UNIVERSITY OF TARTU
Faculty of Science and Technology
Institute of Technology

Mehin Salimli

3D Face Reconstruction from a Single 2D Image

Bachelor's Thesis (12 ECTS)

Curriculum Science and Technology

Supervisor:
Rain Eric Haamer, MSc

Tartu 2021

3D Face Reconstruction from a Single 2D Image

Abstract:

3D face reconstruction is the process of creating a 3D representation of a real human face. 3D face models have several applications like face recognition, 3D games, human-machine interaction, and plastic surgery simulations. Recently there has been a lot of research on deep learning methods for 3D face reconstruction from 2D face images. In this thesis, three deep learning-based methods for 3d reconstruction from a single image are reviewed. A new texturing method for 3D face models that uses the input photo as a UV texture image is proposed. Image warping is used to modify the input photo for this purpose. Warping is achieved using facial landmark detection and triangle meshes. A survey is conducted to assess the three face reconstruction methods and the proposed texturing method.

Keywords:

Computer vision, Deep learning, 3D reconstruction, 3D modeling, UV mapping, image warping, texturing

CERCS:

T111: Imaging, image processing

3D Nägude Rekonstrueerimine ühest 2D pildist

Lühikokkuvõte:

3D nägude rekonstruktsioon on protsess, kus luuakse 3D representatsioon inimese näost. 3D näomudelitel on erinevaid rakendusi nagu näotuvastus, 3D videomängud, inimese-arvuti interaktsioon ja plastilise kirurgia simulatsioonid. Viimasel ajal on tehtud palju uurimistööd süvaõppe-põhisel 3D rekonstruktsioonil 2D näopiltidest. Käesolevas bakalaureusetöös antakse ülevaade kolmest süvaõppe-põhisest 3D rekonstruktsiooni meetodist, mis kasutavad ühte 2D sisendpilti. Samuti loodi 3D näomudelite tekstureerimise meetod, mis kasutab sisendpilti mudeli UV tekstuurina. Selle saavutamiseks kasutati sisendpildi moonutamist, mis tehti näo ankurpunktide abil. Et hinnata kolme nägude rekonstrueerimise meetodit ja loodud tekstureerimist, tehti uuring kus paluti osalejatel neid oma arvamuse põhjal järjestada.

Võtmesõnad:

Tehisnägemine, masinõpe, 3D rekonstruktsioon, 3D modelleerimine, UV tekstuur, pildi moonutamine, 3D tekstuur

CERCS:

T111: Pilditehnika

Table of Contents

1 Introduction

3D modeling is a technique in computer graphics used for generating a three-dimensional representation of an object or surface. 3D modeling is a growing field and is highly in demand. 3D modeling has a wide variety of use cases ranging from generating interactive representations of organs to creating virtual architectural tours. Other applications include animation, gaming, virtual reality, science, geoscience, engineering, etc. [1]

The more precise task of creating 3D models of existing human faces is called 3D face reconstruction. 3D face reconstruction is used in cosmetic surgery simulations, virtual reality simulations, face morphing, face recognition, human-machine interactions, and animation. [2]

There are several methods for producing 3D models[1], which can be either manual or automatic. Manual modeling is similar to sculpting; an artist deforms a geometric shape like a cube or sphere until the model is in the desired shape. However, this kind of modeling is expensive and time-consuming. Automatic image-based 3D reconstruction is another method of 3D modeling. It is the process of automatically building a three-dimensional representation of objects from one or multiple 2D images. [3]

Traditional methods for 3D reconstruction are stereo-based techniques, monocular cues methods, and 3D laser scanning. Stereo-based techniques use multiple images captured from slightly different viewpoints. Monocular cues methods are used to create 3D reconstruction by using 2D characteristics like shading, silhouettes, texture, etc. [3]. These two methods are often not practical as they require multiple images of the same object captured with well-calibrated cameras [3]. 3D laser scanning[2] uses the process of analyzing real objects to create point clouds from the collected data. There are several drawbacks to using 3D laser scanners for 3D face reconstructions, such as cost and size of the scanners and the use of the laser can be dangerous for the eyes during scanning [4].

Humans can infer the 3D geometry of any object by just looking at a 2D image of the object and even tell what it would look like from different angles. It is because we have built prior knowledge from previously seen objects and scenes. 3D reconstruction of objects can be achieved from a single or multiple 2D images by leveraging prior knowledge using deep neural networks. Using neural networks is a fairly recent technique in the 3D modeling field,

[1] https://en.wikipedia.org/wiki/3D_modeling

[2] https://en.wikipedia.org/wiki/3D_scanning

but it has already shown exciting and promising results. In this thesis different deep learning methods for 3D face reconstruction will be analyzed. [3]

In this thesis, three existing neural network-based 3D reconstruction methods will be discussed and compared. As these deep learning methods focus primarily on face shape, the textures of the generated models are typically relatively low quality. In this thesis, some of the existing texture improvement methods will be described, and a new texture generation method will be proposed.

A survey will be conducted to compare the three different face reconstruction methods and also the proposed improved texture method. This research could help professionals choose the best-suited method for themselves while also providing a simple method for highly detailed textures.

This thesis is structured as follows. In Chapter 2, existing 3D methods of face reconstruction from 2D images with deep learning are described. In Chapter 3, existing texturing methods are described, and a new texture generation method is proposed. In Chapter 4, the conducted survey is described. In Chapter 5, the results of the thesis are discussed and analyzed. In Chapter 6, the thesis is summarized. In Appendix I, the column charts of the survey results are given. In Appendix II, the source code for the improved texturing method is included.

2 3D Reconstruction Methods

The goal of 3D reconstruction from a single or multiple 2D images is to infer the 3D structure of natural objects. There are two general approaches to creating 3D face models from 2D images using neural networks [3]. These approaches are described in the following chapters.

2.1 Model-based Methods

Recently, most successful 3D face reconstruction with deep learning methods have used the 3D Morphable Model of Blanz and Vetter [5] to create a 3D face shape from a 2D image [3]. 3D Morphable Model (3DMM) is a generative model that is derived from a dataset of 3D face models. It combines 3D shape and texture information of all the example faces into one vector space of faces. New faces can be generated as linear combinations of these examples. A pretty close 3D model of the input face image can be created by finding a vector in the 3DMM vector space that represents the input face [5].

2.1.1 Accurate 3D Face Reconstruction with Weakly-Supervised Learning: From Single Image to Image Set

As ground truth 3D data is scarce and approaches to using synthetic data may yield bad results due to imperfect training labels, Deng et al. (2020) [6] in their paper use weakly supervised learning where they train the network without shape labels where the loss function measures the discrepancy between the input photo and the rendered image. For example, one method for this is using pixel-wise photometric difference as a training loss. Another method for this is using perception-level loss by measuring distances between facial features.

Using only pixel-wise loss may suffer from an issue where low error can be obtained with unsatisfactory shapes and using only perceptual loss can yield suboptimal results due to ignoring pixel-wise consistency with the raw image signal. Therefore Deng et al. use a hybrid-level loss function that integrates both of them. They also use a skin color-based photometric error attention strategy, which allows the reconstruction of 3D faces where the input faces have occlusion, beard, or heavy make-up. The model trains an off-the-shelf CNN to predict 3D Morphable Model (3DMM) coefficients. Moreover, they regress illumination and face pose coefficients for rendering during training.

For rendering, apart from the 3D face model, they use an illumination model to approximate the scene illumination. They also use a perspective camera with an empirically selected focal length for the 3D-2D projection.

Additionally, Deng et al. propose a method to learn 3D face aggregation from multiple images in an unsupervised fashion. However, this method will not be described further, as this thesis only focuses on 3D face reconstruction from a single 2D image.

The neural network utilized in this paper is called ResNet-50 (referred to as R-Net here) to regress coefficients, which are represented by vectors. Figure 1 illustrates the used framework and training pipeline.

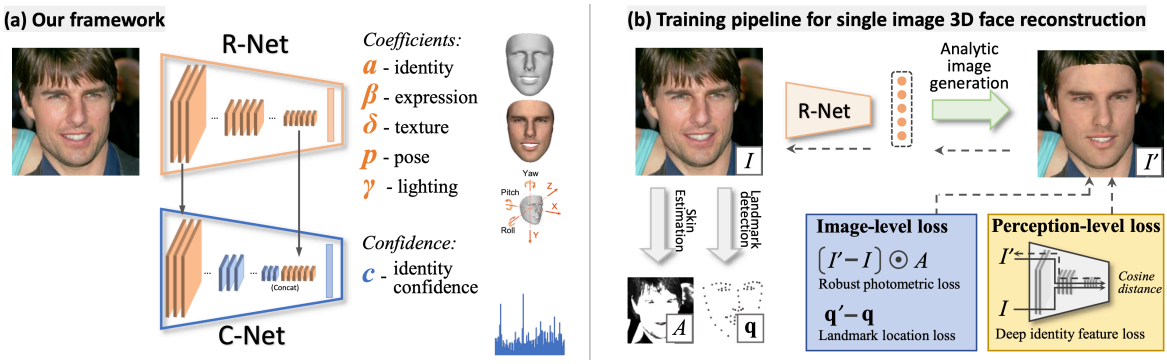


Figure 1. A) The framework of Deng et al's method. B) Training pipeline for single image 3D face reconstruction. [6]

For training, a loss function with pixel-value loss is used to compare pixel values of the original photo and the generated image. The loss function for this was a robust, skin-aware photometric loss instead of a naive one. Also, to gain robustness to occlusions and other challenging appearances such as beard and heavy make-up, a skin-color probability is computed for each pixel. Figure 2 shows the benefit of using this skin attention mask.

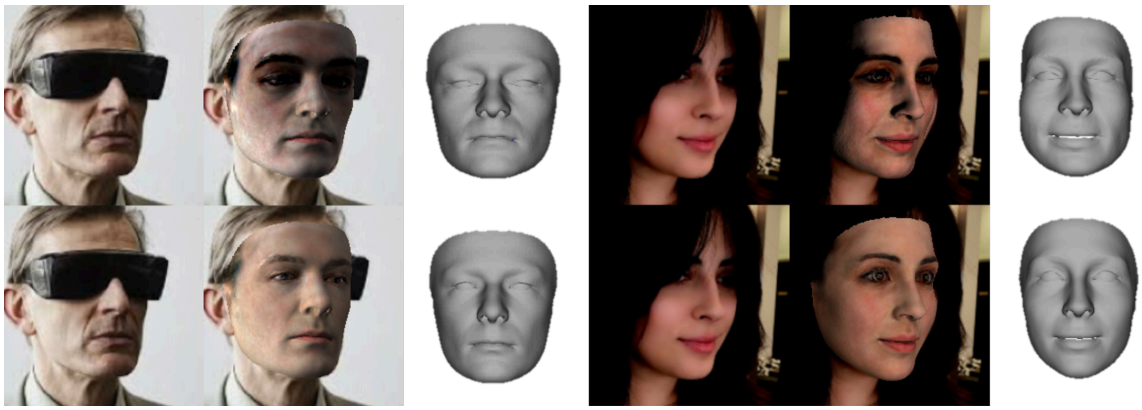


Figure 2. Comparison between using (top row) and not using (bottom row) the skin attention mask. [6]

Additionally, a landmark loss function is utilized, where they use landmark locations on the 2D images as weak supervision to train the network. They run a state-of-the-art 3D face alignment method to detect 68 landmarks of the training images. During training, they use the 3D landmark vertices of the reconstructed shape to compute the loss function value.

For perception-level loss, a facial recognition network is used to compare the facial features of the two images. For this purpose, FaceNet is trained using a face recognition dataset with 3 million face images of 50 thousand identities obtained from the Internet. In Figure 3, with the perceptual loss, the textures appear sharper, and the shapes are more faithful to the original photo.

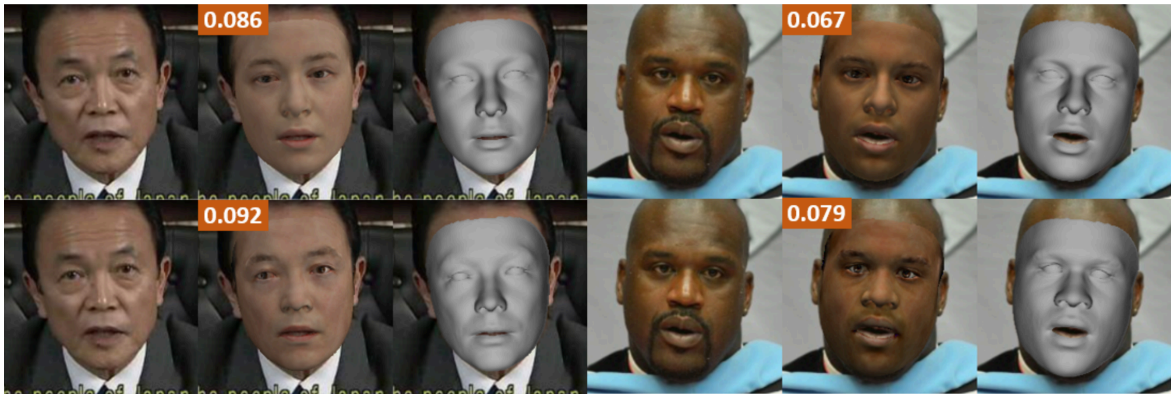


Figure 3. Comparison between using only image-level loss (top row) and using image-level loss and perceptual loss together (bottom row). The numbers represent photometric errors. [6]

Overall, Deng et al. proposed a method for single image 3D face reconstruction with a Convolutional Neural Network without using 3D ground truth data. The method uses a hybrid-level loss function for unsupervised learning and shows a state-of-the-art performance.

2.1.2 Towards Fast, Accurate, and Stable 3D Dense Face Alignment

Guo et al. (2021) [7] in their paper use a neural network called MobileNet to regress 3DMM parameters using supervised learning. The goal of the paper is to accelerate the speed to real-time (CPU) and demonstrate a state-of-the-art performance at the same time. They use two loss terms to handle the optimization problem of parameter regression and propose an optimization method called meta-joint to combine their advantages.

They also propose a virtual synthesis method to transform one still image into a short video that incorporates in-plane and out-of-plane face movement. However, this approach will not be described further, as this thesis focuses only on 3D face reconstruction from a single 2D image.

Figure 4 illustrates the overall pipeline for this method. The architecture consists of the following steps. Firstly, by using a lightweight network like MobileNet, the 3DMM parameters are predicted. Next, meta-joint optimization and landmark regression regularization are applied for training.

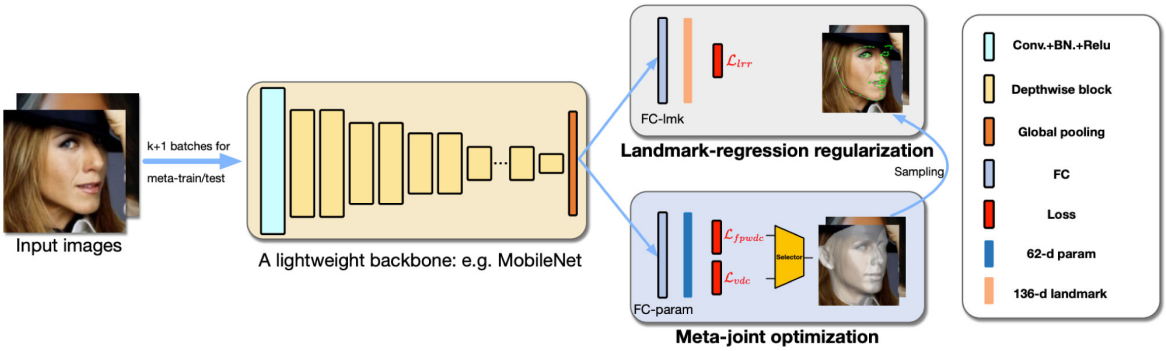


Figure 4. The framework of Guo et al's method. [7]

They use a Vertex Distance Cost loss function (VDC) that minimizes the vertex distances between the fitted 3D face and the ground truth model. Parameter distance cost minimizes the differences in the fitted and ground truth model 3DMM parameters. Weighted Parameter Distance Cost (WPDC) loss function that they use assigns different weights to each 3DMM coefficient. However, WPDC can perform slowly, which is a bottleneck for fast training. Thus, they design a fast implementation of WPDC named fWPDC, which fWPDC performs 10 times faster than the WPDC, while preserving the same outputs.

WPDC/fWPDC is suitable for parameter regression since each parameter is appropriately weighted, while VDC can directly reflect the quality of the 3D face reconstructed from parameters. With their meta-joint optimization strategy, they dynamically combine fWPDC and VDC in the training process. Figure 5 illustrates the overview of the meta-joint optimization method.

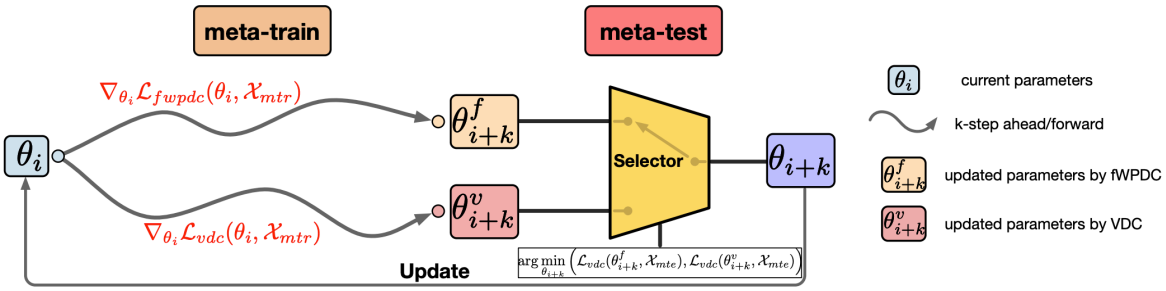


Figure 5. Meta-joint optimization [7]

As 2D sparse landmarks are often used in 3D face reconstruction for easier regression of parameters, they additionally propose a landmark-regression regularization method to achieve higher accuracy.

Overall, Guo et al. proposed a method for single image 3D face reconstruction that achieved fast and accurate 3D dense face alignment. The method uses a meta-joint optimization and landmark-regression regularization.

2.2 Model-free methods

The downside of using model-based methods is the restricted performance due to the limitation of 3D space defined by face model basis or template [8]. Model-free methods such as volumetric grids do not suffer from this problem [3].

2.2.1 Joint 3D Face Reconstruction and Dense Alignment with Position Map Regression Network

Feng et al. (2018) [8] in their paper, propose a model-free method called Position Map Regression Network. The main goal of their method is to solve the problems of face alignment and 3D face reconstruction together in an end-to-end fashion without the restriction of a low-dimensional solution space like 3DMM.

Because the method is model-free, they regress coordinates of all 3D points, unlike the model-based methods, which perform regression of only the model parameters. To represent the 3D face geometry properly so that it can be predicted directly with the neural network, they use a UV position map technique. The UV position map is a 2D image recording positions of 3D points in UV space (Figure 6: second image). Here, UV space is used to store the 3D coordinates of points from the face model. The structure of the neural network is a simple encoder-decoder. UV coordinates were created based on 3DMM to preserve the semantic meaning of points in the position map.

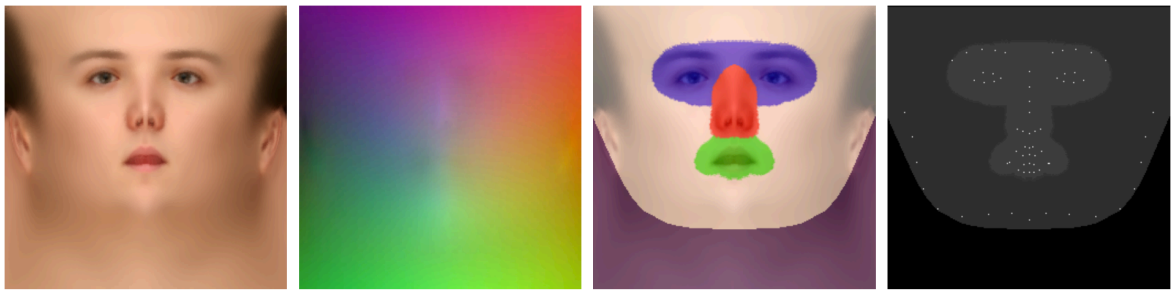


Figure 6. Maps used by the method. From left to right: the UV texture map, the UV position map, the segmentation map and the weight mask. [8]

Feng et al. propose a novel loss function for training. They use a weight mask in their loss function to put more focus on the discriminative features of the face's central part. The weight mask is a 2D grey image (Figure 6: last image) recording the weights of each point on a UV position map. It has the same size and pixel-to-pixel correspondence to the position map. The points are categorized into four groups. Each of them has its own weight in the loss function. The positions of 68 facial landmarks have the highest weight for the network to learn accurate locations of these points. Neck region points are assigned no weight because they want to ignore that part in the reconstruction.

For training the model, they use the 300W-LP dataset, which contains 2D images and estimated 3DMM parameters. The training process needs 2D face images and their corresponding 3D point clouds.

In this paper, Feng et al. proposed an end-to-end joint 3D reconstruction and dense alignment. Their method has achieved fast and accurate 3D face reconstruction. For reconstruction, they used a UV position map to regress the 3D shape along with the semantic meanings. Their comprehensive experiments have shown the effectiveness and efficiency of the method, which is robust to poses, illuminations, and occlusions.

3 Texturing

The 3D reconstruction methods described in the previous chapter use low fidelity vertex-based texturing and it was evident that texturing was not part of their focus. Furthermore, Deng et al. [6] use 3DMM-based texturing, which lacks fine details such as wrinkles and birthmarks, because these details get lost when merging models with different textures.

Much research is being done to improve the facial textures of 3D reconstruction models. In the following subchapters, some of the existing texturing methods are described, and a texture generation method is proposed.

3.1 Existing Texturing Methods

Lin et al. (2020) [9] proposed a method to reconstruct 3D facial shapes with high-fidelity textures from single-view images, without the need to capture a large-scale face texture database for training. They refined the initial texture generated by a 3DMM-based method with facial details from the input image using graph convolutional networks (GCN) [10]. They use Deng et al.'s (2020) [6] approach of regressing 3DMM parameters to build a model with simple texture and implement their GCN to refine the texture. Basel Face Model (BFM) [11] is used as the 3D morphable face model.

Quantitative and qualitative comparisons with the most recent methods were conducted. Qualitative comparison results showed that this method outperforms others with more detailed texture information. For quantitative comparison, they employ the metrics on 2D images since they do not have 3D ground-truth data. They use peak signal-to-noise ratio (PSNR) and structural similarity index (SSIM) to measure the results at pixel level. They use LightCNN and evoLVe face recognition networks for measurements at the perception level. The results are better than others in multiple evaluation metrics. A lower L1 distance and higher PSNR and SSIM indicate that the reconstructed 3D face textures are closer to the input images in pixel level. Both state-of-the-art face recognition networks show that the rendered results and the input images are more likely to be the same person than previous methods. Although their method outputs a full textured model, it does not match the fidelity of the original image.

Deng et al. (2017) [12] achieved 94.05% state-of-the-art verification accuracy by training a Deep Convolutional Neural Network to complete the facial UV map extracted from in-the-wild images to obtain visually realistic, identity-preserved, and semantically correct textures. They proposed a Generative Adversarial Network for UV completion, which uses

a UV generation module, two discriminators, and an additional module for face identity preservation. Their study included a large dataset of completed UV maps. While they can achieve high fidelity textures, their method requires a large-scale database of high-resolution UV maps, which is not publicly available.

3.2 Texture Generation

As the input for the texture generation proposed in this thesis, a face photo and the corresponding face model produced by Deng et al.'s method is used. Deng et al.'s method was selected as it seemed to generate more realistic and accurate face shapes than the other methods described in the previous chapters. This choice was later confirmed by the survey that will be described in Chapter 5.

The input model uses a vertex-based texture, which means that each vertex in the mesh has its own color, and each face gets its texture by interpolating its vertices' colors. This means that the fidelity of the texture depends on the number of vertices. The method proposed in this thesis for texturing uses an image-based texture, which means an image is used to store the texture information of the model. In this case, a modified version of the input photo is used as the texture image.

The texture created using this method preserves the fine details that are present on the input photo. However, it should be noted that only the parts of the model that are visible on the photo get the desired texture. The proposed method is described in detail in the following subchapters.

3.2.1 Generating a UV map

UV mapping[3] is the first step in the proposed texture generation method. UV mapping is the process of projecting a 2D texture onto a 3D model's surface. The letters "U" and "V" denote the two axes of the 2D texture.

The UV map is made up of vertices and faces. Each face of the UV map corresponds to a face on the 3D model and an area on the texture image. As the input model is represented as an OBJ-file, the UV mapping information will also be stored in this file. The link to the texture image is added to a MTL-file which will accompany the OBJ-file.

In this thesis, a UV map is generated by using vertex and face information from the input model. When looking at the input model from the Z direction in a 3D scheme, the model's

[3] https://conceptartempire.com/uv-mapping-unwrapping/

face is rotated the same way it was in the input photo (Figure 7). For texturing the parts of the model's face that were visible on the input photo, a UV mapping can be created by using the vertices of the model without their Z coordinates as the UV vertices, and the input model's faces as the UV faces. When the input photo is aligned to the created UV mapping, it can be used to texture the model.

Figure 7. An input photo and its corresponding input model produced using Deng et. al's method.

In order to create a new OBJ-file with the UV information, we first need to read all the vertex and face values from the input OBJ-file. The model's vertices' x and y coordinates are then used to create the UV vertices. As the model's vertex coordinates are in the range -1 to 1, they are converted to the UV range of 0 to 1. The UV faces are just duplicates of the model's faces. Finally, a new OBJ-file is saved containing both the original shape and the new UV information. Figure 8 illustrates an input model with vertex color-based texture, replaced with a UV-based texture.

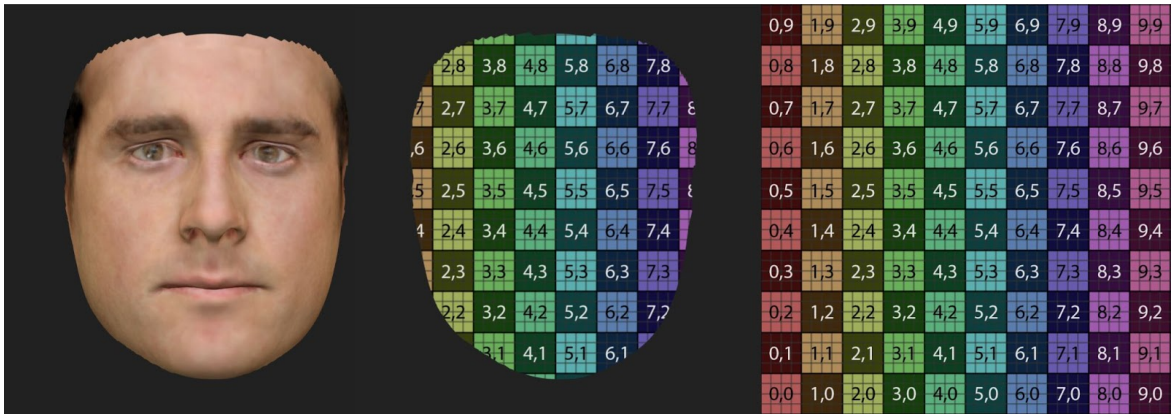


Figure 8. Models viewed from the Z-direction with an orthographic camera. Left: an input model created using Deng et al.'s method. Middle: the input model with a UV-based texture. Right: a sample UV texture image[4].

To use the input image as the texture, it needs to be modified so that it aligns with the created UV map. To achieve this, a rendered image of the model (Figure 8: left), can be used as a guide. This process will be described in the following chapters.

3.2.2 Rendering the input 3D face

If the input model is rendered so that the edges of the scene are at the coordinates -1 and 1 for both the x and y axes and an orthographic camera is used, then the rendered image will match perfectly with the UV map created in the previous chapter. Having the edges of the scene at the specified coordinates ensures that the face in the input image aligns with the UV map because the UV vertices were created using the input model's vertices converted from the range -1 to 1. An orthographic camera is required so that the rendered model would not be distorted due to perspective, which means the Z coordinates of the vertices would essentially be ignored, as they were when the UV map was created.

Figure 9. Input model rendered using pyrender

[4] Image source: *https://www.biocinematics.com/resources*

To actually create the rendered image, the pyrender[5] library is used for rendering. To import and process the OBJ model the trimesh[6] library is used. Figure 9 represents an input model rendered using the aforementioned method.

3.2.3 Aligning the input image with the rendered image

After the rendering process, the input image can be aligned with the render. In this thesis, facial landmark detection is used as a basis for alignment. Facial landmark detection is the process of detecting key features on a face (Figure 10) [13]. The landmarks are used to find the location and size of the faces in the render and the input photo. Then the input is cropped based on that information so that it matches the render.

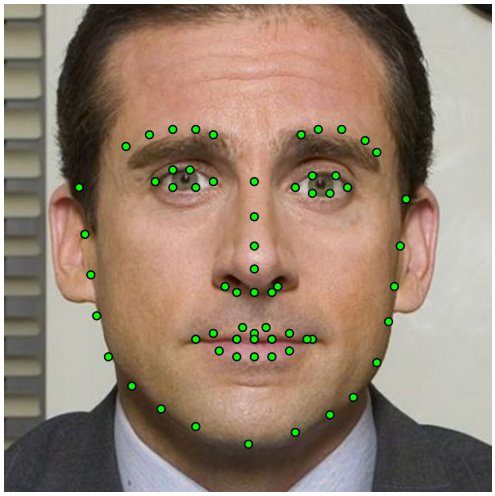


Figure 10. 68 facial landmarks (green points) found using the face_alignment package.

To find the landmarks of the rendered face, the face_alignment[7] package is used. It was chosen as it is one of the most recent and popular methods for landmark detection. Landmark detection is applied to the input and render images, and 68 landmarks are found for both. To crop the input image, a python imaging library called Pillow[8] is used. For cropping, the coordinates of the new sides of the image on the input image need to be found.

First, the landmarks are used to create a centre point for the cropping process. For this, the average landmark of all the 68 landmarks is found by summing the vector representations of the landmarks together and dividing the sum by 68 (blue points on Figure 11). This distance will be referred to as d_{Render} for the render image and d_{Photo} for the input photo. Next, the average distance[9] from the centre point to all the other facial landmarks is found (red circles on Figure 11). This point will be referred to as C_{Render} for the render image and C_{Photo} for the input photo.

[5] https://pyrender.readthedocs.io

[6] https://trimsh.org/trimesh.html

[7] https://github.com/1adrianb/face-alignment

[8] https://pillow.readthedocs.io/en/stable/

[9] http://mathonline.wikidot.com/the-distance-between-two-vectors

Figure 11. Centre points (blue points), average distances from the center point (red circles), and the corresponding crop result. Left: rendered model. Centre: input image. Right: cropped result.

The centre point C_{Render} and the average landmark distance d_{Render} can be used to find the distances to all four sides from C_{Render} relative to d_{Render}. These relative distances are then used with d_{Render} to find the distances to each of the sides of the input image. With these distances and C_{Photo}, the coordinates of the sides of the crop can be found and then used on the input image with Pillow's crop function. Figure 11 illustrates a rendered model, the input image, and the resulting cropped image.

3.2.4 Warping

Figure 12. An input model textured using a cropped input image.

After the cropping process, the cropped and rendered face images are quite well aligned, and the cropped image could already be used as the UV texture image to produce a somewhat acceptable texture (Figure 12). However, there are some apparent problems with the texture. Due to the difference in perspective of the orthographic render and the input photo, there is usually a bit of background visible on the model as the face on the photo appears slimmer than in the orthographic render. Moreover, the face models produced by the current reconstruction methods do not match the input image perfectly, which can cause unaligned facial features and further background bleed. The method used in this thesis for solving the aforementioned alignment problems is warping the cropped face image.

The method proposed in this thesis warps the image so that the landmark positions of the cropped image match the landmarks of the render. However, warping the image like this is a difficult task. The approach in this thesis uses the idea of representing the cropped and rendered images as triangle meshes[10] and warping the cropped image's mesh to match the render's mesh. In other words, a new warped photo is created by taking the shape from the render image mesh and the color information from the cropped photo mesh.

The triangle mesh will use landmarks found in Chapter 3.2.3 as its vertices. As the mesh needs to cover the whole image, additional landmarks (vertices) are added to the corners of the image. To reduce the chance of overlapping triangles on angled images, more landmarks were added to enable creating a better mesh. These landmarks were added to the middle of all the sides of the image, and a landmark was added on each cheek. The cheek landmarks were found by finding the averages of specific eye and mouth landmarks. The added landmarks are illustrated in green in Figure 13.

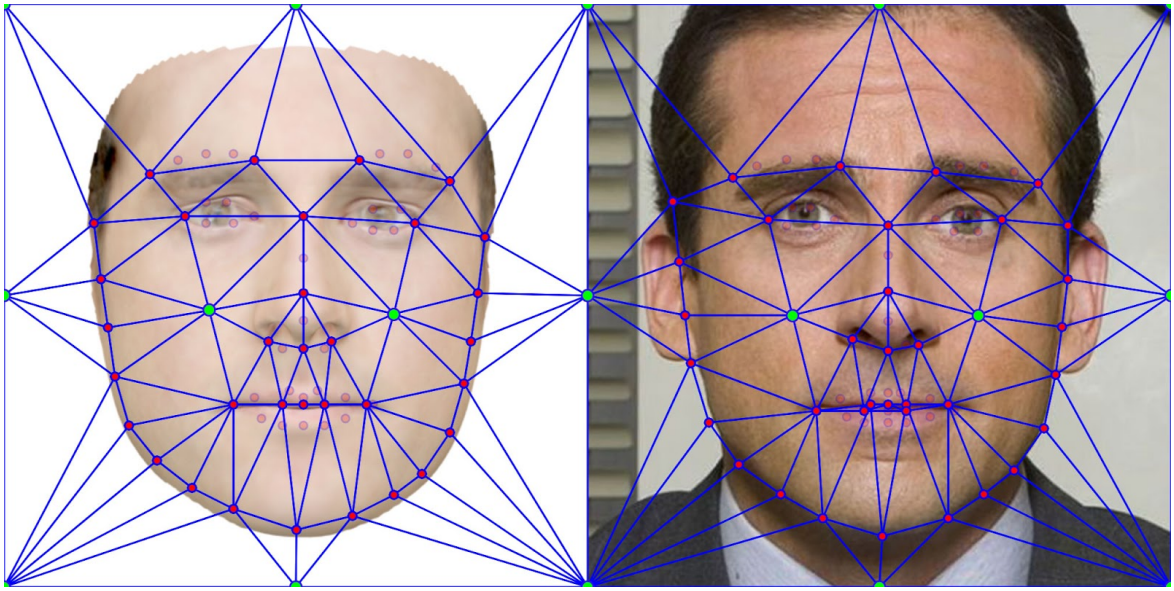


Figure 13. Triangular meshes created using landmarks. Red circles represent the landmarks found using landmark detection, green circles represent the added landmarks and blue lines represent the triangles of the triangle mesh.

Triangles (faces) of the mesh were selected manually so that the sides of the head and mouth area would be warped to match the render while ignoring most of the other landmarks. The other unused landmarks, which are around the eye, nose and mouth area are ignored to avoid creating small triangles that can create very noticeable warping defects when landmark detection is imprecise. The chosen faces of the mesh can be seen in blue on Figure 13.

[10] https://en.wikipedia.org/wiki/Triangle_mesh

The warped image is created pixel-by-pixel by sampling pixel values from the cropped image. Each pixel's position vector is used to find a position vector for sampling the cropped image. The pixels position vector will be referred to as the *pixel vector,* and the vector for sampling will be referred to as the *sample vector*. These vectors are illustrated as P and S in Figure 14. The idea is to use the relative pixel vector (A_RP on Figure 14) on a triangle in the render image mesh to find the relative sample vector (A_PS on Figure 14) on the cropped photo mesh, which can then be used to find the actual sample vector (S on Figure 14).

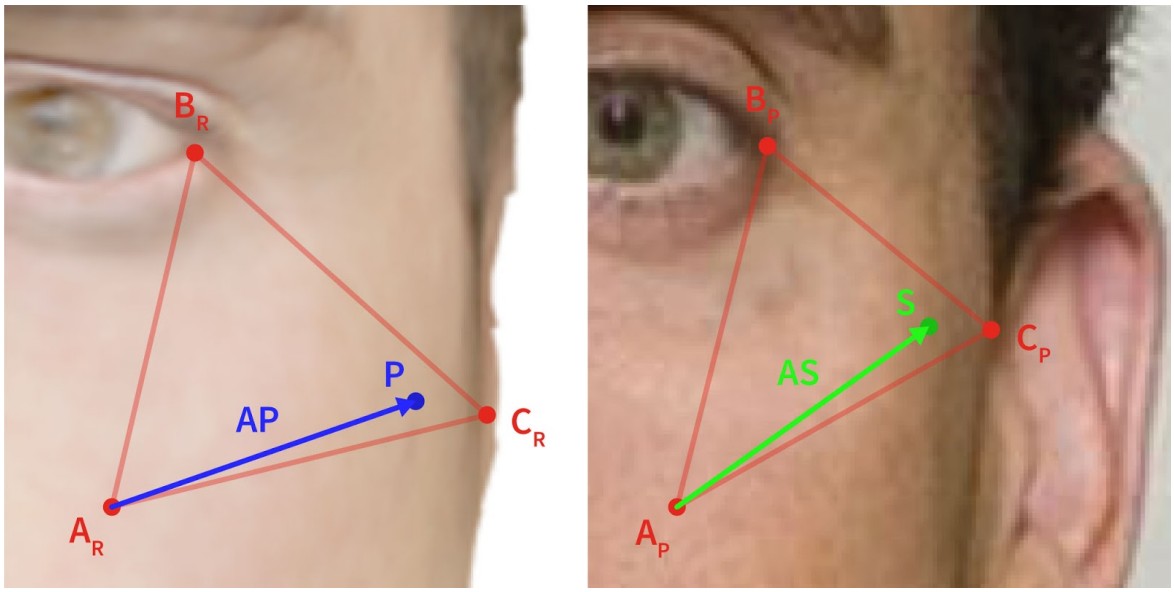


Figure 14. Pixel and sample vectors relative to the render and photo meshes. A_R, B_R, and C_R are render mesh triangle corners. A_P, B_P, and C_P are photo mesh triangle corners. P is the pixel's position and A_RP is its relative position to the render mesh. S is the pixel's position and A_PS is its relative position to the photo mesh.

To find the relative pixel vector on a triangle in the render mesh, first the triangle that the pixel vector is inside needs to be found. The triangle can be found by iterating through all the triangles in the mesh and checking whether the pixel vector is inside. Checking if a point is inside a polygon[11] is a common and well-researched topic in computational geometry and therefore will not be discussed in detail in this thesis. When the triangle that contains the pixel vector is found, the relative pixel vector can be calculated. Now the relative vector will be represented using vectors from one of the triangle's vertices to the other vertices (A_RB_R and A_RC_R on Figure 14). These vectors will be referred to as the triangle's *side vectors* and the vertex they are originating from as the *origin vertex*. The relative pixel will

[11] https://erich.realtimerendering.com/ptinpoly

be represented as the linear combination[12] of the two side vectors. This is illustrated in Figure 15 by the vectors *AB* and *AC* being multiplied by the scalars *m* and *n* to represent the relative pixel vector *AP*.

Finding the scalars can be represented with the linear system $\begin{cases} am + bn = x \\ cm + dn = y \end{cases}$ where $\langle a|b\rangle$ and $\langle c|d\rangle$ are the side vectors, *m and n* are the scalar values representing the pixels' relative position, and $\langle x|y\rangle$ is the pixel's relative position. The system is then solved using Cramer's rule[13] to find the values of the scalars *m* and *n*. The scalars can then be used to find the relative sample vector on the cropped photo. From the relative sample vector, the actual sample vector can be found by adding it to the origin vertex (*AP on* Figure 14) of the triangle in the cropped photo.

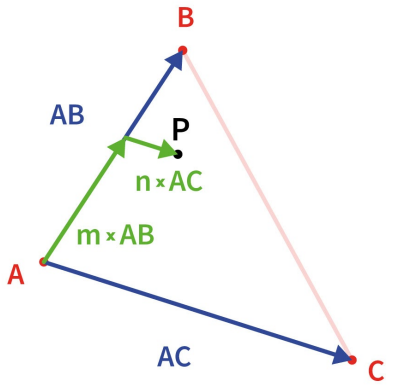


Figure 15. The relative pixel vector AP represented as a linear combination of the side vectors AB and AC. Scalars m and n are the scalars used to represent the pixel vector in the linear combination.

Therefore, we have found the sample vector, and we can use it to sample a color value for the pixel from the input photo. After doing this process for each pixel, a new warped image is created. An input image and its corresponding warped image are illustrated in the top row of Figure 16.

[12] https://www.mathbootcamps.com/linear-combinations-vectors/

[13] https://www.cliffsnotes.com/study-guides/algebra/algebra-ii/linear-sentences-in-two-variables/linear-equations-solutions-using-determinants-with-two-variables

Figure 16. A cropped input image and its warped counterpart being used as the UV texture. Left: Unwarped texture image and the corresponding textured model. Right: Warped texture image and the corresponding model.

When the warped image is used to texture the model, some of the alignment problems described at the beginning of Chapter 3.2.4 are remedied. Figure 16 illustrates a cropped image, and its warped counterpart used as the UV texture for a model.

4 Survey

The goal of 3D face reconstruction from 2D images is to produce models that look realistic and resemble the original face as closely as possible. However, these factors are very difficult to assess, as they can be quite subjective. Thus, a survey was conducted to get an understanding on the quality of the models produced by different methods. The survey consisted of rendered outputs of three different methods and the proposed texture generation method.

The survey was created with Google Forms[14] and was shared via email lists and direct messages. There were 32 questions in the survey, and it was made of 3 separate parts. The whole survey took about 10 minutes to complete. Input image set included face photos of 21 celebrities chosen to have varying poses, races, ages, occlusions, and expressions. All the questions were created so that participants could rank the 3D face models from best to worst Figure 17.

Figure 17. The input for the participants used for ranking models.

The 3D model files were created using the code provided on Github[15] for each method. In order to assess the models, they needed to be rendered. Rendering was done using a browser-based rendering library called Three.js. The three parts of the survey are described in following chapters.

4.1 Part 1

In the first part of the survey, each question had images of four rendered models. Three of them were the models produced by the methods described in Chapter 2, and one was the model with the proposed texture generation. In this part, the participants were asked to rank the face models in order of their preference based only on the visual quality of the models, and thus the questions did not include the input images. The goal in this part of the survey was to understand which method produces the most realistic, human-like, and visually appealing models based on people's opinions. There were 10 questions in Part 1, and one of them is illustrated in the left image of Figure 18.

[14] https://www.google.com/forms/about/

[15] https://github.com/

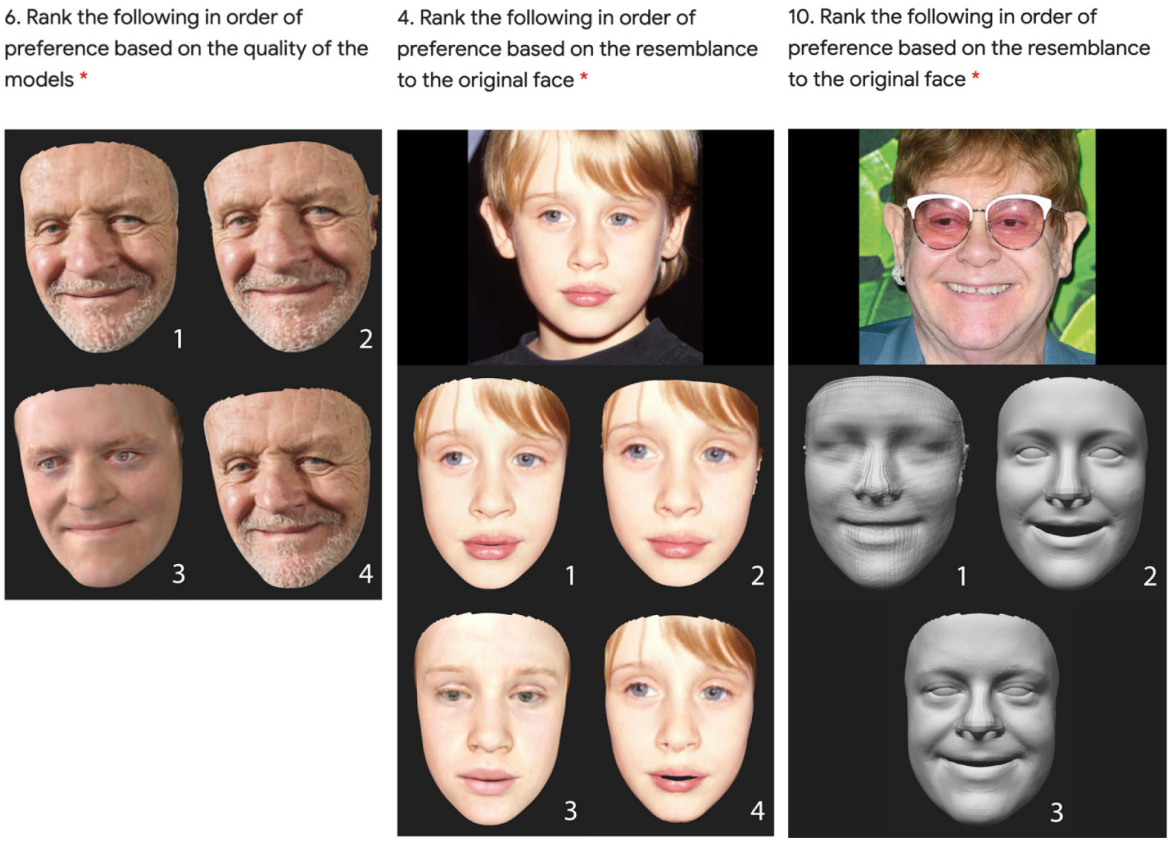


Figure 18. Question examples for the participants. Left: Part 1. Centre: Part 2. Right: Part 3.

4.2 Part 2

In the second part of the survey, the participants were asked to rank the models based on their resemblance to the original face. The goal was to see how well these three methods resemble the shape and texture of the original face. In this part of the survey, the question included the input images, and there were 12 questions in total. One of the input photos was used twice, with the models having different angles to see how varying poses affect people's choices. A question example from Part 2 is illustrated in the centre image of Figure 18.

4.3 Part 3

In the third part of the survey, the participants were asked to rank untextured models based on their resemblance to the original face. In this part, the models were untextured. This part was made to compare only the shape of the models and to see people's preferences when texture is not present. This part of the survey contained 10 questions with images of 3 models. The right image of Figure 18 illustrates one of the questions from this part of the survey.

5 Results

In this chapter, the results of the thesis are analyzed. The goal of this thesis was to compare the best existing methods for 3D reconstruction from 2D images that use deep learning and to improve the texturing of one of these methods by creating a more realistic and detailed texture. In the next chapters, results of the improved texturing method and survey will be analyzed.

5.1 Texturing

In this chapter, the results of the proposed texturing method are shown. Examples of rendered models using the proposed texture can be seen in Figure 19. The generated textures preserve the fine details of the original face.

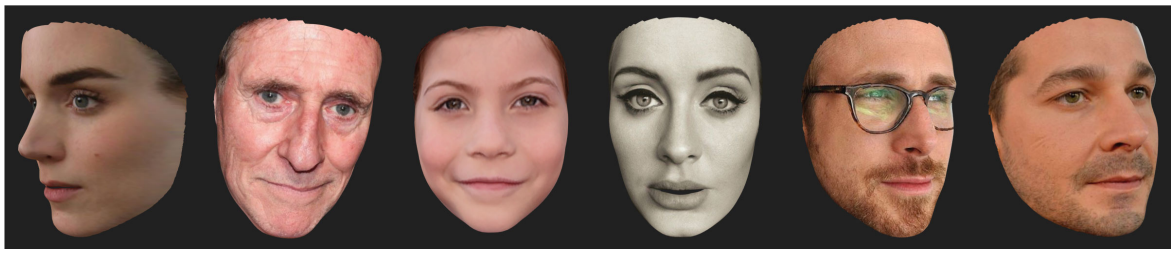


Figure 19. Output models with improved texture.

Some of the textured models have visible warping defects. These are caused by imprecise landmark detection. The defects were especially visible around the mouth due to small triangles in the triangle mesh that was used for warping. An example of this can be seen in Figure 20.

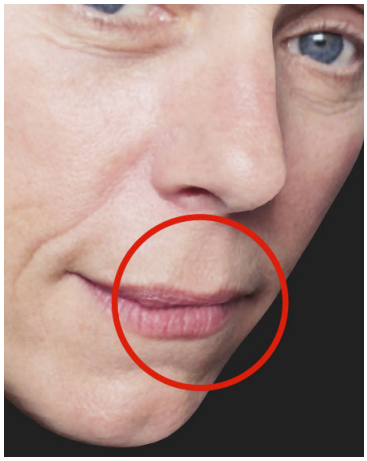


Figure 20. Output model with a defect (highlighted with a red circle).

5.2 Survey

The main tool for comparing the existing methods is results of the survey that was carried out in this thesis. The survey was completed by 50 people who gave 1600 votes in total. Survey responses were analyzed using Google Sheets[16]. Data is presented as the number of votes using column charts, which are chosen as they give a visual presentation

[16] https://www.google.com/sheets/about/

of the data that is easy to comprehend. The survey's results are analyzed in the following subchapters.

5.2.1 Part 1

In this chapter, the results of the first part of the survey are analyzed. In this part, models were ranked based on their quality. The chart in Figure 21 illustrates the number of times each model was ranked first. Out of the 500 total votes, the models with the improved texture got the most (44.2%) votes. Feng et al. was a close second (34.2%), Deng et al. came third (11.6%), and Guo et al. came in last (10%).

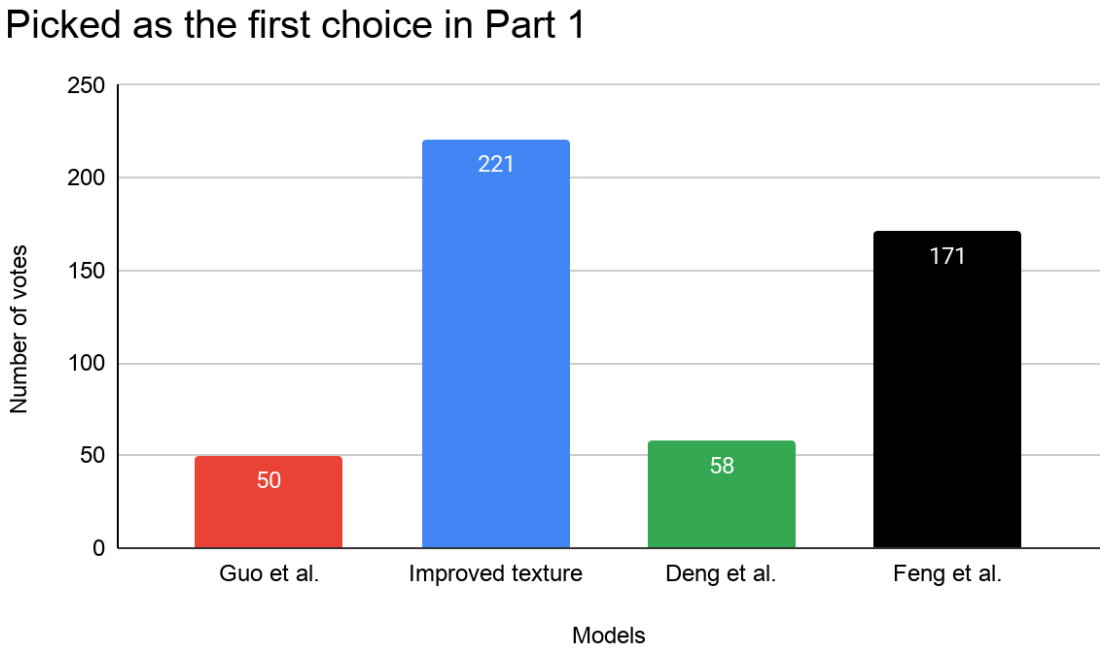


Figure 21. The number of votes as the first choice for the methods in Part 1.

The charts illustrating the votes for the second, third and fourth choice can be found in Appendix I. In order to give a simple overview of all the votes in this part, a weight system is used. Each choice (first, second, third, fourth) was assigned an arbitrary weight which was multiplied by the number of times it was chosen. The weights assigned were 1, ⅔, ⅓, and 0 for the first, second, third, and fourth choices, respectively. This allows us to see the overall results obtained in the first section of the survey. Figure 22 shows the weighted total for the four models.

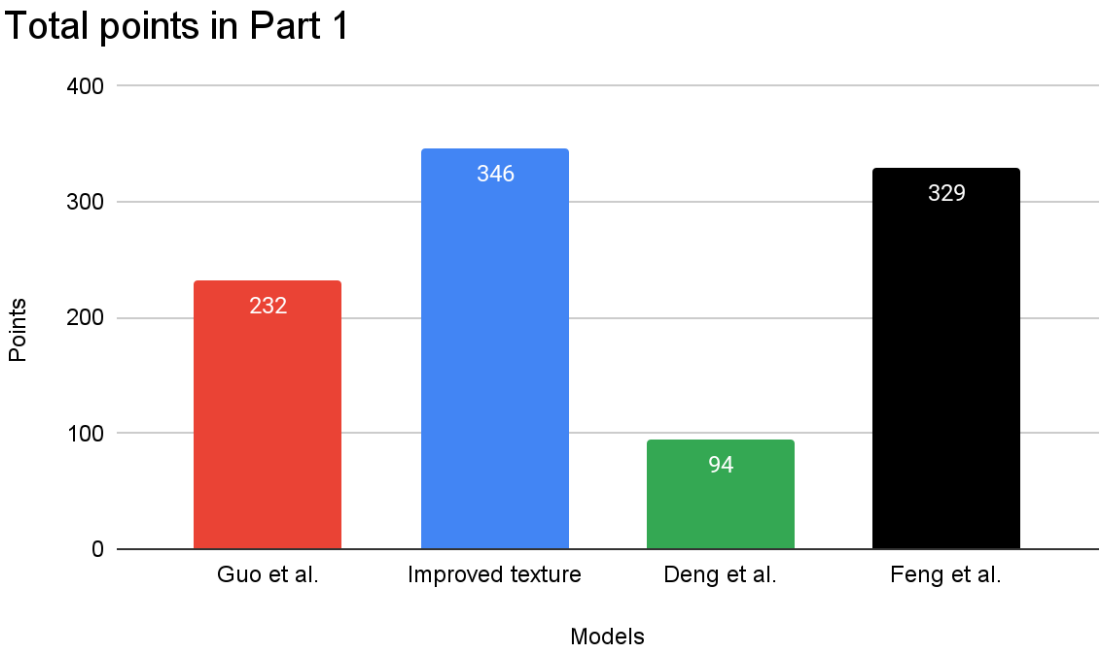


Figure 22. Total points for the methods in Part 1.

From the two previous charts, we can deduct that the model with the improved texture is perceived to have the highest quality. Feng et al.'s method came in as a close second mostly due to the fact that it performed very well on pictures with teeth showing. Out of the total 10 image sets, 4 contained teeth. Feng et al.'s method was chosen as the first choice the most in all 4. Two image sets with teeth showing are illustrated in Figure 23.

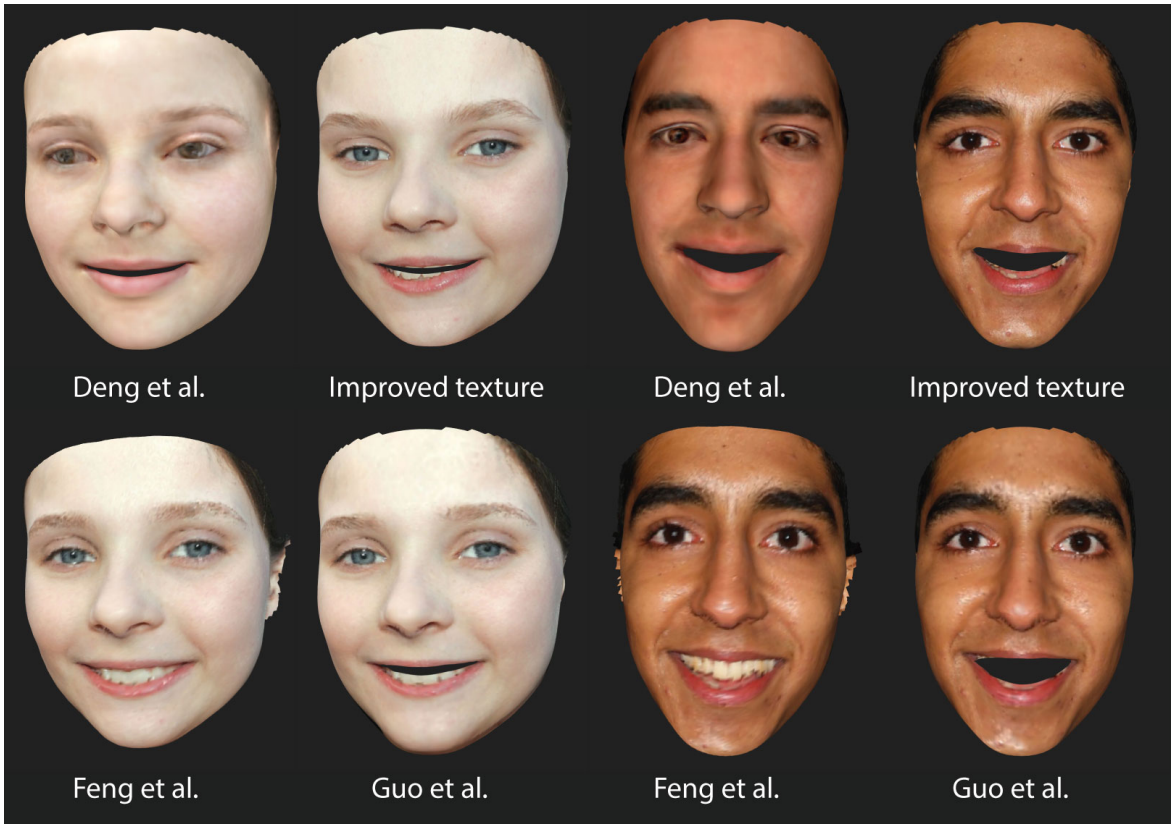


Figure 23. Two sets of models from input photos that show teeth.

Deng et al.'s method ranked 3rd, with Guo et al. closely behind on the number of times it was chosen as the first choice but performed the worst by far based on the point system. This was likely due to the 3DMM-based texture, which yields smooth, undetailed textures which look unrealistic.

5.2.2 Part 2

In this chapter, the results of the second part of the survey are analyzed. There were 12 sets of images with 3D face models in the second part of the survey. In this part, models were ranked based on their resemblance to the original face. The model with the improved texture has considerably more votes as the best model with 46.7% of the total 600 votes, which can be seen in Figure 24. The votes are distributed more or less in the same fashion as in Part 1, except that Feng et al.'s method has got a lot fewer votes for the first choice, this time with 28.9% of the total votes. Again Guo et al. and Deng et al.'s methods are quite close with 11.7% and 12.7% of the votes, respectively.

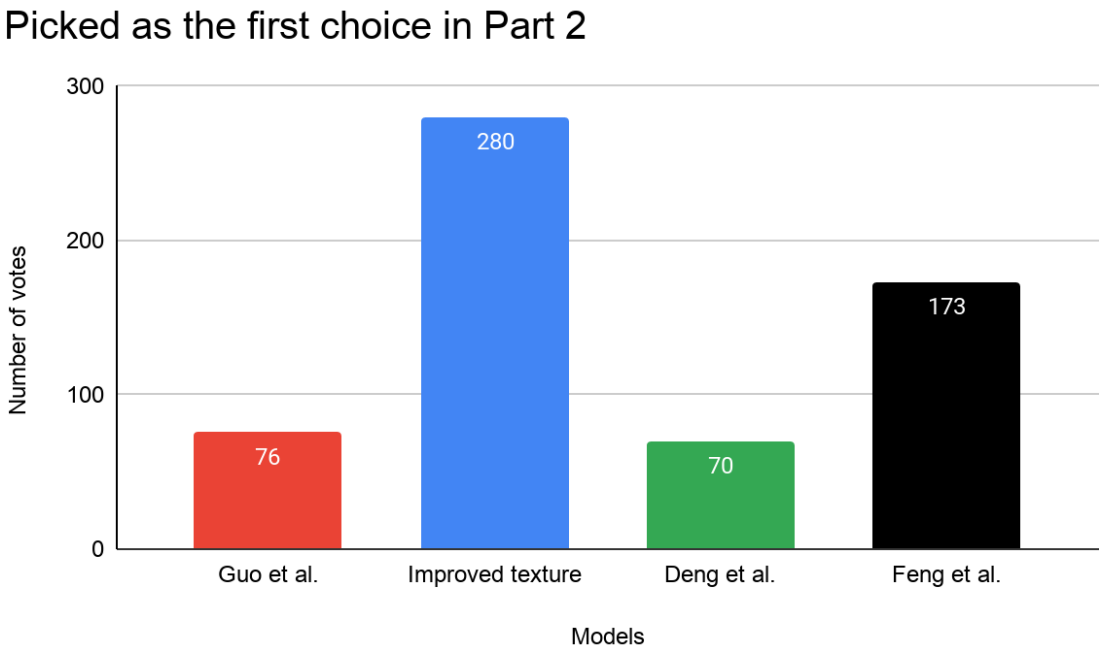


Figure 24. The number of votes as the first choice for the methods in Part 2.

The charts for the second, third and fourth choices can be found in Appendix I. Figure 25 illustrates the results of Part 2 using the same weight system described in the previous chapter. Again, the results are quite similar to the first part.

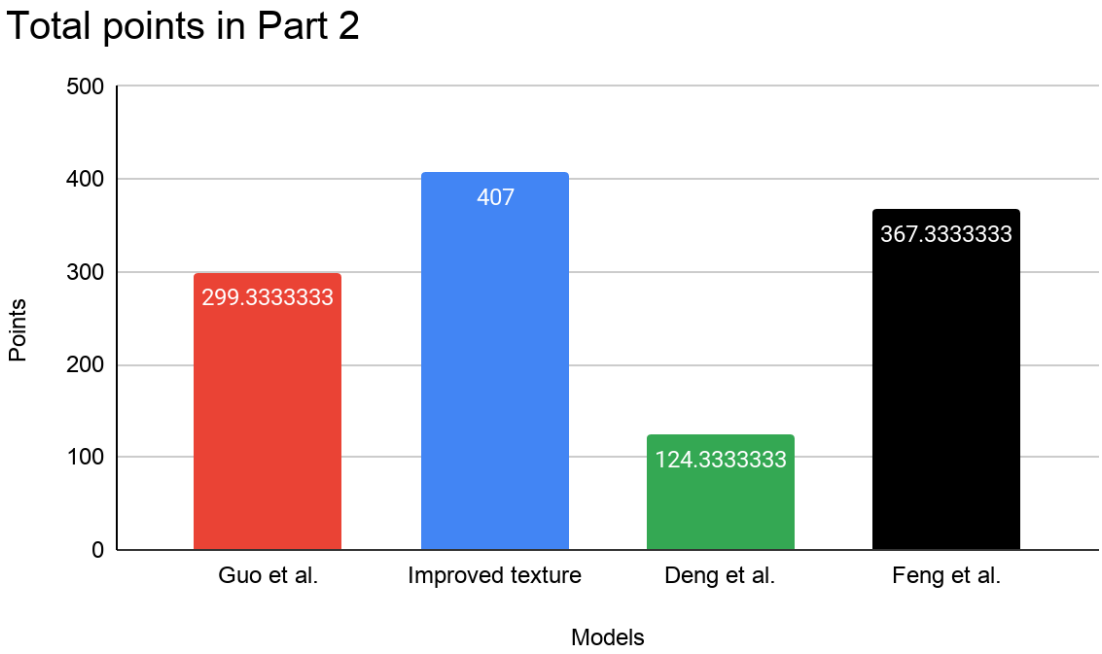


Figure 25. Total points for the methods in Part 2.

Overall, in Part 2, the results were mostly the same as in Part 1. Feng et al.'s method performed a bit worse this time, probably due to there being no images that show teeth. In this part of the survey, there were two images that were rotated from their original angle. These images are shown in Figure 26. Two sets of models rotated from their original angles.. On these images, Deng et al. performed better than usual because the method does not have a problem with texturing the unseen side of the model as the other methods do. Guo et al. did especially badly in this part of the survey, likely due to a large number of the models having a mouth hole that does not align with the texture. An example of this can be seen in the left set of images in Figure 26.

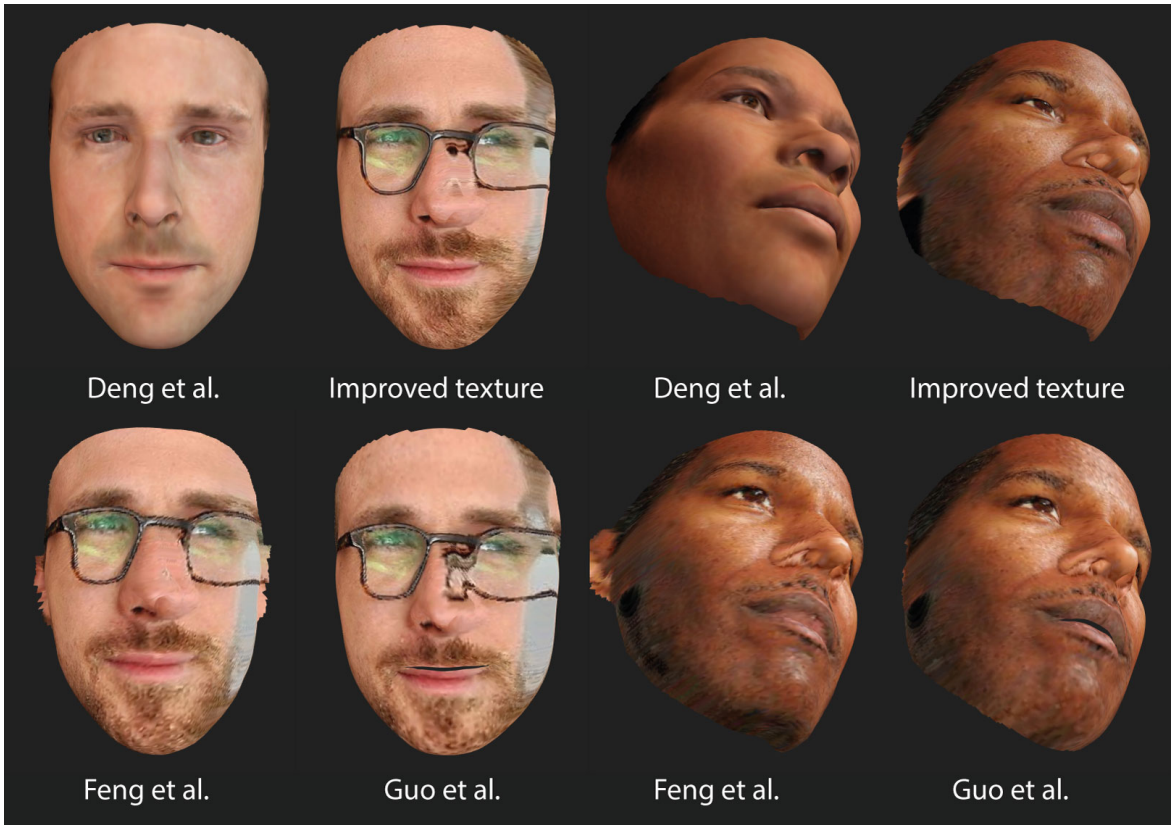


Figure 26. Two sets of models rotated from their original angles.

5.2.3 Part 3

In this chapter, the results of the third part of the survey are analyzed. There were 10 questions containing an input image and untextured models of the three face reconstruction methods. In this part, participants were asked to rank the models shape-wise based on their resemblance to the original face.

Figure 27 illustrates the number of votes received as the first choice. Deng et al. got significantly more votes than the other two with 60.4% of the votes. Guo et al. and Feng et al. got 26% and 13.6% of votes, respectively.

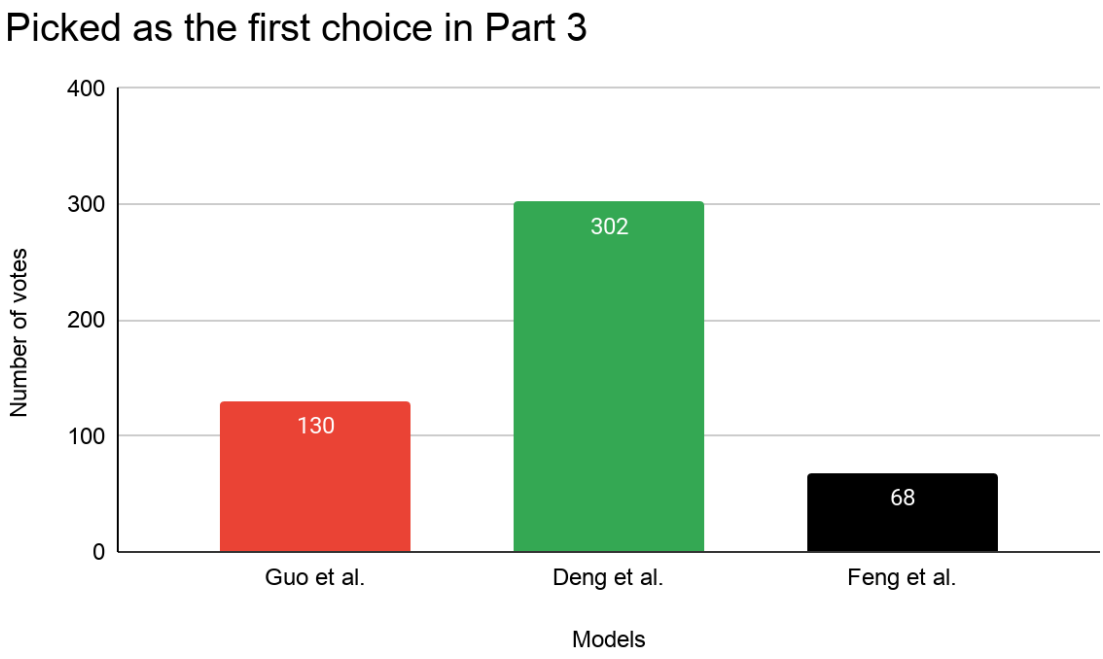


Figure 27. The number of votes as the first choice for the methods in Part 3.

The charts for the second and third choices can be found in Appendix I. Figure 28 illustrates the results of Part 3 using the weight system.

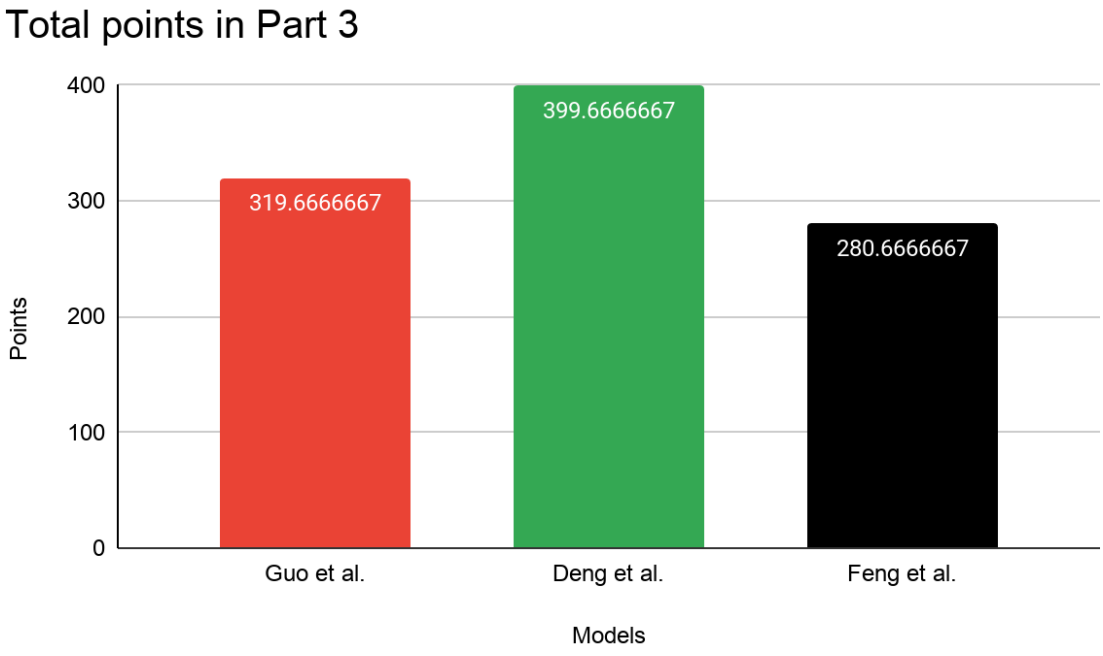


Figure 28. Total points for the methods in Part 3.

In Part 3 the results show that Deng et al.'s method outputs shapes that succeed to resemble the original face shape more accurately compared to the other methods. Feng et al.'s method performed the worst, likely due to the checkerboard-like artifacts on the models' surfaces and the lack of details around the eyes which made the faces look less human-like. The untextured models that were generated by Guo et al.'s method had less detailed face shapes compared to Deng et al., which could explain why it performed worse (Figure 29).

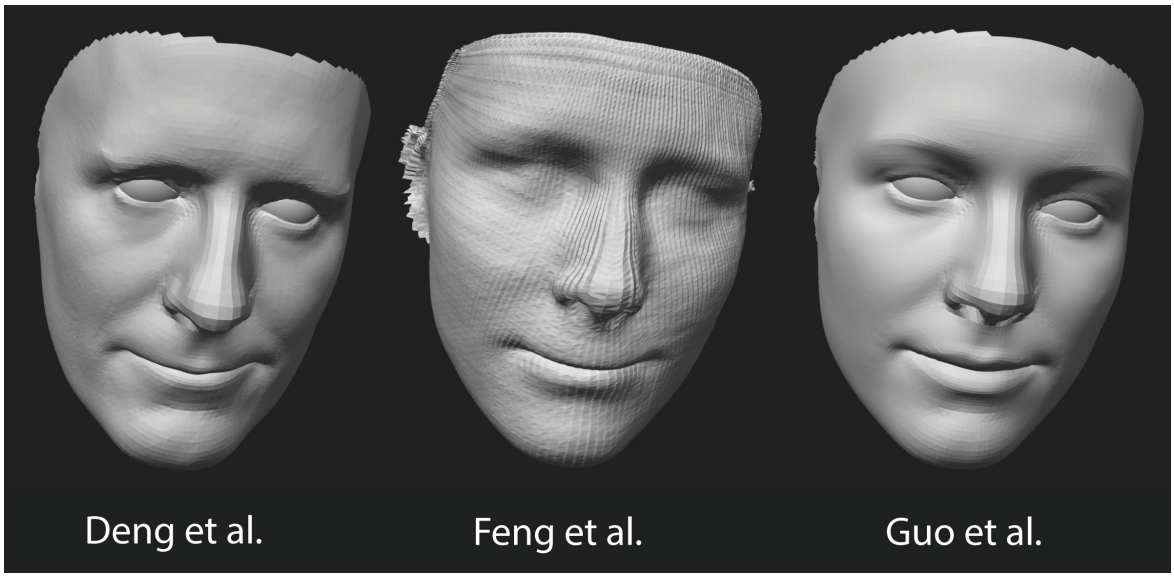


Figure 29. A set of untextured models.

5.2.4 Survey Conclusion

As the results of Part 1 and Part 2 of the survey had very similar results, their weight values are combined in blue in Figure 30. From these two parts of the survey, we can conclude that the improved texturing method was successful as it performed better than the other methods. This was most likely due to the texture being much higher quality, even though there were some warping artifacts visible on a number of the produced textures. Feng et al.'s method comes in as a close second, which performed especially well on images with teeth. On images without teeth, Guo et al. performed mostly on par with Feng et al., but has texture alignment problems with the model's and texture's mouth, which are very noticeable.

The red bars in Figure 30. Total points for the textured and untextured parts of the survey. Models are given on the x-axis, and points are given on the y-axis. Blue bars indicate total points from the first and second part of the survey, and red bars indicate the points received in the third part of the survey illustrate the weight values for Part 3 of the survey. In this

part, Deng et al. was the clear winner, which shows that using this method as the base of the improved texturing method was a good choice.

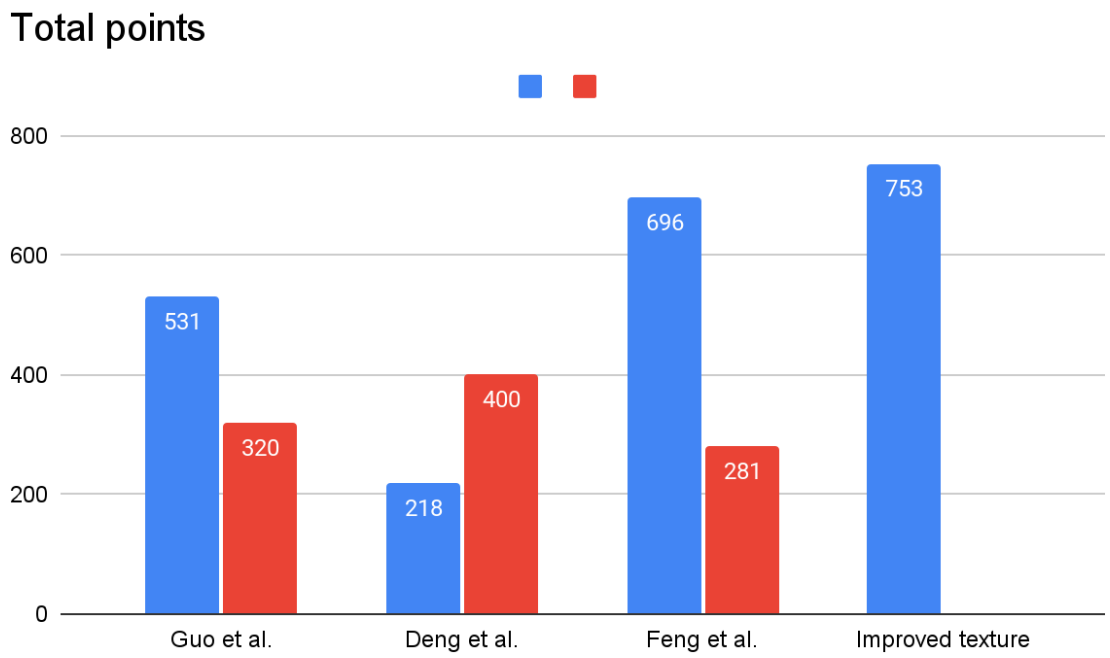


Figure 30. Total points for the textured and untextured parts of the survey. Models are given on the x-axis, and points are given on the y-axis. Blue bars indicate total points from the first and second part of the survey, and red bars indicate the points received in the third part of the survey

6 Conclusion

In this thesis, three different deep learning algorithms for 3D face reconstruction from 2D images were discussed, and a texturing method for 3D face models was proposed. Two of the face reconstruction methods were 3DMM-based, and one was a model-free method. A 3DMM-based method by Deng et al. was chosen as the base for the texturing method proposed in this thesis. The proposed texturing method uses the input photo as a UV texture image to produce a high-fidelity texture.

A survey was conducted to assess the three 3D reconstruction methods mentioned above and the proposed texturing method. The survey consisted of three parts, and it had 50 participants in total. From the parts of the survey that assessed textured models, it was observed that Feng et al.'s method was the best out of the three methods. When untextured models were assessed, it was found that Deng et al.'s performed the best.

The texture improvement proposed in this thesis uses warped input photos as the UV textures for face models. Warping was performed using landmark detection and triangular meshes. From the survey, it was observed that models generated by Deng et al.'s method with the improved texturing method were the best quality and most realistic compared with the other three methods.

The results obtained from this thesis can be helpful to understand the performance and accuracy of different kinds of 3D face reconstruction with deep learning methods. Deng et al.'s method, paired with the proposed texturing method can be a useful tool for 3D face reconstruction from 2D images. In continuation of this research, the proposed texturing method could be tested on all three face reconstructions methods. Moreover, the performance of the texturing methods described in Chapter 3.1 could be assessed as well.

References

[1] J. Petty, „What is 3D Modeling & What's It Used For?," https://conceptartempire.com/what-is-3d-modeling/ (20.05.2021).

[2] W. N. Widanagamaachchi, A. T. Dharmaratne, „3D Face Reconstruction from 2D Images A Survey," *Digital Image Computing: Techniques and Applications,* 2008.

[3] X.-F. Han, H. Laga, M. Bennamoun, „Image-based 3D Object Reconstruction: State-of-the-Art and Trends in the Deep Learning Era," *Computing Research Repository (CoRR),* 2019.

[4] G. Stylianou, A. Lanitis, "Image Based 3D Face Reconstruction: A Survey," 2008.

[5] V. Blanz, T. Vetter, „A Morphable Model for the Synthesis of 3D Faces," *SIGGRAPH '99: Proceedings of the 26th annual conference on Computer graphics and interactive techniques,* pp. 187-194, 1999.

[6] Y. Deng et al., „Accurate 3D Face Reconstruction with Weakly-Supervised Learning: From Single Image to Image Set," *Computing Research Repository (CoRR),* 2020.

[7] J. Guo et al., „Towards Fast, Accurate and Stable 3D Dense Face Alignment," *Computing Research Repository (CoRR),* 2021.

[8] Y. Feng, F. Wu, X. Shao, Y. Wang, X. Zhou, „Joint 3D Face Reconstruction and Dense Alignment with Position Map Regression Network," *Computing Research Repository (CoRR),* 2018.

[9] J. Lin, Y. Yuan, T. Shao, K. Zhou, „Towards High-Fidelity 3D Face Reconstruction from In-the-Wild Images Using Graph Convolutional Networks," *Computing Research Repository (CoRR),* 2020.

[10] S. Zhang et al., „Graph Convolutional Networks: A Comprehensive Review," *Computational Social Networks,* 2019.

[11] P. Paysan et al., „A 3D Face Model for Pose and Illumination Invariant Face Recognition," *Sixth IEEE International Conference on Advanced Video and Signal Based Surveillance,* 2009.

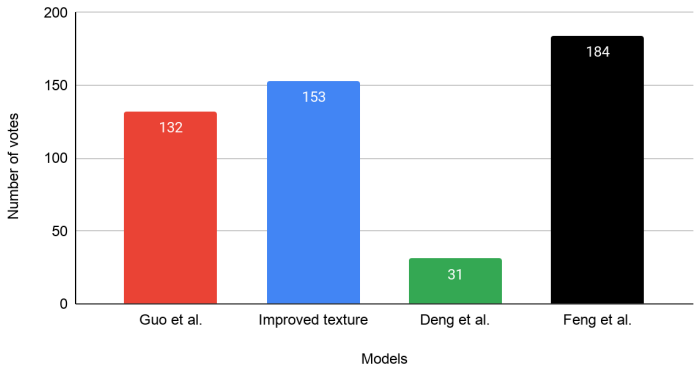
[12] J. Deng, S. Cheng, N. Xue, Y. Zhou, S. Zafeiriou, „UV-GAN: Adversarial Facial UV Map Completion for Pose-invariant Face Recognition," *Computing Research Repository (CoRR),* 2017.

[13] X. Dong, Y. Yan, W. Ouyang and Y. Yang, “Style Aggregated Network for Facial Landmark Detection,” *Computing Research Repository (CoRR),* 2018.

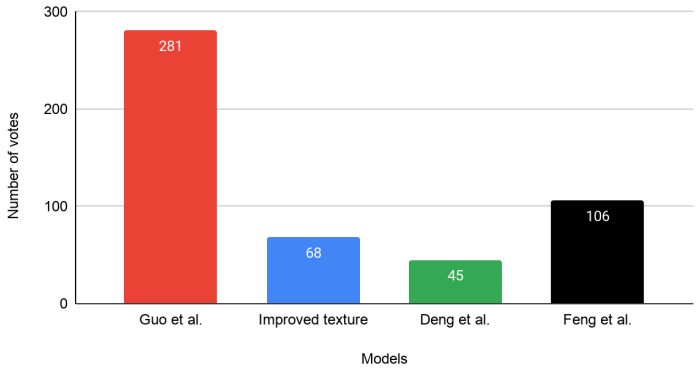
Appendix

I. Graphs from survey results

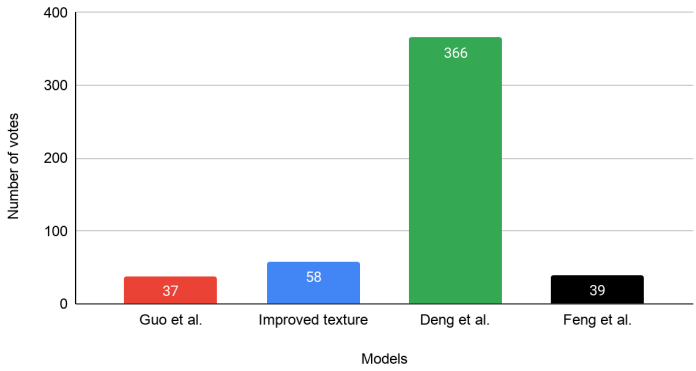
Picked as the second choice in Part 1



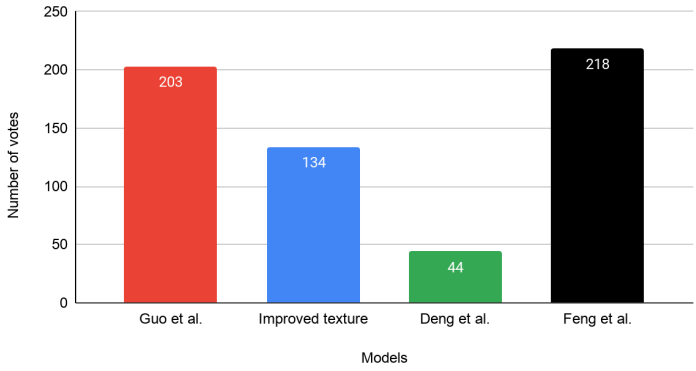
Picked as the third choice in Part 1



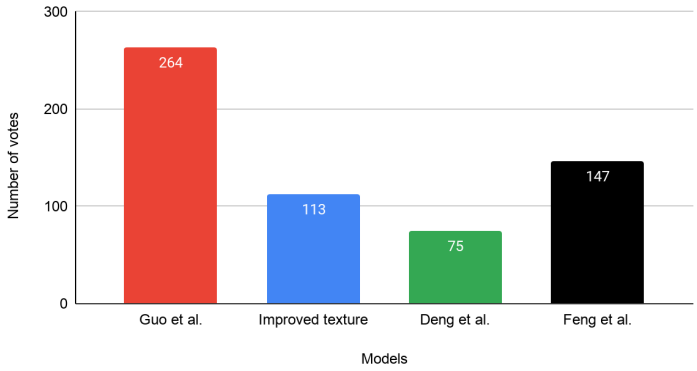
Picked as the fourth choice in Part 1



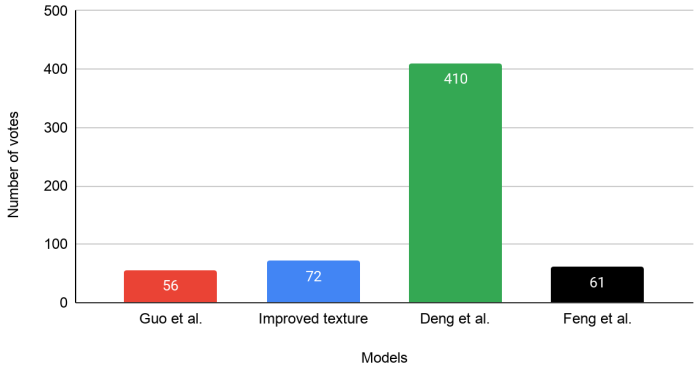
Picked as the second choice in Part 2



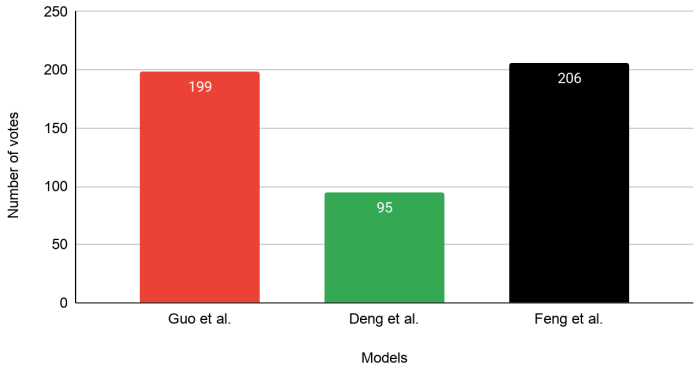
Picked as the third choice in Part 2



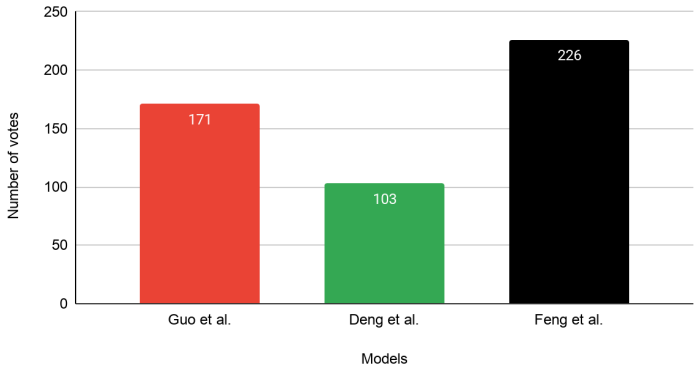
Picked as the fourth choice in Part 2



Picked as the second choice in Part 3



Picked as the third choice in Part 3



II. Source Code

The source code of the thesis can be found in the included ZIP-file.

Contents of the *ZIP-file*:

- *Texturing.ipynb* – the Jupyter[17] notebook for the proposed texturing method.
- *Texturing.py* – the python file for the texturing method generated from the notebook.
- *Input* – the folder containing an example input.obj and input.jpg file.
- *Output* – the folder containing an example output model and its accompanying texture and material files.
- *WarpPolygons.txt* – the text file containing triangle mesh faces for texturing.

The texturing method can be used by running the texturing.py file with python. All the used packages seen on lines 7 to 13 of the python file need to be installed. The script has been tested with python 3.9.2.

[17] https://jupyter.org

III. License

Non-exclusive licence to reproduce thesis and make thesis public

I, **Mehin Salimli**

1. herewith grant the University of Tartu a free permit (non-exclusive licence) to reproduce, for the purpose of preservation, including for adding to the DSpace digital archives until the expiry of the term of copyright,

3D Face Reconstruction from a Single 2D Image,

supervised by Rain Eric Haamer,

2. I grant the University of Tartu a permit to make the work specified in p. 1 available to the public via the web environment of the University of Tartu, including via the DSpace digital archives, under the Creative Commons licence CC BY NC ND 3.0, which allows, by giving appropriate credit to the author, to reproduce, distribute the work and communicate it to the public, and prohibits the creation of derivative works and any commercial use of the work until the expiry of the term of copyright.

3. I am aware of the fact that the author retains the rights specified in p. 1 and 2.

4. I certify that granting the non-exclusive licence does not infringe other persons' intellectual property rights or rights arising from the personal data protection legislation.

Mehin Salimli

20.05.2021