3D human tongue reconstruction from single “in-the-wild” images

Stylianos Ploumpis\textsuperscript{1,2} * Stylianos Moschoglou\textsuperscript{1,2} * Vasileios Triantafyllou\textsuperscript{2}
Stefanos Zafeiriou\textsuperscript{1,2}

\textsuperscript{1}Imperial College London, UK \hspace{1cm} \textsuperscript{2}Huawei Technologies Co. Ltd
\textsuperscript{1}\{s.ploumpis, s.moschoglou, s.zafeiriou\}@imperial.ac.uk \hspace{1cm} \textsuperscript{2}vasilios.triantafyllou@huawei.com

Figure 1. We propose a framework that accurately derives the 3D tongue shape from single images. A high detailed 3D point cloud of the tongue surface and a full head topology along with the tongue expression can be estimated from the image domain. As we demonstrate, our framework is able to capture the tongue shape even in adverse “in-the-wild” conditions.

Abstract

3D face reconstruction from a single image is a task that has garnered increased interest in the Computer Vision community, especially due to its broad use in a number of applications such as realistic 3D avatar creation, pose invariant face recognition and face hallucination. Since the introduction of the 3D Morphable Model in the late 90’s, we witnessed an explosion of research aiming at particularly tackling this task. Nevertheless, despite the increasing level of detail in the 3D face reconstructions from single images mainly attributed to deep learning advances, finer and highly deformable components of the face such as the tongue are still absent from all 3D face models in the literature, although being very important for the reality of the 3D avatar representations. In this work we present the first, to the best of our knowledge, end-to-end trainable pipeline that accurately reconstructs the 3D face together with the tongue. Moreover, we make this pipeline robust in “in-the-wild” images by introducing a novel GAN method tailored for 3D tongue surface generation. Finally, we make publicly available to the community the first diverse tongue dataset, consisting of 1,800 raw scans of 700 individuals varying in gender, age, and ethnicity backgrounds*. As we demonstrate in an extensive series of quantitative as well as qualitative experiments, our model proves to be robust and realistically captures the 3D tongue structure, even in adverse “in-the-wild” conditions.

1. Introduction

Recently, 3D face reconstruction from single “in-the-wild” images has been a very active topic in Computer Vision with applications ranging from realistic 3D avatar creation to image imputation and face recognition [12, 14, 22, 37, 39, 44]. Nevertheless, despite the improvement in the quality of the 3D reconstructions, all of these methods do not accommodate any statistical variations in the oral cavity let alone a tongue template mesh. As a result, the oral region is completely disregarded from the final result. Being able to reconstruct the tongue expression has multiple advantages in various applications. First of all, the generated avatars would be more realistic and would be able

*Authors contributed equally.

*Project url: www.github.com/steliosploumpis/tongue
to mimic many more facial expressions. Moreover, speech animation tasks would be improved as the inclusion of the oral cavity plays a significant role. Finally, face recognition applications could be enhanced as more extreme poses and expressions would be modeled.

However, as we already pointed out, all of the current state-of-the-art (SOTA) methods [14, 39, 44] do not contain the tongue component in their implementations. This is because of two reasons: a) there is no publicly available tongue dataset, and b) it is very challenging to carry out 3D reconstruction of the face together with the tongue in “in-the-wild” conditions, because of the highly deformable nature of the human tongue.

To tackle the absence of tongue data, we collected a large and diverse dataset of textured 3D tongue point-clouds (more info about the data in Section 3). Having captured the data, we created a pipeline which is comprised of the following parts: a) a tongue point-cloud autoencoder (AE) which is used to derive useful 3D features of our raw collected 3D data, b) a tongue image encoder optimized based on the aforementioned 3D features, c) a shape decoder which translates the encoder outputs to the parameter space of the Universal Head Model (UHM) [33]. We should note that the UHM in our case is further rigged/modified so that it can model various tongue shapes/expressions, as explained in Section 3. We begin by training the AE in step a) and then we train steps b-c) in an end-to-end fashion so that the output tongue expression of the UHM model is as close as possible to the corresponding ground-truth 3D tongue point-cloud of the 2D tongue image.

Since there is a lack of ground-truth 3D tongue data corresponding to “in-the-wild” 2D tongue images, the pipeline we described so far is only trained using our collected data which were captured under controlled conditions. This results in sub-optimal performance in “in-the-wild” conditions. To remedy this, we developed a novel conditional GAN framework that is able to generate accurate 3D tongue point-clouds based on the image encoder outputs (step b) of the pipeline). Having created new image/point-cloud pairs of “in-the-wild” tongue data, we re-train the pipeline using also these new data. As we show in Section 4, this addition substantially improves the quality of the tongue reconstructions. To summarize, the contributions of our work are the following:

- We release a dataset of 1,800 raw tongue scans of various shapes and positions, corresponding to around 700 subjects. Being the first such diverse tongue dataset, it can be proven very useful to the community.
- We present a complete pipeline trained in an end-to-end fashion that is able to reconstruct the 3D face together with the tongue from a single image.
- To make this pipeline robust to “in-the-wild” images, we introduce a novel GAN framework which is able to accurately reconstruct 3D tongues from “in-the-wild” images with an increasing level of detail.

2. Related Work

Single view 3D reconstruction of human facial/head parts is undeniably an extremely valuable task in Computer Vision. However, it has posed many challenges to the research community, due to the fundamental depth ambiguities and the ill-posed nature of the problem. In order to constrain the ambiguity of the problem, many statistical parametric models have been introduced for different parts of the human face/head [2, 5, 26, 34].

Due to the increasing interest of facial analysis over the years the research community has mainly focused on human facial reconstructions. Since the inception of facial 3D Morphable Models (3DMMs) in [2], a myriad of scientific papers have been published focusing solely on the reconstruction of facial shape and appearance [3, 4, 14, 22]. Only recently with the emergence of 3D scanning data has the research interest shifted to other significant parts of the human head. A few head models have been introduced during the recent years but without any statistical craniofacial correlations [26, 36]. The first craniofacial 3DMM of the human head was introduced in [10] and later extended and leveraged into a 3D head reconstruction setting from unconstrained single images [34]. A few recent works tried to align a skull structure of the the human head with the facial topology [27, 28] in order to obtain a distribution of plausible face shapes given a skull shape.

Finer details of the human face/head started to appear with the introduction of 3D human ear modeling [46]. Ears are key structures of the human head that have an important contribution to the biometric recognition and general appearance of a person. The two foremost examples of ear models were introduced in [9, 47] but none of them was fused to a face/head in order to create a complete appearance.

Moreover, in an attempt to overcome the “uncanny valley” problem, a few approaches have tried to model the independent variations/movements of the human eye and the the facial eye region [1, 42]. These efforts are challenging due to the limited amount of data around the eye region and the extreme level of detail required for this task. Moving towards the oral cavity, teeth modeling was introduced in [40, 43], where the 3D structure of the teeth was recovered from 2D images via an elaborate optimization scheme.

Only very recently, a few approaches [25, 33] have tried to combine all of these aforementioned attributes of the human head (eyes, ears, teeth, and inner oral cavity) in order to build a complete model in terms of shape and texture, which
accurately represents the human head. Although these models include an oral topology, none of them deals with the dynamics of the tongue, something which is really important for speech animation and the overall realism of the avatar representation. To this end, in this work we aim at extending these approaches and paving the way towards a realistic human appearance by releasing a diverse 3D tongue dataset to the research community. We also present the first framework for accurate 3D human tongue reconstruction from single images.

3. 3D human tongue reconstruction

In this Section we present the complete tongue reconstruction pipeline. We begin by describing our collected 2D/3D tongue dataset and our manually rigged tongue dataset which is based on the UHM [33] template. We further provide details about the point-cloud AE, the image encoder, the shape decoder and the overall loss functions we used to optimize the pipeline for the tongue reconstruction. Moreover, we present the novel conditional GAN method which is able to accurately reconstruct 3D tongue point-clouds of “in-the-wild” tongue images. Finally, we explain how we used the generated point-clouds of the GAN to retrain the pipeline to achieve better results in “in-the-wild” conditions.

3.1. Tongue datasets

**TongueDB: the first 3D tongue dataset.** As mentioned in Section 1, we collected a large dataset comprising of textured 3D tongue scans. Our point cloud database, dubbed TongueDB, contains approximately 1,800 3D tongue scans which were captured during a special exhibition in the Science Museum, London. The subjects were instructed to perform a range of tongue expressions (i.e., tongue out left and right, tongue out center, tongue out center round, tongue out center extreme open mouth, tongue inside left and right, etc.). Some example images can be seen in Fig. 6. The capturing apparatus utilized for this task was a 3dMD 4 camera structured light stereo system, which produces high quality dense meshes. We recorded a total of 700 distinct subjects with available metadata about them, including their gender (42% male, 58% female), age, and ethnicity (82% White, 9% Asian, 3% Black and 6% other).

**Rigged tongue database.** In order to carry out 3D tongue and face reconstruction, we would need to use a face/head model. Nevertheless, one major drawback of all of the currently used face/head models [10, 26, 36] is that they are missing the tongue component. This is because it is a challenging task to non-rigidly capture in a fixed template the 3D topology of the oral cavity. These challenges include: a) the highly deformable nature of the tongue, b) the non-convexity of the mouth region, c) the specular texture of the teeth. In order to alleviate this issue, we constructed a synthetic 3D head and tongue dataset rigged by 3D artists. The artists used a proportion of the raw tongue scans as a guide for the manual sculpting by tracing the 3D details of the original scans. The raw scans chosen for 3D tracing were carefully selected to represent most of the shape variance existing in TongueDB. For our neutral mesh template we utilize the mean template of the UHM [33] as it provides all the necessary components of the human oral cavity in accordance with the entire head statistical structure. The resulting rigged tongue expressions rise at 75 distinct meshes. In order to further augment our synthetic dataset, we performed trilinear interpolation between the closest expression meshes and generated a total of \( n_t = 720 \) tongue expressions. Some example synthetic expressions are shown in Fig. 2. A standard PCA was applied on the interpolated meshes resulting in an orthogonal basis matrix \( U \in \mathbb{R}^{3N \times n_t} \) (where \( N \) are the mesh vertices and \( n_t = 110 \) the kept components). The PCA is performed on the entire set of head vertices and not solely on the oral cavity. In this way, it is more efficient afterwards to transfer the tongue expression from the mean head to a head with a different facial identity.

3.2. Method

**Tongue point-cloud AE.** In order to accurately reconstruct a tongue in its 3D form based on a 2D image, our image encoder needs to be guided by meaningful target labels which can capture all the desired 3D point-cloud information. These labels, denoted as \( y \in \mathbb{R}^{256} \), are learned by autoencoding the raw point-clouds of our dataset (i.e., the raw 3D tongue scans). For this task, we utilize a self organizing-map framework for hierarchical feature extrac-
The task of the tongue image encoder is to produce features which are close to the target 3D features \( y \) of the AE. To make the encoder robust to various camera angles or illuminations, we employ a rendering framework where we utilize the textured raw scans (TongueDB in Section 3.1). We render our 1.8K textured meshes with a pre-computed radiance transfer technique using spherical harmonics which efficiently represent global light scattering. Additionally, we use more than 15 different indoor scenes coupled with random light positions and mesh orientations around all 3D axes, resulting in approximately 100K images. As an encoder we used a ResNet-50 [17] model pre-trained on ImageNet [11] and fine-tuned it on our dataset. In particular, we modified the last layer of the network to output a vector \( \tilde{y} \in \mathbb{R}^{256} \) similar to the dimension of the ground truth vector \( y \).

**Shape decoder.** In order to decode the encoder \( \tilde{y} \) labels into meaningful tongue shapes, we use the synthetic PCA model \( U_t \) of the rigged tongue expression dataset. To this end, after producing the \( \tilde{y} \) labels, we utilize a standard multi-layer perceptron (MLP) which works as a regression scheme to the latent parameters \( p_t \in \mathbb{R}^{110} \) of the synthetic PCA tongue model. The statistical nature of the PCA model helps us constrain the final result during training and ensures meaningful deformations which lie inside the spectrum of our rigged/modified UHM model.

**Pipeline training.** During training, we first train the point-cloud AE on its own and then train the pipeline of the image encoder and shape decoder in an end-to-end fashion. To optimize the pipeline, we apply a total of 6 losses with each one contributing to the quality of the final result. The first 2 losses are calculated between the predicted tongue expression of the rigged/modified UHM model and the ground-truth tongue point-cloud of the corresponding input image. Similarly to [41], we adopt a Chamfer loss [13] \( L_{CD} \) to optimize the position of the resulting template points as well as a normal loss \( L_n \) to correct the orientation of the mesh. In order to compute an accurate Chamfer loss, we only utilize a small area around the oral cavity which is defined based on the ground-truth point-cloud. Additionally, we calculate a Laplacian regularization \( L_l \) loss between our predicted mesh and the mean shape of the PCA model in order to prevent the vertices from moving too freely outside the mean positions and constrain the resulting shape to be smooth. An edge length loss \( L_e \) is also introduced which penalizes any flying vertices (outliers). Finally, we employ a collision loss \( L_c \) which prevents the points of the tongue to penetrate the surface of the oral cavity and is formulated as the sum of each collision error around the 12 mouth landmarks of the UHM template (as illustrated in the supplementary material):

\[
L_c = \frac{1}{N} \sum_{k=0}^{11} \sum_{i=0}^{N-1} \max \left( 0, d_k^i \right) \tag{1}
\]

\[
d_k^i = r^2 - (q_k^1 - x_k)^2 - (q_k^2 - y_k)^2 - (q_k^3 - z_k)^2
\]

the \( L_c \) is calculated as the sum of distances of each collided point \( q^i = \{q_k^1, q_k^2, q_k^3\} \) to the sphere \( k \) with center the landmark coordinates \( x_k, y_k, z_k \) and radius \( r = 1, 5cm \). Lastly, we impose a final L2 loss \( L_y \) in the intermediate step of our pipeline where we constrain the predicted \( \tilde{y} \) encoded features to be as close as possible to the ground-truth features \( y \) of the corresponding autoencoded point-cloud. This loss is of paramount importance because: a) the \( \tilde{y} \) features in this way contain rich 3D information invariant to texture/illumination variations and b) our “in-the-wild” ex-

### Figure 3
An illustration of our tongue reconstruction framework. First we train the point-cloud AE on its own to get meaningful 3D features \( \tilde{y} \) and then we trained the image encoder /shape decoder using a number of different losses as explained in Section 3.
tension which we introduce later is based on a generative point-cloud framework that relies on such rich 3D features.

The final loss function $L_{total}$ is given by:

$$L_{total} = \lambda_1 L_{CD} + \lambda_2 L_n + \lambda_3 L_l + \lambda_4 L_c + \lambda_5 L_e + \lambda_6 L_y$$

(2)

where $\lambda_1, \ldots, \lambda_6$ are training hyper-parameters. During inference, the encoder network takes as an input a single tongue image and predicts a 3D embedding $\tilde{y}$, which is later transformed to the corresponding $p_t$ parameters of the synthetic expression model, through the MLP between the two latent spaces. Finally, we apply the PCA model of the rigged head model on these $p_t$ parameters to derive the final mesh of the head with the tongue expression. An overview of the methodology can be seen in Fig. 3.

3.3. TongueGAN for “in-the-wild” reconstruction

Although the pipeline presented in Section 3.2 provides a good estimation of the tongue pose in the test set of our collected data, it does not perform very well in “in-the-wild” images (Fig. 9). This behavior is expected because our collected data were captured in controlled conditions and the training of the encoder was carried out only with rendered images which do not fully mimic “in-the-wild” conditions. To make our approach robust in “in-the-wild” images too, we would need to further train the pipeline by using also such data. However, for “in-the-wild” collected images from the web, we do not have their corresponding 3D tongue point-clouds. As a result, to use “in-the-wild” data in the pipeline, we would first need to have a method that can learn the distribution of our collected 3D tongue data and generalize well.

Finding a method to generate novel 3D tongues is tricky. This is because of several unique properties of the human tongue: a) it is a highly deformable object, so we cannot register our collected data in a reference template and apply relevant methods [6,31,35], b) it is a non-watertight surface (i.e., it contains holes) so we cannot also use any implicit function approximations methods [29,32,38] or volumetric approaches [19,20,45]. Therefore, having excluded the aforementioned categories, we decided to use GANs [15] for the 3D tongue surface generations.

In order to generate accurate point-clouds that correspond to certain tongue images, our GAN, dubbed as TongueGAN, needs to be guided by meaningful labels which can capture all the desired 3D surface information. These labels are provided by the trained point-cloud AE as described in Section 3. Since the generation is driven by labels, TongueGAN is a conditional one [30]. In particular, given a label denoted as $y$ and a random Gaussian noise $z \in \mathbb{R}^{128}$, the generator $G$ produces a novel point-cloud point $G(z, y) \in \mathbb{R}^3$, which we denote as $\tilde{x}_t$, that belongs to the tongue surface represented by the label $y$. On the other hand, the discriminator $D$ receives as inputs the label $y$, a real point-cloud point $x_t$ (which belongs to the tongue represented by the label $y$) and the generator output $\tilde{x}_t$ and tries to discriminate the fake (i.e., generated) from the real point. In the mathematical parlance, this is described as:

$$L_D = \mathbb{E}_{x_t} [\log D(x_t, y)] - \mathbb{E}_{\tilde{x}_t} [\log D(\tilde{x}_t, y)],$$

$$L_G = \mathbb{E}_{\tilde{x}_t} [\log D(\tilde{x}_t, y)]$$

(3)

where $D$ tries to maximize $L_D$, whereas $G$ tries to minimize $L_G$. Please note that instead of generating whole point-clouds for every provided pair $(z, y)$ of noise and label, respectively, we merely generate a point corresponding to the surface which the label $y$ represents. That confers several advantages in comparison to the rest of the methods in the literature, such as: a) we do not need to have in our training set point-clouds with the same number of points and as a result we can train our GAN without any data pre-processing on the raw point-clouds, which do not have a
fixed number of points among them, b) when it comes to generating point-clouds corresponding to a particular label, we can generate on demand as many points as we want and, contrary to the rest of the literature, we are not constrained by any initially fixed resolution.

For the TongueGAN loss we chose the Wasserstein loss with Gradient Penalty (WGAN with GP) [16] due to its stability and good performance throughout the training process. As far as the architecture is concerned, we turned our attention to the recently proposed II-Nets [7, 8], which are easy to implement. In particular, we use our own, custom modification of II-Nets to accommodate the needs of our task. A graphical presentation of the network structure is provided in Fig. 4.

### 3.3.1 GAN loss for accurate surface approximation

Even though, as can be clearly seen in the experiments, our custom II-Nets modification along with the WGAN-GP loss significantly improve upon the point-cloud GAN [23], there is still room for improvement as the point-cloud representations are not ideally reconstructed (see Table 1).

We primarily attribute this to the strict behavior of the discriminator in GANs (i.e., deciding in our case whether a generated point matches exactly a point of the target point-cloud). This rigidity, especially in the early steps of the training process, is not very helpful, as the generator struggles to learn the real distribution of the point-clouds (i.e., all of the generated points are discarded as fake by the discriminator with high confidence). To remedy this, we slightly softened the discriminator, especially in the initial steps, by slightly modifying the real points fed to it. To achieve this, instead of directly feeding a real point $x_t$ corresponding to a label $y$ to the discriminator, we feed the following:

$$x_{t|y} \sim \mathcal{N}(x_t|y, \sigma_e I) \quad (4)$$

where $\mathcal{N}(x_{t|y}, \sigma_e)$ is a multi-variate normal distribution with mean $x_t$ and (isotropic) variance $\sigma_e$. The variance $\sigma_e$ is not dependent on the label $y$. It is only dependent on the epoch $e$. By employing (4), especially when the training process commences, the generator can better learn the actual distribution as it does not get severely punished by the discriminator when it slightly misses out the actual surface (see the accompanying Fig. 5 for a visual understanding). As can be also seen in the experiments, this addition yields better results and stabilizes the training even further. We begin the training with a relatively small value for the variance and further linearly reduce it as we go along with the training process till it basically becomes zero towards the final epochs. This is verified empirically in Section 4.

### 3.3.2 Re-training the pipeline

After the training is complete, we use the trained generator together with the trained encoder from Section 3.2. We create “in-the-wild” pairs of 2D/3D data as follows: we feed the 2D “in-the-wild” image to the encoder and get the label $\tilde{y}$. We then use this label $\tilde{y}$ and the generator to produce a 3D point-cloud of the input image. As we can see in Fig. 8, although TongueGAN is trained only on our collected data, it is able to generalize very well in “in-the-wild” images and as a result we can use it to create 2D/3D tongue pairs. We apply this process to a number of “in-the-wild” images to create multiple pairs. Finally, we re-train the pipeline in Section 3.2, using also the aforementioned pairs.

### 4. Experiments

In this section we provide details regarding the training and we outline a series of quantitative as well as qualitative experiments under control and “in-the-wild” conditions.

The MLP utilized for the regression between the labels $\tilde{y}$ and the PCA parameters $p_t$ in Section 3.2 has a structure of $(256, 128, 110)$ with a ReLU activation in the intermediate layers. The hyper parameters of (2) which balance the losses are $\lambda_1 = 1.2, \lambda_2 = 1.6e-4, \lambda_3 = 0.4, \lambda_4 = 0.2, \lambda_5 = 0.8$ and $\lambda_6 = 1.5$. As described in Section 3.3, for TongueGAN we used a variant of WGAN with GP [16], which includes the injection mechanism [8], as well as the surface loss function presented in Section 3.3.1. More specifically, we utilized a 9-layer Generator ($G$) and a 8-layer Discriminator ($D$) with a total number of parameters of about $8 \times 10^6$ and $4 \times 10^6$, respectively. We trained TongueGAN using the Adam optimizer [21] with $(\beta_1 = 0, \beta_2 = 0.9)$. We also trained with a batch size of 2048 for a total of $10^6$ iterations. Following the idea introduced in [18], we use individual learning rates for $D$ and $G$. 

![Figure 5. A visual description of the mechanism explained in Section 3.3.1. a) a raw tongue mesh along with a green section. b) 3D points (green color) that belong to the tongue surface and lie upon the section and with lighter green we depict the accepted areas from which points can be sampled. c) a zoomed-in area.](image-url)
with values of $1 \times 10^{-4}$ and $1 \times 10^{-5}$, respectively. Finally, we start training with the variance $\sigma_e$ in (4) being $5 \times 10^{-3}$ and we linearly decrease it by 10% every $50 \times 10^3$ steps. The exact network structures are deferred to the supplementary material with more details.

### 4.1. 3D reconstruction in control conditions

In this set of experiments, we used 90% of TongueDB for training and the rest for testing. Due to the intricacy of the tongue as a surface (as we explained in detail in Section 3.3), we decided to use a GAN for the tongue surface generation part. Moreover, an extra reason to utilize a GAN for the training is the fact that it is able to generalize very well in unseen labels during testing. To the best of our knowledge, the only method which has been introduced in the literature and is able to carry out point-cloud generations based on unseen labels is PointCloud GAN (PC-GAN) [23]. Consequently, in what follows we draw comparisons against PC-GAN [23] and another two variants of TongueGAN, namely: a) TongueGAN_v1, which is the same as TongueGAN with the only difference being that the novel loss function (Section 3.3.1) is not available in this version, and b) TongueGAN_v2, which is the typical GAN structure where, instead of the injections we have simple concatenations along the layers. Finally, we also report the results for the regressed tongue expression of the shape model after retraining the pipeline (3.2) as explained in Section 3.3.2. (referred to as Tongue-Reg). For this, we only take into account a small patch around the oral cavity defined by the ground truth point cloud, in order to deduce a correct error.

Quantitative results are provided in Table 1 and qualitative results are presented in Fig. 7. For the quantitative results, we utilize the test set of our TongueDB and we measure the error based on the two commonly used type of distances when it comes to unordered 3D data, namely Chamfer Distance (CD) and Earth Mover’s Distance (EMD) [13]. As can be clearly seen in all of the comparisons, TongueGAN outperforms the compared methods by a large margin whereas Tongue-Reg outperforms the rest of the methods.

### 4.2. 3D reconstruction in “in-the-wild” conditions

In this Section, we attempt to reconstruct the 3D surface of the tongue together with the entire head structure from “in-the-wild” images. In this set of experiments, we used all of TongueDB for training. We also added to our training data another 5K “in-the-wild” tongue images and created their 3D point-clouds using TongueGAN. Using all these data, we re-trained the pipeline according to Section 3.3.2. The results are only visual as we do not have ground-truth point-cloud data to report quantitative comparisons. Regarding the comparisons, we should note that PointCloud GAN [23] cannot be used in these experiments, as in order to work in the conditional setting, it needs as input the actual ground-truth point-cloud it attempts to reconstruct, something which we do not have at our disposal. Given that the TongueGAN variations (i.e., TongueGAN_v1 and TongueGAN_v2) perform worse than PointCloud GAN [23], we only present results in this Section regarding TongueGAN.
Figure 8. 3D head reconstructions with tongue animations from “in-the-wild” images. From left to right, we depict the “in-the-wild” image, then the point-cloud generations from two viewpoints and finally the 3D head reconstruction with a zoomed-in area around the oral cavity.

Since our tongue regression method is based on the mean template mesh of UHM we can easily utilize the pipeline presented in [33] in order to extent our approach to a particular facial identity. We begin by fitting a facial mesh to the image domain in order to get the 2D/3D landmarks and the identity of the subject and then we regress to the full head topology based on the UHM model. After reconstructing the head shape we crop the image around the projected 2D mouth landmarks. We then feed this cropped image to the re-trained pipeline and get the mean head shape with the tongue expression as mentioned in Section 3. Finally we merge the predicted tongue shape with the associated identity by treating the predicted tongue expression as a separate blend-shape. Some 3D reconstructions can be seen in Fig. 8. As evidenced, our pipeline is able to accurately reconstruct the 3D tongue details even in “in-the-wild” conditions. Additional tongue reconstructions of our method before and after the re-training framework against state-of-the-art methods can be seen in Fig. 9. To further empirically validate that TongueGAN is able to capture the 3D structures of random tongues that are not included in the training set we provide linear interpolations between unseen latent features in the supplementary material.

5. Conclusion and Limitations

In this work, we presented the first pipeline which is able to perform 3D head and tongue reconstruction from a single image. To achieve this, we collected the first diverse tongue dataset with various tongue shapes and positions which we make publicly available to the research community. To also make this pipeline robust in “in-the-wild” images and to mitigate the absence of their corresponding ground-truth 3D tongue data, we introduced the first GAN method that is tailored for accurately reconstructing the 3D surface of a tongue from 2D images. As we show in a series of experiments, we are now able to accurately carry out 3D head reconstruction together with the tongue from a single image and thus create more realistic 3D avatars. However, people with extreme mouth and tongue poses in combination with images taken from difficult angles will cause failures in: i) the detection of the face, ii) the mouth landmark localization, iii) the regression of the tongue expression. Extreme tongue expressions such as tongue folding in between the teeth and inward/sideways twisting or rolling are not possible to reconstruct.

Acknowledgements: S. Zafeiriou acknowledges funding from the EPSRC Fellowship DEFORM (EP/S010203/1) and S. Moschoglou from the EPSRC DTA (EP/N509486/1).
References


[31] Stylianos Moschoglou, Stylianos Ploumpis, and Mihalis Nicolaou. 3dfacegan: Adversarial nets for 3d face representation, generation, and translation. 5