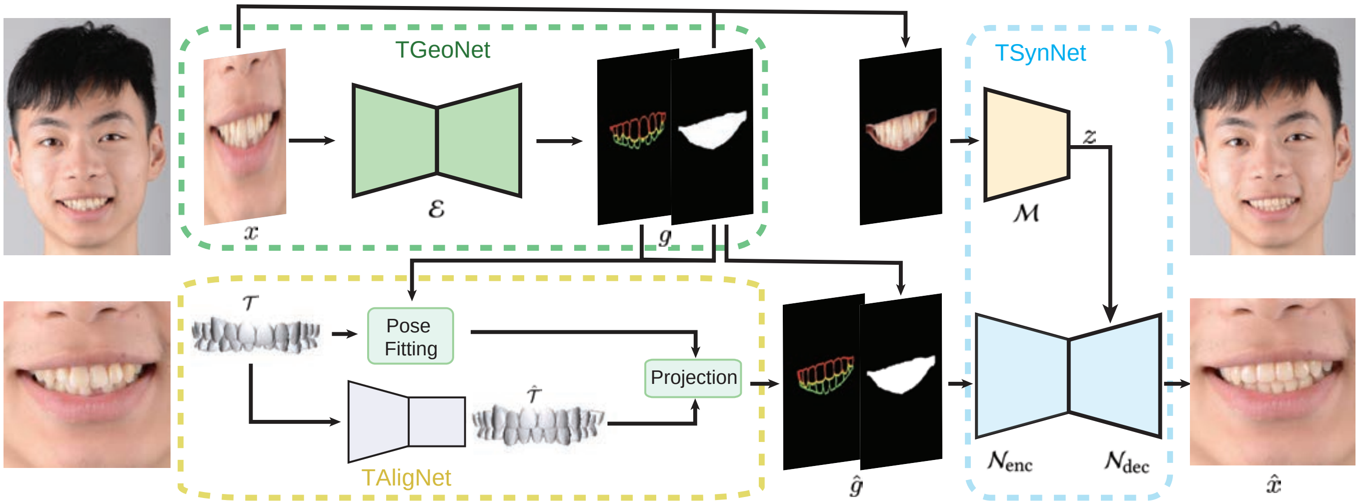

Fig. 2. The pipeline of our method. Given a frontal face photo with visible malpositioned teeth and a corresponding 3D teeth model $\mathcal{T}$, we first use a face detection algorithm [Cao et al. 2014] to extract the mouth image x, which is then fed into the proposed *TGeoNet* to obtain teeth silhouette maps and a mouth cavity mask. They are then used to optimize the global pose of $\mathcal{T}$. Our teeth alignment network *TAligNet* automatically infers an aligned 3D teeth model $\hat{\mathcal{T}}$ from $\mathcal{T}$. Then, $\hat{\mathcal{T}}$ with the optimized global pose of $\mathcal{T}$ is projected to synthesize after-orthodontics teeth silhouette maps, which are concatenated with the mouth mask extracted by *TGeoNet* to guide our *TSynNet* to generate a final after-orthodontics mouth image $\hat{x}$ based on the appearance posteriori from x. The three networks are trained separately.

et al. 2017; Qiao et al. 2018; Song et al. 2018]. Yet, these methods lack the ability to simultaneously control both the teeth geometry and appearance in output images. To overcome this, a set of works build upon conditional variational autoencoder (cVAE) [Zhu et al. 2017b] to learn disentangled representations [Esser et al. 2018; Lample et al. 2017; Qian et al. 2019; Shu et al. 2018; Watters et al. 2019]. Inspired by these works, we exploit a similar disentangled encoder-decoder architecture in the design of our *TSynNet*. In our architecture, we further exploit the decoder block from StyleGAN2 [Karras et al. 2019] and reshape it to fit into our decoder.

Deep Image-to-image Translation. Deep learning techniques have been extensively studied on the topic of image-to-image translation [Jing et al. 2019]. The influential work of Pix2Pix [Isola et al. 2017] performs image translation with conditional GANs in a supervised manner. Followed-up approaches extend this idea to unsupervised one-to-one image translation [Amodio and Krishnaswamy 2019; Liu et al. 2017; Wu et al. 2019a; Yi et al. 2017; Zhu et al. 2017a], multi-modal image translation [Alharbi et al. 2019; Choi et al. 2018; Huang et al. 2018; Wu et al. 2019b; Zhao et al. 2018], and attribute transfer [He et al. 2019; Perarnau et al. 2016; Pumarola et al. 2018; Siddiquee et al. 2019]. These methods, however, are designed for texture transfer or attribute mapping across multiple domains. Since their control of attributes is often implicitly encoded as latent codes, these methods are not directly applicable to our problem to enable the explicit fine-scale control of the teeth shape while retaining the in-mouth appearance.

Learning Spatial Transformations. One essential step in our task is to accurately predict the aligned teeth shape and their arrangement. Learning explicit geometric transformations is known to be difficult for CNNs [Dai et al. 2017; Wu et al. 2019c]. Although differentiable spatial transformation modules have been introduced in the literature for both 2D and 3D processing [Jaderberg et al. 2015; Qi et al. 2017; Zhu et al. 2019a] and applied in a wide range of vision tasks such as recognition, classification, and registration [Arar et al. 2020; Rawat and Wang 2017; Zhao et al. 2019], they are not directly applicable for computing a target teeth shape, since the shape transformation for each tooth involved in the alignment can be highly nonlinear in 2D, and subject to occlusions (e.g., occluded by the lips and the neighboring teeth). Moreover, unlike faces, shape transformations of teeth cannot be easily parameterized (e.g., via landmarks) or encoded as coefficients. Hence, we resort to a 3D teeth model to help us learn the explicit tooth transformations in 3D and then project the transformed teeth back to 2D to generate the target teeth shape.

3 METHODOLOGY

As illustrated in Fig. 2, the input to our system is a face photograph of a patient with visible misaligned teeth and a corresponding 3D teeth model of the patient, both of which are obtained before the patient receives orthodontic treatment. The 3D teeth models can be obtained by a dental scanner (e.g., 3Shape Trios-4[1]) or reconstructed using example-based methods [Wu et al. 2016]. In our context, we use the former way to collect them from the dental company. Our goal is to regenerate the mouth area of the facial image with aligned teeth while retaining the appearance of the original teeth and gums, i.e., the aligned teeth should appear to be from the same patient. Moreover, we need to enable orthodontists in-the-loop to let them

[1] https://www.3shape.com/en/scanners/trios-4

have control over the generation of alignment effect by editing the arrangement of the 3D teeth (Section 4).

3.1 Problem Formulation

We consider the in-mouth synthesis problem by modeling the intricate interplay of teeth geometry g and in-mouth appearance z. Essentially, g indicates the 2D geometry of the teeth $\mathcal{T}$ and reflects the arrangement of $\mathcal{T}$, while z characterizes the in-mouth appearance that could vary due to different surface attributes and lighting conditions. In practice, the geometry of the teeth can be explicitly represented (e.g., with a teeth boundary map [Wu et al. 2016]) while the appearance is more abstract. Thus we consider the appearance as a latent variable to be inferred from data. Assuming that we already have the function to produce g from the mouth image x, e.g., $g = \mathcal{E}(x)$, we can conversely reconstruct x based on g so as to speculate the latent variable z. To achieve this goal, we can maximize the conditional log-likelihood as follows:

$$\begin{aligned}\log p(x|g) = \log \int_z p(x,z|g)\,dz &= \log \int_z \frac{p(x,z|g)}{q(z|x,g)} q(z|x,g)dz \\ &\geq \mathbb{E}_q \log \frac{p(x|z,g)p(z|g)}{q(z|x,g)} \\ &= \mathbb{E}_q \log \frac{p(x|z,g)p(z)}{q(z|x)}, \qquad (1)\end{aligned}$$

where the second line comes with the evidence lower bound (ELBO) from Jensen's inequality while the third line is under the assumption that the in-mouth appearance z does not depend on the teeth geometry g. Inspired by the VAE literature [Zhu et al. 2017b], we model $p(z)$ as an isotropic Gaussian distribution while $p(x|z,g)$ and $q(z|x)$ are respectively estimated by a generative network $\mathcal{N}_\theta$ and an appearance encoder $\mathcal{M}_\phi$ (θ and ϕ denote their corresponding parameters). The loss function derived from Eqn. (1) has the following form:

$$\mathcal{L}(x,\theta,\phi) = \mathbb{E}_{q(z|x)}(-\log p(x|z,g)) + D_{kl}(q(z|x)||p(z)), \qquad (2)$$

where the first term can be reduced to the distance between $\mathcal{N}(g,z)$ and x [Kingma and Welling 2013], and D_{kl} refers to the KL divergence between two distributions.

As we can see, as long as we obtain a set of face images and a way to extract g from x, we can optimize the above loss function. After training, we wish z and g together capture all variations of interest and are fully disentangled so that we can synthesize the realistic target mouth area $\hat{x} = \mathcal{N}(\hat{g}, z)$ based on the appearance $z = \mathcal{M}(x)$ from the original image and the target teeth geometry $\hat{g}$, leaving the remaining difficulty to the accurate estimation of the target teeth geometry $\hat{g}$, for which we resort to the 3D teeth model. The pipeline shown in Fig. 2 illustrates the entire formulation step by step. The next sections will describe our method in detail.

3.2 Conditional Geometry Generation

In this subsection, we describe the 2D geometry maps representing g and their generation by the proposed *TGeoNet* (i.e., $\mathcal{E}$).

2D Geometry Maps. Since we want to bake the spatial details endowed by g into the output image, we represent g in the form of an image with the same resolution as x. We use 2D teeth silhouettes g_y including an upper teeth silhouette map g_u and a lower teeth silhouette map g_l, and a mouth cavity mask g_m to represent g (Fig. 3).

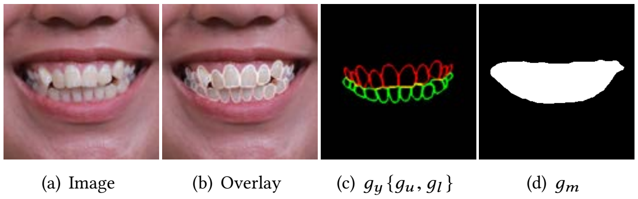
(a) Image (b) Overlay (c) $g_y\{g_u, g_l\}$ (d) g_m

Fig. 3. To train *TGeoNet*, we manually label a set of training images. For each image (a), we label the mouth cavity mask (d) and teeth silhouettes (c). We further distinguish the upper teeth silhouette map g_u (red) from the lower teeth silhouette map g_l (green).

The advantages of using teeth silhouettes are three-fold. First, since we derive the transformed teeth geometry from the aligned 3D teeth model during testing, the silhouette maps can bridge the gap between geometry maps extracted from the real image data and synthesized ones from 3D teeth models (as in training we only use real images). Second, the silhouette maps reflect the arrangement and shapes of 3D teeth after projection in a fine-scale way. Finally, a dataset of silhouette maps are much easier to acquire than other geometry representations, such as a normal map, since no 3D models and 3D fitting are involved during the labeling procedure, and all we need is to extract the visible 2D teeth boundaries.

We include the mouth cavity mask g_m (inner mouth region) to guide the network $\mathcal{N}$ to synthesize the full oral cavity. To our best knowledge, there exists no well-designed algorithm to robustly extract g_y or predict g_m. Although one possibility to compute g_y is to extract it from the teeth model $\mathcal{T}$, this requires an accurate fitting of $\mathcal{T}$ to x, which is non-trivial given only the mouth image with unknown camera pose (see Fig. 11). In addition, g_m cannot be extracted from $\mathcal{T}$. Thus, we propose a neural network *TGeoNet* to extract g_y and g_m. On the other hand, the estimated g_y and g_m can be used as geometric cues to help the fitting of $\mathcal{T}$ (Sec. 3.4).

TGeoNet. The input to *TGeoNet* is the mouth image x and the output $\mathcal{E}(x)$ are three binary maps $\{\bar{g}_u, \bar{g}_l, \bar{g}_m\}$, where the upper bar is used to differentiate them from the ground truth. We use the U-Net architecture [Ronneberger et al. 2015] consisting of an encoder and a decoder with skip connections between them (Fig. 4).

The loss function to optimize *TGeoNet* is derived from Multinomial Logistic Regression in view of each pixel possibly being classified into more than two labels. It has the following form:

$$\mathcal{L}_m = \mathbb{E}[\sum_k g_k \odot \log(\bar{g}_k) + (1-g_k) \odot \log(1-\bar{g}_k)] + \lambda_m \mathbb{E}[\omega^2], \quad (3)$$

where $\odot$ denotes element-wise multiplication, λ_m the balancing term for the L_2 regularization term, ω the parameters of $\mathcal{E}$, $g_k \in \{g_u, g_l, g_m\}$ and $\bar{g}_k \in \{\bar{g}_u, \bar{g}_l, \bar{g}_m\}$.

Training Data. We first collect 3,000 front face photos from orthodontic cases which encompass diverse teeth geometry and appearance. Such photos were taken before patients received the treatment with their consensus. Each photo is cropped to its mouth region identified by the method of Cao et al. [2014] and resized to

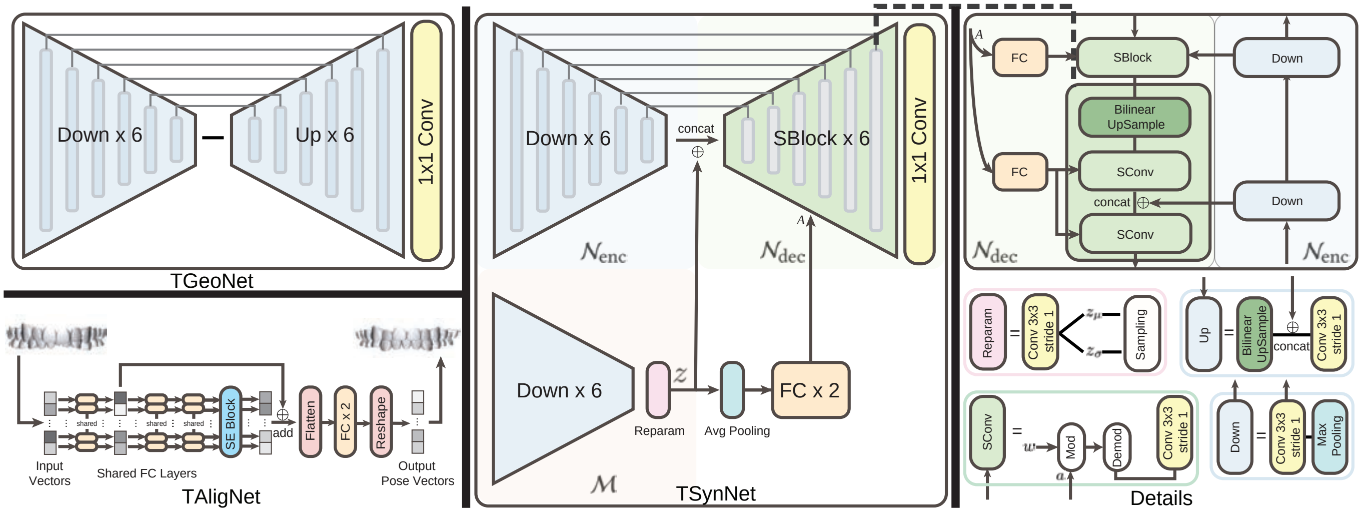

Fig. 4. The architectures of our *TGeoNet*, *TAligNet*, and *TSynNet*. More details can be found in Section 3.5.

the resolution of 256 × 256. Each mouth image x is then sampled from this dataset. We manually label the mouth cavity mask g_m for each x, and also the corresponding contour of every visible tooth in it. We further manually classify the semantic position of each contour as belonging either to the upper or the lower teeth. Fig. 3 shows an example of g and x. We in total label 1,100 face images instead of the entire set of 3,000 images, since this subset is sufficient for this task. These labeled images are randomly split into a training set of size 1,000 and a testing set of size 100.

3.3 TSynNet

In this subsection, we describe *TSynNet*, which combines the aforementioned appearance encoder $\mathcal{M}$ and the generative network $\mathcal{N}$. The generative network $\mathcal{N}$ consists of a geometry encoder $\mathcal{N}_{\text{enc}}$ which extracts a geometry code from the input geometry maps, and a decoder $\mathcal{N}_{\text{dec}}$ that combines the coming geometry code from $\mathcal{N}_{\text{enc}}$ and the appearance code from $\mathcal{M}$ to synthesize a realistic mouth area. *TSynNet* only synthesizes the mouth cavity and the other parts are directly copied from the original image to produce an entire face image with the aligned teeth.

Architecture. Our goal is to enforce the spatial details of the teeth geometry so that it can be reflected in the output image. In light of this, we let $\mathcal{N}_{\text{enc}}$ hierarchically encode the geometry information and inject it into $\mathcal{N}_{\text{dec}}$ through skip connections. For the in-mouth appearance z, we represent it as a compact bottleneck representation, encoded from the mouth image by $\mathcal{M}$. To enforce the disentanglement of teeth geometry and appearance, a straightforward way to embed z into $\mathcal{N}_{\text{dec}}$ is through the concatenation of $\mathcal{N}_{\text{enc}}(g)$ and z. However, such concatenation can only manipulate high-level information [Zhu et al. 2019b]. Inspired by the style transfer literature [Huang and Belongie 2017; Yang et al. 2018], we treat the in-mouth appearance as a style code, which is injected into each decoding block so as to control the low-level feature maps without being entangled with geometry. We utilize the state-of-the-art weight modulation and demodulation proposed by [Karras et al. 2019] to achieve this goal.

Specifically, for a convolution layer with its original kernels w and input activations (assuming unit standard deviation), the modulation operation effectively scales each input feature map based on the incoming style a, dynamically generated by an MLP from z:

$$w'_{i,j,k} = a_i \cdot w_{i,j,k}, \tag{4}$$

where w' represents a set of modulated kernels, and i, j and k enumerate the input feature maps, output feature maps, and spatial sizes of the convolution, respectively. Then, in order to restore the output feature maps back to the unit standard deviation, the weight demodulation is applied to w':

$$w''_{i,j,k} = w'_{i,j,k} / \sqrt{\sum_{i,k} {w'_{i,j,k}}^2 + \epsilon}. \tag{5}$$

We implement this mechanism in each decoding block in $\mathcal{N}_{\text{dec}}$ to hierarchically control the decoding process. Fig. 4 show the details of the architecture. Different from StyleGAN2 [Karras et al. 2019], which is an unconditional generator whose input is a random noise vector, our network is conditional on two orthogonal attributes of teeth.

Loss. We include three loss functions. The first is the reconstruction loss, which penalizes the perceptual difference between the original image x and the output $\bar{x} = \mathcal{N}(\mathcal{E}(x), \mathcal{M}(x))$:

$$\mathcal{L}_{\text{rec}} = \sum_k \lambda_k ||\Phi_k(x) - \Phi_k(\bar{x})||, \tag{6}$$

where Φ is a pretrained VGG19 network [Simonyan and Zisserman 2014], Φ_k indicates the feature maps of its kth layer, and λ_k is a hyperparameter for each used layer. The second loss $\mathcal{L}_{kl}$ measures the KL divergence between $\mathcal{M}(x)$ and $N(0, I)$ [Kingma and Welling 2013] and the third is an adversarial loss $\mathcal{L}_{adv}$, which makes generated results appear indistinguishable from real samples. We adopt WGAN-GP [Gulrajani et al. 2017] for stable training.

Our overall objective function is as follows:

$$\mathcal{L}_{app} = \mathcal{L}_{adv} + \mathcal{L}_{\text{rec}} + \lambda_{kl}\mathcal{L}_{kl}, \tag{7}$$

where λ_{kl} is a hyperparameter controlling the weight of $\mathcal{L}_{kl}$.

Training Data. Only a massive collection of face images with visible teeth is needed for training, which is already obtained in the previous section. We randomly split it into a training set of 2800 images and a testing set of 200 images. Note that for training *TSynNet*, we use the output of *TGeoNet*, i.e., $\mathcal{E}(x)$, instead of the projected silhouette maps from the aligned teeth models (the latter is only used in the test stage).

3.4 Aligned Teeth Silhouette Maps Generation

To synthesize the final mouth image $\hat{x}$ with the aligned teeth, we need the target teeth silhouette maps $\{\hat{g}_u, \hat{g}_l\}$. To this end, we first optimize the global pose of the 3D teeth model $\mathcal{T}$ to match the extracted silhouette maps $\{\bar{g}_u, \bar{g}_l\}$ and then introduce *TAligNet* to automatically compute the aligned pose of each individual tooth, i.e., their arrangement after alignment. Afterwards, the aligned teeth model $\hat{\mathcal{T}}$ is projected back to the mouth area to generate the transformed teeth silhouette maps $\{\hat{g}_u, \hat{g}_l\}$, which will be concatenated with the mouth cavity map $\bar{g}_m$ to guide the generative network $\mathcal{N}$ to produce $\hat{x}$. This process consists of two main steps: global teeth pose fitting and 3D teeth alignment.

Global Teeth Pose Fitting. The input 3D teeth model $\mathcal{T}$ consists of individual pre-segmented 3D tooth models, which can be divided into two rows of teeth, namely, the upper teeth $\mathcal{T}_u$ and the lower teeth $\mathcal{T}_l$. We use $\{\bar{g}_u, \bar{g}_l, \bar{g}_m\}$ and $\mathcal{T}$ to fit two transformation matrices for the upper and lower teeth rows separately. We adopt a similar Expectation Maximization (EM) based method and modify it to fit more tightly into our context to estimate the transformation matrices. As this is a well-studied and more or less solved problem, we put our details in Appendix A. In most cases our method can automatically get desired fitting results. We also develop a simple interface allowing users to interactively fix the fitting errors (e.g., the examples shown in the first row of Fig. 6 and the second person in the third row of Fig. 9).

3D Teeth Alignment. This step is performed by *TAligNet* which takes a 3D teeth model as input and outputs an aligned pose of each tooth. The pose of a tooth is represented as a 7 dimensional vector $\mathbf{v} = (\mathbf{v}^p, \mathbf{v}^q)$, where $\mathbf{v}^p$ denotes its 3D position while $\mathbf{v}^q$ its orientation represented as a 4D quaternion. *TAligNet* jointly regresses the target global pose vector of each tooth, based on a large set of paired unaligned-aligned 3D teeth models.

As the teeth alignment is not geometry-agnostic (e.g., gaps between adjacent teeth, occlusion relations are all related to teeth geometry), we use the PointNet autoencoder [Achlioptas et al. 2018] to independently encode the geometry of each tooth. Specifically, the autoencoder consists of a PointNet [Qi et al. 2017] as an encoder and a simple MLP as a decoder. The input to the encoder is the uniformly sampled $n_x = 1024$ points $\mathbf{X} = \{\mathbf{x}_i|_{i=1}^{n_x}\}$ on a tooth while the output is an $|\mathbf{c}| = 100$ dimensional vector $\mathbf{c}$ representing the tooth geometry code, based on which the decoder reconstructs the point set $\hat{\mathbf{X}}$. The autoencoder is trained by minimizing the Chamfer distance:

$$\sum_{\mathbf{x}\in\mathbf{X}} \min_{\hat{\mathbf{x}}\in\hat{\mathbf{X}}} ||\mathbf{x} - \hat{\mathbf{x}}||_2 + \sum_{\hat{\mathbf{x}}\in\hat{\mathbf{X}}} \min_{\mathbf{x}\in\mathbf{X}} ||\mathbf{x} - \hat{\mathbf{x}}||_2. \tag{8}$$

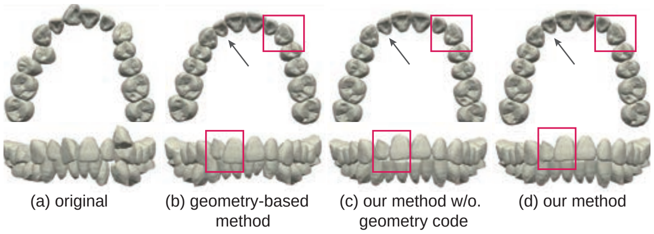

Fig. 5. Visual comparisons of our teeth alignment method with alternatives. Our method with geometry encoding achieves the best result.

Since the extracted geometry code $\mathbf{c}$ only represents the local tooth shape, we also include its initial global pose vector $\bar{\mathbf{v}}$. With the concatenated $(|\mathbf{c}| + 7)$ dimensional feature vector $(\mathbf{c}, \bar{\mathbf{v}})$ for each tooth, we construct a 1D image of teeth with the shape of $N\times(|\mathbf{c}|+7)$, N being the total number of teeth of a patient (usually 28). This 1D image is then fed into the MLP-based decoder to predict the aligned target poses with the shape of $N \times 7$. The structure of this MLP is shown in Fig. 4. The goal is to minimize the following loss:

$$\sum_{i}^{N} -(\mathbf{v}_i^q \cdot \hat{\mathbf{v}}_i^q)^2 + w_r||\mathbf{v}_i^p - \hat{\mathbf{v}}_i^p||_2, \tag{9}$$

where i denotes the i-th tooth, $\hat{\mathbf{v}}^q$ and $\hat{\mathbf{v}}^p$ are the respective predicted rotation and position while $\mathbf{v}^q$ and $\mathbf{v}^p$ the ground truth, and w_r is a hyperparameter balancing the importance of the rotation and position errors.

We collect 8,995 pairs of unaligned-aligned 3D teeth models from orthodontic cases, with a training-test split of 8000 : 995. Fig. 5 shows the aligned teeth generated using our algorithm. We find that this step is sufficient for generating the target teeth silhouette maps. Note that from *TAligNet* we can compute the local transformation matrix of each individual tooth, which can then be applied to our teeth model under the optimized global pose to obtain the aligned teeth.

3.5 Implementation Details

Network Details. We describe the structures of our aforementioned networks in Fig. 4. The details of the used main blocks (“Up”, “Down”, and “SBlock”) are illustrated in the right column. Concretely, *TGeoNet* is composed of a contracting part (6 "Down" blocks) and an expanding part (6 "Up" blocks) with (32, 64, 128, 256, 256, 256) and (256, 256, 256, 128, 64, 32) feature channels, respectively. The final convolution layer is added to obtain the 3 binary maps. The resolutions of input and output images are 256 × 256.

For *TSynNet*, the geometry encoder $\mathcal{N}_{\text{enc}}$ consists of 6 consecutive “Down” blocks with feature channels of (8, 16, 32, 32, 32, 32) to progressively encode geometry information. The appearance encoder $\mathcal{M}$ uses 6 consecutive “Down” blocks followed by a reparametrization layer [Kingma and Welling 2013] to obtain the appearance code z. All these “Down” blocks have feature channels of (32, 64, 128, 128, 128, 128). The size of appearance latent code z

and the feature channels of 2 subsequent fully connected (FC) layers are all 128. The decoder $\mathcal{N}_{\text{dec}}$ utilizes 6 consecutive "SBlocks" with the respective feature channels of (128, 128, 128, 128, 64, 32). The final convolution layer is added to obtain the output image. The resolutions of input and output images are 256 × 256.

In *TAligNet*, several shared FC layers, a squeeze-and-excitation (SE) block [Hu et al. 2018] with reduction ratio 4, and skip connection are sequentially applied, followed by the flatten layer and the 2 FC layers. The feature channels of 3 shared FC layers and 2 subsequent FC layers are respectively (100, 100, 100, 1000, 196).

Training Details. For all the networks, we use Adam solver [Kingma and Ba 2015] as the optimizer. As *TGeoNet* is a classification network while *TSynNet* is generative, we train them separately with different strategies. For *TGeoNet*, we use $\lambda_m = 0.00001$. The network is trained from scratch with a batch size of 8 and an initial learning rate of 0.0002, which is linearly decayed to 0 during the whole training procedure (50K iterations). To make *TGeoNet* more robust against input variations, we augment our image dataset by applying some random image manipulations, including Gaussian blur, Gaussian noise, rotation, scaling, and mirroring.

For *TSynNet*, we further reshape it into a skip generator as in [Karras et al. 2019] to progressively learn the details. We use the image domain ($\lambda_0 = 1.0$) as well as VGG19 layers $\{relu1\_2, relu2\_2, relu3\_4\}$ (with the same balancing weight $\lambda_k = 0.001$) to compute $\mathcal{L}_{\text{rec}}$. λ_{kl} is set to 0.5. *TSynNet* is trained from scratch with a batch size of 8 and a learning rate of 0.001 (250K iterations). We further augment our image dataset by applying some random image manipulations, including rotation and mirroring.

For teeth alignment, the PointNet autoencoder is trained from scratch with a batch size of 50 and a learning rate of 0.001 (4.5K iterations). We further augment the data by applying some random noise on the point clouds. For *TAligNet*, we use $w_t = 0.01$. The network is trained from scratch with a batch size of 100 and a learning rate of 0.0001 (40K iterations).

4 RESULTS

We show the results of our full approach on a variety of orthodontic cases with teeth in diverse shapes, appearance, and arrangements. All the results presented in this paper, except explicitly indicated, are generated without any user intervention.

Runtime Performance. As mentioned earlier, iOrthoPredictor consists of several steps. The detection of silhouette maps and mouth cavity mask by *TGeoNet* takes 6 ms per image. The synthesis of the after-orthodontics mouth image takes 6 ms per image. The prediction of 3D teeth alignment takes around 0.03 ms for a teeth model. Network training takes 6 hours for *TGeoNet*, 36 hours for *TSynNet*, and 5 minutes for *TAligNet* (not including the part of geometry encoding). 3D teeth pose fitting takes about 2 seconds for 30 optimization iterations. All the other steps of our pipeline incur a negligible time penalty. All the tests are conducted on a PC with an i7-8700 3.2GHz CPU, 16 GB main memory, and a GeForce 2080Ti GPU (11 GB memory).

Alignment Effect Prediction. Our main application is the visual effect prediction of orthodontic treatment in a full-face photo. By disentangling the geometry and appearance representations, *TSynNet* can faithfully synthesize a high-quality after-orthodontics mouth image based on the appearance from a before-orthodontics mouth image and the geometry from the predicted after-orthodontics teeth. Also, *TSynNet* is agnostic to any input identity and thus can be used to generate after-orthodontics faces for any patients. Fig. 6 shows several representative results. Thanks to the pose fitting, the synthesized after-orthodontics 2D silhouette maps retain the coarse position and relative size for each original tooth. Note that *TSynNet* learns to contain not only the teeth attributes but also the global lighting conditions, such that the synthesized teeth regions, even hidden in the original images, are consistent in the output images, e.g., the left case in the first row of Fig. 9. For extremely irregular teeth, our method is still able to generate high-quality regular teeth, e.g., the examples shown in Fig. 6. Our method is also able to handle missing teeth, holes, and occlusions well (see the examples in the second and third rows of Fig. 9, where some teeth are missing or occluded by lips). In Fig. 9, we also show the generated images with the aligned 3D teeth model edited by an orthodontist (starting from the unaligned teeth). It can be observed that our generated results resemble closely to that of the orthodontists' results, indicating the efficacy of *TAligNet*.

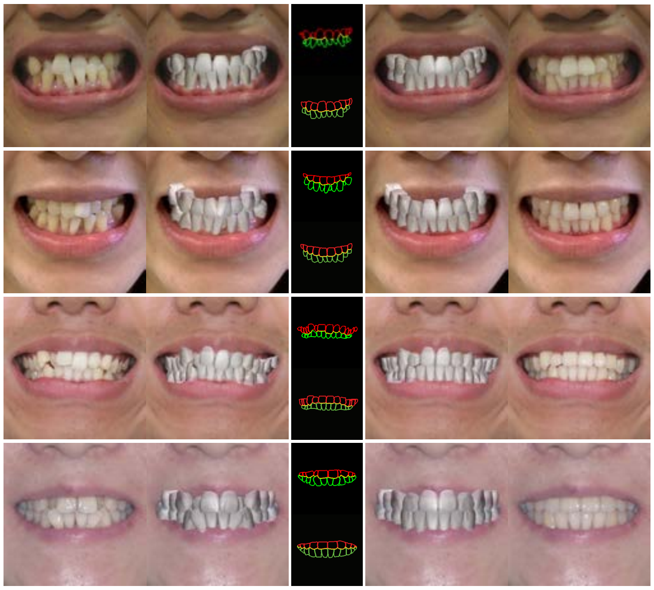

Fig. 6. Diverse results generated by our method. For each row from left to right: the original mouth image, the fitting result, the detected silhouette maps (top) and the silhouette maps for the projection of the aligned teeth (bottom), the aligned 3D teeth, and the synthesized after-orthodontics mouth image.

User Edits. Due to the explicit use of geometry maps as input to *TSynNet*, our system naturally supports user edits to generate the alignment effect. They are reflected in two aspects. First, we allow orthodontists to edit the aligned teeth in 3D (e.g., by using software such as ClinCheck[2]), such that they can control the alignment effect

[2] https://www.invisalign.com/the-invisalign-difference/clincheck-software

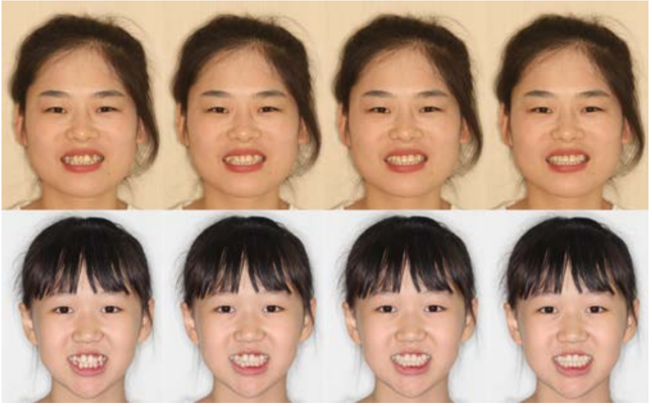

Fig. 7. Our method is able to progressively synthesize the alignment effect at different orthodontic stages. The corresponding 3D teeth models at different stages are manually edited by an orthodontist.

for medical accuracy. For example, in the third row of Fig. 9 the patient on the left has missing teeth. Orthodontic planning for missing teeth is rather complicated and subject to the specific conditions of patients. The missing part can be either filled by neighboring teeth during the planning or left untouched for dental implant. Such prior knowledge is not considered in the design of *TAligNet*, resulting in imperfect alignment: our prediction leaves the space between neighboring teeth but the space is not medically correct. In such cases, orthodontists can further edit the 3D aligned teeth to achieve more correct results if desired. Alternatively, orthodontist can also directly edit the arrangement of teeth from scratch without using *TAligNet*. Two examples are shown in Fig. 7, where the orthodontist edits the 3D teeth at various orthodontic stages to enable a visualization of how the teeth are progressively aligned. Thanks to our appearance conditioning, the generated results are consistent in shades of color. Despite this editable feature, all the results shown in this paper, except those in Fig. 7 and Fig. 8, are obtained without user edits.

Second, our system also supports edits from novice users in 2D. Fig. 8 shows two such examples, where the target teeth are generated by user editing over existing silhouette maps derived from a manually fitted 3D teeth model. The edit operations include redrawing of the teeth silhouettes or erasing of parts. Interestingly, with our appearance conditioning, here the teeth appearance can be extracted from either the original image or a reference image, enabling a flexible control over the synthesized results. Note also that our algorithm is agnostic to facial appearance and thus applicable to people with different skin colors. More editing results are presented in the supplementary document.

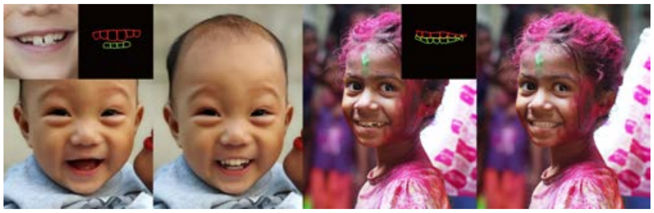

Fig. 8. Flexible control of synthesis results by novice users. The user is able to synthesize new teeth by editing on existing silhouette maps (projected from a template 3D teeth model). The conditional teeth appearance can be taken from either the original image (in the second example) or a reference one (in the first example, where the reference teeth image is shown at the top-left corner). Original image courtesy of Arunachal Art and Himanshu Singh Gurjar respectively.

5 EVALUATION

We conduct qualitative and quantitative experiments to evaluate the impact of various algorithmic components of our approach.

5.1 Evaluation of *TGeoNet*

Detection. To evaluate the accuracy and amount of training data required for *TGeoNet*, we iteratively increase the size of used training data, ranging from 10% to 100% of the full training set, and calculate the F1-Score[3] on the full test set. Fig. 10 shows that more training data generally leads to higher accuracy. Nevertheless, there is no significant improvement from over 40% of the training set for mouth cavity mask and 80% for silhouette maps. The best results are obtained with the full dataset. The accuracy for g_l is relatively low due to the fact that the lower teeth often suffer more from occlusion than the upper teeth in the input images.

Necessity. We assess the necessity of *TGeoNet* for generating the silhouette maps and the mouth cavity map. Specifically, we use the facial landmark detection method of Cao et al. [2014] to extract the mouth cavity mask and use the edge detection method [Xie and Tu 2015] to extract an edge map from a mouth region. These two maps are then used as geometry maps to train our *TSynNet*. Since here edge maps are used for training *TSynNet*, an edge map instead of a silhouette map is required at the time of testing. To do so, we first use the detected edge map to fit the global pose of the teeth, and then render the aligned 3D teeth model (using basic OpenGL rendering) to obtain a shaded image, which is fed into the method of Xie and Tu [2015] to obtain an aligned edge map.

Comparative results are shown in Fig. 11. Three disadvantages of the alternative maps are revealed. First, the detected edge map (lower corner of Fig. 11 (a)) is unreliable, and thus the pose fitting step would easily fail, resulting in weird results (Fig. 11 (b)). Second, the coarse and inaccurate mouth cavity map generated from the landmarks might not erase the original irregular teeth area entirely or preserve lips well, resulting in some artifacts (Fig. 11 (c)). In contrast, using our mouth cavity mask does not generate these artifacts (Fig. 11 (d)). Yet, there is a big domain gap between the edge map extracted from the rendered image and the one from the real data (see in Fig. 11 (a) and (d)). Compared with the edge map, our silhouette maps better bridge the input gap and thus enable us to obtain more realistic results, where the teeth shapes match the original teeth better (Fig. 11 (e)).

Silhouette Maps. We use the combination of an upper silhouette map and a lower silhouette map instead of a single silhouette map for two advantages. First, these two maps can benefit pose fitting

[3] https://en.wikipedia.org/wiki/F1_score

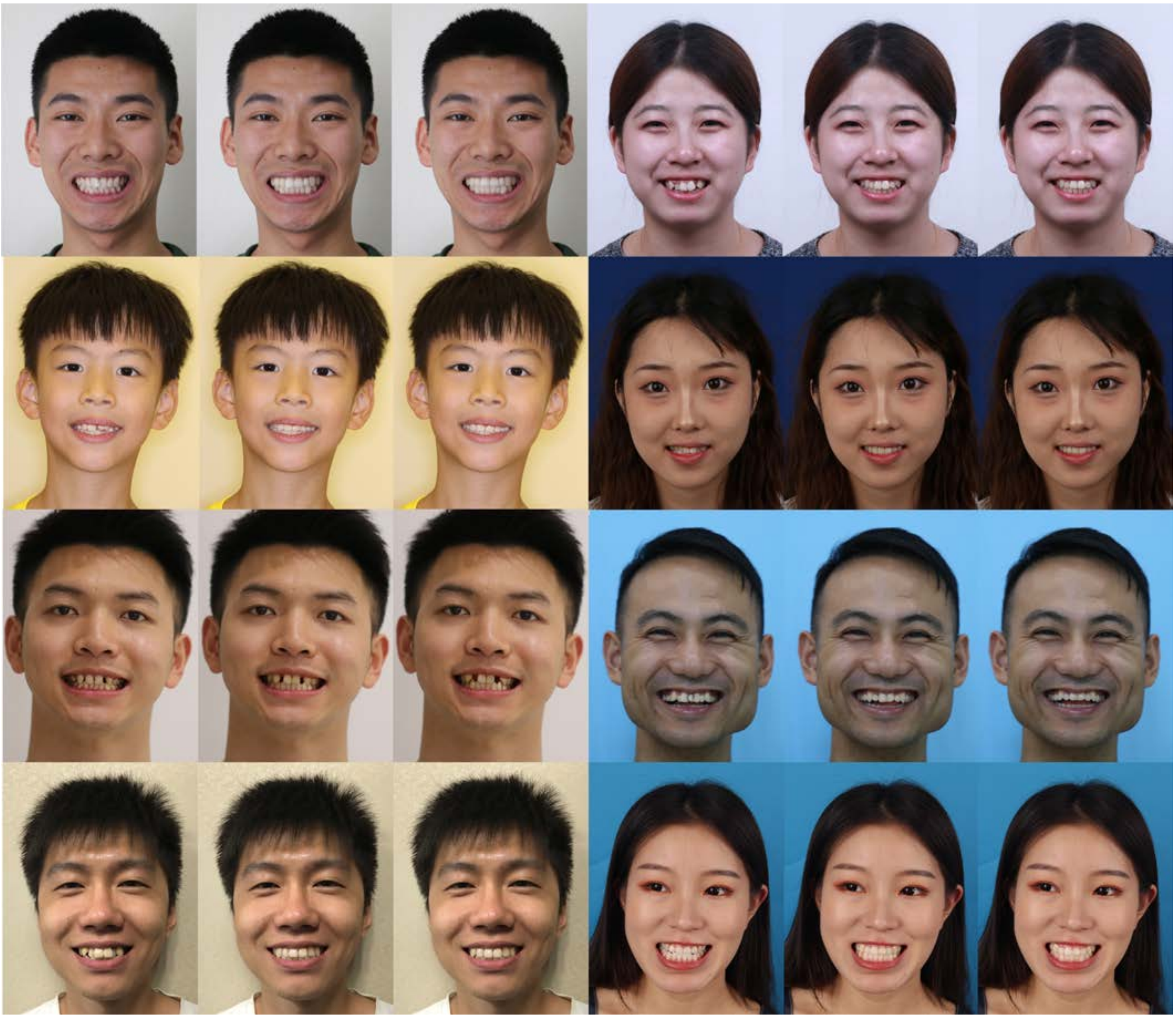

Fig. 9. Diverse results generated by our method. For each patient, from left to right are the original image, our automatically generated result with the 3D teeth model aligned by *TAligNet*, and the generated result with the 3D teeth model edited (from the unaligned teeth model) by an orthodontist.

(see Appendix A) as the correspondence search space for each teeth row is confined to either the upper or the lower region. Otherwise, finding the correspondence for one teeth row would be easily spoiled by the silhouette region of the other. The second advantage is demonstrated in the Fig. 12. For a single silhouette map (Fig. 12 (a)), the contour formed from the occluding part of the upper and lower teeth might cause ambiguity since one cannot differentiate whether it indicates a tooth or a hole (highlighted in the gray box of Fig. 12 (a)). This issue is resolved by using two maps separately for the upper and the lower teeth (see Fig. 12 (b) and (c)). Note in (b) the tooth silhouette in the lower map is not closed, thus indicating a hole. Furthermore, *TSynNet* has learned to generate holes and teeth based on different conditions (see Fig. 12 (b) and (c)).

5.2 Evaluation of *TSynNet*

Latent Space for Appearance. There is a trade-off between appearance retention and representation disentanglement, which hinges on the resolution of the latent space z. On one hand, if the resolution is set to be too low, e.g., 1×1, the texture mapping network $\mathcal{M}$ is liable to obliterate appearance details, thus compromising the synthesized results. On the other hand, if the resolution is too high, e.g., 16×16, the disentanglement of geometry and appearance would fail. This is because a higher resolution would incur the preservation of

Table 1. Quantitative comparisons of different resolutions for the appearance latent space on the test set. We use the reconstruction loss $\mathcal{L}_{\text{rec}}$ in Eqn. (6) for comparison.

Resolution	relu1_2	relu2_2	relu3_4	total ($\mathcal{L}_{\text{rec}}$)
1 × 1	0.0481	0.0710	0.0370	0.1561
4 × 4	0.0436	0.0661	0.0337	0.1434
16 × 16	0.0347	0.0516	0.0243	0.1106

more layout information. Thus, during training the whole network would find it easier to reconstruct the image x through the path $\mathcal{N}_{\text{dec}}(\mathcal{M}(x))$ and ignore the geometry information from $\mathcal{N}_{\text{enc}}(e)$.

Therefore, we examine synthesis results with different latent space resolutions ranging from 1 × 1 to 16 × 16. For each resolution, we change only the number of down-sampling modules used in *TSynNet* and adjust the number of up-sampling modules correspondingly. Table 1 shows that the reconstruction loss ($\mathcal{L}_{\text{rec}}$) decreases as the resolution increases. However, the synthesized after-orthodontics images in Fig. 13 substantiate that the network with a high-resolution latent space would collapse the disentanglement of geometry and appearance, and consequently it just learns to copy and paste the appearance from the input to the output. Therefore, we choose 4 × 4 as the resolution of our latent space for its good synthesized quality and generalization.

Note that we do not model our latent space as a mixture of Gaussians as done in the method of Qian et al. [2019] but with a single Gaussian. This is due to the lack of multimodal labeled data. On the other hand, it is also difficult to split the teeth appearance into different attributes. Besides, we do not use a similar mechanism used in FaderNet [Lample et al. 2017] to automatically find the latent space for appearance. The key enabler of FaderNet is its explicit representation of the attributes as an abstract vector, and it achieves the disentanglement of the attributes from the salient information via adversarial learning in the latent space. In our case, the teeth geometry is represented as an image containing the fine-grained teeth geometry, which cannot be easily represented as such an abstract vector, making it unclear how to disentangle the geometry from the appearance code using adversarial learning.

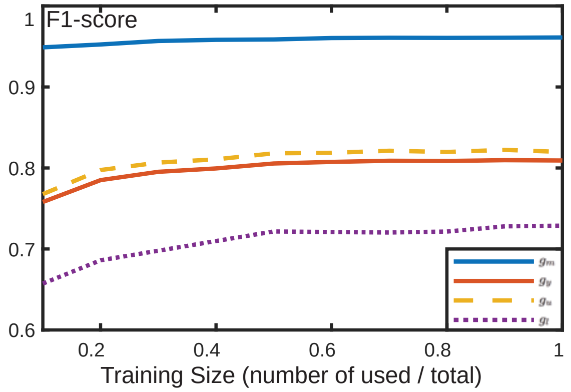

Fig. 10. Accuracy curves for mouth cavity mask g_m, upper silhouette map g_u, lower silhouette map g_l, and the combination g_y of g_u and g_l.

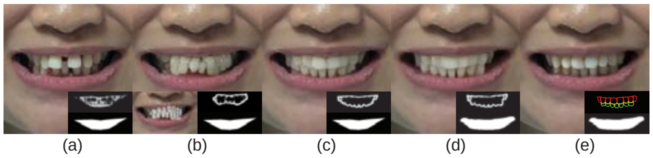

Fig. 11. Comparisons of our geometry maps with alternative geometry maps. (a) input image, edge map of Xie and Tu [2015] (denoted as e) and cavity map of Cao et al. [2014] (denoted as $\bar{m}$), (b) $e + \bar{m}$ + global teeth pose fitted with (e, $\bar{m}$), (c) $e + \bar{m}$ + global teeth pose τ fitted with our (g_y, g_m), (d) $e + g_m + \tau$, (e) ours. The input maps for synthesizing each result in (b)-(e) are shown at the corresponding lower corner.

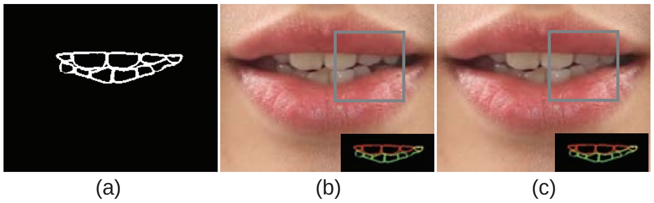

Fig. 12. For a single silhouette map (a), there is ambiguity in the gray box region (hole or tooth). Our semantic silhouette maps can help resolve the ambiguity (corner of images (b) and (c): green for lower teeth, red for upper teeth, and yellow for both). Furthermore, our *TSynNet* has learned to generate holes and teeth based on different conditions (images (b) and (c)).

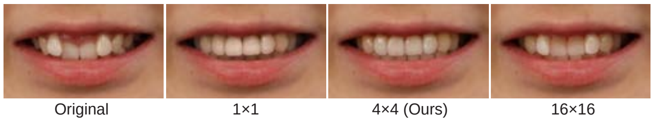

Fig. 13. Comparisons of different network architectures in terms of latent space resolution. We use after-orthodontics results for comparisons.

Table 2. Frechet inception distance (FID) [Heusel et al. 2017] for different generators.

Method	Baseline-NANS	Baseline-NS	Ours
FID	12.95	6.34	4.77

Ablation Study. We evaluate the significance of the adversarial loss $\mathcal{L}_{adv}$ and the style modulation technique (weight modulation and demodulation) adapted from [Karras et al. 2019]. To this end, we compare our method with two simplified alternatives on the testing set. The first simplification ("Baseline-NANS") removes both $\mathcal{L}_{adv}$ and style modulation while the second one ("Baseline-NS") removes only style modulation. As shown in Fig. 14, "Baseline-NANS" suffers from over-smoothing while "Baseline-NS" generates more details due to the adversarial mechanism between the generator and the discriminator. By virtue of style modulation, which hierarchically controls the synthesis process, ours could produce more vivid and realistic teeth appearance (see more natural teeth highlights) compared to "Baseline-NS". The quantitative comparisons shown in Table 2 also corroborate this.

5.3 Evaluation of *TAligNet*

We qualitatively and quantitatively evaluate *TAligNet*. First, we examine the geometry code used in *TAligNet*. Fig. 5 and Table 3 show

Table 3. Average angular error and average translation error, evaluated on the test set of 995 teeth models.

	Angular Error (degrees)	**Translation Error** (mm)
Geometric Method	10.5	2.03
w/o. Geometry Code	6.46	1.02
w. Geometry Code	5.64	0.97

that the performance of *TAligNet* degenerates (cf. the angular error and translation error) if the geometry code is not considered in the regression. This proves the concept that the teeth alignment is not geometry-agnostic. We also show a comparison with a traditional geometry-based teeth alignment method [Li et al. 2019]. Since their method fits teeth arches as guidance for global alignment, their results are sensitive to the original poses of the teeth. Second, since our teeth pose vectors are wrapped into a 1D image of $N \times (n_g + 7)$, we can also alternatively consider it as an image of $2 \times N/2$ with $n_g + 7$ independent channels and use a CNN-based method to regress the target pose vectors. However, in practice we receive comparable performance in this setting. This might be due to the fact that the image is compressed too much. Nevertheless, our entire pipeline is not sensitive to this step as long as satisfactory results are obtained in this stage.

6 COMPARISONS

In this section, we conduct several comparisons of *TSynNet* with different alternatives. We first compare *TSynNet* with three general image-to-image translation networks, including Pix2Pix [Isola et al. 2017] for paired single-modal translation and two state-of-the-art unpaired methods, Fixed-Point GAN [Siddiquee et al. 2019] and MUNIT [Huang et al. 2018]. All methods are based on the author-provided implementations with the default settings.

Since Pix2Pix requires supervised data, we use our $(\bar{g}_y, \bar{g}_m)$ as its input and the mouth image x as the output to let the network learn to synthesize the appearance. At inference time, we send $\hat{g}_y$ generated from our aligned teeth model as its input to synthesize the final result. Note that since it does not have any appearance reference, the generated results are out of control in terms of lighting and texture, thus deviating from the original teeth attributes. Thanks to the appearance conditioning, *TSynNet* can retain the in-mouth appearance much better.

To train Fixed-Point GAN and MUNIT, we divide the whole dataset into a regular teeth set and an irregular teeth set, each of which contains about 1,000 images. During testing, we feed an image of irregular teeth into them. Fig. 15 shows the comparison results. Note that, without explicit guidance or control of the geometry, both Fixed-Point GAN and MUNIT are unable to learn the transformed teeth geometry. They tend to “blur” or partially adjust the input to make it appear “aligned”. In contrast, ours is more controllable and realistic.

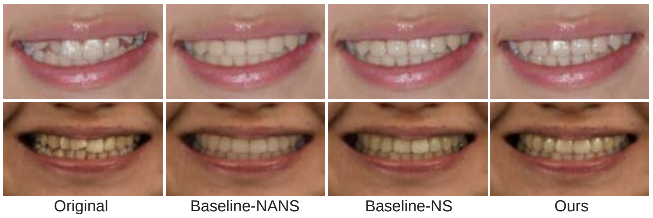

Fig. 14. Comparisons with the baseline models.

We also conduct a comparison against a texture mapping method. We first use the fitted 3D teeth to query the colors for the visible vertices directly from the before-orthodontics image. Then, we render the aligned teeth model (by projecting back the queried colors to the image space) to generate an image where invisible teeth parts of the original image are simply represented as blank. Such a direct texture mapping mechanism ignores the lighting and shading changes during the teeth transformations. Even with an inpainting network (we use a cGAN structure akin to [Isola et al. 2017]) to fill in the holes, the results still appear misaligned and unrealistic (Fig. 15).

7 USER STUDY

To further evaluate the quality of our generated images, we conducted a web-based user study with 80 participants, consisting of 60 ordinary people in the age range of 20-36 (a majority of them were college students) and 20 dentists. In the study, we presented two groups of facial images with different identities: a group A of 20 real face images with well-aligned teeth, and a group B of 20 face images with synthesized teeth by iOrthoPredictor. The participants were presented with a web page, displaying the real/synthetic images one by one in a random order, and were asked to respond to the statement “To what extent do you think the teeth in the image are real” on a 5-point Likert scale (1-not real, 2-likely not real, 3-cannot tell, 4-likely real, 5-real). The images were further divided into two sets: Set 1 containing the entire faces while Set 2 only the mouth regions. As shown in Table 4, for ordinary people, 71.6% and 60.8% of the real images (in Set 1 and Set 2 respectively) were rated as real (score 4 or 5) while 67.4% and 63.4% of our generated images were rated as real. Even for dentists, the rating difference is relatively small (64.4% and 68.7% for the real images while 62.9% and 67.0% for our results). The statistics indicate that our method can generate high-quality images so that even dentists find them convincingly real. Observing the images with low ratings, we found that they were often of low-quality (e.g., with back lighting conditions). Please see the images in the supplementary document.

8 DISCUSSION

Here we discuss some limitations of our method. First, our method only focuses on the mouth region, thus the facial growth that could alter during the orthodontic treatment is untouched. Simulation-based methods [Koch et al. 1996] may be exploited to solve this issue by taking facial bones into consideration. However, this is orthogonal to our work. Also, since our method does not estimate the gum geometry, the occluding relation between the teeth and the gums is not modeled. In some cases, the transformation of teeth might lead to a false occluding relation between the teeth and the gums. For example, part of a tooth, supposed to be covered by the

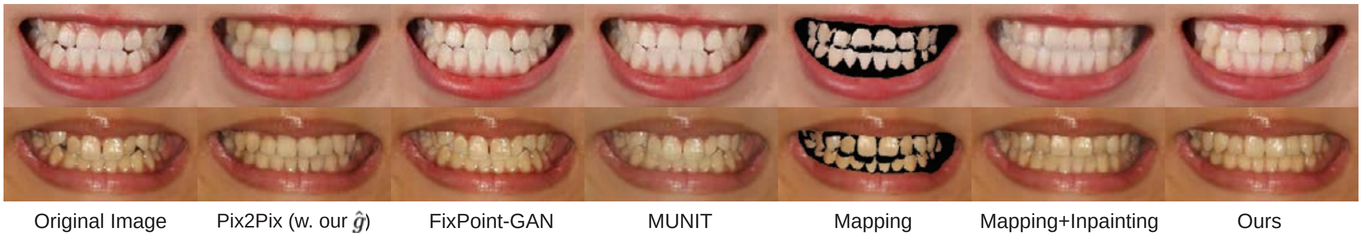

Fig. 15. Comparisons of our approach with alternative image translation, texture mapping, and inpainting methods.

Table 4. Statistics of our user study. The scores are from 1 (not real) to 5 (real). We give the percentage for each score and the percentage that the image was rated as 'real' (a score of 4 or higher).

	Ordinary People						**Dentists**						**Average**		
	Scores						Scores						Scores		
	1	2	3	4	5	'real'	1	2	3	4	5	'real'	4	5	'real'
set 1 (real)	6.9	13.9	7.6	22.8	48.8	71.6%	11.8	14.9	8.9	29.7	34.7	64.4%	26.3	41.7	68.0%
set 2 (real)	10.4	16.5	12.3	14.6	46.2	60.8%	9.6	7.2	14.5	24.1	44.6	68.7%	19.3	45.4	64.7%
set 1 (fake)	8.9	15.8	7.9	19.3	48.1	67.4%	16.9	13.5	6.7	25.8	37.1	62.9%	22.5	42.6	65.1%
set 2 (fake)	11.1	16.0	9.5	17.6	45.8	63.4%	9.9	7.7	15.4	41.7	25.3	67.0%	29.7	35.5	65.2%

gums, might become visible after transformation (see Fig. 16 (left) for an example), leading to small artifacts. A similar issue exists with the tongue. As the geometry of tongue is not explicitly modeled, incorrect estimation could happen during regeneration as shown in Fig. 16. In addition, when the mouth is largely open, the teeth area is comparatively small. As a result, the extracted teeth appearance code might be spoiled by the other parts of the mouth cavity (Fig. 16). Using semantic masks and partial convolution might be a solution.

Second, our method requires a 3D dental teeth model to accurately predict the teeth geometry transformations, which could be an overhead. Image-based teeth reconstruction methods (e.g., [Wu et al. 2016]) may be applied to ease this procedure. On the other hand, although *TAligNet* enables us to predict the aligned teeth model that are close to orthodontists' edits (Section 5.3), it does not fully resolve the orthodontic planning problem since medical orthodontic planning involves many rules. For example, the gap and the collision between two adjacent teeth should be precise for real treatment, and the occluding relations between the upper and the lower teeth should also be correct. Any of these issues will require further adjustment of the teeth model. Since adjusting one tooth could affect all the rest, this problem is very challenging. Thus, in our current solution, we only focus on generating visually correct alignment results (Fig. 9). Besides, *TAligNet* cannot synthesize a missing tooth and thus it may fail to handle missing parts well, as indicated in Sec. 4. We consider it as our future work.

Third, our *TSynNet* does not extract shading parameters. As a consequence, inconsistent shading and incorrect subsurface scattering and translucency might be visible on certain generated photographs (e.g., Fig. 6, the third row). Estimating the appearance model [Velinov et al. 2019] and augmenting *TSynNet* with depth data might alleviate this problem.

Fourth, since our method requires a frontal face image with visible teeth, it does not perform well on face images where the head pose is non-frontal or the teeth are invisible. In addition, our current implementation focuses on people with permanent teeth and thus is not applicable to babies.

9 CONCLUSION

In this paper, we introduced for the first time a deep learning based framework, named iOrthoPredictor, to predict the visual outcome of orthodontic treatment in a face photo. The key formulation is to disentangle the in-mouth appearance synthesis from the teeth geometry transformation. To accurately compute a target teeth shape, whose projection generates the aligned 2D teeth shape, our method leverages a given 3D teeth model of a patient and learns teeth alignment in 3D. Three neural networks have been introduced in iOrthoPredictor and seamlessly integrated to fulfill the disentanglement and synthesize the final result. Our system allows a flexible control over the fine-scale teeth geometry as well as the retention of the in-mouth appearance. Extensive experiments and a user study show the effectiveness of our method in predicting the treatment effect in digital orthodontics.

Our work lies in the series of image-based facial editing techniques, with a particular focus on orthodontic alignment. Although we focus on a feasible solution to a specific problem, which is not well explored in the graphics community, we believe that the high

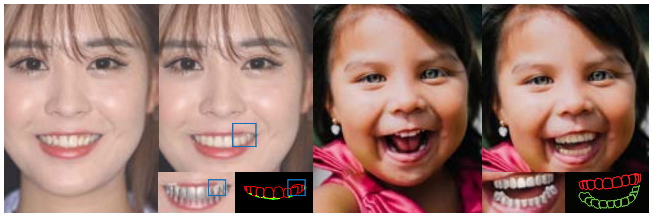

Fig. 16. Imperfect results generated by our algorithm. The right image is collected from the Internet, thus we manually fit a 3D teeth model to it. The right image original courtesy of Joel Danielson.

practical value of our work shows a proof of concept that any physically correct edits of an object should eventually obey the underlying geometry. This is also true in mesh editing techniques where to achieve physically correct deformations one should essentially take the underlying muscle structures into consideration. We hope that this proof of concept could be inspiring for future works in the areas of image and mesh editing.

ACKNOWLEDGMENTS

We thank the anonymous reviewers for the constructive comments. Hongbo Fu was partially supported by grants from the Research Grants Council of the Hong Kong Special Administrative Region, China (Project No. CityU 11212119).

REFERENCES

Panos Achlioptas, Olga Diamanti, Ioannis Mitliagkas, and Leonidas Guibas. 2018. Learning representations and generative models for 3D point clouds. In *International Conference on Machine Learning (ICML)*. 40–49.

Yazeed Alharbi, Neil Smith, and Peter Wonka. 2019. Latent Filter Scaling for Multimodal Unsupervised Image-to-Image Translation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 1458–1466.

Matthew Amodio and Smita Krishnaswamy. 2019. TraVeLGAN: Image-to-image Translation by Transformation Vector Learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 8983–8992.

Moab Arar, Yiftach Ginger, Dov Danon, Ilya Leizerson, Amit Bermano, and Daniel Cohen-Or. 2020. Unsupervised Multi-Modal Image Registration via Geometry Preserving Image-to-Image Translation. *arXiv preprint arXiv:2003.08073* (2020).

Milton B Asbell. 1990. A brief history of orthodontics. *American Journal of Orthodontics and Dentofacial Orthopedics* 98, 2 (1990), 176–183.

Coloma Ballester, Marcelo Bertalmio, Vicent Caselles, Guillermo Sapiro, and Joan Verdera. 2001. Filling-in by joint interpolation of vector fields and gray levels. *IEEE transactions on image processing* 10, 8 (2001), 1200–1211.

Connelly Barnes, Eli Shechtman, Adam Finkelstein, and Dan B Goldman. 2009. PatchMatch: A randomized correspondence algorithm for structural image editing. In *ACM Transactions on Graphics (ToG)*, Vol. 28. ACM, 24:1–24:11.

Connelly Barnes, Eli Shechtman, Dan B Goldman, and Adam Finkelstein. 2010. The generalized PatchMatch correspondence algorithm. In *European Conference on Computer Vision*. Springer, 29–43.

Marcelo Bertalmio, Guillermo Sapiro, Vincent Caselles, and Coloma Ballester. 2000. Image Inpainting. In *Proceedings of the 27th Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH '00)*. 417–424.

Volker Blanz, Curzio Basso, Tomaso Poggio, and Thomas Vetter. 2003. Reanimating faces in images and video. *Computer Graphics Forum* 22, 3 (2003), 641–650.

Chen Cao, Qiming Hou, and Kun Zhou. 2014. Displaced dynamic expression regression for real-time facial tracking and animation. *ACM Transactions on Graphics (ToG)* 33, 4 (2014), 43.

Duygu Ceylan, Niloy J. Mitra, Youyi Zheng, and Mark Pauly. 2014. Coupled structure-from-motion and 3D symmetry detection for urban facades. *ACM Trans. Graph.* 33, 1, Article Article 2 (Feb. 2014), 15 pages.

Menglei Chai, Tianjia Shao, Hongzhi Wu, Yanlin Weng, and Kun Zhou. 2016. AutoHair: Fully automatic hair modeling from a single image. *ACM Trans. Graph.* 35, 4 (2016), 116:1–116:12.

Huiwen Chang, Jingwan Lu, Fisher Yu, and Adam Finkelstein. 2018. PairedCycleGAN: Asymmetric style transfer for applying and removing makeup. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 40–48.

Yunjey Choi, Minje Choi, Munyoung Kim, Jung-Woo Ha, Sunghun Kim, and Jaegul Choo. 2018. StarGAN: Unified generative adversarial networks for multi-domain image-to-image translation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 8789–8797.

Antonio Criminisi, Patrick Pérez, and Kentaro Toyama. 2004. Region filling and object removal by exemplar-based image inpainting. *IEEE Transactions on image processing* 13, 9 (2004), 1200–1212.

Zhiming Cui, Changjian Li, and Wenping Wang. 2019. ToothNet: Automatic tooth instance segmentation and identification from cone beam CT images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 6368–6377.

Jifeng Dai, Haozhi Qi, Yuwen Xiong, Yi Li, Guodong Zhang, Han Hu, and Yichen Wei. 2017. Deformable convolutional networks. In *Proceedings of the IEEE International Conference on Computer Vision*. 764–773.

Kevin Dale, Kalyan Sunkavalli, Micah K Johnson, Daniel Vlasic, Wojciech Matusik, and Hanspeter Pfister. 2011. Video face replacement. *ACM Trans. Graph.* 30, 6 (2011), 130:1–130:10.

Yue Deng, Qionghai Dai, and Zengke Zhang. 2011. Graph Laplace for occluded face completion and recognition. *IEEE Transactions on Image Processing* 20, 8 (2011), 2329–2338.

Hui Ding, Kumar Sricharan, and Rama Chellappa. 2018. ExprGAN: Facial expression editing with controllable expression intensity. In *Thirty-Second AAAI Conference on Artificial Intelligence*.

Iddo Drori, Daniel Cohen-Or, and Hezy Yeshurun. 2003. Fragment-based image completion. *ACM Transactions on Graphics (ToG)* 22, 3 (2003), 303–312.

Alexei A Efros and William T Freeman. 2001. Image quilting for texture synthesis and transfer. In *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*. ACM, 341–346.

Alexei A Efros and Thomas K Leung. 1999. Texture synthesis by non-parametric sampling. In *Proceedings of the seventh IEEE international conference on computer vision*, Vol. 2. IEEE, 1033–1038.

Patrick Esser, Ekaterina Sutter, and Björn Ommer. 2018. A variational U-Net for conditional appearance and shape generation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 8857–8866.

Yaroslav Ganin, Daniil Kononenko, Diana Sungatullina, and Victor Lempitsky. 2016. Deepwarp: Photorealistic image resynthesis for gaze manipulation. In *European Conference on Computer Vision (ECCV)*. Springer, 311–326.

Pablo Garrido, Levi Valgaerts, Hamid Sarmadi, Ingmar Steiner, Kiran Varanasi, Patrick Perez, and Christian Theobalt. 2015. VDub: Modifying face video of actors for plausible visual alignment to a dubbed audio track. *Computer Graphics Forum* 34, 2 (2015), 193–204.

Jiahao Geng, Tianjia Shao, Youyi Zheng, Yanlin Weng, and Kun Zhou. 2018. Warp-guided GANs for single-photo facial animation. *ACM Trans. Graph.* 37, 6, Article 231 (2018), 12 pages.

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. 2014. Generative adversarial nets. In *Advances in Neural Information Processing Systems (NIPS)*. 2672–2680.

Lee W Graber, Robert L Vanarsdall, Katherine WL Vig, and Greg J Huang. 2016. *Orthodontics: current principles and techniques*. Elsevier Health Sciences.

Qiao Gu, Guanzhi Wang, Mang Tik Chiu, Yu-Wing Tai, and Chi-Keung Tang. 2019. LADN: Local adversarial disentangling network for facial makeup and de-makeup. *Proceedings of the IEEE International Conference on Computer Vision* (2019).

Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, and Aaron C Courville. 2017. Improved training of wasserstein gans. In *Advances in neural information processing systems*. 5767–5777.

James Hays and Alexei A Efros. 2007. Scene completion using millions of photographs. *ACM Transactions on Graphics (ToG)* 26, 3 (2007), 4–es.

Zhenliang He, Wangmeng Zuo, Meina Kan, Shiguang Shan, and Xilin Chen. 2019. AttGAN: Facial attribute editing by only changing what you want. *IEEE Transactions on Image Processing* (2019).

Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. 2017. Gans trained by a two time-scale update rule converge to a local nash equilibrium. In *Advances in neural information processing systems*. 6626–6637.

Jie Hu, Li Shen, and Gang Sun. 2018. Squeeze-and-Excitation Networks. *IEEE Conference on Computer Vision and Pattern Recognition*.

Jia-Bin Huang, Sing Bing Kang, Narendra Ahuja, and Johannes Kopf. 2014. Image completion using planar structure guidance. *ACM Transactions on Graphics (ToG)* 33, 4 (2014), 129.

Xun Huang and Serge Belongie. 2017. Arbitrary style transfer in real-time with adaptive instance normalization. In *Proceedings of the IEEE International Conference on Computer Vision*. 1501–1510.

Xun Huang, Ming-Yu Liu, Serge Belongie, and Jan Kautz. 2018. Multimodal unsupervised image-to-image translation. In *Proceedings of the European Conference on Computer Vision (ECCV)*. 172–189.

Satoshi Iizuka, Edgar Simo-Serra, and Hiroshi Ishikawa. 2017. Globally and locally consistent image completion. *ACM Transactions on Graphics (ToG)* 36, 4 (2017), 107.

Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. 2017. Image-to-image translation with conditional adversarial networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 1125–1134.

Max Jaderberg, Karen Simonyan, Andrew Zisserman, and koray kavukcuoglu. 2015. Spatial transformer networks. In *Advances in Neural Information Processing Systems 28*, C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett (Eds.). 2017–2025.

Yongcheng Jing, Yezhou Yang, Zunlei Feng, Jingwen Ye, Yizhou Yu, and Mingli Song. 2019. Neural style transfer: A review. *IEEE transactions on visualization and computer graphics* (2019).

Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. 2019. Analyzing and Improving the Image Quality of StyleGAN. *CoRR* abs/1912.04958 (2019).

Masahide Kawai, Tomoyori Iwao, Daisuke Mima, Akinobu Maejima, and Shigeo Morishima. 2013. Photorealistic inner mouth expression in speech animation. In *ACM SIGGRAPH 2013 Posters*. ACM, 9:1–9:1.

Masahide Kawai, Tomoyori Iwao, Daisuke Mima, Akinobu Maejima, and Shigeo Morishima. 2014. Data-driven speech animation synthesis focusing on realistic inside of the mouth. *Journal of information processing* 22, 2 (2014), 401–409.

Hyeongwoo Kim, Pablo Garrido, Ayush Tewari, Weipeng Xu, Justus Thies, Matthias Niessner, Patrick Pérez, Christian Richardt, Michael Zollhöfer, and Christian Theobalt. 2018. Deep video portraits. *ACM Trans. Graph.* 37, 4, Article 163 (2018), 14 pages.

Diederik Kingma and Max Welling. 2013. Auto-Encoding Variational Bayes. In *ICLR*.

Diederik P. Kingma and Jimmy Lei Ba. 2015. Adam: A method for stochastic optimization. In *International Conference on Learning Representations (ICLR)*.

Norman William Kingsley. 1880. *A treatise on oral deformities as a branch of mechanical surgery*. D. Appleton.

Rolf M Koch, Markus H Gross, Friedrich R Carls, Daniel F von Büren, George Fankhauser, and Yoav IH Parish. 1996. Simulating facial surgery using finite element models. In *Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*. 421–428.

Rolf Köhler, Christian Schuler, Bernhard Schölkopf, and Stefan Harmeling. 2014. Mask-specific inpainting with deep neural networks. In *German Conference on Pattern Recognition*. Springer, 523–534.

Johannes Kopf, Wolf Kienzle, Steven Drucker, and Sing Bing Kang. 2012. Quality prediction for image completion. *ACM Transactions on Graphics (ToG)* 31, 6 (2012), 131.

Iryna Korshunova, Wenzhe Shi, Joni Dambre, and Lucas Theis. 2017. Fast face-swap using convolutional neural networks. In *The IEEE International Conference on Computer Vision*. 3697–3705.

Claudia Kuster, Tiberiu Popa, Jean-Charles Bazin, Craig Gotsman, and Markus Gross. 2012. Gaze correction for home video conferencing. *ACM Trans. Graph.* 31, 6 (2012), 174:1–174:6.

Vivek Kwatra, Irfan Essa, Aaron Bobick, and Nipun Kwatra. 2005. Texture optimization for example-based synthesis. *ACM Transactions on Graphics (ToG)* 24, 3 (2005), 795–802.

Guillaume Lample, Neil Zeghidour, Nicolas Usunier, Antoine Bordes, Ludovic DENOYER, et al. 2017. Fader Networks: Manipulating Images by Sliding Attributes. In *Advances in Neural Information Processing Systems*. 5963–5972.

Anat Levin, Assaf Zomet, and Yair Weiss. 2003. Learning how to inpaint from global image statistics. In *International Conference on Computer Vision*. IEEE, 305–312.

Zhanli Li, Jingding Fu, Hongan Li, Kang Zhou, and Qiaojuan Hui. 2019. Automatic arrangement method of misaligned teeth in virtual orthodontic treatment. In *Journal of Graphics (in Chinese)*, Vol. 40. 225–234.

Ming-Yu Liu, Thomas Breuel, and Jan Kautz. 2017. Unsupervised image-to-image translation networks. In *Advances in neural information processing systems*. 700–708.

Si Liu, Xinyu Ou, Ruihe Qian, Wei Wang, and Xiaochun Cao. 2016. Makeup like a superstar: Deep localized makeup transfer network. *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence, IJCAI'16* (2016), 2568—-2575.

Mehdi Mirza and Simon Osindero. 2014. Conditional generative adversarial nets. *arXiv preprint arXiv:1411.1784* (2014).

Umar Mohammed, Simon JD Prince, and Jan Kautz. 2009. Visio-lization: Generating novel facial images. *ACM Transactions on Graphics (ToG)* 28, 3 (2009), 57.

Kyle Olszewski, Zimo Li, Chao Yang, Yi Zhou, Ronald Yu, Zeng Huang, Sitao Xiang, Shunsuke Saito, Pushmeet Kohli, and Hao Li. 2017. Realistic dynamic facial textures from a single image using GANs. In *IEEE International Conference on Computer Vision (ICCV)*. 5429–5438.

Darko Pavić, Volker Schönefeld, and Leif Kobbelt. 2006. Interactive image completion with perspective correction. *The Visual Computer* 22, 9-11 (2006), 671–681.

Guim Perarnau, Joost Van De Weijer, Bogdan Raducanu, and Jose M Álvarez. 2016. Invertible conditional gans for image editing. *NIPS Workshop on Adversarial Training* (2016).

Robert J Peterman, Shuying Jiang, Rene Johe, and Padma M Mukherjee. 2016. Accuracy of Dolphin visual treatment objective (VTO) prediction software on class III patients treated with maxillary advancement and mandibular setback. *Progress in orthodontics* 17, 1 (2016), 19.

G Power, J Breckon, M Sherriff, and F McDonald. 2005. Dolphin Imaging Software: an analysis of the accuracy of cephalometric digitization and orthognathic prediction. *International journal of oral and maxillofacial surgery* 34, 6 (2005), 619–626.

William R Proffit, Henry W Fields Jr, and David M Sarver. 2006. *Contemporary orthodontics*. Elsevier Health Sciences.

Albert Pumarola, Antonio Agudo, Aleix M Martinez, Alberto Sanfeliu, and Francesc Moreno-Noguer. 2018. GANimation: Anatomically-aware facial animation from a single image. In *Proceedings of the European Conference on Computer Vision (ECCV)*. 818–833.

Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. 2017. Pointnet: Deep learning on point sets for 3D classification and segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 652–660.

Shengju Qian, Kwan-Yee Lin, Wayne Wu, Yangxiaokang Liu, Quan Wang, Fumin Shen, Chen Qian, and Ran He. 2019. Make a face: Towards arbitrary high fidelity face manipulation. In *International Conference on Computer Vision (ICCV)*.

Fengchun Qiao, Naiming Yao, Zirui Jiao, Zhihao Li, Hui Chen, and Hongan Wang. 2018. Geometry-contrastive generative adversarial network for facial expression synthesis. *arXiv preprint arXiv:1802.01822* (2018).

Waseem Rawat and Zenghui Wang. 2017. Deep convolutional neural networks for image classification: A comprehensive review. *Neural computation* 29, 9 (2017), 2352–2449.

Jimmy SJ Ren, Li Xu, Qiong Yan, and Wenxiu Sun. 2015. Shepard convolutional neural networks. In *Advances in Neural Information Processing Systems*. 901–909.

Yurui Ren, Xiaoming Yu, Ruonan Zhang, Thomas H Li, Shan Liu, and Ge Li. 2019. StructureFlow: Image inpainting via structure-aware appearance flow. In *Proceedings of the IEEE International Conference on Computer Vision*. 181–190.

Olaf Ronneberger, Philipp Fischer, and Thomas Brox. 2015. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*. Springer, 234–241.

Zhixin Shu, Mihir Sahasrabudhe, Riza Alp Guler, Dimitris Samaras, Nikos Paragios, and Iasonas Kokkinos. 2018. Deforming autoencoders: Unsupervised disentangling of shape and appearance. In *The European Conference on Computer Vision (ECCV)*.

Md Mahfuzur Rahman Siddiquee, Zongwei Zhou, Nima Tajbakhsh, Ruibin Feng, Michael B. Gotway, Yoshua Bengio, and Jianming Liang. 2019. Learning fixed points in generative adversarial networks: From image-to-image translation to disease detection and localization. In *The IEEE International Conference on Computer Vision (ICCV)*.

Denis Simakov, Yaron Caspi, Eli Shechtman, and Michal Irani. 2008. Summarizing visual data using bidirectional similarity. In *2008 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 1–8.

Karen Simonyan and Andrew Zisserman. 2014. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556* (2014).

Lingxiao Song, Zhihe Lu, Ran He, Zhenan Sun, and Tieniu Tan. 2018. Geometry guided adversarial facial expression synthesis. In *ACM Multimedia Conference on Multimedia Conference*. ACM, 627–635.

Jian Sun, Lu Yuan, Jiaya Jia, and Heung-Yeung Shum. 2005. Image completion with structure propagation. *ACM Transactions on Graphics (ToG)* 24, 3 (2005), 861–868.

Supasorn Suwajanakorn, Steven M Seitz, and Ira Kemelmacher-Shlizerman. 2017. Synthesizing obama: learning lip sync from audio. *ACM Transactions on Graphics (ToG)* 36, 4 (2017), 95.

Justus Thies, Michael Zollhöfer, Matthias Nießner, Levi Valgaerts, Marc Stamminger, and Christian Theobalt. 2015. Real-time expression transfer for facial reenactment. *ACM Trans. Graph.* 34, 6 (2015), 183:1–183:14.

Zdravko Velinov, Marios Papas, Derek Bradley, Paulo Gotardo, Parsa Mirdehghan, Steve Marschner, Jan Novák, and Thabo Beeler. 2019. Appearance capture and modeling of human teeth. *ACM Transactions on Graphics (ToG)* 37, 6 (2019), 207.

Shuyang Wang and Yun Fu. 2016. Face behind makeup. In *Thirtieth AAAI Conference on Artificial Intelligence*. 58–64.

Nicholas Watters, Loïc Matthey, Christopher P. Burgess, and Alexander Lerchner. 2019. Spatial Broadcast Decoder: A Simple Architecture for Learning Disentangled Representations in VAEs. *CoRR* abs/1901.07017 (2019). arXiv:1901.07017 http://arxiv.org/abs/1901.07017

Yonatan Wexler, Eli Shechtman, and Michal Irani. 2007. Space-time completion of video. *IEEE Transactions on Pattern Analysis & Machine Intelligence* 3 (2007), 463–476.

Oliver Whyte, Josef Sivic, and Andrew Zisserman. 2009. Get out of my picture! Internet-based inpainting.. In *BMVC*, Vol. 2. 5.

Chenglei Wu, Derek Bradley, Pablo Garrido, Michael Zollhöfer, Christian Theobalt, Markus Gross, and Thabo Beeler. 2016. Model-based teeth reconstruction. *ACM Transactions on Graphics (ToG)* 35, 6 (2016), 220.

Po-Wei Wu, Yu-Jing Lin, Che-Han Chang, Edward Y Chang, and Shih-Wei Liao. 2019b. RelGAN: Multi-Domain Image-to-Image Translation via Relative Attributes. In *Proceedings of the IEEE International Conference on Computer Vision*. 5914–5922.

Ruizheng Wu, Xin Tao, Xiaodong Gu, Xiaoyong Shen, and Jiaya Jia. 2019c. Attribute-driven spontaneous motion in unpaired image translation. In *The IEEE International Conference on Computer Vision (ICCV)*.

Wayne Wu, Kaidi Cao, Cheng Li, Chen Qian, and Chen Change Loy. 2019a. TransGaGa: Geometry-aware unsupervised image-to-image translation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 8012–8021.

Chaohao Xie, Shaohui Liu, Chao Li, Ming-Ming Cheng, Wangmeng Zuo, Xiao Liu, Shilei Wen, and Errui Ding. 2019. Image inpainting with learnable bidirectional attention maps. In *Proceedings of the IEEE International Conference on Computer Vision*. 8858–8867.

Junyuan Xie, Linli Xu, and Enhong Chen. 2012. Image denoising and inpainting with deep neural networks. In *Advances in neural information processing systems*. 341–349.

Saining Xie and Zhuowen Tu. 2015. Holistically-nested edge detection. In *Proceedings of the IEEE International Conference on Computer Vision*. 1395–1403.

Wei Xiong, Jiahui Yu, Zhe Lin, Jimei Yang, Xin Lu, Connelly Barnes, and Jiebo Luo. 2019. Foreground-aware image inpainting. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 5840–5848.

Xiaojie Xu, Chang Liu, and Youyi Zheng. 2018. 3D tooth segmentation and labeling using deep convolutional neural networks. *IEEE transactions on visualization and computer graphics* 25, 7 (2018), 2336–2348.

Chao Yang, Xin Lu, Zhe Lin, Eli Shechtman, Oliver Wang, and Hao Li. 2017. High-resolution image inpainting using multi-scale neural patch synthesis. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition.* 6721–6729.

Lingchen Yang, Lumin Yang, Mingbo Zhao, and Youyi Zheng. 2018. Controlling Stroke Size in Fast Style Transfer with Recurrent Convolutional Neural Network. In *Computer Graphics Forum*, Vol. 37. 97–107.

Raymond Yeh, Ziwei Liu, Dan B Goldman, and Aseem Agarwala. 2016. Semantic facial expression editing using autoencoded flow. *arXiv preprint arXiv:1611.09961* (2016).

Zili Yi, Hao Zhang, Ping Tan, and Minglun Gong. 2017. Dualgan: Unsupervised dual learning for image-to-image translation. In *Proceedings of the IEEE International Conference on Computer Vision.* 2849–2857.

Jiahui Yu, Zhe Lin, Jimei Yang, Xiaohui Shen, Xin Lu, and Thomas S Huang. 2018. Generative image inpainting with contextual attention. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition.* 5505–5514.

Bo Zhao, Bo Chang, Zequn Jie, and Leonid Sigal. 2018. Modular generative adversarial networks. In *Proceedings of the European Conference on Computer Vision (ECCV).* 150–165.

Zhong-Qiu Zhao, Peng Zheng, Shou-tao Xu, and Xindong Wu. 2019. Object detection with deep learning: A review. *IEEE transactions on neural networks and learning systems* 30, 11 (2019), 3212–3232.

Chuanxia Zheng, Tat-Jen Cham, and Jianfei Cai. 2019. Pluralistic image completion. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition.* 1438–1447.

Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. 2017a. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Proceedings of the IEEE international conference on computer vision.* 2223–2232.

Jun-Yan Zhu, Richard Zhang, Deepak Pathak, Trevor Darrell, Alexei A Efros, Oliver Wang, and Eli Shechtman. 2017b. Toward multimodal image-to-image translation. In *Advances in Neural Information Processing Systems.* 465–476.

Xizhou Zhu, Han Hu, Stephen Lin, and Jifeng Dai. 2019a. Deformable convnets v2: More deformable, better results. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition.* 9308–9316.

Zhen Zhu, Tengteng Huang, Baoguang Shi, Miao Yu, Bofei Wang, and Xiang Bai. 2019b. Progressive pose attention transfer for person image generation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition.* 2347–2356.

A GLOBAL TEETH POSE FITTING

In this appendix, we describe the fitting process of our teeth model $\mathcal{T}$ to the detected geometry maps $\{\bar{g}_u, \bar{g}_l, \bar{g}_m\}$. We only describe the fitting process of the upper teeth $\mathcal{T}_u$, since the process for $\mathcal{T}_l$ is similar. As in Wu et al. [2016], we employ an EM algorithm to alternate multiple times between the following two steps: estimating the point-wise correspondence between the projected contour of $\mathcal{T}_u$ and that of $\bar{g}_u$, and optimizing the transformation matrix by minimizing the re-projection error of the corresponding points. Different from Wu et al. [2016], where the shape of the 3D teeth model is simultaneously optimized in the EM process, our 3D teeth shape is known and matches $\bar{g}_u$ well, enabling us to add more specific constraints into correspondence searching to achieve more robust fitting.

Specifically, we solve the 2D point-to-point correspondences $\Theta : \{\boldsymbol{P}^T\} \rightarrow \{\boldsymbol{P}^D\}$, where $\{\boldsymbol{P}^T\}$ is a set of sampled points (100 sampled points in our implementation) on the projected silhouette map of $\mathcal{T}_u$ (orthogonal projection is used here), while $\{\boldsymbol{P}^D\}$ an entire set of points on $\bar{g}_u$. For each point $\boldsymbol{P}_i^T$, $\boldsymbol{P}_{\Theta(i)}^D$ is the optimal correspondence, with Θ being computed by minimizing the following energy function:

$$\arg\min_{\Theta} \sum_i (E_p(\boldsymbol{P}_i^T) + E_e(\boldsymbol{P}_i^T, \boldsymbol{P}_{i+1}^T)), \tag{10}$$

where E_p helps to find the point not only close to but also similarly oriented as $\boldsymbol{P}_i^T$, and E_e takes Markov properties into consideration, ensuring the continuity for the correspondences of two geodesically adjacent points $\boldsymbol{P}_i^T$ and $\boldsymbol{P}_{i+1}^T$:

$$E_p(\boldsymbol{P}_i^T) = ||\boldsymbol{p}_i^T - \boldsymbol{p}_{\Theta(i)}^D|| \cdot exp(-|\langle \boldsymbol{t}_i^T, \boldsymbol{t}_{\Theta(i)}^D \rangle|), \tag{11}$$

$$E_e(\boldsymbol{P}_i^T, \boldsymbol{P}_{i+1}^T) = ||(\boldsymbol{p}_i^T - \boldsymbol{p}_{\Theta(i)}^D) - (\boldsymbol{p}_{i+1}^T - \boldsymbol{p}_{\Theta(i+1)}^D)||, \tag{12}$$

where $\boldsymbol{p}$ and $\boldsymbol{t}$ respectively denote position and tangent vectors. We solve the above problem by optimizing a Hidden Markov Model with E_p and E_e treated as the emission and transition probabilities respectively, as in Chai et al. [2016]. Fig. 17 shows the benefits of adding these constraints.

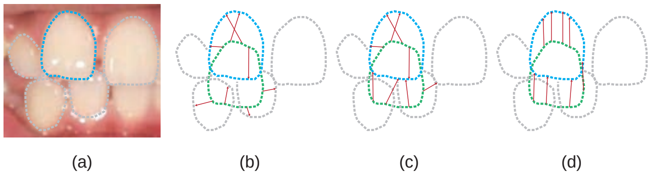

Fig. 17. Point-wise correspondence search between the projected silhouette (shown in green dots in (b)-(d)) of a 3D tooth and the detected teeth silhouettes in the image (a). Ideally, the green silhouette should match with the blue one in the upper row of teeth (a). Without semantic information for the upper and lower teeth, the searching process is likely spoiled by the lower teeth boundaries (b), while with such information it is confined to the upper teeth silhouettes (c). However, only with E_p as the energy function, the continuousness of the correspondences is lost (c). Our full configuration achieves a more robust result (d).

This is a highly nonlinear optimization, and any bad initialization of the global transformation would cause failure. To automate the process, we additionally let *TGeoNet* extract the semantic contours of the four incisors, trained over the same dataset with extra semantic annotations on the four incisors. To obtain a good initial pose preceding the EM optimization, we perform normalized cross correlation (NCC) [Ceylan et al. 2014] to align the 3D teeth silhouettes with respect to the predicted four 2D incisor contours. In practice, we find that this step largely facilitates the subsequent optimization to find a good global optimum. For cases where some incisors are missing or not detected well due to occlusion, the fitting might fall into local minima. In such cases, we include similar human interaction as in [Wu et al. 2016] by asking users to identify one tooth per row. In practice, we perform such user corrections only for the example shown in the first row of Fig. 6 and the 2^{nd} person in the third row of Fig. 9, where the silhouettes of the incisors are incompletely detected due to occlusion. For other examples in the paper, the fitting is fully automatic.