OrthoAligner: Image-Based Teeth Alignment Prediction via Latent Style Manipulation

Beijia Chen, Hongbo Fu, Kun Zhou, *Fellow, IEEE*, and Youyi Zheng

Abstract—In this article, we present OrthoAligner, a novel method to predict the visual outcome of orthodontic treatment in a portrait image. Unlike the state-of-the-art method, which relies on a 3D teeth model obtained from dental scanning, our method generates realistic alignment effects in images without requiring additional 3D information as input and thus making our system readily available to average users. The key of our approach is to employ the 3D geometric information encoded in an unsupervised generative model, i.e., StyleGAN in this article. Instead of directly conducting translation in the image space, we embed the teeth region extracted from a given portrait to the latent space of the StyleGAN generator and propose a novel latent editing method to discover a geometrically meaningful editing path that yields the alignment process in the image space. To blend the edited mouth region with the original portrait image, we further introduce a BlendingNet to remove boundary artifacts and correct color inconsistency. We also extend our method to short video clips by propagating the alignment effects across neighboring frames. We evaluate our method in various orthodontic cases, compare it to the state-of-the-art and competitive baselines, and validate the effectiveness of each component.

Index Terms—Teeth alignment, GAN inversion, StyleGAN

1 Introduction

Methods for facial image beautification[1], [2], [3], [4] have attracted increasing attention in recent years. Despite the recent progress in auto makeup and wrinkle removal, little attention has been paid to tools that enable the alignment of teeth. As one part of human facial appearance, teeth have a profound influence on one's facial identity (see Fig. 1). A set of aligned teeth in one's smiling portrait help greatly build self-confidence. Besides its great potential in facial image beautification, a teeth alignment tool also enables a quick pre-review of post-treatment effects in orthodontics. Such a tool, even inaccurate in orthodontics, could largely help enhance the engagement between potential patients and dentists if made accessible to commodity users.

Building an effective teeth alignment tool faces several challenges. First, it requires the tool to infer for each individual tooth a series of movements, which are essentially in 3D space. Second, to get holistic alignment effects, one needs to hallucinate the teeth appearance changes along with the teeth shape changes while also maintaining their original characteristics (so that the aligned teeth appear to be a 'real' transform from the original ones). To tackle these challenges, Yang *et al.* [5] proposed to explicitly leverage a 3D teeth model obtained from dental scanning to delegate the geometric transformation of each tooth and further use an appearance-structure disentangled variational autoencoder (VAE) [6] to generate the final teeth image. Although their approach achieves remarkable results, their dependency on an accurate 3D teeth model of a patient leads to the cost of tedious manual geometric preprocessing (e.g., scanning, 3D reconstruction, denoising, manual segmentation, etc.), and also prevents their tool from being used by commodity users.

In this paper, we develop a novel system named OrthoAligner to produce realistic teeth alignment effects in images with no additional user or 3D input. The central idea of our method is inspired by the recent findings in unsupervised generative adversarial networks (GANs) [7], [8], [9], [10]: the manifold of 2D images, usually parameterized by certain latent space in unsupervised GANs, contains rich 3D geometric clues [11], such that walking along certain paths on the manifold could lead to meaningful geometric transformation of an underlying object (e.g., rotation and translation [12], [13], [14], [15]). In essence, our goal is to discover in the latent space a particular path which corresponds to the teeth alignment process.

To this end, we face two sub-problems: 1) GAN inversion problem [16], [17], [18]: given a malpositioned teeth image extracted from a patient portrait, we need to invert it to the latent space of a pretrained unsupervised GAN. 2) The editing problem: after finding an optimal latent representation of a specific image, we need to find an editing path that corresponds to the "alignment" process. Both problems are non-trivial. First, current GAN inversion methods [16], [17], [18] often suffer from the well-known reconstruction-or-editing dilemma [18], [19], i.e., the latent vector that produces faithful reconstruction of a given image might perform poorly when feeding to downstream editing methods. This is because that most inversion methods simply expand the original latent space to achieve better reconstruction quality without considering the latent space statistics [18]. On the

- *Beijia Chen, Kun Zhou, and Youyi Zheng are with the State Key Laboratory of CAD&CG, Zhejiang University, Hangzhou 310058, China. E-mail: {beibeijia, youyizheng}@zju.edu.cn, kunzhou@acm.org.*
- *Hongbo Fu is with the School of Creative Media, City University of Hong Kong, China. E-mail: hongbofu@cityu.edu.hk.*

Manuscript received 11 Aug. 2021; revised 20 Feb. 2022; accepted 27 Mar. 2022. Date of publication 11 Apr. 2022; date of current version 30 June 2023.
This work was supported in part by the National Key Research & Development Program of China under Grant 2018YFE0100900.
(Corresponding author: Youyi Zheng.)
Recommended for acceptance by W. Wang.
Digital Object Identifier no. 10.1109/TVCG.2022.3166159

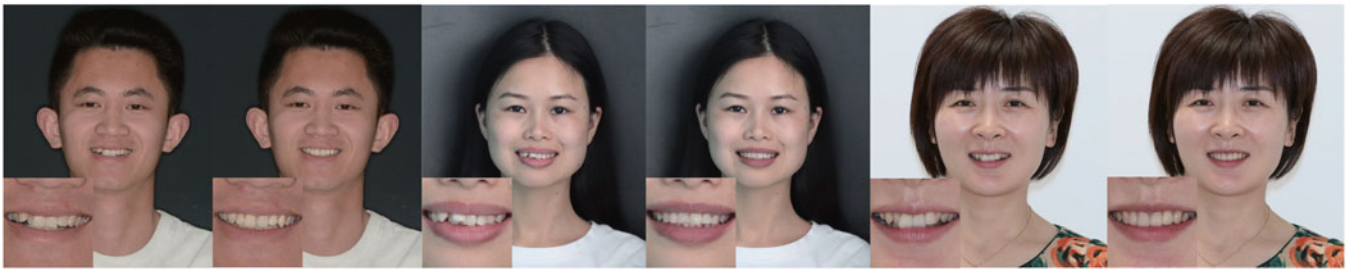


Fig. 1. Representative teeth alignment results for real portrait images by our method. For each case, we show the original portrait image with unaligned teeth (Left) and the edited portrait image with aligned teeth (Right). The edited area has been enlarged for each case. We can observe that teeth alignment positively influences one's facial appearance.

other hand, the editing problem poses additional challenges. First, it requires the editing path to be solely geometric, i.e., only adjusting the overall geometry between different teeth. The appearance (such as color, lighting) and identity (e.g., shape) of each individual tooth should remain intact. Second, unlike other concepts (such as the head orientation in a portrait image) that can be explicitly defined as numeric numbers, "alignment" is a holistic status that describes the complex interplay between teeth.

We present a set of techniques to address the aforementioned challenges. We employ StyleGAN [9] to model the manifold of teeth images, since it provides natural disentangling of different levels of image features via controlling the generation process by layer-wise latent codes [9], [10]. To address the reconstruction-or-editing dilemma, we adopt the state-of-the-art optimization-based inversion method that incorporates the latent space statistics, thus achieving a better trade-off between reconstruction and editing quality. To discover the editing path corresponding to the alignment process, we divide the style codes into three groups, one of them corresponding to geometric transformations of teeth. Then, to enable our optimizer to find an optimal "alignment" editing path, we manually annotate each teeth image with a scalar score (1-10) to define its mal-occluding level and pretrain a ScoreNet. The optimal editing path, parametrized as a transformation in the latent space, is obtained by minimizing the score evaluated by the ScoreNet. The ScoreNet stores the concrete concepts of what a malpositioned or an aligned teeth should look like and drives the optimization to find the meaningful editing path. It also enables us to visualize how an unaligned teeth gradually evolves into an aligned one in the image space (see Fig. 11).

The imperfect reconstruction in GAN inversion often brings in boundary artifacts and color inconsistency when mapping the edited teeth image back to the original portrait image. To address this issue, we further introduce a BlendingNet for boundary blending and color correction. Fig. 2 visualizes the overall pipeline. We also extend our teeth alignment method to short video clips. Extensive results shown that our method produces realistic aligned teeth images, and it is comparable to the 3D-assisted method [5] and surpasses competitive baselines.

In summary, our contributions are:

- We present the first method for teeth alignment prediction in portrait images without explicitly employing any user or 3D input, making our method available to average users.
- We present a novel latent editing method in StyleGAN for finding the transformation path corresponding to the teeth alignment process.
- We introduce a BlendingNet to remove visible artifacts and color inconsistency at boundaries when mapping the edited teeth image back to the original portrait. We also extend our align method to short video clips.

2 Related Works

2.1 Image-Based Teeth Alignment

Yang *et al.* [5] are among the first to develop a system to enable preview of teeth alignment effects in images. Their method relies on a pre-scanned 3D teeth model of a patient, which is then aligned with an input image to predict the 3D transformations of individual teeth. The transformed 3D teeth are then rendered back to the 2D image space to generate a teeth silhouette map, which represents the teeth geometry, followed by an appearance-and-geometry disentangled VAE to synthesize aligned results. Such a 3D teeth model, requiring a tedious procedure of digital processing (scanning, reconstruction, segmentation), hinders their tool to be used for average users. On the other hand, without explicitly modeling the gum, their method sometimes suffers from visual artifacts between the teeth-gum parts caused by inaccurate rendering of the 3D teeth (as shown in Fig. 13). In this paper, we build the first system for image-based teeth alignment that relies on an input portrait image, without any additional inputs.

2.2 Image-to-Image Translation

Image-to-image translation, which aims to convert a given image across different domains, has attracted increasing attentions. Starting from the groundbreaking work by Isola *et al.* [20], which learns a deterministic mapping from paired data in a supervised manner, diverse translating models adapt the original idea to unsupervised situations [21], [22], [23], [24], multi-label image attribute editing [25], [26], [27], and multi-style image generation [21], [28], [29], etc. Instead of directly conducting translation in the image space, our work learns transformations in the latent space of an unsupervised GAN. We benefit from this choice in two important aspects: 1) The image quality generated by unsupervised GANs (such as StyleGAN [9] and StyelGAN2 [10]) often surpasses that of previously-mentioned translating models [21], [22], [27], [28]. Therefore, by employing a well-trained StyleGAN as our generator, we naturally

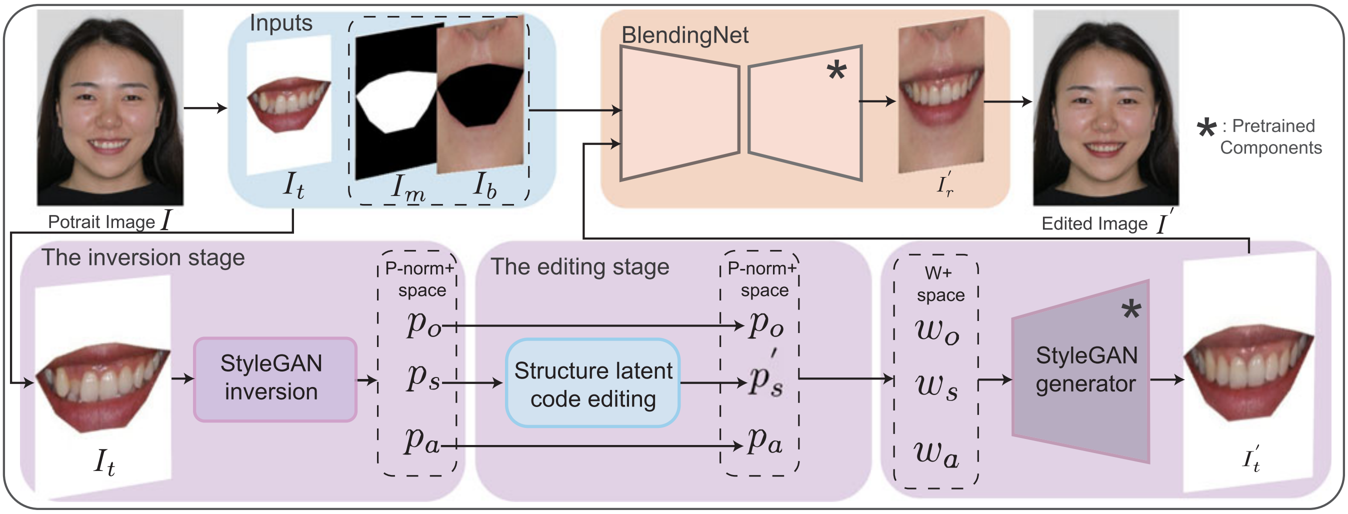


Fig. 2. The overall pipeline of our method. Given a portrait image I with visible malpositioned teeth we preprocess it to obtain the teeth image I_t, the mouth mask I_m and the mouth background image I_b according to Fig. 3. The inversion stage inverts the teeth image I_t to the $P\text{-}norm^+$ space of a pretrained StyleGAN. The teeth structure latent codes p_s are manipulated at the structure latent code editing stage. The edited structure codes p_s' with the original outer-mouth latent codes p_o and appearance codes p_a are then transformed into the W^+ space by passing them through a fixed function $T(\cdot)$. The StyleGAN generator outputs the corresponding edited teeth image I_t'. Finally, we use a BlendingNet, which blends the edited teeth image I_t', mouth mask I_m, and background image I_b, to obtain a realistic aligned image. In the figure, we use $*$ to label the components that require pre-training and remain fixed during the whole pipeline.

inherent the ability of generating high-quality images without the complex training of a discriminator during the translation process. 2) By decoupling several types of image semantics in the latent space, we obtain a model that can be applied to multiple tasks without redesigning the network. For example, we can also transfer appearance from one teeth image to another, as shown in Fig. 5.

2.3 Image Manipulation via GAN Latent Space Editing

Recent works have demonstrated that well-trained unsupervised GANs [7], [8], [9], [10] encode rich semantics in their latent space [12], [30], [31], [32]. Numerous methods [12], [14], [15], [30] have been proposed to uncover such semantics by identifying meaningful editing paths in latent space that yield editing effects along one attribute while remaining other image attributes intact. Generally, these methods can be classified into the unsupervised ones [30], [33] and the supervised ones [12], [14], [15]. Unsupervised methods find the meaningful editing paths by analyzing the characteristics encoded in pretrained GANs, such as imposing constraints on latent space [33], examining the weights of a pretrained GAN [30], and analyzing activation maps [34]. Differently, supervised methods [12], [14], [15] require a large amount of semantic annotation of sampled data. They rely on learning paradigms to force their models to disentangle meaningful paths via carefully designed losses and architectures. Differently, our method stays in the middle ground between the two classes of the methods. We resort to a pretrained ScoreNet, which evaluates the maloccluding level of a teeth image, to uncover the meaningful editing paths in an easy and effective manner.

2.4 GAN Inversion

To apply meaningful edits on real images, one needs to invert given images to the latent space of a pretrained unsupervised GAN (we mainly focus on discussing StyleGAN below), which is usually termed as the GAN inversion problem[35]. Currently, GAN inversion methods can be classified into 1) optimization-based methods [16], [17], [18] and 2) learning-based methods [19], [36], [37], [38], [39], [40], [41]. The optimization-based methods find the optimal latent codes via minimizing the reconstruction loss. The learning-based inversion methods train an additional encoder that maps a given real image into the latent space. Generally, the optimization-based methods are slow at inference and perform poorly in downstream editing applications, but are better at reconstruction, while the learning-based methods are fast in inference and yield better editing quality but are limited in reconstruction. In this paper, we consider employing the optimization-based methods since the characteristics (such as teeth shape and appearance) of the original image are required to be preserved as much as possible for subsequent editing. The slow performance of the optimization-based methods is less significant compared to image quality.

3 Methodology

3.1 Overview

The overall pipeline of our method is shown in Fig. 2. Given a portrait image I of a patient with visible malpositioned teeth, our goal is to generate a portrait image I' with aligned teeth. Specifically, our approach follows four main steps: 1) data preprocessing, 2) GAN inversion, 3) latent code editing, and 4) background-foreground blending.

As shown in Fig. 3, in the data preprocessing step, we extract the mouth region I_r, a teeth image I_t, a mouth mask I_m, along with a mouth background image I_b with the help of an off-the-shelf landmark extractor [42]. We then focus on editing the teeth image I_t to obtain its aligned counterpart I_t', which will be blended with the background image I_b to obtain the aligned result.

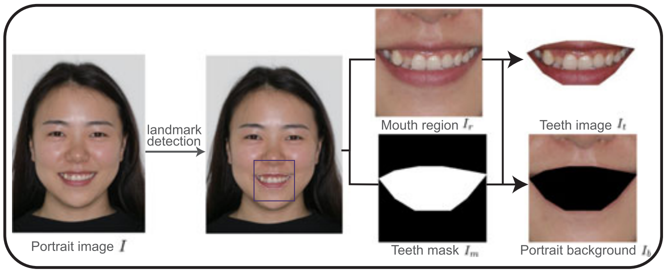


Fig. 3. Given a portrait image with visible malpositioned teeth, we first obtain the mouth landmarks and a bounding box around the mouth region using an off-of-shelf landmark predictor [42]. Then we extract the mouth mask I_m, the mouth background image I_b, and the teeth image I_t from the original portrait I.

To edit a teeth image I_t, we first invert it to the latent space of StyleGAN [9] and then we seek in the StyleGAN latent space an optimal editing path that corresponds to the "alignment" process to manipulate the latent code. The whole editing process can be formulated as

$$I_t' = G(E(V(I_t))), \tag{1}$$

where $V(\cdot)$ denotes the GAN inversion process that converts a teeth image I_t to its corresponding latent code, $E(\cdot)$ represents the latent code manipulator, and $G(\cdot)$ is the pre-trained StyleGAN generator, which outputs the manipulated teeth image according to the edited latent codes.

We find that a straightforward copy-and-paste combination of the edited I_t' and the original background I_b will introduce visible artifacts and color inconsistency at boundaries (see Fig. 9). Therefore, we propose to train a BlendingNet adversarially to blend the edited teeth image I_t' with the background image I_b and obtain I'.

In the following, for the completeness of our paper, we will first introduce the StyleGAN generator in Section 3.2. Then we will present the inverting method and the editing method in Sections 3.3 and 3.4, respectively. The details of BlendingNet will be given in Section 3.5. Moreover, we also extend the alignment effects in single images to short video clips in Section 3.6.

3.2 The StyleGAN Generator

A basic structure of the StyleGAN generator is shown in Fig. 4. Given a latent code $z \in R^{512}$ sampled from the latent space $Z \subset R^{512}$, StyleGAN [9] first transforms it into a style vector $w \in R^{512}$ using a nonlinear mapping f, i.e., $w = f(z)$. To generate a teeth image $I_t \in R^{256\times256}$, we can inject the same style vector w to n different AdaIn [43] blocks of the generator G ($n = 14$ is the number of AdaIn layers), which is highlighted as red arrows in Fig. 4. In this way, we obtain a latent space W formed by all possible style vectors $w \in R^{512}$.

Alternatively, the StyleGAN generator allows us to feed a different style vector into each layer to generate an image. To achieve so, we first extend $W \subset R^{512}$ space to a larger latent space $W^+ \subset R^{14\times512}$ by concatenating n different latent vectors sampled from W and then feed a different style vector into each AdaIn block, as highlighted as blue arrows in Fig. 4. This is termed as StyleMixing in the original paper [9].

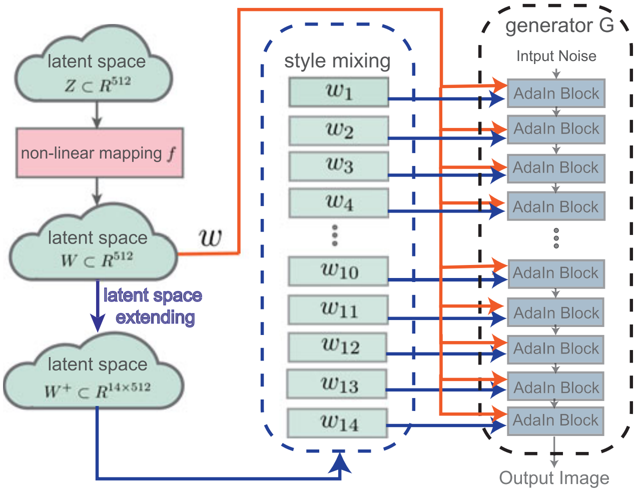


Fig. 4. The basic structure of the StyleGAN generator used in our method. Please refer to Section 3.2 for more details.

3.3 The Inverting Method

In this subsection, we consider solving the latent vectors for a given teeth image I_t. Several works [16], [18] have shown that the original W space $\subset R^{512}$ in StyleGAN is limited in reconstructing a real image. Compared to the original image, the reconstructed image often exhibits changes in structure, appearance, and moreover, lost fine-grained details. Although $W^+ \subset R^{14\times512}$ space gives a faithful reconstruction of a real image [16], the solved latent codes perform poorly when fed into the downstream editing methods [18], [44]. Following [18], we seek a balance between the editing and reconstruction quality of the inversion process by considering the P-norm$^+$ space, which is transformed from W by inverting the last LeakyReLU layer [45] in the nonlinear transformation f. We use the P-norm$^+$ space for inversion since its unit-Gaussian prior helps to regularize the inverted latent codes to lie in a valid manifold. In particular, we adopt the spherical constraint used in [18] as our baseline.

Therefore, the optimal latent codes in the P-norm$^+$ space can be solved by

$$p = argmin\|VGG(I_t) - VGG(G(T(p)))\|_1 + \lambda S(p), \tag{2}$$

where $VGG(\cdot)$ is an image feature extraction network [46], $T(\cdot)$ is a fixed transformation from the P-norm$^+$ space to the W^+ space and consists of a re-normalization layer and a LeakyRelu layer, $G(\cdot)$ is the StyleGAN generator, $S(\cdot)$ is the spherical constraint on latent codes p, and λ controls the trade-off between reconstruction and regularization. Specifically, increasing λ imposes a stronger constraint during inversion so that the reconstruction quality decreases while the editability increases (see Section 2 in the supplementary material for illustration), which can be found on the Computer Society Digital Library at http://doi.ieeecomputersociety.org/10.1109/TVCG.2022.3166159. The detailed evaluations of various latent spaces and the choice of the values of λ on our problem can be found in the supplementary material, available online.

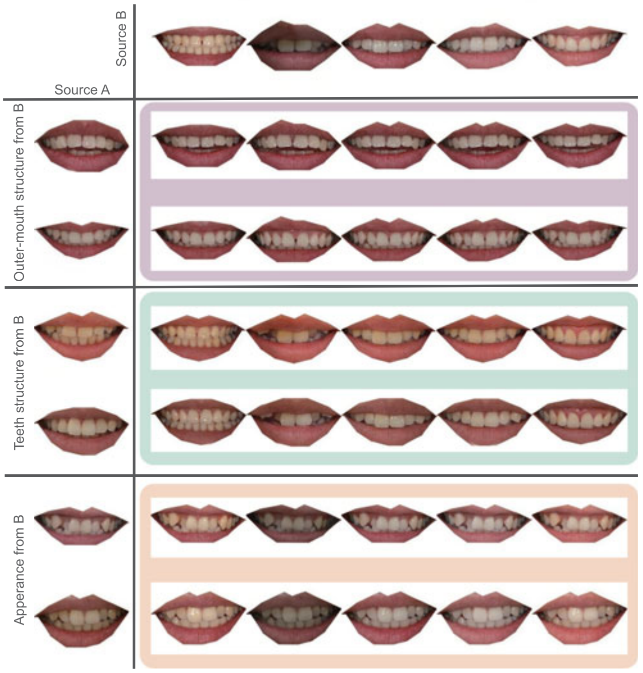


Fig. 5. We generate two sets of images (source A and source B) using their respective latent codes (The first row displays images of source B and the first column shows the images of source A). The images shown in the middle are classified into 3 subsets (arranged in purple, green and orange boxes respectively), each of which were generated by copying a subset of styles from source B and taking the rest from source A. Specifically, the images in the purple box borrow the outer-mouth codes (the latent codes of the first 4 layers) from source B; the images in the green box borrow the teeth structure codes (the latent codes of the 5th-to-9th layers) from source B; the images in the orange box borrow the appearance codes (the latent codes of the 10th-to-14th layers) from source B. We can observe that the pretrained StyleGAN generator provides a natural and spontaneous disentanlement of different image features(i.e., outer-mouth structure, teeth structure, and overall appearance).

3.4 The Editing Method

In this stage, we aim to find an editing path in the P-norm$^+$ space that transforms a malpositioned teeth image to an aligned one without changing other image features (such as the mouth shape and its overall appearance).

3.4.1 Disentangling Teeth Structure From Other Image Features

As shown in Fig. 4, the style vector w_i provides a layer-wise control on the activations of the feature maps in StyleMixing. Karras *et al.* [9] have shown that style vectors at different layers control the generation of images in a coarse-to-fine manner: the latent codes at lower layers often control the low-level information of generated objects (such as location, shape, and structure), while style vectors at higher layers control the high-level features (such as appearance and color) of generated images. Based on this precondition, we examine how the latent code at each layer influences the generation and further manually group all 14 style vectors into three categories, as shown in Fig. 6: 1) structure latent codes of outer-mouth w_o: style vectors of the first 4 layers; 2) structure latent codes of teeth w_s: style vectors from the 5th to 9th layers; and 3) the overall appearance latent codes w_a: style vectors of the last 5 layers.

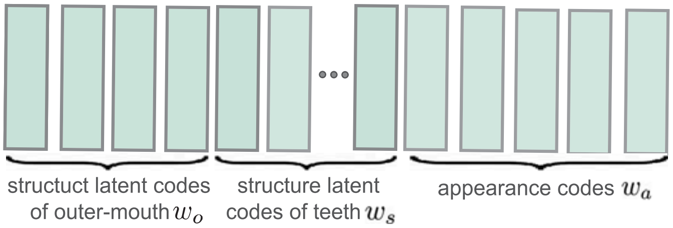

structuct latent codes of outer-mouth w_o
structure latent codes of teeth w_s
appearance codes w_a

Fig. 6. The latent code grouping criterion for image feature disentanglement.

To validate our choice, we show several mixing results in Fig. 5, where we respectively generate three subsets of images by alternatively mixing the three types of codes from source A and source B. The results show that the pretrained StyleGAN generator offers a natural, spontaneous, and fine-grained disentangling of outer-mouth structure, teeth structure, and appearance features, thus enabling us to edit the teeth structure solely while leaving the other features (i.e., the overall mouth shape and the teeth appearance) unchanged. To further validate our choice, we also provide a theoretical explanation and a rich gallery of experiment results in Section 4 of the supplementary material, available online.

Note that the relationships between the p-$norm^+$ space and the W^+ space are layer-wise. We thus can group the latent codes in the p-$norm^+$ space into the outer-mouth latent codes p_o, the teeth structure latent codes $p_s \in R^{5\times 512}$, and the appearance latent codes p_a accordingly.

Specifically, to only edit the teeth structure and keep the other image features intact, we only edit p_s in the P-norm$^+$ space while leaving p_o and p_a unchanged.

3.4.2 The Editing of Teeth Structure

In the following, we explore the teeth structure latent space to discover a specific editing path corresponding to an "alignment" transformation.

Defining the aligning transformation in a StyleGAN generator to encode complex interplay between individual teeth is nontrivial and essentially ill-posed since we have no prior knowledge about the individual tooth shape, not to mention how they transform to achieve aligned poses. One possible consideration is to handcraft a series of rules that an aligned teeth image should satisfy (e.g., the symmetry constraint). However, hard-coded rules show limited flexibility when dealing with diverse situations. Instead, we resort to a pretrained ScoreNet S, which receives a teeth image as input and outputs a scalar value (1-10) to rate its malocclusion level, to drive the exploration. By showing a large number of malpositioned teeth at different levels during training, the ScoreNet S stores the concrete visual definition of what an aligned teeth should look like implicitly. We show that minimizing the sequential malocclusion scores not only gives us the dynamic process showing how the input unaligned teeth is transformed progressively into the aligned one (see Fig. 11), but also eases the underlying optimization.

Following previous works [12], [47], we formulate the editing path as a linear transformation towards a specific direction, there concludes the editing process as

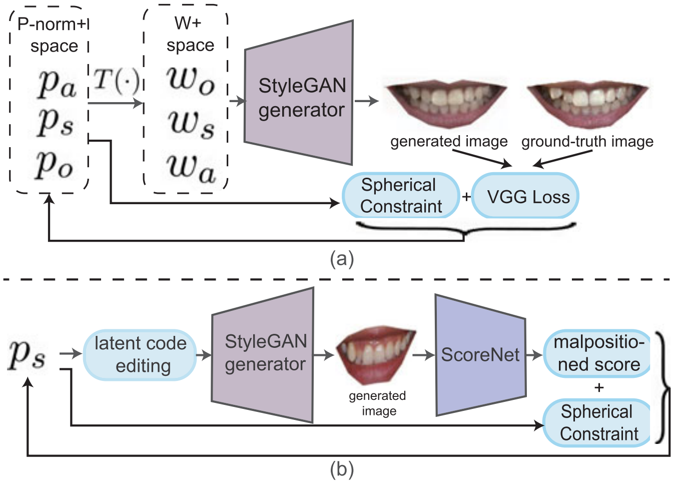


Fig. 7. The inversion process (a) and the editing process (b) of our method.

$$v = argmin\ S(G(T(concat(p_o, p_s + \beta v, p_a)))) + \alpha C(v), \tag{3}$$

where $S(\cdot)$ is the pretrained ScoreNet, β is the initial maloccluding level of I_t evaluated by ScoreNet $S(\cdot)$, and $concat(\cdot)$ represents the concatenation operation. We move the structure latent codes p_s along a specific direction v with a path length measured by β. The edited latent codes in P-norm$^+$ are then obtained by combining the edited structure latent codes $p_s + \beta v$ with the original outer-mouth codes p_o and appearance codes p_a, which are further transformed to the W^+ space using the fixed transformation T. The pretrained StyleGAN generator $G(\cdot)$ takes the edited structured latent codes in W^+ space to generate the edited teeth image I_t'. Then the optimal moving direction v can be solved by minimizing the unalignment score evaluated by S. Note that we also impose a spherical constraint, denoted as C, on the editing direction v to ensure the edited latent codes to lie in the valid manifold (the strength of spherical constraint is controlled by α). The editing process is illustrated in Fig. 7 (The details of training ScoreNet S can be found in Section 1.3 of the supplementary material, available online.)

3.5 BlendingNet

To obtain the full edited portrait image I', we need to map the generated teeth image I_t' back to the original portrait I. However, we observe noticeable boundary artifacts and color inconsistency in edited results if we simply employ a copy-and-paste strategy (see Column (c) in Fig. 9 for illustration). We anticipate that these artifacts are not caused by the editing of latent codes but are mainly introduced by the imperfect reconstruction in the inverting stage. To validate this, we visualize the results of directly copying-and-pasting the reconstructed teeth image (i.e., without any editing) into the original background I_b (see Column (b) in Fig. 9), which exhibit similar boundary artifacts with results in column (c). As shown in Fig. 8, an alternative solution of Poisson blending [48] also has difficulties in blending fine structure details and yields blurry results at boundaries.

To remove such artifacts, we introduce a self-supervised BlendingNet, which exploits adversarial learning to generate high-quality results. Specifically, the BlendingNet consists of a U-net generator G and a patch discriminator D. To train the BlendingNet, we forge training pairs, one of them containing synthesized artifacts. Specifically, for each image in the training dataset, we extract the teeth image I_t, the background image I_b, and the mouth mask I_m, as shown in Fig. 3. Then we obtain a degraded version I_t^d of I_t by eroding [49] its boundaries and adding random color jitter to it. Then the BlendingNet is trained to recover the original image I. During inference, we replace the degraded teeth image I_t^d with the edited result I_t' , and feed the image set $\{I_t', I_m, I_b\}$ to the generator so that the generator outputs a realistic blended image I'. More details about the training and architectures of BlendingNet can be found in Section 1.2 of the supplementary material, available online.

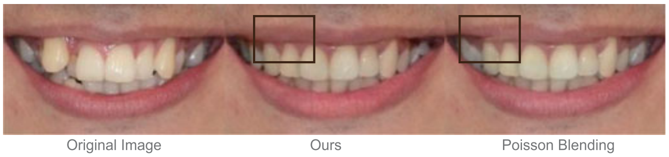


Fig. 8. A visual comparison between Poisson blending [48] and our BlendingNet. (Zoomed in for better details.)

3.6 Video Teeth Alignment

To extend our editing method to short video clips, one can run the editing process for each frame independently. However, such a naive extension does not generalize well to varying head poses since our method is built on the data collected in a frontal pose (see Fig. 17 and Section 6 for further illustration). On the other hand, the temporal incoherence issue could arise if we edit each frame independently.

To create the alignment effect for video $V = \{I_1, I_2, \dots, I_n\}$, we resort to a modified method of [50]. The general idea is to edit one frame and propagate the edits to the rest (we assume that there exists one frame I_i which is taken in a frontal pose). The end-to-end method [50] takes a source image (I_i in our case) and a reference frame (say I_j) and computes a set of corresponding keypoints, which are fed into a motion network to predict a dense warping field ϕ and an occlusion map o between the two images. The computed ϕ, o, and the source image are then fed into a

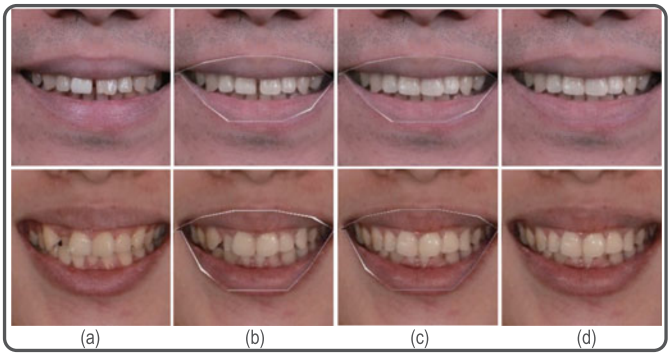


Fig. 9. Visual comparisons of our BlendingNet with the direct copy-and-paste method. (a) The original unaligned image. (b) Directly copying-and-pasting the reconstructed teeth image with the mouth background. (c) Directly copying-and-pasting the edited teeth image with the mouth background. (d) Results obtained by our BlendingNet. The comparisons show that the artifacts are introduced in the GAN inversion stage, and our BlendingNet produces high-quality composition results.

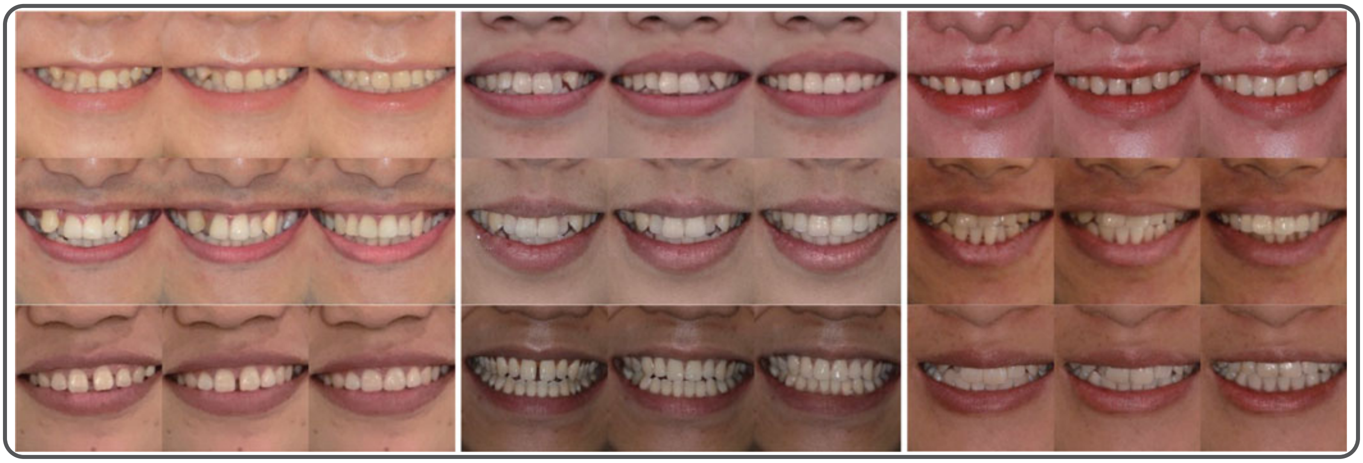


Fig. 10. A series of representative aligning results for real portrait images by our method. For each case, from left to right we show the original unaligned image, the inverted teeth image, and the edited teeth image (blended with the original background).

generator to synthesize the target image whose motion comes from the reference frame. We refer to [50] for more details.

We find that a direct deployment of their method on our portrait image leads to deficient results. First, their method fails to hallucinating textures that do not exist in the original image (e.g., when turning the head from front to left). Moreover, their method generates low-quality teeth movements since most of the detected keypoints are scattered at the whole face, not at the teeth. To solve the above problems, we make two modifications to the original pipeline. First, we only synthesize the mouth region while leaving the background untouched. Second, to make the network focus on detect the movement of teeth, we constrain their keypoint detector within the mouth mask area. We retrain the modified model on 400 video clips (containing 44,000 frames).

To evaluate the effectiveness of our improvements, we compare the original pipeline and our improved one in terms of the per-frame reconstruction loss on the test set. Specifically, our improved pipeline achieves a lower reconstruction error (0.0069) than the original approach (0.0206).

Fig. 12 shows a visual comparison. Observable artifacts (e.g., enlarged incisors, blurry background) are presented with a direct employment of [50] (Fig. 12a). On the other hand, editing each frame might cause temporally inconsistent results (Fig. 12c, highlighted with purple boxes). Our method generates higher quality and temporally more coherent results. Currently, our method does not support large head pose changes (such as turning head 90 degree to the left/right from frontal pose). For more details, please refer to Section 5 in supplementary material, available online, in which we include more results and a discussion for video teeth alignment.

4 Experiments

We conduct both qualitative and quantitative experiments to evaluate our method. We first introduce the implementation details of our method in Section 4.1. Qualitative and quantitative comparisons with other methods are presented in Sections 4.2 and 4.3 respectively.

4.1 Implementation Details

Our method contains several different components. Due to the limited paper length, we only provide the details of the used dataset, the inversion process, and the editing method here. The training details of our StyleGAN Generator, BlendingNet, and ScoreNet can be found in Section 1 of the supplementary material, available online.

Dataset. We obtain a large dataset consisting of 6,000 pretreatment patient portraits from a dental company. Each portrait image was taken in the front head pose, with the malpositioned teeth visible. To model the manifold of aligned teeth images, we additionally collect a set of 1,000 images with their teeth aligned using the method of [5] (We pick those high-quality results without artifacts). We refer to the whole dataset data as T.

The Inverting and Editing Processes. Both the inverting process and the editing process are formulated as iterative optimization. We use Adam optimizer [51] with a learning rate of 0.01 for both steps. We set the maximum iteration number for the inverting process as 1,000 and 200 for editing. Specifically, we set the parameter λ in Eq. (2) as 0.0002 and

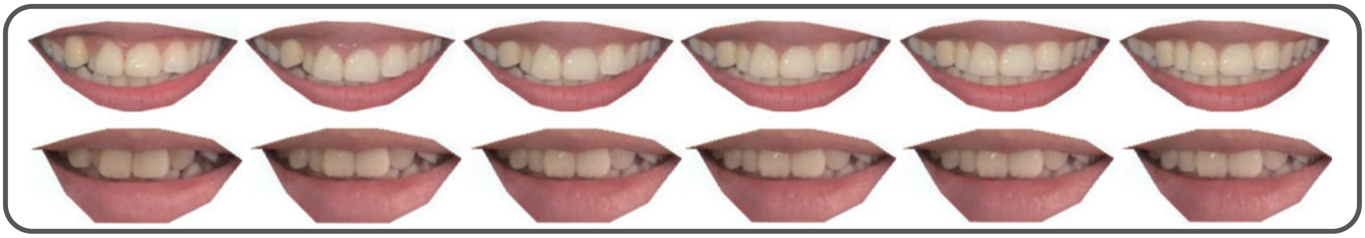


Fig. 11. The iterative optimization enables us to progressively visualize how the given unaligned teeth images (Leftmost) evolve into the aligned ones (rightmost).

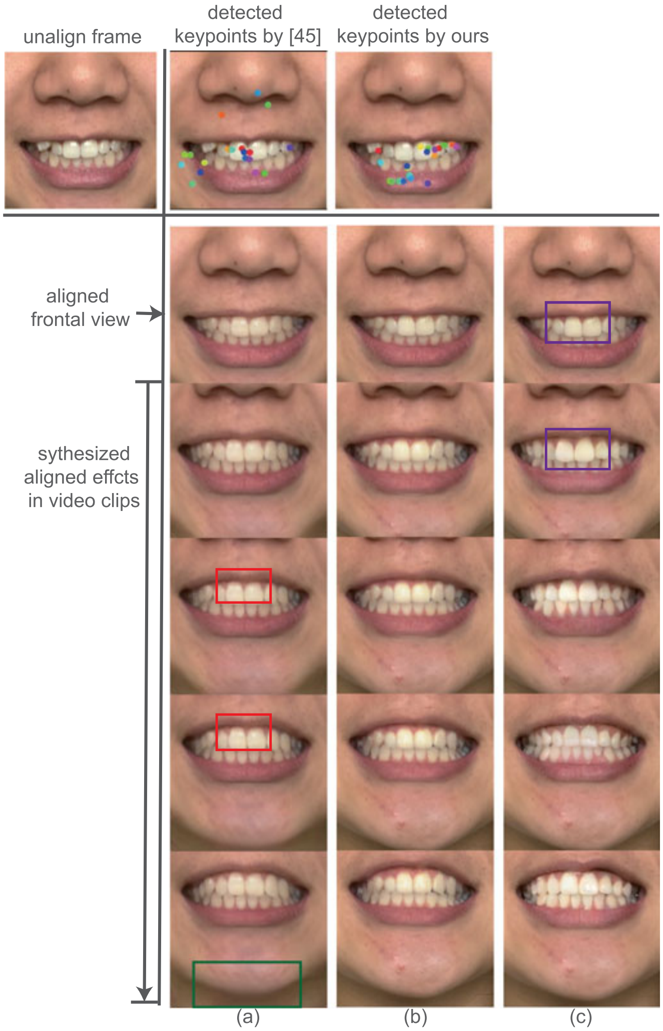

unalign frame
detected keypoints by [45]
detected keypoints by ours
aligned frontal view
sythesized aligned effcts in video clips
(a)
(b)
(c)

Fig. 12. An example of video teeth alignment. The original unaligned frame is shown in the first row. (a) the alignment results by [50] (b) the alignment results by our improved method upon [50]; (c) the alignment results by running our single-frame alignment method frame-by-frame. We highlight the artifacts, i.e., the blurriness of background, the distorted incisors and temporal incoherence in (a) and (c) with red, green and purple bounding boxes respectively.

the parameter α in Eq. (3) as 0.002. Both steps are implemented using PyTorch with one NVIDIA RTX 3090ti (24 GB memory).

4.2 Qualitative Results

Fig. 10 shows diverse aligned results using our method. For each case, we show the original image, the inverted teeth image blended with the original background, and the edited teeth image blended with the original background. For easy comparison between the teeth before and after alignment, we only show the edited areas around the mouth. We observe that for most cases, our inversion method produces a faithful reconstruction of the original images. Our editing method further transforms the inverted unaligned teeth images to their aligned counterparts with high fidelity. The synthesis results preserve not only the low-level semantics of the original teeth images (such as the total number of the visible teeth, the shape and location of each individual tooth, and the shape of the outer mouth shape), but also the high-frequency details (such as appearance, textures, and highlight). For more qualitative results in diverse cases, please refer to the Section 4 in the supplementary material, available online.

We also show the holistic aligned effect by our method on one's facial appearance in Fig. 1, from which we can observe that the edited area is consistent with the whole facial appearance. Moreover, the iterative formulation of our editing process allows us to visualize how a set of unaligned teeth gradually evolve into the aligned ones in Fig. 11. Our editing method gives each individual tooth a different transformation so that these transformations form convinced intermediate results. For example, for case in the first row, our editing method pushes the tooth highlighted with a black bounding box from outside to inside and pulls tooth highlighted with a red bounding box in an opposite way while keeping the other teeth intact.

4.3 Comparisons

In this subsection, we present a series of qualitative and quantitative comparisons to evaluate the superiority of our full model. Specifically, we compare our approach with the 1) state-of-the-art image-based teeth alignment method: iOrthoPredictor [5]; 2) the deep-learning based image translation methods [52], [53], [54]; 3) other image manipulation methods built on manipulating the latent space of StyleGAN [30], [55].

For deep image translation methods, we only consider the ones developed under unpaired settings due to the absence of the ground-truth training pairs (i.e., pre-orthodontic treatment portraits and their post-orthodontic counterparts). Specifically, we choose Cycle-GAN [52], contrastive learning for unpaired image-to-image translation [53], and HiSD [54]. Cycle-GAN [52] is a popular image-to-image translation framework that transforms images of different domains via a cycle-consistency loss. More recently, contrastive learning [53] has been introduced for unsupervised domain translation. HiSD [54] is the state-of-the-art image-to-image translation method that unifies both multi-style and multi-modal image translation. To adapt these methods for teeth alignment, we manually classify all teeth images in datatset T into two domains: images of low malpositioned level (i.e., the malpostioned score from 1 to 5) and images of high malpositioned level (i.e., the malpostioned score from 6 to 10), and retrain these methods with their public codes with the default training settings.

We also compare our latent editing strategy with alternative latent editing methods [30], [55], which also built upon StyleGAN. There are extensive editing schemes [14], [15], [30], [55] along this line. We choose InterFaceGAN [55] and Sefa [30] for their generality since other methods [14], [15] are tailored for face editing and require domain-specific models (such as the 3D morphable face models (3DMM) [56] and pretrained face recognition models [15]).

4.3.1 Comparison With iOrthoPredictor

Visual comparisons with iOrthoPredictor [5] are shown in Fig. 13, from which we can observe that guided by ground-truth 3D teeth models, iOrthoPredictor generates high-quality teeth alignment results in inner mouth areas. However, as claimed in their paper, their method has difficulties in synthesizing the areas between gums and lips since these

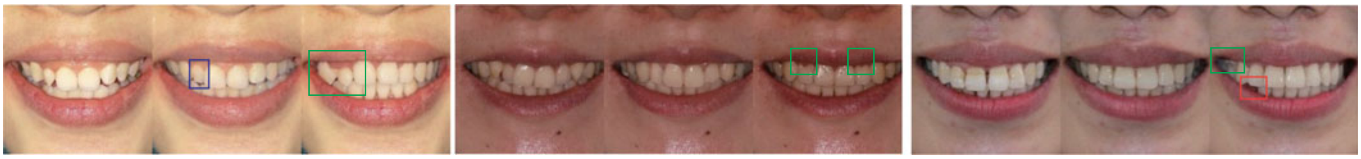


Fig. 13. Comparison with Yang *et al.* [5]. For each case, from left to right are the original image, the aligned result by our method and the aligned result by Yang *et al.* [5].

areas are not modeled by the 3D teeth models (such as the areas highlighted by green bounding boxes in Fig. 13). Moreover, due to the imperfect scanning of 3D teeth models, their method also results in missing teeth in certain cases (such as areas highlighted by red bounding box in case (c)). In contrast, our method does not rely on additional input and generates high quality details.

We also provide the quantitative comparisons with iOrthoPredictor. For each method, we generate 1,000 examples and compute the Fréchet inception distance [57] (FID) of generated images against the whole dataset (excluding those synthetic results generated by iOrthoPredictor). As shown in Table 1, we obtain lower FID score than iOrthoPredictor [5], indicating the superiority in the realism of our results.

One possible weakness of our method compared with iOrthoPredictor is that the shape of some teeth in the images generated by our method might be changed slightly (see the tooth highlighted by a blue bounding box in case (a) in Fig. 13.) Although we try to preserve the characteristics of the original teeth as much as possible, such an artifact might still happen since we cannot guarantee the transformation for each individual tooth to be rigid in our formulation. However, we argue that, for portrait image beautification, this problem is less severe since it is more important to build a method that can be used by average users. For more discussion about the applications of our method, please refer to Section 7.

4.3.2 Comparisons With Image Translation Methods

We provide several visual comparisons in Fig. 14. From this figure, we can observe that 1) Cycle-GAN [52] and contrastive learning [53] have difficulties in capturing complex geometric transformations, thus failing to transform images from unaligned teeth to aligned ones; 2) HiSD [54], which is designed for transferring images between different structures (such as adding bangs or glasses to portrait images), is able to generate alignment effects to some extent. However, their method cannot disentangle high-level appearance from low-level structures, thus still causing both structural and color distortions (such as the case in the 2nd row). Moreover, HiSD generates blurry textures at teeth boundaries, as highlighted in black bounding boxs in Fig. 14. On the contrary, our method is capable of generating aligned results of high quality and preserves the characteristics in the original images better. Quantitative comparisons, as shown in Table 1, also validate the superiority of our method.

TABLE 1
Quantitative Comparisons With Other Methods

	iOrthoPredictor [5]	Cycle-GAN	Contrastive Learning	HiSD	InterFace-GAN	Sefa	Ours
FID	9.97	11.65	10.67	13.82	**8.92**	10.84	9.19

4.3.3 Comparisons With Different Latent Manipulation Strategies

Both InterFaceGAN [12] and Sefa [58] methods analyze the latent space of an unsupervised GAN and aim to uncover the semantic meaningful editing paths. Their difference lies in that InterfaceGAN trains a linear support vector machine (SVM) to identify a desirable editing path while Sefa achieves such a goal in a purely unsupervised manner by decomposing the weights of a pretrained GAN. However, one shared drawback of these two methods is that they only identify the editing path while leaving the edit length (i.e., the extent of edit along the path) unresolved. Thus these methods rely on a manual decision on the edit length. Our method, in contrast, determines the edit length automatically. In comparison, we manually set the edit length of their methods to be a reasonable fixed value for simplicity and for a fair comparison with these alternative editing strategies, we reshape their methods to fit them tightly into our pipeline.

To adapt InterfaceGAN for teeth alignment, we first generate 2,000 images using our StyleGAN generator. Both the generated images and their corresponding latent codes in the W space are recorded. The generated images are further classified into aligned images and unaligned images. A linear SVM is trained to identify the ideal hyper-plane that separates aligned and unaligned latent codes. Like our method, we force their method to edit the teeth structure latent codes while leaving the latent codes of other layers intact. For Sefa, we decompose the weights of teeth structure layers and choose the direction that causes the most significant variation as the editing path.

Visual comparisons between our method and InterFaceGAN as well as Sefa are shown in Fig. 14. Generally, InterFaceGAN is able to generate certain alignment effects on images. However, it causes some undesirable artifacts. For example, in the 1st row, the result of InterFaceGAN exhibits distortion artifacts. For Sefa, we surprisingly find that the editing path obtained in such an unsupervised way can generate some aligned effects on images, but limited in some cases (such as the case in the 2nd row). In general, our method generates more convincingly aligned results and is better at preserving the characteristics of the original teeth images during transformation.

Quantitative comparisons between our method and InterFaceGAN as well as Sefa are shown in Table 1. It is

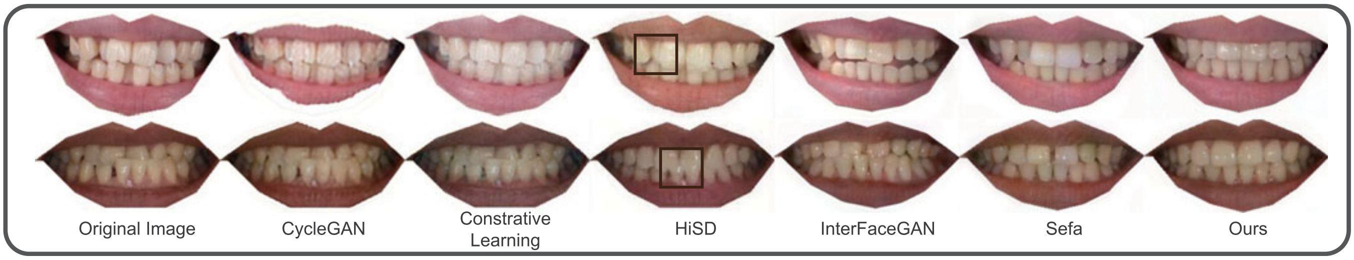


Fig. 14. Several visual comparisons between different methods. For each case, from left to right are the original image, the results generated by Cycle-GAN [52], Contrastive learning [53], HiSD [54], InterFaceGAN [47], Sefa [30], and our method.

worth noticing that the FID score of our method is slightly higher than that of InterFaceGAN. It is because that InterFaceGAN only identifies the editing path, while our method carefully sets the edit length to ensure the high-quality of edited images for fair comparison. Although InterFaceGAN generates images of higher quality, their method produces limited alignment effect as shown in Fig. 14.

5 Perceptive Study

We also evaluate the quality of the results by our method using a web-based perceptive study by two groups of human viewers: participants with no knowledge in dentistry (group 1) and professional orthodontists (group 2). To do this, we first randomly select a subset Ω of 15 real images from the test set and generate a set of 15 portrait images from Ω using our method and a set of 15 portrait images using iOrthoPredictor [5]. We further collect a set of 15 real portrait images with well-aligned teeth. All the images excluding those in Ω (so that all teeth appear to be aligned) are used for user evaluation. Here we only compare to iOrthoPredictor since it is the SOTA method that can generate high-quality results closest to ours.

We shuffle these images randomly and present them to 38 invited volunteers (group 1). Most of the volunteers are university students with no knowledge in dentistry. For each image presented, each volunteer was asked to classify it into either “real” or “fake” category (they were asked to focus more on the teeth area). For each image set, we calculate the average percentage of its images being classified to “real”. The detailed results are shown in Table 2. It can be observed that 1) both the iOrthoPredictor and our method achieve scores close to the real image set, indicating that the two methods are capable of generating realistic images; 2) in general, our method generates more realistic results than iOrthoPredictor.

We also invite 11 professional orthodontists (group 2) to do the same test. Table 2 shows that even professional orthodontists find it hard to distinguish between our generated images with the real images, again proving the effectiveness of our method. Moreover, compared to the statistics in group 1, our method achieves a higher score than iOrthoPredictor at a larger margin, indicting that the professional orthodontists are more likely to perceive the small unnaturalness produced by iOrthoPredictor (such as the artifacts illustrated in Fig. 13) than inexperienced university students.

TABLE 2
Statistics of Our User Study

	Real images	iOrthoPredictor [5]	Ours
AP (group 1)	73.86%	68.07%	71.23%
AP (group 2)	75.09%	67.64%	74.30%
Average	74.47 %	67.85 %	72.76%

Here we compute the average percentage (AP) for both groups (i.e., group 1: participants with no knowledge in dentistry; group 2: professional orthodontists).

5.1 Evaluation

In this section, we evaluate the effectiveness of the design of our method.

5.1.1 The Design of Our Editing Method

Here we evaluate the influence of the spherical constraint during the editing stage. We show several visual comparisons between the edited results of with and without the spherical constraint in Fig. 15. It can be easily observed that the editing method fails to generate high-quality results without the spherical constraint in some cases. This is mainly due to that, without this constraint, the optimization process in the editing stage purely seeks a solution that minimizes the malpositioned score so that the solved latent codes no longer guarantee the visual plausibility of generated results. In contrast, the spherical constraint in our method limits the size of the valid latent space and ensures the high-quality of the generated images.

5.1.2 Effectiveness of Multi-Score Optimization

We also show that our method benefits from using the multi-score optimization strategy than a two-score optimization. Specifically, we label the teeth images with malpositioned scores of 1 to 5 as the ”align” data and 6-10 as “unalign” data, and retrain the ScoreNet for two-class scoring. We observe that a two-score optimization could sometimes get stuck at local optima and fail to align the teeth, as shown in Fig. 16. On the contrary, multi-score optimization can greatly help the optimizer in generating more aligned effects.

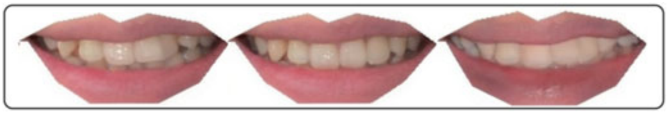


Fig. 15. A visual comparison between the edited results with (middle) and without (right) the spherical constraint.

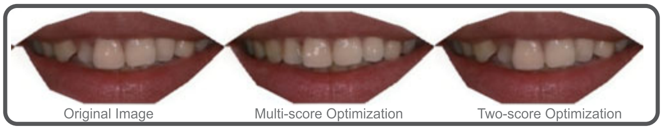


Fig. 16. A visual comparison between the multi-score optimization and two-score optimization.

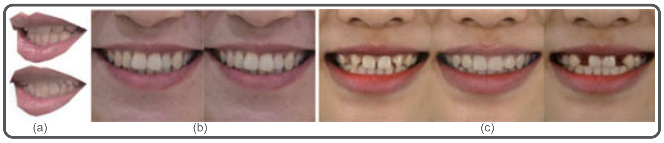


Fig. 17. Illustrations of several limitations of our method. (a) Top: the original image, Bottom: the reconstructed image by inversion. (b) Left: the image that needs to be edited; Right: the edited result. (c) From left to the right are: the original image, the edited result by our method, and the edited result by iOrthoPredictor [5].

6 Limitations

Here we discuss several limitations of our method. First, our method assumes that the portrait images of patients are taken in the frontal pose, it generalizes poorly to poses that far from this setting (see Fig. 17a) for an illustration).

Second, since both the inversion stage and the editing stage are formulated as iterative optimization, our method does not run instantly. Tested on an RTX 3090ti, our method takes about 5 seconds to finish the editing of one image on average.

Third, since our method relies on a ScoreNet to guide the editing, the performance of our method degrades in wrong classification situations. For example, in Fig. 17b, the teeth image is mis-classified as "aligned" (i.e., with score 10) so that the ScoreNet does not provide useful signals for optimization. We believe that this phenomenon can be relieved if the ScoreNet is trained on a larger dataset with higher diversity.

Fourth, our method may deviate from real orthodontic treatment. In real orthodontics, many cases will need to pull out patients' teeth for a suitable alignment, however, our method always tends to inpaint the missing teeth. Fig. 17c shows an example, whereas the actual orthodontic planning will leaving a missing tooth for in-plant. On the contrary, iOrthoPredictor [5] is capable of handling such cases since they have the 3D aligned teeth as guidance.

Fifth, similar to iOrthoPredictor [5], our approach only considers the synthesis of the mouth region while leaving the other areas (such as the contour profile of face, the relative location of upper and lower jaw, and the changes of soft-tissues muscles) untouched. In fact, orthodontic treatment in one's early age has a more profound influence on facial appearance (such as the growth of maxilla and mandible, and the improvements on the occluding relationship between the upper and lower jaws) than simply aligning the teeth. More factors should be modeled in future works.

7 Conclusion and Discussion

In this paper, we introduced the first model-free method that generates the visual outcome of orthodontic treatment for a portrait image.

Our key innovation is that we formulate the image translation problem from malpositioned teeth to aligned teeth as a latent space exploration problem, where we first model the teeth image manifold with the state-of-the-art unsupervised GAN (i.e., StyleGAN in our paper) and find the geometrically meaningful editing path that corresponds to "alignment" in its latent space. To achieve so, we first disentangle the teeth structure from other image features and guide the editing using a ScireNet.

Different from iOrthoPredictor [5], which requires a corresponding 3D teeth model as input, our method does not rely on any additional 3D input, making our method accessible for commodity users. Here we discuss some real-world applications of our method. First, our method can be used as a built-in tool in portrait retouching software. Smiling portraits widely exist in our daily life. However, aligning one's teeth in portrait images requires complex local edits even for professional users with traditional image retouching tools like Photoshop. Our method serves as a strong and useful backbone in such a situation since our method automatically generates high-quality teeth alignment results and does not need any human intervention. In such scenarios, the real post-treatment effects are not important. On the other hand, for orthodontic treatment, although iOrthoPredictor can achieve more correct alignment results under the strong guidance of a 3D teeth model, their results are still not orthodontically accurate, as indicated in their paper [5] due to the complex process of real orthodontic treatment. Along this vain, we believe a lightweight tool that releases the need for a 3D scanner and professional preprocessing is more user-friendly and can quickly help end users to perceive the effects and build their confidence in taking real orthodontic treatment.

References

[1] T. Li *et al.*, "BeautyGAN: Instance-level facial makeup transfer with deep generative adversarial network," in *Proc. 26th ACM Int. Conf. Multimedia*, 2018, pp. 645–653.

[2] H.-J. Chen, K.-M. Hui, S.-Y. Wang, L.-W. Tsao, H.-H. Shuai, and W.-H. Cheng, "BeautyGlow: On-demand makeup transfer framework with reversible generative network," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 10 042–10 050.

[3] Q. Gu, G. Wang, M. T. Chiu, Y.-W. Tai, and C.-K. Tang, "LADN: Local adversarial disentangling network for facial makeup and de-makeup," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2019, pp. 10 481–10 490.

[4] S. Liu *et al.*, "PSGAN++: Robust detail-preserving makeup transfer and removal," *IEEE Trans. Pattern Anal. Mach. Intell.*, to be published, doi: 10.1109/TPAMI.2021.3083484.

[5] L. Yang *et al.*, "iOrthoPredictor: Model-guided deep prediction of teeth alignment," *ACM Trans. Graph.*, vol. 39, no. 6, pp. 1–15, 2020.

[6] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," 2013, *arXiv:1312.6114*.

[7] A. Brock, J. Donahue, and K. Simonyan, "Large scale GAN training for high fidelity natural image synthesis," 2018, *arXiv: 1809.11096*.

[8] T. Karras, T. Aila, S. Laine, and J. Lehtinen, "Progressive growing of GANs for improved quality, stability, and variation," 2017, *arXiv:1710.10196*.

[9] T. Karras, S. Laine, and T. Aila, "A style-based generator architecture for generative adversarial networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 4401–4410.

[10] T. Karras, S. Laine, M. Aittala, J. Hellsten, J. Lehtinen, and T. Aila, "Analyzing and improving the image quality of StyleGAN," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 8110–8119.

[11] X. Pan, B. Dai, Z. Liu, C. C. Loy, and P. Luo, “Do 2D GANs know 3D shape? Unsupervised 3D shape reconstruction from 2D image GANs,” 2020, *arXiv:2011.00844*.

[12] Y. Shen, J. Gu, X. Tang, and B. Zhou, “Interpreting the latent space of GANs for semantic face editing,” in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 9243–9252.

[13] R. Abdal, P. Zhu, N. J. Mitra, and P. Wonka, “StyleFlow: Attribute-conditioned exploration of StyleGAN-generated images using conditional continuous normalizing flows,” *ACM Trans. Graph.*, vol. 40, no. 3, pp. 1–21, 2021.

[14] A. Tewari *et al.*, “StyleRig: Rigging StyleGAN for 3D control over portrait images,” in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 6142–6151.

[15] A. Tewari *et al.*, “PIE: Portrait image embedding for semantic control,” *ACM Trans. Graph.*, vol. 39, no. 6, pp. 1–14, 2020.

[16] R. Abdal, Y. Qin, and P. Wonka, “Image2StyleGAN: How to embed images into the StyleGAN latent space?,” in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2019, pp. 4432–4441.

[17] R. Abdal, Y. Qin, and P. Wonka, “Image2StyleGAN++: How to edit the embedded images?,” in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 8296–8305.

[18] P. Zhu, R. Abdal, Y. Qin, J. Femiani, and P. Wonka, “Improved StyleGAN embedding: Where are the good latents?,” 2020, *arXiv:2012.09036*.

[19] O. Tov, Y. Alaluf, Y. Nitzan, O. Patashnik, and D. Cohen-Or, “Designing an encoder for StyleGAN image manipulation,” 2021, *arXiv:2102.02766*.

[20] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros, “Image-to-image translation with conditional adversarial networks,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 1125–1134.

[21] A. Almahairi, S. Rajeshwar, A. Sordoni, P. Bachman, and A. Courville, “Augmented CycleGAN: Learning many-to-many mappings from unpaired data,” in *Proc. 35th Int. Conf. Mach. Learn.*, 2018, pp. 195–204.

[22] Z. Yi, H. Zhang, P. Tan, and M. Gong, “DualGAN: Unsupervised dual learning for image-to-image translation,” in *Proc. IEEE Int. Conf. Comput. Vis.*, 2017, pp. 2849–2857.

[23] M.-Y. Liu, T. Breuel, and J. Kautz, “Unsupervised image-to-image translation networks,” 2017, *arXiv:1703.00848*.

[24] Y. Taigman, A. Polyak, and L. Wolf, “Unsupervised cross-domain image generation,” 2016, *arXiv:1611.02200*.

[25] Y. Choi, M. Choi, M. Kim, J.-W. Ha, S. Kim, and J. Choo, “StarGAN: Unified generative adversarial networks for multi-domain image-to-image translation,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 8789–8797.

[26] Z. He, W. Zuo, M. Kan, S. Shan, and X. Chen, “AttGAN: Facial attribute editing by only changing what you want,” *IEEE Trans. Image Process.*, vol. 28, no. 11, pp. 5464–5478, Nov. 2019.

[27] M. Liu *et al.*, “StGAN: A unified selective transfer network for arbitrary image attribute editing,” in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 3673–3682.

[28] X. Huang, M.-Y. Liu, S. Belongie, and J. Kautz, “Multimodal unsupervised image-to-image translation,” in *Proc. Eur. Conf. Comput. Vis.*, 2018, pp. 172–189.

[29] H.-Y. Lee, H.-Y. Tseng, J.-B. Huang, M. Singh, and M.-H. Yang, “Diverse image-to-image translation via disentangled representations,” in *Proc. Eur. Conf. Comput. Vis.*, 2018, pp. 35–51.

[30] Y. Shen and B. Zhou, “Closed-form factorization of latent semantics in GANs,” 2020, *arXiv:2007.06600*.

[31] L. Goetschalckx, A. Andonian, A. Oliva, and P. Isola, “GANalyze: Toward visual definitions of cognitive image properties,” in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2019, pp. 5744–5753.

[32] A. Jahanian, L. Chai, and P. Isola, “On the “steerability” of generative adversarial networks,” 2019, *arXiv:1907.07171*.

[33] X. Chen, Y. Duan, R. Houthooft, J. Schulman, I. Sutskever, and P. Abbeel, “InfoGAN: Interpretable representation learning by information maximizing generative adversarial nets,” 2016, *arXiv:1606.03657*.

[34] E. Collins, R. Bala, B. Price, and S. Susstrunk, “Editing in style: Uncovering the local semantics of GANs,” in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 5771–5780.

[35] W. Xia, Y. Zhang, Y. Yang, J.-H. Xue, B. Zhou, and M.-H. Yang, “GAN inversion: A survey,” 2021, *arXiv:2101.05278*.

[36] J.-Y. Zhu, P. Krähenbühl, E. Shechtman, and A. A. Efros, “Generative visual manipulation on the natural image manifold,” in *Proc. Eur. Conf. Comput. Vis.*, 2016, pp. 597–613.

[37] S. Guan, Y. Tai, B. Ni, F. Zhu, F. Huang, and X. Yang, “Collaborative learning for faster StyleGAN embedding,” 2020, *arXiv:2007.01758*.

[38] E. Richardson *et al.*, “Encoding in style: A StyleGAN encoder for image-to-image translation,” 2020, *arXiv:2008.00951*.

[39] G. Perarnau, J. Van De Weijer, B. Raducanu, and J. M. Álvarez, “Invertible conditional GANs for image editing,” 2016, *arXiv:1611.06355*.

[40] Y. Alaluf, O. Patashnik, and D. Cohen-Or , “Only a matter of style: Age transformation using a style-based regression model,” 2021, *arXiv:2102.02754*.

[41] Y.-D. Lu, H.-Y. Lee, H.-Y. Tseng, and M.-H. Yang, “Unsupervised discovery of disentangledmanifolds in GANs,” 2020, *arXiv:2011.11842*.

[42] A. Bulat and G. Tzimiropoulos, “How far are we from solving the 2D & 3D face alignment problem? (and a dataset of 230,000 3D facial landmarks),” in *Proc. IEEE Int. Conf. Comput. Vis.*, 2017, pp. 1021–1030.

[43] X. Huang and S. Belongie, “Arbitrary style transfer in real-time with adaptive instance normalization,” in *Proc. IEEE Int. Conf. Comput. Vis.*, 2017, pp. 1501–1510.

[44] A. Tewari, M. Elgharib, B. Mallikarjun, P. Patrick, M. Zollhöfer, and T. Christian, “Pie: Portrait image embedding for semantic control,” *ACM Trans. Graph.*, vol. 39, no. 6, pp. 1–14, 2022.

[45] A. L. Maas *et al.*, “Rectifier nonlinearities improve neural network acoustic models,” in *Proc. Int. Conf. Mach. Learn.*, 2013, Art. no. 3.

[46] K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” 2014, *arXiv:1409.1556*.

[47] Y. Shen, C. Yang, X. Tang, and B. Zhou, “InterFaceGAN: Interpreting the disentangled face representation learned by GANs,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 4, pp. 2004–2018, Apr. 2022.

[48] P. Pérez, M. Gangnet, and A. Blake, “Poisson image editing,” in *Proc. ACM SIGGRAPH Papers*, 2003, pp. 313–318.

[49] J. Serra and P. Soille, *Mathematical Morphology and its Applications to Image Processing*, vol. 2. Berlin, Germany: Springer Science & Business Media, 2012.

[50] A. Siarohin, S. Lathuilière, S. Tulyakov, E. Ricci, and N. Sebe, “First order motion model for image animation,” in *Proc. 33rd Int. Conf. Neural Inf. Process. Syst.*, 2019, pp. 7137–7147.

[51] D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” 2014, *arXiv:1412.6980*.

[52] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros, “Unpaired image-to-image translation using cycle-consistent adversarial networks,” in *Proc. IEEE Int. Conf. Comput. Vis.*, 2017, pp. 2223–2232.

[53] T. Park, A. A. Efros, R. Zhang, and J.-Y. Zhu, “Contrastive learning for unpaired image-to-image translation,” in *Proc. Eur. Conf. Comput. Vis.*, 2020, pp. 319–345.

[54] X. Li *et al.*, “Image-to-image translation via hierarchical style disentanglement,” in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 8639–8648.

[55] Y. Shen, P. Luo, J. Yan, X. Wang, and X. Tang, “FaceID-GAN: Learning a symmetry three-player GAN for identity-preserving face synthesis,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 821–830.

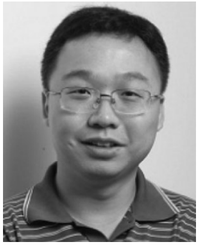
[56] B. Egger *et al.*, “3D morphable face models–past, present, and future,” *ACM Trans. Graph.*, vol. 39, no. 5, pp. 1–38, 2020.

[57] M. Heusel, H. Ramsauer, T. Unterthiner, B. Nessler, and S. Hochreiter, “GANs trained by a two time-scale update rule converge to a local nash equilibrium,” in *Proc. 31st Int. Conf. Neural Inf. Process. Syst.*, 2017, pp. 6629–6640.

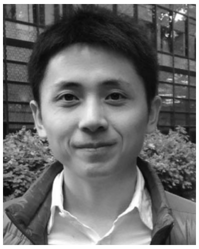
[58] Y. Shen and B. Zhou, “Closed-form factorization of latent semantics in GANs,” in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 1532–1540.

Beijia Chen received the BS and MS degrees from the Nanjing University of Science and Technology, China. She is currently working toward the PhD degree with the State Key Lab of CAD&CG, Zhejiang University, China. Her research interests include 3D human recovery, image manipulation, and deep learning.

Hongbo Fu received the BS degree in information sciences from Peking University, China, in 2002, and the PhD degree in computer science from the Hong Kong University of Science and Technology, Hong Kong, in 2007. He is a full professor with the School of Creative Media, City University of Hong Kong, Hong Kong. His primary research interests fall in the fields of computer graphics and human computer interaction. He has served as an associate editor of *The Visual Computer*, *Computers & Graphics*, and *Computer Graphics Forum*.



Kun Zhou (Fellow, IEEE) received the BS and PhD degrees in computer science from Zhejiang University, China, in 1997 and 2002, respectively. He is a Cheung Kong professor with the Computer Science Department, Zhejiang University, China. His research interests are in visual computing, parallel computing, human computer interaction, and virtual reality. He currently serves on the editorial advisory boards of *ACM Transactions on Graphics* and *IEEE Spectrum*.



Youyi Zheng received the BS and MS degrees in mathematics, both from Zhejiang University, China, in 2005 and 2007, respectively, and the PhD degree in computer science from the Hong Kong University of Science & Technology, Hong Kong, in 2011. He is currently a researcher with the State Key Lab of CAD&CG, Zhejiang University, China. His research interests include geometric modeling, imaging, and human-computer interaction. He has served as an associate editor of *The Visual Computer* and *Frontiers of Computer Science*.