ORIGINAL ARTICLE

Building anatomically realistic jaw kinematics model from data

Wenwu Yang[1] · Nathan Marshak[2] · Daniel Sýkora[3] · Srikumar Ramalingam[2] · Ladislav Kavan[2]

Abstract

Recent work on anatomical face modeling focuses mainly on facial muscles and their activation. This paper considers a different aspect of anatomical face modeling: kinematic modeling of the jaw, i.e., the temporomandibular joint (TMJ). Previous work often relies on simple models of jaw kinematics, even though the actual physiological behavior of the TMJ is quite complex, allowing not only for mouth opening, but also for some amount of sideways (lateral) and front-to-back (protrusion) motions. Fortuitously, the TMJ is the only joint whose kinematics can be accurately measured with optical methods, because the bones of the lower and upper jaw are rigidly connected to the lower and upper teeth. We construct a person-specific jaw kinematic model by asking an actor to exercise the entire range of motion of the jaw while keeping the lips open so that the teeth are at least partially visible. This performance is recorded with three calibrated cameras. We obtain highly accurate 3D models of the teeth with a standard dental scanner and use these models to reconstruct the rigid body trajectories of the teeth from the videos (markerless tracking). The relative rigid transformations samples between the lower and upper teeth are mapped to the Lie algebra of rigid body motions in order to linearize the rotational motion. Our main contribution is to fit these samples with a three-dimensional nonlinear model parameterizing the entire range of motion of the TMJ. We show that standard principal component analysis (PCA) fails to capture the nonlinear trajectories of the moving mandible. However, we found these nonlinearities can be captured with a special modification of autoencoder neural networks known as nonlinear PCA. By mapping back to the Lie group of rigid transformations, we obtain a parametrization of the jaw kinematics which provides an intuitive interface allowing the animators to explore realistic jaw motions in a user-friendly way.

Keywords Jaw kinematics · Teeth motion · TMJ

✉ Wenwu Yang
wwyang@mail.zjgsu.edu.cn

Nathan Marshak
tauneutrino3@gmail.com

Daniel Sýkora
sykorad@fel.cvut.cz

Srikumar Ramalingam
srikumar.ramalingam@gmail.com

Ladislav Kavan
ladislav.kavan@gmail.com

[1] School of Computer and Information Engineering, Zhejiang Gongshang University, Hangzhou 310012, China

[2] University of Utah, Salt Lake City, Utah, USA

1 Introduction

Anatomical modeling of the face has been explored in the pioneering work of [32,35], but recent years witnessed a resurgence of interest in anatomically based facial animation [5,15,19,21]. Naturally, the primary focus is accurate modeling of facial muscles and their ability to generate facial expressions. However, the shape and expressions of the face are significantly affected by two major bones: the skull and the mandible (lower jaw). This is evidenced by people who underwent jaw surgery, which is a relatively frequent surgery to correct congenital malformations such as the overbite. We argue that realistic anatomically based facial animation needs to start with an accurate jaw kinematics model. With physics-based simulation of facial soft tissues, the relative rigid transformation between the skull and the mandible has

[3] Faculty of Electrical Engineering, Czech Technical University in Prague, Prague, Czech Republic

a significant effect on the result, because the bones are used as Dirichlet boundary conditions. Even though, strictly speaking, the bones' attachment to the teeth is elastic, in normal physiological motions these deformations are negligible and we can safely assume that hard tissues behave as rigid bodies. However, the rigid motion of the mandible relative to the skull is not arbitrary, but is constrained by the anatomy of the temporomandibular joint (TMJ). The TMJ is a very complicated joint and enables functions such as chewing of food or talking. Due to its complicated anatomy, the TMJ is also prone to pathologies which are a common concern in medicine [40].

In this paper, we focus on accurate modeling of the kinematics of the TMJ for the purposes of computer animation. An ideal interface for animation (known as a "rig") should be user-friendly. The most prominent mode of motion is opening of the mouth. Even this common, everyday motion is, kinematically, a non-trivial composition of rotation and translation (sliding). This sliding occurs on a curve which reflects the geometry of the mandibular condyle and the zygomatic process which are held in close proximity by connective tissues (Fig. 1). Additionally, the jaw also allows for some amount of sideways and front–back translation, even without opening the mouth. Normally, when the mouth is closed, the upper teeth rest naturally in front of the lower teeth. We invite the reader to try translating their lower teeth forward—they can be moved *in front of* the upper teeth. Similarly, it is also possible to move the lower teeth from left to right. All of these motions are combined together to endow the jaw with its basic functions, such as chewing or talking. Our goal is to provide an intuitive animation interface allowing the users to explore realistic jaw motions. In particular, the users can synthesize realistic jaw motions by just varying three parameters that correspond to anatomically prominent modes of motion: opening, sideways sliding (known as "lateral excursion" in medical terminology), and front–back translation (known as "protrusion" and "retrusion"). In addition to synthesizing anatomically accurate jaw motions using an intuitive interface, our modeling will also allow us to validate if a given jaw motion is anatomically accurate or not.

Previous kinematic models for the jaw were generally qualitative, designed by researchers who observed the relevant anatomy and proposed models for its geometric behavior. We propose a different, data-driven approach, taking advantage of the fortuitous fact that the kinematics of the jaw can be measured with high accuracy using optical methods. This is because the skull and the mandible bones are connected with the upper and lower teeth, and the motion of the teeth is directly visible if the lips are open. To obtain data to train our model, we have asked an actor to exercise the entire range of motion of the jaw while keeping the lips at least partially open. This performance is recorded by multiple (we use three) calibrated cameras. We also create a highly
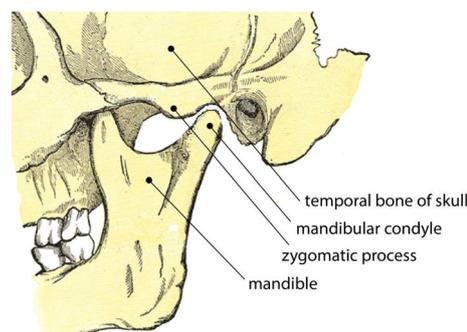
Fig. 1 The temporomandibular joint is the joint between the mandible and the temporal bone of the skull

Fig. 2 3D models (meshes) of the lower and upper teeth

accurate 3D models of the teeth of the actor using a standard dental 3D scanner, see Fig. 2. We use the 3D models of the upper and lower teeth for tracking the video, reconstructing the 3D position and orientation of the upper and lower teeth in each frame. Computing the relative rigid motion between the upper and lower teeth poses in each frame gives us point samples of physiologically possible jaw motions. These point samples are rigid transformations, i.e., points on the Lie group of rigid motions, often denoted as $SE(3)$. To simplify our task of creating an intuitive parameterization of jaw kinematics, we map our point samples from $SE(3)$ to the corresponding Lie algebra $se(3)$ via the logarithmic mapping. The Lie algebra $se(3)$ is a standard linear (vector) space and therefore permits standard principal component analysis (PCA). However, performing the PCA on our point samples fails to capture the nonlinearity of trajectories of motions such as mouth opening, where the mandibular condyle slides along the zygomatic process along a curved path.

In order to extract anatomically meaningful modes of motion from our input data ($se(3)$ samples), we turned to autoencoder neural networks. In particular, we applied a special type of autoencoder termed nonlinear PCA (NLPCA) [30]. A modified version of NLPCA allowed us to explain all of our input data with a generative model with only three parameters, corresponding to the main modes of jaw motion:

(1) mouth opening, (2) lateral excursion, (3) protrusion and retrusion. Furthermore, we use our data to obtain explicit boundaries on each of these three modes of motion, with the bounds on lateral excursion and pro/retrusion depending on the amount of mouth opening. The result is an intuitive and anatomically realistic 3-parameterization for jaw kinematics. There are two potential applications of our resulting jaw kinematics model: (1) a control interface allowing animators to intuitively synthesize meaningful jaw motions, e.g., for special effects animation [15]; (2) allow computer vision researchers to automatically discard invalid jaw poses while tracking recorded facial performances [39].

2 Related work

The need for jaw kinematic modeling was identified in previous work in computer graphics. Sifakis et al. [32] created a jaw kinematics rig by designing sliding tracks of the condyles identified from magnetic resonance images. Because MRI is not always available and may be even medically contraindicated, Wu et al. [39] proposed a geometric rig offering two rotational degrees of freedom along a common pivot point and one translational degree of freedom along a fixed axis. This model was slightly generalized by Ichim et al. [15] who proposed using three translational degrees of freedom instead of just one, in addition to the two original rotational degrees of freedom. Li et al. [23] use an articulated model including the neck and eyeballs, with three rotational degrees of freedom for each of the joints, including the jaw. Our approach does not require MRI but still produces highly accurate jaw kinematics model for a given actor.

The mechanical function of the jaw has been studied in biomechanics, often using full six degrees of freedom for the rigid body motion of the jaw relative to the skull [18,36], even though reduced models were also considered, e.g., De Zee et al. [7] who proposed a planar constraint along which the condyles can slide, resulting in four degrees of freedom. A simulation platform can be used to create computational models using variables for modeling gravity, external forces, and jaw muscle activity [11]. These models were shown to be capable of predicting jaw movements for mundane actions, but inappropriate for complex actions like chewing [10] or post-reconstruction surgery [11].

An established tool to study bone kinematics is fluoroscopy (X-ray videos). Fluoroscopic studies of the temporomandibular joint kinematics have been carried out on rabbits [13], but the use of ionizing radiation (X-ray) for research purposes on humans is not acceptable. Fortunately, the motion of the mandible relative to the skull can be captured using optical methods if the lips are at least partially open. Tracking in videos is a well-studied computer vision problem [22], and popular methods include template-based tracking

such as Lucas-Kanade [25], active appearance models [6], feature-based tracking [24], and edge or boundary-based tracking [17,27,37]. A key challenge in teeth tracking and pose estimation problem is that it violates many common assumptions that are commonly satisfied in a tracking framework. Tracking methods such as Lucas–Kanade [25] and active appearance models [6] require the object to remain free from occlusion during the tracking process and undergo only relatively small appearance changes with respect to the original template. Unfortunately, in practice teeth are usually highly occluded and their appearance changes considerably due to glossy nature of enamel. More robust keypoint-based methods such as SIFT [24] are also hardly applicable since teeth are usually smooth and self-similar; therefore, it is difficult to find sufficient number of distinct feature points that can be tracked consistently.

There are a few tracking algorithms that are customized for teeth [2,37,38], and these tracking methods are primarily developed for augmented reality applications, where the goal is to overlay an AR image on the patient for providing additional assistance during dental surgery. In [37], a stereo camera is used to capture and reconstruct the 3D contour of a patients teeth. This 3D contour is registered with the 3D model obtained using CT scans using iterative closest point (ICP) algorithm. Using this registration, we can have augmented reality (AR) overlay of 3D image on the patient during dental surgery. Nevertheless, these techniques still assume avoidance of teeth occlusion. To alleviate the occlusion problem, Zoss et al. [42] used specifically designed markers for the teeth tracking. These markers are mounted on steel pins, which are glued to the teeth using UV-hardened dental composite. Our teeth tracking approach is noninvasive, which does not attach marker tags to the teeth but employs a model-based registration process to obtain high-quality teeth poses.

In this work, we are primarily interested in principal component analysis (PCA) on 6-dimensional jaw motions in $se(3)$. Since linear PCA fails to capture the nonlinear trajectories of motions, we resort to nonlinear PCA (NLPCA) methods [20]. Popular nonlinear methods include principal curves [12], locally linear embedding (LLE) [29], and Isomap [34]. In particular, we show that the input data can be explained using only three parameters corresponding to three main modes of jaw motions using a special type of autoencoder termed nonlinear PCA (NLPCA) [30].

3 Method

3.1 Data acquisition and preparation

We start by obtaining the models of the upper and lower teeth of our actor by a standard dental scanning procedure, produc-

Fig. 3 Our recording setup, featuring three tripod-mounted GoPro cameras and diffuse light sources

Fig. 4 Sample teeth segmentation results from synchronized images in three video cameras

ing detailed tooth geometry with distances in millimeters, see Fig. 2.

We capture the dynamic teeth performances using three tripod-mounted GoPro Hero 5 Black cameras, see Fig. 3. To reconstruct the teeth poses from the three camera videos, we exploit the teeth's position and shape information that are implicitly encoded in the video frames. We extract this information by segmenting out the teeth from the video frames, as shown in Fig. 4. The segmentation of the video frames is performed by employing the Roto brush tool in Adobe Effect [4]. The tool required minimal user interaction (i.e., only a few strokes) to achieve the teeth segmentation. Please note that gums between the teeth are allowed to be part of the teeth segmentation, since the color variation between them may be very small, as shown in Fig. 4. For additional technical details regarding our data capture and processing please see Sect. 5.

Only the relative motion between the skull and the mandible is relevant for further processing. The actual pose or orientation of the head is not important. To remove the global pose of the head from our study, we proceed as follows. For a given image, let (R^L, t^L) and (R^U, t^U) be the rigid motion (rotation, translation) of the lower and upper teeth, respectively. Let us denote (R, t) to represent the relative motion between the skull and mandible bones, and it can be derived as follows:

$$\begin{bmatrix} R^U & t^U \\ 0 & 1 \end{bmatrix} \begin{bmatrix} R & t \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} R^L & t^L \\ 0 & 1 \end{bmatrix}. \tag{1}$$

From Eq. (1), we can compute

$$R = (R^U)^T R^L \tag{2}$$
$$t = (R^U)^T (t^L - t^U) \tag{3}$$

3.2 Learning jaw kinematics from data

Let (R_i, t_i) be the relative jaw transformation of the frame indexed by i from the input video. Our goal is to construct an anatomically realistic jaw kinematics model which can explain all of the measured relative jaw transformations $\{(R_i, t_i)\}_{i=1}^n$ (we used $n = 833$ frames from our input training video) while providing the user with intuitive parameters for controlling the jaw's poses.

Each of our input data points (R_i, t_i) lies on a $SE(3)$ manifold (the manifold of rigid body transformations), which is nonlinear and thus not ideal for further analysis. Therefore, we map each of our data points from $SE(3)$ to the corresponding Lie algebra $se(3)$ using the logarithmic mapping, which has a closed-form expression in the case of $SE(3)$ [26]. Geometrically, this corresponds to unfolding the relevant part of the nonlinear manifold $SE(3)$ to a linear space (the Lie algebra). All of our subsequent analysis will be performed on the data points in $se(3)$. We denote the resulting vectors as $\{v_i\}_{i=1}^m$, where $v_i \in \mathbb{R}^6$.

3.2.1 PCA learning

Our first attempt to analyze the input training data $v_1, \ldots, v_m$ is to apply principal component analysis (PCA), producing six principal components $\mathbf{p}_i \in \mathbb{R}^6$ with $i = \{1, 2 \ldots, 6\}$, where their corresponding six singular values are given by the set $\{10.38, 0.73, 0.44, 0.34, 0.25, 0.07\}$. With this PCA basis, each jaw pose can be written as

$$\mathbf{p} = \sum_{j=1}^6 w_j \mathbf{p}_j, \tag{4}$$

where $w_j \in \mathbb{R}$ is weight (PCA loading) of each of the principal components. The resulting $\mathbf{p} \in se(3)$ can be mapped to $SE(3)$ by computing the exponential mapping (closed-form expression [26]). The resulting rigid body transformation can be then used to transform the mandible mesh, producing visualization of the jaw motion represented by the linear combination of the principal components.
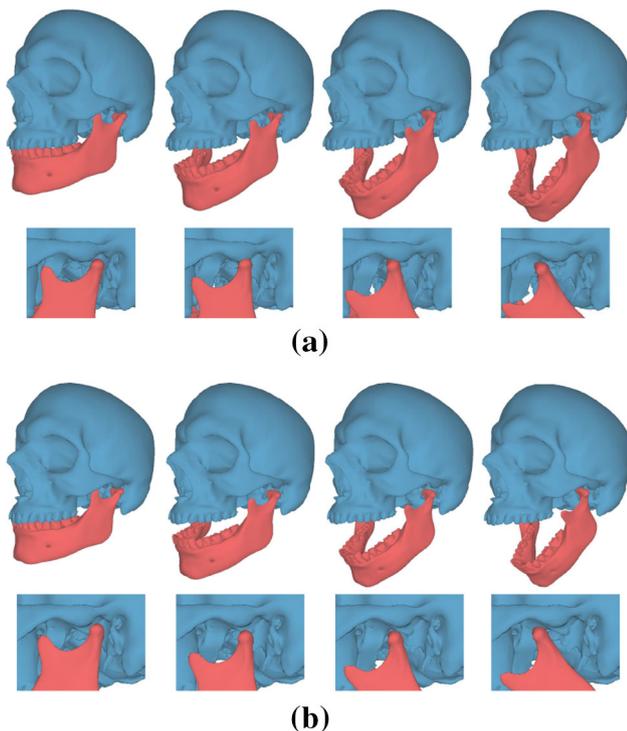
Fig. 5 Motions of mouth opening that are generated using PCA (**a**) and NLPCA autoencoder (**b**). The jaw poses in **a** are generated by scaling the first principal component: $w_1\mathbf{p}_1$, for $w_1 = \{-1.6157, -0.5717, 0.4723, 1.5165\}$ (left to right), while the jaw poses in **b** are generated as $\Phi_{gen}(c_1, 0, 0, 0, 0)$ where c_1 is $\{-0.1943, -0.0195, 0.1552, 0.3300\}$ (left to right). Note that the zoom out in the bottom row of **a** illustrates that the PCA does not learn the correct nonlinear trajectory and the mandibular condyle intersects the zygomatic process, while in **b** the condyle slides along the zygomatic process, which is the desired anatomically realistic behavior

In Fig. 5a, we visualize the effect of the first principal component $\mathbf{p}_1$ which, not surprisingly, captures the dominant motion—opening of the mouth. However, the visualization in Fig. 5a also reveals the limitations of PCA (please see also the animation in the accompanying video). Specifically, even though the jaw motion generated by scaling the first principal component roughly corresponds to opening of the mouth and includes both rotation and translation, it fails to accurately capture the trajectory of the mandibular condyle. When opening the mouth, the mandibular condyle slides over the surface of the zygomatic process (see Fig. 1). Because the zygomatic process is curved (as opposed to flat), this results in a nonlinear trajectory. The nonlinearity of this trajectory can be observed by real-time magnetic resonance imaging of a human subject performing voluntary mouth opening and closing [41] (please see the accompanying video). Unfortunately, the first principal component produces only a crude approximation of this nonlinear trajectory (specifically, this approximation corresponds to a straight line in $se(3)$). There are other limitations with PCA.

The second and the remaining principal components do not correspond to anatomically meaningful motions such as lateral excursions and pro/retrusion. It is also challenging to generate anatomically realistic motions by considering the 5 or 6 principal components within their boundary values.

3.2.2 NonLinear PCA learning

To be able to capture the correct anatomical behavior of the temporomandibular joint during mouth opening with a single parameter, we turn to a more powerful, nonlinear data analysis tool: autoencoder neural networks [9]. We trained various autoencoder architectures using TensorFlow [1] and found that substantial dimensionality reduction can be achieved, e.g., using only three dimensions can reproduce the training data $\{v_1, \ldots, v_m\}$ very accurately (unlike PCA, which produces relatively large error with only three principal components). However, the reduced parameters produced by the encoder part of an autoencoder do not have any hierarchical meaning as is the case in PCA. In PCA, the first principal component explains the largest variance in the data, but this is not the case for the first component produced by the encoder. Fortunately, we found that a solution to this problem has already been described by Scholz and colleagues [30,31], who proposed a modified autoencoder network which mimics the hierarchical property of the principal components. Their approach is called NLPCA (for nonlinear PCA).

Let $\mathcal{S}$ be a data space given by our 6-dimensional data points in $se(3)$ and $\mathcal{P} \subseteq \mathbb{R}^n$ a component space with $n \leq 6$. NLPCA learns two nonlinear functions Φ_{extr} and Φ_{gen}, where $\Phi_{extr} : \mathcal{S} \mapsto \mathcal{P}$ is called *component (or feature) extraction* function (corresponding to the encoder part of an autoencoder) and $\Phi_{gen} : \mathcal{P} \mapsto \mathcal{S}$ is called *data generation* function (corresponding to the decoder part). The extraction function Φ_{extr} transforms a 6-dimensional data point into the corresponding component representation $\mathbf{c} = (c_1, c_2, \ldots, c_n)$, while the generation function Φ_{gen} performs the reverse, i.e., reconstructs the original data points from its lower-dimensional component representation.

With our training data $v_1, \ldots, v_m$, we found that only five dimensions are needed with NLPCA; the sixth dimension introduces only minimal modifications which can be attributed to noise in the input data. Furthermore, visualizing the effect of the individual components, we observe that the nonlinear characteristics of the mouth opening motion are effectively captured by the first component (c_1). Visualizing the results of $\Phi_{gen}(c_1, 0, 0, 0, 0)$ for varying c_1 produces anatomically realistic nonlinear trajectory of the mandibular condyles, see Fig. 5b. Most importantly, the mandibular condyle now slides along the zygomatic process, as opposed the unrealistic intersection produced by PCA (Fig. 5a). Even though this five-dimensional model can accurately represent all possible motions of the jaw, we found it is possible to
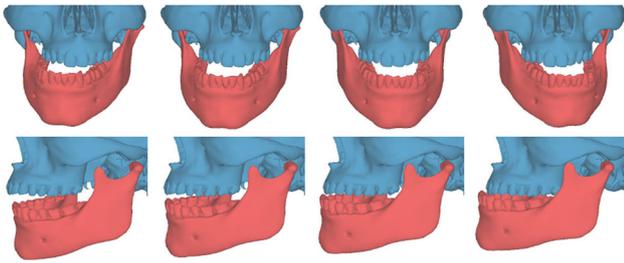
Fig. 6 Top row: lateral excursions learned by nonlinearly reducing D_{23} to a single dimension; bottom row: the same approach applied to D_{45} produces one-dimensional parameterization of protrusion and retrusion
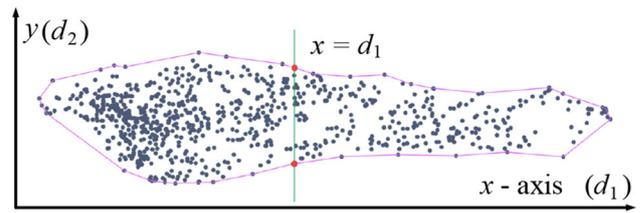

Fig. 7 The range of the parameter d_2 is determined with respect to a given d_1 (mouth opening). The parameters (d_1, d_2) of our original data points are projected onto the x-y plane, and the valid range of d_2 is determined by y-coordinates of the intersection points between the line $(x = d_1)$ and a polygonal boundary of the 2D projection points

reduce the number of parameters further, providing a more compact and intuitive interface to the user while increasing the error only negligibly.

By visualizing the effect of the components c_2 and c_3, we found that both of them correspond to lateral excursions (moving the jaw sideways, left and right). Similarly, the last two components c_4 and c_5 correspond to forward and backward motion of the jaw, known as protrusion and retrusion. This observation motivates the final refinement of our model, which we describe next. First, we generate a set of data points (denoted as D_{23}) which correspond to zeroing all components except for c_2 and c_3, i.e., $\Phi_{gen}(0, c_2, c_3, 0, 0)$, where the values of c_2 and c_3 are given by encoding of our training data. Similarly, we construct another set of data points (denoted as D_{45}) in terms of the components of $(0, 0, 0, c_4, c_5)$. Next, we run NLPCA [30] on the D_{23} dataset to nonlinearly reduce its dimensionality to one. We denote the resulting one-dimensional generation function (decoder) as $\Phi_2(d_2)$, where d_2 is a scalar input parameter. We observe that by varying d_2, the function Φ_2 can explain the entire range of lateral excursions (sideways motions of the jaw from left to right), during which the trajectory of the condyles again succeeds to avoid inter-penetrations with the zygomatic process (similarly as before for jaw opening). Again, we execute NLPCA on the D_{45} dataset and obtain a one-dimensional generation function $\Phi_3(d_3)$, such that varying the scalar parameter d_3 produces an entire range of anatomically realistic protrusion and retrusion. These motions are visualized in Fig. 6. To make our notation succinct, let us introduce a shorthand $\Phi_1(d_1) := \Phi_{gen}(d_1, 0, 0, 0, 0)$ where d_1 is a parameter corresponding to mouth opening.

Our final model parameterizing the jaw kinematics with only three dimensions is a linear combination of the individual parts:

$$\Phi(d_1, d_2, d_3) = \Phi_1(d_1) + \Phi_2(d_2) + \Phi_3(d_3) \tag{5}$$

Linear combination is justified by the fact that the Lie algebra $SE(3)$ is a linear space and the maximal relative rotations are bounded (far below 180°) [3]. As we discuss in more details in Sect. 5, this final $\Phi(d_1, d_2, d_3)$ introduces only marginally higher error compared to the five-dimensional NLPCA model, while using fewer parameters and being much more intuitive to the user.

Valid component representation It is evident that for a valid parameter representation $\mathbf{d} = (d_1, d_2, d_3)$, each of its parameters should be in a specified range. Since each of our original data points has a corresponding parameter representation, a simple way to determine the parameter ranges would be to bound each parameter using the parameter representations of the given data points. However, in real jaw motion, the ranges of parameters d_2 and d_3 are dependent on the current value of d_1. To demonstrate this, we invite the reader to try sliding their lower teeth to the right and then opening the mouth—the lower teeth are automatically moved back to the center. Therefore, we need to dynamically determine the valid ranges for the parameters d_2 and d_3, with respect to current d_1. As shown in Fig. 7, we use d_2 as an example to describe our solution. First, we project the parameters (d_1, d_2) of our original data points onto a x-y plane. Then, a polygonal boundary is formed using the alpha shape ($alpha = 0.12$ in our experiments) of these planar points. For a given d_1, the current range of d_2 is determined by the intersection points between the polygonal boundary and the line $x = d_1$.

4 Model-based teeth motion tracking

Essentially, our teeth motion tracking problem is a 3D–2D matching problem [8,28] where the task is to find the best pose of a 3D model so that its 2D projections are well aligned with the 2D images. In our scenario, the 3D model is a teeth model, corresponding to either the upper or lower teeth of the performer, and the model pose corresponds to a rigid motion. More formally, we denote the teeth model as $\mathcal{M}$ and represent the rigid motion as a 6-tuple $T = \{\theta_x, \theta_y, \theta_z, t_x, t_y, t_z\}$, where $\theta_x, \theta_y, \theta_z$ are the rotation angles around the x-, y-, and z-axes, respectively, and t_x, t_y, t_z are the translation offsets along the x, y, and z directions, respectively. To make the projections
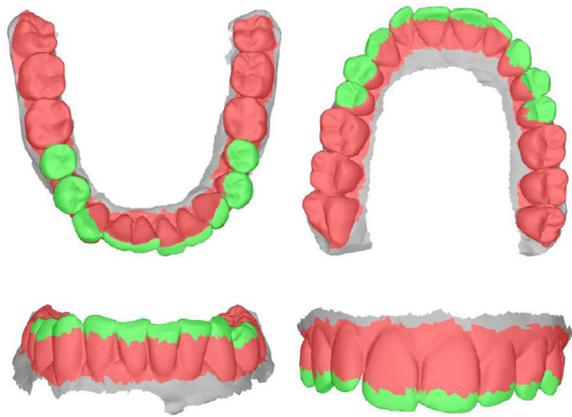
Fig. 8 The teeth model is roughly and manually marked out from the 3D scans of the performer's teeth rows. The geometry of the teeth model consists of the triangles in the red and green regions, where the green parts correspond to the boundary of interest of the teeth model

of $\mathcal{M}$ most consistent with the (three in our case) images that corresponds to a common video frame, we can solve the following optimization problem:

$$\arg\min_{T} \sum_{i=1}^{3} E(\mathbb{P}_i(R\mathcal{M} + t), I_i), \tag{6}$$

where $R = R_{\mathbf{z}}(\theta_z)R_{\mathbf{y}}(\theta_y)R_{\mathbf{x}}(\theta_x)$ with $R_{\mathbf{x}}(\cdot)$, $R_{\mathbf{y}}(\cdot)$, and $R_{\mathbf{z}}(\cdot)$ being the rotation matrices about the x-, y-, and z-axes, respectively, with a specified angle, $t = (t_x, t_y, t_z)$ is a translation vector, $\mathbb{P}_i(\cdot)$ is an operation that projects a 3D model to the image plane of the ith camera, I_i corresponds to a video frame image that is captured by the ith camera, and $E(\cdot, \cdot)$ defines an energy function which measures the inconsistency or discrepancy between the 2D projection of the 3D model and the real 2D image. In our implementation, we use hardware pipeline via OpenGL to conduct the projection operation $\mathbb{P}_i(\cdot)$, with respect to the cameras' intrinsic and extrinsic parameters.

To define E, our idea is to fully exploit the geometry information of the teeth model (see Fig. 8) as well as the shape and position information of the performer's teeth that are encoded in the teeth segment of the captured images. We define E as follows:

$$E = E_{SP} + \lambda E_{BND}. \tag{7}$$

where E_{SP} measures the overlap of the teeth model with the actual visible teeth segment, while E_{BND} ensures alignment of the roof region of frontal teeth which we call boundary of interest (BOI). We use fixed weight $\lambda = 0.4$ for all of the results in this paper.

To compute E_{SP} for every captured image, we render its teeth segmentation with white color into clean (black) screen

buffer (see Fig. 9a). Then, the teeth model at current pose is rendered in red color (with alpha blending $\alpha = 0.5$), i.e., projected with respect to the intrinsic and extrinsic parameters of the corresponding camera. Thanks to this second rendering pass the pixels of the teeth segmentation that are overlapped with the projection of the teeth model will change to pink color (see Fig. 9b). The closer the poses of the teeth model are to the actual performer's teeth, the fewer pixels with white color remain. Therefore, we define the energy term E_{SP} as follows:

$$E_{SP} = \#Pixel\_White, \tag{8}$$

where $\#Pixel\_White$ refers to the number of white pixels in the screen buffer.

Due to the occlusion by lip or tongue, the teeth segmentation of the performer is usually smaller than the 2D projection of its teeth model, such that the energy term E_{SP} may lead to sub-optimal result, as shown in Fig. 9c. We use the energy term E_{BND} to tackle this problem. To compute the boundary term E_{BND}, during the rendering of teeth model we change the color of corresponding BOI to green (see Fig. 9d, e). This ensures that when the 2D projection of the roof region is perfectly aligned with the teeth segmentation, the number of pixels with light green color is increased, while the number of pixels with dark green color is minimal. Thus, we compute the energy term E_{BND} as:

$$E_{BND} = \#Pixel\_DGreen, \tag{9}$$

where $\#Pixel\_DGreen$ refers to the number of dark green pixels in the screen buffer. Please note that the energy term E_{SP} will prevent the 2D projection of the roof region being pulled into the segmentation too much, since it will increase the number of white pixels.

To minimize E, we combine a gradient descent-like approach with dense sampling which, in practice, ensures good local optima while being computationally efficient. During the descent, we estimate the direction of the energy gradient relative to T using finite differences. Along this search direction, we then use golden section search [16] to find the new optimal value for T. To reduce the number of degrees of freedom, we first fix translation (t_x, t_y, t_z) and update rotation $(\theta_x, \theta_y, \theta_z)$; then, rotation is fixed and translation updated. We repeat this process until convergence. To avoid getting stuck in poor local minima, we subsequently refine the pose T by sampling the space of possible configurations more densely in a small neighborhood around the current pose T.

To further improve the optimization process, we use the pose from previous frame as the initial pose T. For the first frame, we bootstrap the process by manually selecting four vertices on the teeth model and find their corresponding

(a) Segmentation **(b)** Initial Pose **(c)** E_{SP} **(d)** $E_{SP} + E_{BND}$ **(e)**
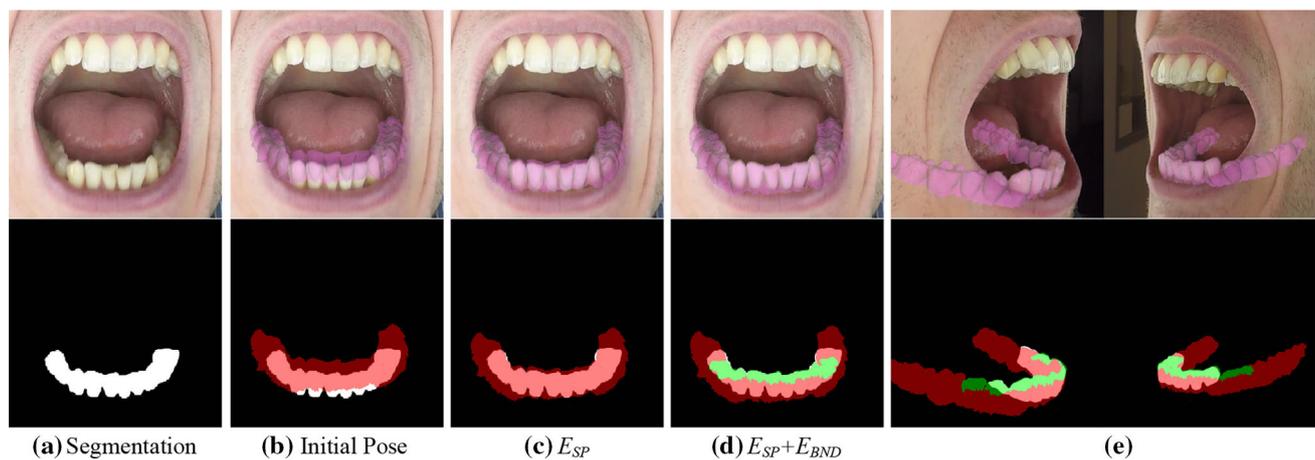
Fig. 9 Optimization energy of the (lower or upper) teeth pose. Given the captured images and the teeth segmentation (**a**) in Fig. 4 and the initial pose of the teeth model, we render for each camera the 2D projection of the model's geometry and the teeth segmentation into the same screen buffer (**b**). The overlapping pixels are used to define energy terms (**c**, **d**) which measure how consistent is between the poses of the teeth model and the real teeth at the current video frame. In **d**, **e**, the projections of the optimal teeth pose in all of the three cameras are shown

positions in two captured images. Through triangulation in stereo analysis, we obtain the corresponding 3D positions for selected vertices; then, an initial guess T for the first frame can be obtained by fitting the initial positions of the selected vertices to their new positions [33].

5 Experimental results

The algorithms were implemented in MATLAB/C++ and were run on a 2.9 GHz Intel Core(TM)i7 processor (32 GB RAM) with a single CPU thread.

The teeth performance capture is achieved using three GoPro Hero 5 cameras at 1920×1080 resolution, and 24p linear video mode settings. The performer places his mouth approximately 20 cm from each of the three cameras to allow for optimum coverage and larger overlap between the cameras. Using planar checkerboard patterns, the cameras are calibrated for both intrinsic and extrinsic parameters. The three cameras are synchronized by turning on and off lights. We locate the frames with sudden intensity changes to match the frames from different cameras. An accurate hardware synchronization could also be employed to eliminate this step.

Teeth motion tracking On average, the total optimization time is about 15 s per frame and the main bottleneck is the computational time for computing the energy function E in Eq. (7) which involves counting the number of pixels of specific color in the screen buffer. In our current implementation, we use 940×480 screen buffer with 24 bits per pixel, where the pixels are transferred from the GPU into CPU and counted. The computational efficiency can be improved by considering parallel counting on the GPU, e.g., using histogram operation available in CUDA SDK.

We recorded 833 teeth poses of the actor and performed teeth tracking on all these frames for jaw kinematic modeling. We show the results in the supplementary video; a sample of teeth tracking results is shown in Fig. 10. Error in teeth tracking can come from several sources: calibration, errors in 3D scans of the teeth, and segmentation errors. Unfortunately, it is difficult to get the ground truth for this experiment, and therefore, the validation is mainly done qualitatively. Since we have images from three camera views, we are able to visualize the tracked poses without depth ambiguity, as shown in Fig. 10. Furthermore, we apply the tracked poses of the upper and lower teeth to the skull and mandible models of the performer, respectively. As shown in Fig. 11, we can create augmented reality sequence by overlaying the skull and mandible models on the captured images. From the reconstructed motions of the skull and mandible models, we can also stabilize skull motions and visualize only the relative motion of the mandible as shown, e.g., in Fig. 14. Note that the skull and mandible models are segmented and reconstructed from the performer's MRI data. While these models are helpful for the visual evaluation of the tracking results, they are not necessary for our tracking approach. It can be seen from the present examples and the supplementary video that the tracked poses of the teeth model are visually well aligned with the real teeth of the performer.

In addition to visual comparison, we provide an approximate metric for measuring the accuracy. Since feature points and their correspondences are difficult to obtain in the case of teeth images, we instead measure the difference in the area from the projection of the teeth model and the teeth in the images. The difference in area can be approximately computed as the ratio between energy term E_{SP} in Eq. (8) and the total pixel number of the teeth segmentation. The average error on the captured images is about 0.754%.

Fig. 10 Tracking results from several teeth poses. Each row corresponds to a specific teeth pose and its 2D projections in the three cameras are shown. Note that the 2D projection (pink overlaid region) from the 3D model covers both the visible and the occluded teeth

Fig. 11 The tracked poses of the upper and lower teeth can also be illustratively shown using the projection of the skull (blue) and mandible models (red) on the captured images

Jaw kinematics model With the tracked poses of the performer's lower and upper teeth, the jaw motion extraction, including the data mapping from $SE(3)$ to $se(3)$, can be computed efficiently at the rate of 4.6 s for all the 833 teeth poses. In MATLAB, the nonlinear PCA takes 108 s for learning the jaw kinematics model.

In Fig. 12, we use the learned jaw kinematics model to reconstruct the originally captured jaw poses. Then, we measure the reconstruction error by computing the difference between each captured pose and its reconstruction. To measure the difference between a pair of given poses, we apply them to the mandible mesh, respectively, and then compute the maximum vertex distance between the two transformed

mandible meshes. As shown in Fig. 12, our jaw kinematics model can reconstruct the jaw kinematics of a real person up to an error of 1.73 millimeters on average. To further validate that our jaw kinematics model generates the anatomically realistic poses, we randomly synthesized 1000 jaw poses using our model. For each synthesized pose, we measure how realistic it is by computing the difference between the pose and its closest one in the real jaw poses that are captured from the performer. The result is shown in Fig. 13.

Finally, using the jaw kinematics model, we can then synthesize anatomically realistic and visually pleasing jaw poses, i.e., by adjusting the parameters $\{d_i\}_{i=1}^3$ of the model in their respective valid ranges. To facilitate this step, we designed a simple user interface where the user can easily explore the constrained nonlinear space of the jaw kinematics through three sliders, each of which is used to tune a parameter and thus corresponds to a semantically meaningful mode of the jaw motion (see Fig. 14).

Discussion In [42], a jaw kinematics model was also constructed, by parameterizing the jaw poses over a so-called Posselt's envelope of motion which is formed by the trajectories of a reference point at the anterior of the jaw. However, while the surface of envelope can be successfully traced from the jaw poses which are densely and uniformly sampled in

(a) 4-dim NLPCA

(b) 5-dim NLPCA

(c) our model

(d) pose with maximum error
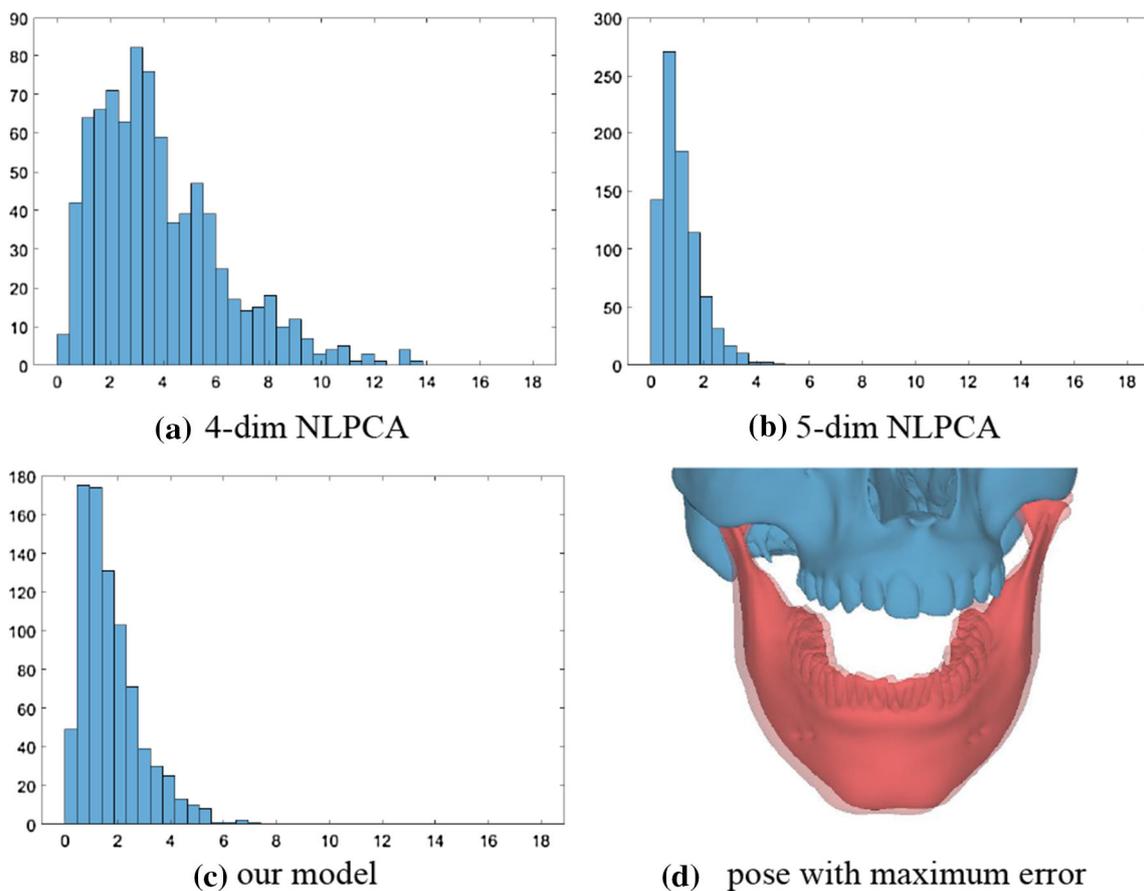
Fig. 12 For each of captured poses, we use NLPCA with different number of component dimensions to reconstruct its original pose in **a**, **b**. In **c**, the original pose is reconstructed using our final kinematic model according to Eq. (5). The histogram of the reconstruction errors of the 833 captured poses is shown (horizontal axis: error in millimeter; vertical axis: frequency). For the pose with maximum reconstruction error (i.e., 7.2216) using our final kinematics model, its original and reconstructed poses are shown in an overlapping way (**d**), where the transparent part corresponds to the difference between the two poses

Fig. 13 One thousand jaw poses are randomly synthesized using our kinematics model with and without the dynamic adjustment of the valid ranges for the parameters d_2 and d_3. The histogram of the accuracy of the 1000 synthesized poses is given (left is with the dynamic adjustment, while right is without the dynamic adjustment), whose layout is similar to that of Fig. 12
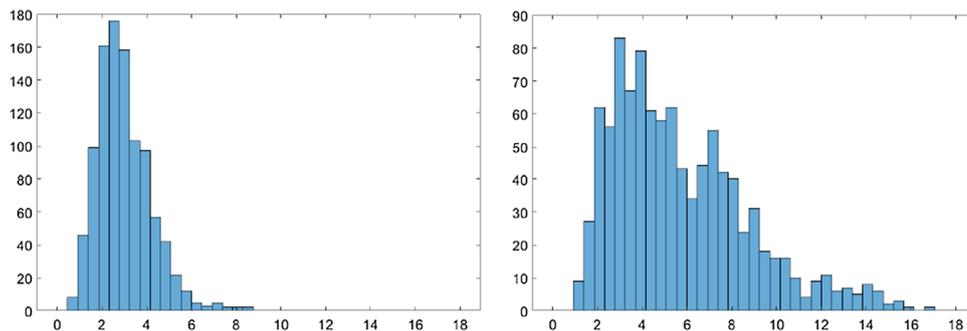
the range of motion of the jaw, it may fail for the case where the distribution of the sampled jaw poses is not so dense and uniform, e.g., the tracked jaw poses in our case, as shown in Fig. 15.

Inverse kinematics (IK) In addition, we allow the user to generate the jaw poses via direct manipulation. To edit the jaw pose, the user can pick a vertex on the mandible mesh and mouse-drag it to a new position. The system then automati-cally updates the parameters of our jaw kinematics model to find the optimal pose that is consistent with this user-specified positional constraint. Given the new position of the selected vertex (p'), our goal is to find parameters $\mathbf{d} = (d_1, d_2, d_3)$ so that the transformed position of the selected mesh vertex ($p_\mathbf{d}$) is as close as possible to p':

$$\arg \min_{\mathbf{d}} \| p' - p_\mathbf{d}) \|^2 \quad \text{subject to valid } \mathbf{d}. \tag{10}$$
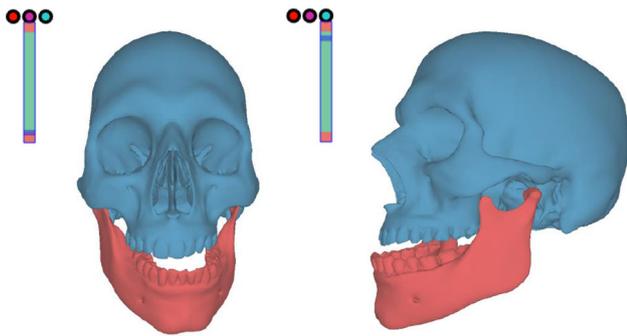
Fig. 14 Our user interface includes three circle points which correspond to the modes of mouth opening/closing, lateral excursions, and pro/retrusion, respectively. When a circle point is selected, a slider appears and the user can move the slider to adjust the corresponding parameter for the desired jaw pose
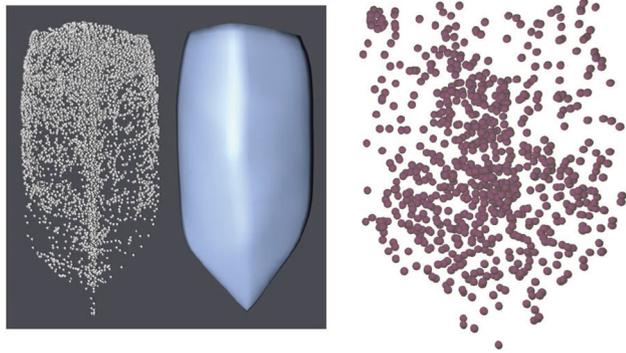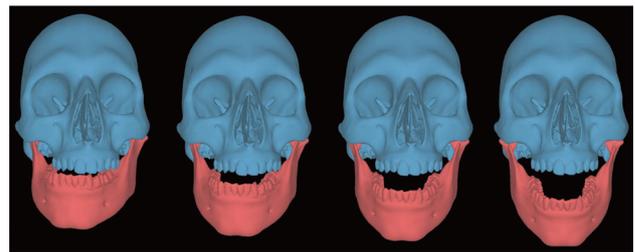

Fig. 15 While the envelope surface can be traced from the 9000 densely and uniformly sampled poses (left which is depicted from [42]), it is difficult to trace the surface from the 833 sparsely tracked poses in our case (right)

Fig. 16 Using our interface, the user can create two jaw poses (left and right) and the system then automatically generates a jaw animation by keyframe interpolation

(a)   **(b)**
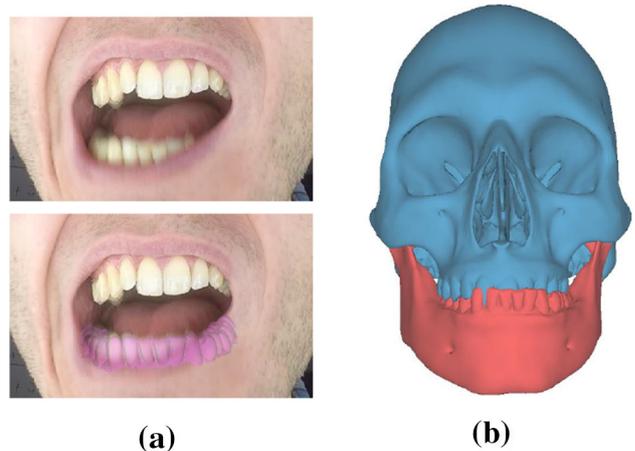
Fig. 17 Limitations and failure cases: **a** we show poor teeth tracking results in the case of images with blur. **b** Invalid jaw poses can be produced when the parameters $\{d_1, d_2, d_3\}$ are chosen near their respective boundary values. In particular, the boundary conditions for parameters d_2 and d_3 vary and they depend on the current value of d_1; however, such boundary conditions are just linear approximations and may not be accurate

We can efficiently solve Eq. (10) with a brute-force approach. We sample the space of possible configurations densely in a small neighborhood around the current pose and evaluate the best candidate. This takes only few milliseconds. An example of an interactive IK session is presented in the accompanying video.

Animation To generate continuous jaw motions, the user can design several key poses of the jaw through our user interface, and then the intermediate poses can be computed using simple linear interpolation between the parameters of the key poses. As shown in Fig. 16, such a simple interpolation scheme already generates high-quality jaw motions which are comparable to real jaw motions.

Limitations Rapid motion of the performer can lead to blurred images, which in turn can lead to segmentation errors and incorrect teeth tracking as shown in Fig. 17a. The use of cameras with higher frame rate can eliminate this problem.

Another limitation stems from the valid ranges of the parameters in our jaw kinematics model. While current scheme for determining the valid ranges is simple and practical, invalid jaw poses may still happen when extreme poses are synthesized using the parameters that locate near the boundary of their valid ranges. An example of such scenario is shown in Fig. 17b, where the jaw is in the rightmost of the lateral excursion and in the maximum of protrusion. A possible solution might be a smooth and tighter bounding volume, e.g., the isosurface representation [14].

6 Conclusion

In this paper, we build an anatomically realistic jaw kinematics model from data. We use three tripod-mounted GoPro Hero 5 Black cameras to capture the dynamic teeth performances of an actor. Then, the jaw kinematics is learned from these jaw poses using the nonlinear PCA, which effectively captures the nonlinear characteristics of the jaw's motion. The resulting jaw kinematics model has three parameters, each of which corresponds to an intuitive mode of the

jaw's motion, i.e., mouth opening, lateral excursions, and pro/retrusion. Finally, our jaw kinematics model provides an intuitive interface allowing the animators to explore realistic jaw motions in a user-friendly way. Such model is potentially useful for various application scenarios to guarantee anatomically correct results, such as synthesization of meaningful jaw motions, e.g., for special effects animation [15], or automatic discarding of invalid jaw poses for tracking recorded facial performances [39].

In future, we plan to create a library of user-tailored jaw kinematics models since each person has her personalized capacity of jaw motion. We plan to obtain completely automatic segmentation algorithm using graph cuts algorithm, where the parameters of the energy function will be learned from the training data with manually annotated segmentation labels. An interesting line of research would be to actually use the learned jaw kinematics model in the teeth tracking problem and vice versa.

Compliance with ethical standards

References

1. Abadi, M., Barham, P., Chen, J., et al.: Tensorflow: a system for large-scale machine learning. In: Proceedings of the 12th USENIX Conference on Operating Systems Design and Implementation, pp. 265–283 (2016)

2. Aichert, A., Wein, W., Ladikos, A., Reichl, T., Navab, N.: Image-based tracking of the teeth for orthodontic augmented reality. In: Proceedings of the 15th International Conference on Medical Image Computing and Computer-Assisted Intervention, Part II, pp. 601–608 (2012)

3. Alexa, M.: Linear combination of transformations. ACM Trans. Graph. **21**(3), 380–387 (2002)

4. Bai, X., Wang, J., Simons, D., Sapiro, G.: Video snapcut: robust video object cutout using localized classifiers. ACM Trans. Graph. **28**(3), 70:1–70:11 (2009)

5. Cong, M., Bao, M., E, J.L., Bhat, K.S., Fedkiw, R.: Fully automatic generation of anatomical face simulation models. In: Proceedings of the 14th ACM SIGGRAPH/Eurographics Symposium on Computer Animation, pp. 175–183 (2015)

6. Cootes, T.F., Edwards, G.J., Taylor, C.J.: Active appearance models. IEEE Trans. Pattern Anal. Mach. Intell. **23**(6), 681–685 (2001)

7. De Zee, M., Dalstra, M., Cattaneo, P.M., Rasmussen, J., Svensson, P., Melsen, B.: Validation of a musculo-skeletal model of the mandible and its application to mandibular distraction osteogenesis. J. Biomech. **40**(6), 1192–1201 (2007)

8. Gold, S., Rangarajan, A., Lu, C.P., Pappu, S., Mjolsness, E.: New algorithms for 2D and 3D point matching: pose estimation and correspondence. Pattern Recognit. **31**(8), 1019–1031 (1998)

9. Goodfellow, I., Bengio, Y., Courville, A.: Deep Learning. MIT Press, Cambridge (2016)

10. Hannam, A.G., Stavness, I., Lloyd, J.E., Fels, S.: A dynamic model of jaw and hyoid biomechanics during chewing. J. Biomech. **41**(5), 1069–1076 (2008)

11. Hannam, A.G., Stavness, I.K., Lloyd, J.E., Fels, S.S., Miller, A.J., Curtis, D.A.: A comparison of simulated jaw dynamics in models of segmental mandibular resection versus resection with alloplastic reconstruction. J. Prosthet. Dent. **104**(3), 191–198 (2010)

12. Hastie, T., Stuetzle, W.: Principal curves. J. Am. Stat. Assoc. **84**(406), 502–516 (1989)

13. Henderson, S.E., Desai, R., Tashman, S., Almarza, A.J.: Functional analysis of the rabbit temporomandibular joint using dynamic biplane imaging. J. Biomech. **47**(6), 1360–1367 (2014)

14. Herda, L., Urtasun, R., Fua, P., Hanson, A.: An automatic method for determining quaternion field boundaries for ball-and-socket joint limits. In: Proceedings of Fifth IEEE International Conference on Automatic Face Gesture Recognition, pp. 95–100 (2002)

15. Ichim, A.E., Kadleček, P., Kavan, L., Pauly, M.: Phace: physics-based face modeling and animation. ACM Trans. Graph. **36**(4), 153:1–153:14 (2017)

16. Kiefer, J.: Sequential minimax search for a maximum. Proc. Am. Math. Soc. **4**(3), 502–506 (1953)

17. Klein, G., Murray, D.: Full-3D edge tracking with a particle filter. In: Proceedings of the British Machine Vision Conference, pp. 1119–1128 (2006)

18. Koolstra, J.H.: Dynamics of the human masticatory system. Crit. Rev. Oral Biol. Med. **13**(4), 366–376 (2002)

19. Kozlov, Y., Bradley, D., Bächer, M., Thomaszewski, B., Beeler, T., Gross, M.: Enriching facial blendshape rigs with physical simulation. Comput. Graph. Forum **36**(2), 75–84 (2017)

20. Kramer, M.: Nonlinear principal component analysis using autoassociative neural networks. AIChE J. **37**(2), 233–243 (1991)

21. Lan, L., Cong, M., Fedkiw, R.: Lessons from the evolution of an anatomical facial muscle model. In: Proceedings of the ACM SIGGRAPH Digital Production Symposium, pp. 11:1–11:3 (2017)

22. Lepetit, V., Fua, P.: Monocular model-based 3D tracking of rigid objects: a survey. Found. Trends Comput. Graph. Vis. **1**(1), 1–89 (2005)

23. Li, T., Bolkart, T., Black, M.J., Li, H., Romero, J.: Learning a model of facial shape and expression from 4D scans. ACM Trans. Graph. **36**(6), 194 (2017)
24. Lowe, D.: Distinctive image features from scale-invariant keypoints. Int. J. Comput. Vis. **60**(2), 91–110 (2004)
25. Lucas, B.D., Kanade, T.: An iterative image registration technique with an application to stereo vision. In: Proceedings of the 7th International Joint Conference on Artificial Intelligence, vol. 2, pp. 674–679 (1981)
26. Murray, R.M., Sastry, S.S., Zexiang, L.: A Mathematical Introduction to Robotic Manipulation. CRC Press, Boca Raton (1994)
27. Ramalingam, S., Bouaziz, S., Sturm, P., Brand, M.: SKY-LINE2GPS: localization in urban canyons using omni-skylines. In: International Conference on Intelligent Robotics (IROS), pp. 3816–3823 (2010)
28. Rosenhahn, B., Perwass, C., Sommer, G.: Pose estimation of 3D free-form contours. Int. J. Comput. Vis. **62**(3), 267–289 (2005)
29. Roweis, S.T., Saul, L.: Nonlinear dimensionality reduction by locally linear embedding. Science **290**(5500), 2323–2326 (2000)
30. Scholz, M., Fraunholz, M., Selbig, J.: Nonlinear principal component analysis: neural network models and applications. Lect. Notes Comput. Sci. Eng. **58**, 44–67 (2008)
31. Scholz, M., Vigario, R.: Nonlinear PCA: a new hierarchical approach. In: Proceedings of European Symposium on Artificial Neural Networks, pp. 439–444 (2002)
32. Sifakis, E., Neverov, I., Fedkiw, R.: Automatic determination of facial muscle activations from sparse motion capture marker data. ACM Trans. Graph. **24**(3), 417–425 (2005)
33. Sorkine-Hornung, O., Rabinovich, M.: Least-Squares Rigid Motion Using SVD. Technical Notes, ETHZ (2009)
34. Tenenbaum, J.B., de Silva, V., Langford, J.C.: A global geometric framework for nonlinear dimensionality reduction. Science **290**(5500), 2319–2323 (2000)
35. Terzopoulos, D., Waters, K.: Physically-based facial modelling, analysis, and animation. Comput. Anim. Virtual Worlds **1**(2), 73–80 (1990)
36. Villamil, M., Nedel, L., Freitas, C., Macq, B.: Simulation of the human TMJ behavior based on interdependent joints topology. Comput. Methods Progr. Biomed. **105**(3), 217–232 (2012)
37. Wang, J., Suenaga, H., Hoshi, K., Yang, L., Kobayashi, E., Sakuma, I., Liao, H.: Augmented reality navigation with automatic marker-free image registration using 3-D image overlay for dental surgery. IEEE Trans. Biomed. Eng. **61**(4), 1295–1304 (2014)
38. Wang, J., Suenaga, H., Yang, L., Kobayashi, E., Sakuma, I.: Video see-through augmented reality for oral and maxillofacial surgery. Int. J. Med. Robot. Comput. Assist. Surg. **13**(2), e1754 (2017)
39. Wu, C., Bradley, D., Gross, M., Beeler, T.: An anatomically-constrained local deformation model for monocular face capture. ACM Trans. Graph. **35**(4), 115:1–115:12 (2016)
40. Yoon, H.J., Baltali, E., Zhao, K.D., Rebellato, J., Kademani, D., An, K.N., Keller, E.E.: Kinematic study of the temporomandibular joint in normal subjects and patients following unilateral temporomandibular joint arthrotomy with metal fossa-eminence partial joint replacement. J. Oral Maxillofac. Surg. **65**(8), 1569–1576 (2007)
41. Zhang, S., Gersdorff, N., Frahm, J.: Real-time magnetic resonance imaging of temporomandibular joint dynamics. Open Med. Imaging J. **5**(1), 1–9 (2011)
42. Zoss, G., Bradley, D., Bérard, P., Beeler, T.: An empirical rig for jaw animation. ACM Trans. Graph. **37**(4), 59:1–59:12 (2018)

Wenwu Yang is currently an associate professor at the School of Computer Science and Information Engineering, Zhejiang Gongshang University, Peoples Republic of China. He received his Ph.D. degree from the State Key Lab of CAD & CG, Zhejiang University in 2009. His research interests include cartoon animation, image editing, computer graphics, and geometric modeling.
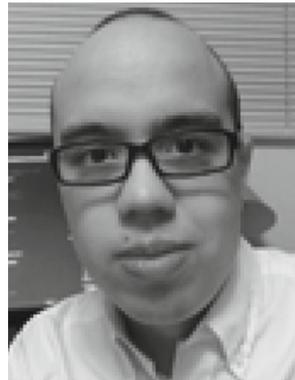
Nathan Marshak is currently a Ph.D. candidate at the University of Utah. The research focus of his work is mainly on physically based simulation. He received his Master of Engineering degree from the University of Pennsylvania.

Daniel Sýkora is an professor at the Department of Computer Graphics and Interaction, Faculty of Electrical Engineering, Czech Technical University in Prague where he also received his M.S. (2003) and Ph.D. (2007). After finishing Ph.D., Daniel spent 2 years as a postdoc at Trinity College Dublin. Daniel's research interests stem from his long-standing passion for traditional animation. He focuses on algorithms that can eliminate repetitive and time-consuming tasks while still being able to preserve full creative freedom of an artist.

Srikumar Ramalingam is an associate professor in the School of Computing at the University of Utah since 2017. Before that, he worked as a senior principal research scientist at Mitsubishi Electric Research Lab (MERL) since 2008. He received his Ph.D. in computer science and applied mathematics from INRIA Rhone-Alpes (France) in 2006, and his Ph.D. was funded by Marie Curie VisionTrain scholarship. His current research interests are in computer vision, machine learning, and discrete optimization.

Ladislav Kavan is an assistant professor of computer science at the University of Utah. Prior to joining Utah, he was an assistant professor at the University of Pennsylvania and research scientist at Disney Interactive Studios. Ladislav research focuses on interactive computer graphics, physics-based animation, and geometry processing. His goal is to combine computer graphics with biomechanics and medicine. Ladislav is a member of the ACM SIGGRAPH community and serves as an associate editor for ACM Transactions on Graphics.