# Evaluating the feasibility and effectiveness of multi-perspective stereoscopy for 3D face reconstruction

Nahuel Manterola
*Faculty of Electrical Engineering, Mathematics & Computer Science*
*University of Twente*
*Supervised by: dr.ir. L.J. Spreeuwers and ing. G.J. Laanstra*
*Examination committee: dr.ir. L.J. Spreeuwers, ing. G.J. Laanstra and dr. B. Alsadik*

***Abstract*—This paper presents research on the feasibility and effectiveness of multi-perspective stereoscopy. A method is introduced to reconstruct a 3D face from a set of images taken by a camera array on a 2D plane. A physical setup is presented to dynamically emulate this array, along with a digital rendering setup. Quantitative measurements are shown on the performance of the physical setup, and the results of the digital setup are qualitatively compared. The influence of the camera baseline, subject surface angle and the number of cameras on the reconstruction quality is determined. Conclusions are drawn, but only for the current setup and methods.**

## 1. Introduction

A decade ago, 3D face recognition was mostly an academic subject. Nowadays, 3D face recognition's inherent advantages over its 2D counterpart has pushed not only academics, but also the industry towards further development in recognition and reconstruction of 3D face models. Many flagship portable devices are now using 3D face recognition as an unlocking method, and many companies use access systems based on 3D face recognition. The main advantages over 2D face recognition are: the minor dependence on (I) light conditions, (II) make-up and facial texture, and (III) head pose, and (IV) its higher spoofing resistance[1].

3D reconstruction can be categorized in two approaches: active and passive. Where the active methods actively project light onto the subject, the passive methods reconstruct depth from 2D images. Not all methods of 3D reconstruction are suitable for face reconstruction. An overview of the methods will be shown in the related works (Section 2). In [2], the predecessor of this paper, Spreeuwers describes a passive 5-camera setup in a '+' configuration, and shows that using a larger amount of cameras significantly increases the accuracy of the reconstruction. An image of this setup is shown in Figure 1. The theoretical limits for cameras in a single plane, as shown in [3], state that the standard deviation on the X- and Y-axes of a reconstruction scales proportionally with $1/\sqrt{C}$, with $C$ as the number of cameras. On the Z-axis this scaling factor is $1/\sqrt{C^3}$. This shows a strong potential for a performance improvement by increasing the amount of cameras. If we look from a consumer perspective, the amount of cameras on handheld devices has seen a sharp increase in the past years, with a current 30% shipment share of quad-camera phones [4]. This opens up possibilities for stereo reconstruction. Considering the practical success in the previous paper, the theoretical potential of improvement and the current trends in technology, there is a strong incentive for further research in the performance of multi-cam 3D face reconstruction. This paper presents an easy-to-construct physical setup to position a single camera on any desired location in a 220mm by 220mm plane, in order to emulate a multi-camera setup. It also presents a digital setup, which offers further flexibility on all camera parameters and imperfections. It then proposes a method to combine data from a set of images taken with a camera array. This method is used to assess the performance of varying amounts of cameras, and estimate the effect of the surface angle of the subject on reconstruction quality. The following main questions are proposed: **(I)** What is the relationship between the amount of cameras and the reconstruction performance? **(II)** What is the effect on the reconstruction performance of the angle between the subject surface and the camera? **(III)** What is the relationship between the baseline in the camera array and the reconstruction performance? **(IV)** What is the maximum performance achieved by using the entire proposed pipeline?

Figure 1: Photo of the 5 camera setup used in [2].

Two minor questions are formulated to assess the performance of the physical setup and the digital setup: **(V)** What is the performance in position placement of the physical setup? **(VI)** How well does the digital setup mirror the real setup in terms of performance?

In Section 2, the related works will be discussed. In Section 3, the methods used in this paper are described. Section 4 describes the performed experiments and their respective results. Section 5 describes the conclusions and Section 6 describes the author's recommended paths for future research.

## 2. Related works

### 2.1. 3D face reconstruction

Research on both passive and active 3D face reconstruction is expanding and broadening rapidly. This section will give an overview of the research field.

#### 2.1.1. Active reconstruction

Active reconstruction methods are all methods that emit light as a means of capturing depth data from a subject. The main three methods are structured light, time-of-flight and laser triangulation. The **structured light** methods use a light source to project a pattern on the subject. A camera captures one or several images of the subject with the pattern and by matching the projected structures in the source and the image, the depth is triangulated. Current research is working on solving issues such as sensitivity to environment illumination [5] and occlusions [6]. [7] combines structured light with passive stereoscopic reconstruction, which partially solves limitations of both methods. **Laser triangulation** is somewhat comparable to structured light, but projects a laser line on the subject. This line is moved over the entire surface while images are captured, which allows for depth triangulation. [8] shows very good precision, with a standard deviation of 0.01mm, and avoids occlusions by using multiple cameras. However, like many laser triangulation methods, it reports capture times of multiple seconds, making it unreliable for scanning the face of a moving subject. **Time-of-flight** based systems project a light pulse and register the time for the light pulse to return to the capturing device. [9] shows one of the few attempts at registering a 3D face. However, the accuracy and information density is currently too low to reliably use for 3D face recognition.

#### 2.1.2. Passive reconstruction

In the last five years, deep learning has become the dominant research topic for passive reconstruction [10], with monocular reconstruction being the most active topic. **Monocular reconstruction** is the reconstruction of a 3D face using a single 2D image. Depth from shading is one of the more notable methods. While monocular reconstruction has an intrinsic information deficit compared to stereoscopy, promising results have been achieved very recently. In 2005, [11] shows a method to extract a 3D model from a single image to use in face recognition, but the results are poor by today's standards. [12], [13] and [14] show deep learning approaches which clearly improve on their predecessors. However, use in 3D face recognition is limited due to insufficient accuracy. **Stereoscopic reconstruction** generates a 3D face model by matching features in two or more images, and triangulating their depth. [15] presents a method with three cameras, which can prevent occlusions, but does not reach the accuracy necessary for 3D face recognition. [16] shows that a stereoscopic system can have very high accuracy under perfect circumstances, with up to pore-scale accuracy. However, the demands of this system are very high, requiring expensive cameras at great angles around the subject, a completely dark room and a 20 minute processing time. [2] shows a 5-camera solution with an iterative reconstruction technique to warp the correlation windows based on depth. Further research on camera arrays for 3D face reconstruction is very scarce, partially due to the inferior results compared to active methods.

### 2.2. Highlighted methods

Some of the related work is relied on heavily in this paper, and will therefore be described more thoroughly in this section.

#### 2.2.0.1 Semi-global matching

In [17], H. Hirschmuller presents semi-global matching (SGM), a method to estimate a dense disparity map from a rectified stereo image pair. It consists of the following steps: (I) A matching cost is calculated for each pixel $\mathbf{p}$ in image 1 at each desired disparity. This is stored in a cost matrix $C(\mathbf{p}, d)$ with dimensions W×H×D. (II) A smoothed cost $S(\mathbf{p}, d)$ is generated by accumulating the cost $C(\mathbf{p}, d)$ of each disparity in 8 directions. The disparity with the lowest smoothed cost is selected for each pixel. (III) Quadratic curve fitting is performed for three disparities around the selected disparity for sub-pixel estimation.

Most methods of calculating disparity cost for a pixel use a matching function on a window around this pixel. This relies on the fronto-parallel assumption, which introduces inaccuracies in real-world applications. SGM rejects this assumption by matching single pixels instead of windows and applying a smoothness constraint afterwards. For the first step, many matching costs have been proposed. Hirschmuller initially proposes using Mutual Information [18] or the Birchfield and Tomasi measure [19], and later proposes the census transform [20] as a suitable candidate. The smoothness constraint is accumulated along 8 paths $r$. pathwise costs are stored in $L_{\mathbf{r}}(\mathbf{p}, d)$, where $\mathbf{p}$ is the pixel for which the cost is being calculated for each disparity $d$. Equation 1 shows the recursive function, which starts at pixel $\mathbf{p}$ and sums the matching cost with the minimum of three options:

1) $L_{\mathbf{r}}$ of the previous pixel on the path with the same disparity.
2) $L_{\mathbf{r}}$ of the previous pixel on the path with a disparity jump of 1, with a jump penalty of $P_1$.

3) $L_{\mathbf{r}}$ of the previous pixel on the path with a disparity jump to the lowest cost, with a jump penalty of $P_2$.

Note that the paths should be traversed in the direction of r, from the edges of the image to the pixel, to make calculation straightforward.

$$L_{\mathbf{r}}(\mathbf{p}, d) = C(\mathbf{p}, d) + \min(L_{\mathbf{r}}(\mathbf{p} - \mathbf{r}, d), L_{\mathbf{r}}(\mathbf{p} - \mathbf{r}, d-1) + P_1, L_{\mathbf{r}}(\mathbf{p} - \mathbf{r}, d+1) + P_1, \min_i L_{\mathbf{r}}(\mathbf{p} - \mathbf{r}, i) + P_2) \quad (1)$$

Afterwards, the 8 pathwise costs are summed to a smoothed cost function $S(p, d)$. See Equation 2.

$$S(\mathbf{p}, d) = \sum_{\mathbf{r}} L_{\mathbf{r}}(\mathbf{p}, d) \quad (2)$$

Finally, a quadratic interpolation is performed to find a sub-pixel minimum using the costs of the minimum disparity and the two surrounding disparities.

#### 2.2.1. Local binary patterns

In [21], the predecessor of local binary patterns is first discussed under the name 'Texture Units', used as a means of texture classification. With this method, the image can be encoded by its local texture. The following algorithm is employed for each pixel: (I) A window is drawn around the center pixel. (II) Each pixel in this window is compared to the center pixel. When the value is higher, a 2 is stored, when equal, a 1, and when lower a 0. (III) These values are then concatenated clockwise in a vector, describing the local pattern around the pixel. [22] introduces local binary patterns, where the comparison with each pixel is binary instead of ternary: a higher value gives a 1, a lower or equal value gives a 0. The resulting vector for a 3x3 window can now be encoded as a byte. This process is shown in Figure 2.

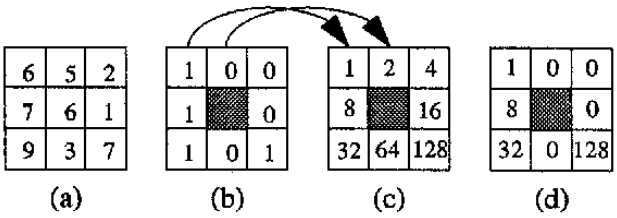
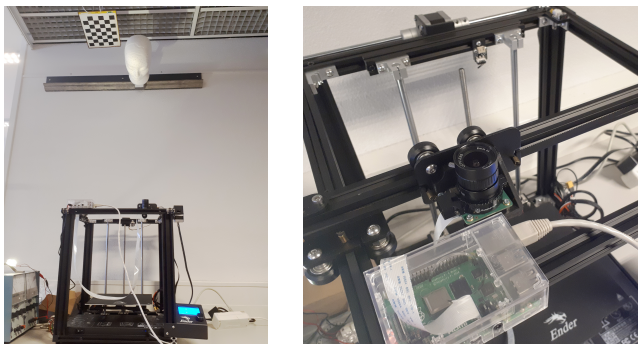



Figure 2: Overview of the generation of a local binary pattern. (a) shows a window of 9 pixels, (b) shows the window, thresholded with the center pixel. (c) shows the weight for each pixel, representing the bit shift to concatenate the bits. (d) shows the weighted values in the window, which can be added to a single byte. [22]

## 3. Methods

### 3.1. Building an experimental setup

There are several options to build an experimental setup that can take images from several different locations. However, considering cost, development time and the desired mutability of camera position, the presented setup moves a single camera to desired positions in an XY-plane. Instead of developing a motorized system, the camera is mounted on the head of a 3D printer. The Creality Ender-5 was selected on the following criteria: (I) the printer head should have two axes of translation. (II) The printing nozzle should be removable to replace it with a camera. (III) The printer head should have high precision and accuracy for its position placement. A Raspberry Pi 4b was used as an interface and a Sony IMX477 sensor with a 6mm lens was used to capture the images. The setup is shown in Figure 3. For each image, the 3D printer is instructed to move to its desired location by directly sending it G-code over USB using Pronsole [23]. After a short delay, to avoid any remaining vibrations, the Raspberry Pi is instructed to take a photograph over ethernet. This method gives the researcher the freedom to assess the performance of any camera arrangement in a 2D plane.



(a) Overview of setup

(b) Close-up of camera mounted on printing head

Figure 3: Photographs illustrating the setup used to emulate cameras at different positions.

### 3.2. Calibration

The calibration of the cameras is a two step process: intrinsic and extrinsic calibration.

#### 3.2.1. Intrinsic calibration

Using 250 images of a checkerboard calibration pattern in different positions and orientations, the intrinsic camera parameters are determined. This is done by first estimating the corners using a sector-based approach [24] and then calibrating the camera using Z. Zhang's calibration method [25]. The lens distortion can now be corrected, and the principal point and focal length are now known with a high precision.

#### 3.2.2. Extrinsic calibration

While the experimental setup approximately positions the camera in the right positions, the accuracy is not high enough to assume the images to be taken from the exact desired location. Therefore, an extrinsic calibration step is necessary to approximate the position and orientation of each camera.

For the extrinsic calibration, the same calibration pattern and corner extraction method is used for each image. The camera extrinsics are first optimized for each camera location individually using a gradient descent optimization over the estimated corner positions. This is done by optimizing the six degrees of freedom of the calibration pattern in each image over the reprojection error, while assuming the camera is at the origin without rotation. The positions of the camera are then found by inverting the position of the calibration pattern after each descent. Now the approximate positions have been found, a joint gradient descent is performed for all images simultaneously, optimizing the already found positions with additional restraints. Because of the use of a single moving camera and the planar motion of the setup, the orientation of all cameras will be considered equal, and will be jointly optimized while the positions are adjusted accordingly.
The found cameras lie on a plane, which is slightly tilted. This is caused by the difference in orientation between the camera and the plane. The angles of the plane are calculated and the images are rectified to the plane.

### 3.3. Digital setup

The digital setup is created in Blender [26]. A textured model of a head [27] is digitally placed in front of a camera array. Some options to add realism can be implemented: a light subsurface scattering effect can be applied to the face surface to simulate real skin, a slight depth-of-field effect is used to simulate a real camera diaphragm, and a noise texture is added to the whole image to simulate camera noise. Additionally, the positions of the cameras can be slightly varied to simulate an imperfect calibration. These options will reduce the chance for a correct match to be found in reconstruction, like imperfections expected in the real world. Figure 4 shows an example of these renders.

Figure 4: Two cropped renders from a central camera and a camera 10cm to the right of it.

### 3.4. Stereo matching

Many methods of multi-camera stereo reconstruction create disparity maps from different perspectives and later fuse these. The method presented in this paper generates a single cost function, generated by using all images, which is later filtered for smoothness. An origin camera $c_O$ with a corresponding origin image $I_O$ is selected which will be the perspective reference for the disparity map. It is preferably, but not necessarily, close to the average position of the cameras. While this method can use an arbitrary number of cameras, the rest of this section will be described with an example of 25 images taken by a 5 by 5 array of cameras, placed approximately 5cm apart. The following stepwise procedure is followed:
**(I)** Each image is first transformed to its local binary pattern, $LBP_c(\mathbf{p}_c)$. With $c$ being the camera index and $\mathbf{p_c} = [x_{p_c}, y_{p_c}]^T$ the pixel coordinates in the image.
**(II)** Using the calibrated camera positions, the epipolar line of each camera with the origin camera is calculated.
**(III)** A disparity step vector $S_c$ along the epipolar line is generated for each camera, containing the step in pixels to be made in an image to increase the disparity by 1. $|S_c|$ will be 1 for a camera next to the origin camera. i.e. a camera with a relative x,y position of $[5cm, 0cm]^T$ will have a step vector of $[1, 0]^T$. A camera at $[-10cm, 5cm]^T$ will then have a step vector of $[-2, 1]^T$.) Because the spacing of the cameras is not exact, these step vectors are stored as floating points.
**(IV)** For each pixel $\mathbf{p_{cO}}$ in the origin image and a selected disparity range $d \in [d_{min}, d_{max}]$, the required pixels are selected and stored in a matrix $LBP_c(\mathbf{p_{cO}}, d)$.

$$LBPd_c(\mathbf{p_{cO}}, d) = LBP_c\left(\mathbf{p_{cO}} + round(d\mathbf{S_c})\right) \quad (3)$$

**(V)** To compare the stored disparity ranges, edges $e = [c_1, c_2]$ are selected between all orthogonally adjacent cameras. For a 5 by 5 camera matrix, 40 edges are available. The choice is made to only compare cameras that are close together. The underlying idea is that pore level structure, which is too small to be detailed in disparity calculations, creates an apparent texture difference from different directions. This effect is minimized by only comparing adjacent cameras. The effect on the styrofoam test objects can be seen in Figure 5. The perspective distortion that can be seen in the last image compared to the other two is of small concern because of the matching method, but the change in texture decreases the chance for a correct match to be found.

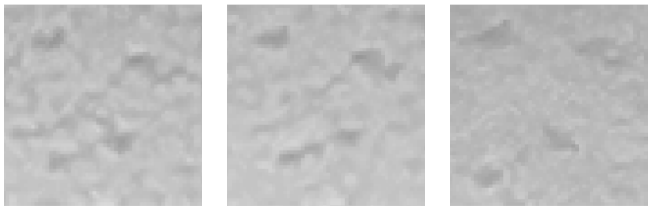


Figure 5: The first two images show the came surface as seen by two adjacent cameras at 5cm distance. The last image shows the surface as seen by a camera which is at 28cm distance from the first.

**(VI)** The initial cost matrix for each edge is determined by

the hamming distance [28] $H$ of the LBP of both cameras connected by the edge.

$$C_e(\mathbf{p_{cO}}, d) = H(LBPd_{c_1}(\mathbf{p_{cO}}, d), LBPd_{c_2}(\mathbf{p_{cO}}, d)) \tag{4}$$

To combine the cost functions of all edges, for each disparity at each pixel the costs under the median cost are selected in $c_{min}(\mathbf{p_{cO}}, d)$ and then averaged to provide a noise rejecting average.

$$C(\mathbf{p_{cO}}, d) = \frac{1}{size\ of\ c_{min}} \sum_{c \in c_{min}(\mathbf{p_{cO}}, d)} C_e(\mathbf{p_{cO}}, d) \tag{5}$$

If the disparity with the lowest cost is used for each pixel, the disparity map is very prone to noise, since a single pixel doesn't contain enough information to find a reliable match. To combine pixel information and enforce smoothness, a smoothed cost is determined using the semi-global matching[17] constraint. For each pixel, the disparity with the lowest smoothed cost is selected.

### 3.5. Orthogonality map generation

Parts of this papers experiments rely on the orthogonality of a reconstructed surface and the ray cast by a camera for each pixel. We define an orthogonality map $O(\mathbf{p})$ of equal size to a depth map, where the value of each pixel lies between 0 and 1. 0 meaning the surface is parallel to the camera ray for that pixel and 1 meaning it is completely orthogonal. $O(\mathbf{p})$ can be generated by taking the pixelwise dot product of the normalized surface normals of the reconstruction $\widehat{\mathcal{N}}(\mathbf{p})$, and the normalized ray vectors from a camera to the reconstructed surface $\widehat{\mathcal{R}}(\mathbf{p})$ The surface normals can be calculated as follows from a depth map $D(\mathbf{p})$:

$$\mathcal{N}(\mathbf{p}) = \begin{bmatrix} D\left(\mathbf{p} + \left[\begin{smallmatrix}1\\0\end{smallmatrix}\right]\right) - D\left(\mathbf{p} - \left[\begin{smallmatrix}1\\0\end{smallmatrix}\right]\right) \\ D\left(\mathbf{p} + \left[\begin{smallmatrix}0\\1\end{smallmatrix}\right]\right) - D\left(\mathbf{p} - \left[\begin{smallmatrix}0\\1\end{smallmatrix}\right]\right) \\ \frac{2s_p}{f} D\left(\mathbf{p}\right) \end{bmatrix} \tag{6}$$

Here $\mathbf{p} = [u_p, v_p]^T$ is the 2D pixel coordinate, $f$ is the camera focal length and $s_p$ is the physical pixel size. The ray vectors are calculated with:

$$\mathcal{R}(\mathbf{p}) = \begin{bmatrix} (u_p - u_0) D\left(\mathbf{p}\right) * s_p / f \\ (v_p - v_0) D\left(\mathbf{p}\right) * s_p / f \\ D\left(\mathbf{p}\right) \end{bmatrix} - c_p + c_r \tag{7}$$

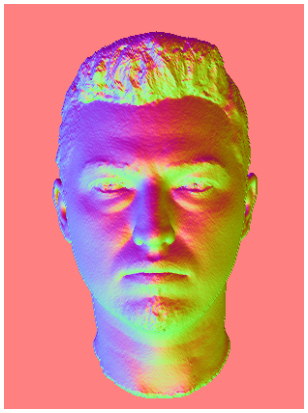
Where $u_0$ and $v_0$ describe the camera principal point, $c_p$ is the position of the camera from which perspective the depth map is generated, and $c_r$ is the camera which is the source of the rays. Both can then be normalized with:

$$\widehat{\mathcal{N}}(\mathbf{p}) = \frac{\mathcal{N}(\mathbf{p})}{||\mathcal{N}(\mathbf{p})||}, \qquad \widehat{\mathcal{R}}(\mathbf{p}) = \frac{\mathcal{R}(\mathbf{p})}{||\mathcal{R}(\mathbf{p})||} \tag{8}$$

Finally, the orthogonality is defined by the dot product:

$$O(\mathbf{p}) = \widehat{\mathcal{N}}(\mathbf{p}) \cdot \widehat{\mathcal{R}}(\mathbf{p}) \tag{9}$$

Figure 6 shows an example of a surface normal map and an orthogonality map.



(a) Normalized surface normals. x, y and z are portrayed in blue, green and red

(b) Orthogonality map from a camera positioned to the top right

Figure 6: Surface normals and orthogonality map, generated using a 3D model

## 4. Experiments and results

### 4.1. Performance of the physical setup

**Experiment**

The goal of the physical setup is to emulate a desired array of cameras. How well this array is being emulated can be described with the standard deviation and the bias of the camera position. In this experiment, the setup is instructed to take 10 series $s$ of photographs at 25 locations $l$ in a 5 by 5 grid with a spacing of 5cm. The precise position $p_{l,s}$ is determined by calibrating the cameras using the calibration step described in 3.2. The standard deviation from the average position per location is defined as:

$$\boldsymbol{\sigma} = \sqrt{\frac{1}{LS} \sum_{l \in L} \sum_{s \in S} (\mathbf{p}_{l,s} - \boldsymbol{\mu}_l)^2} \tag{10}$$

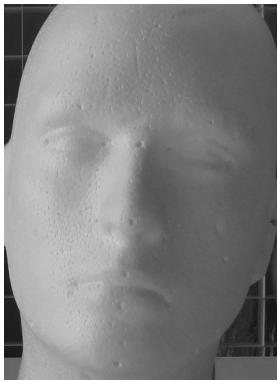
Where $L$ is the number of target locations, $S$ is the number of series and $\boldsymbol{\mu}_l$ is the average position for each location. The average absolute bias for each location is defined as:

$$\mathbf{b}(l) = \frac{1}{S} \sum_{s \in S} |\mathbf{p}_{l,s} - \mathbf{p}_l^t| \tag{11}$$

Here, $\mathbf{p}_l^t$ is the target position for the location. From the bias per location, the average bias and the average Euler distance can be determined.

**Results**

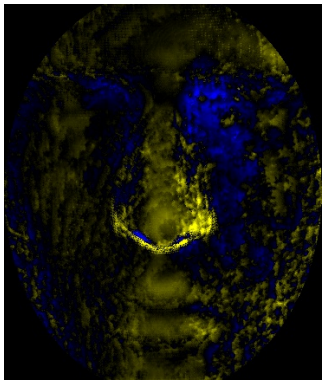
The standard deviation of the camera position in XYZ is $[54, 24, 35]^T \mu m$. This means the position is very reproducible. The average absolute bias is $[260, 560, 480]^T \mu m$. This means that, while the calibrated position for a certain location is very predictable, it is not very precise. With the current physical setup, a deviation of 1mm translates to a deviation of 4 pixels on a subject at 1 meter distance. To maintain precision it is therefore important that the



(a) Center image of the styrofoam face.

(b) Registered depth image of the ground truth face model.

(c) Registered depth image of the reconstructed face model, generated with 25 cameras at a 5cm baseline.



(d) Difference of reconstructed depth with the ground truth. Blue: reconstruction is too close, yellow: too far away.

Figure 7: Overview of the reconstruction and performance metric on a styrofoam face.

reconstruction method can determine and make use of any arbitrary camera position instead of assuming perfect placement.

## 4.2. Reconstruction performance

In this section, the reconstruction of a styrofoam head using the physical setup is discussed. While insufficient data has been collected to assess the performance of the reconstructions in 3D face recognition, an indication can be obtained by measuring the average error in reconstruction depth.

**Experiment**

The physical setup described in Section 3.1 is used to create an array of images taken in a 5 by 5 array with a baseline of 5cm. Using the reconstruction method from Section 3.4, a reconstruction is made of the head. The reconstruction is manually masked to select the facial area and registered to an intrinsic coordinate frame using Spreeuwers' registration method described in [29]. Then it is stored as a depth map. The same registration is performed to a high-quality model of the styrofoam head, to be used as ground truth. This model is generated with the Artec Eva, a high-quality hand-held structured-light reconstruction device which has to be manually rotated around the subject. The error is now found by subtracting the ground truth depth map from the registration depth map.

**Results**

Figure 7 shows an overview of the process and result. The two areas in Figure 7d that are clearly most problematic are the edge of the nose and the (from our perspective) right eye. The error in the eye can be explained by a lack of apparent texture around the area, caused by insufficient angled lighting. Illumination from from an angle to the surface causes the styrofoam to cast a shadow on its pores, while illumination orthogonal to the surface or scattered illumination produces almost no shadow, creating an almost textureless surface. This effect can be seen, albeit to a lesser degree, on the whole right side of the face. The error in the edge of the nose can be explained by multiple factors: (I) Because of the angle, less information is available on these surfaces. (II) Because occlusion is currently not handled by the proposed method, cameras with no vision of a surface are still be taken into consideration. (III) Because of the tendency of semi-global matching to reward low disparity steps, sharp edges will be smoothed. Figure 8 shows a single depth row of the reconstruction and ground truth depth maps, through the part of the nose with the worst results. This clearly shows the advantage of the well lit left side.

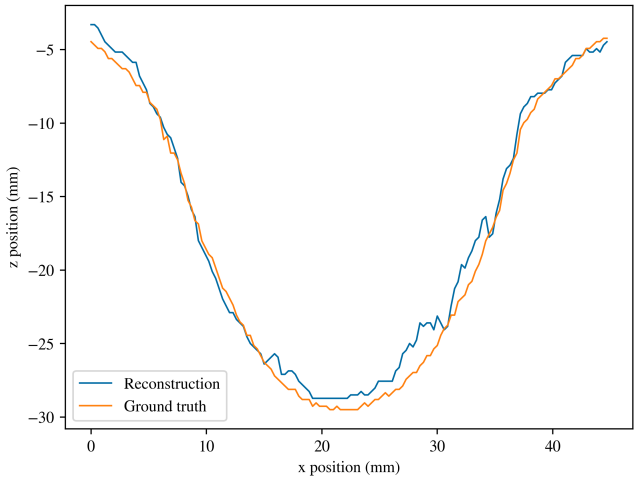



Figure 8: Comparison of nose depth between the reconstruction and the ground truth over the X-axis

To convey a more intuitive meaning of the difference between the reconstruction and the ground truth, Figure 9 shows a 3D rendered version of the reconstruction, with the ground truth in the background. Note that the well-lit side of the reconstructed model is prominently shown. Because the results of the dimly lit side misrepresent the performance of

the system, only the well-lit side is used to assess the results in Table 1.

| Region | Average absolute error |
|---|---|
| Overall | 0.45mm |
| Nose | 0.80mm |
| Overall without Nose | 0.38mm |

TABLE 1: Average absolute error for masked parts of the reconstructed face depth map.

Figure 9: 3D render of the reconstruction in the foreground and the ground truth in the background.

## 4.3. Influence of variables on reconstruction performance

In this section, a series of experiments is executed to measure the influence of several parameters. Each experiment is executed both on (I) images of a 10cm radius styrofoam ball, acquired from a physical setup and on (II) images of a head model [27], rendered with Blender [26] (see Section 3.3). This allows for a result under real-life conditions and under more ideal circumstances. Both subjects are at approximately 75cm from the camera plane. The reconstructions are masked using (I) the shape of the ball and (II) the outline of the face, found using the 68-point face landmark detection [30] in the dlib [31] library. For the render, camera noise and depth-of-field are simulated to increase the similarity to a real-life scenario. The performance metric will be the average absolute deviation (AAD) in depth error for surfaces with similar orthogonality (see Section 3.5). This choice is made because the performance of regions with equal orthogonality is expected to be comparable. For the physical setup, a depth map of an ideal ball is generated by determining the position and radius of the ball on the image, determining its real world position, and ray casting the depth for each pixel. For the rendered head model, the 3D model itself is available and will be used to create a ground truth depth map. The difference in depth between the reconstruction and the ground truth for each pixel will be stored as an error map. Four experiments are executed: (a) measurement of the performance of the full setup with 25 cameras, (b) measurement of the influence of the camera baseline, (c) surface orthogonality, and (d) camera count on the reconstruction performance. Afterwards, the similarities and differences of the physical setup (I) and rendered setup (II) are discussed.

#### 4.3.1. Performance of the full setup

**Experiment**

A reconstruction is made, using a 5 by 5 grid of cameras with a baseline of 5cm. A orthogonality map is generated from the perspective of the center camera with the method described in Section 3.5. The AAD and the bias (average distance to ground truth) are evaluated against the orthogonality.

**Results**

Figures 11 and 12 show the results of the full setup. As expected, the AAD in Figure 11 is lowest for surfaces facing the camera. Figure 12 shows that the system has a bias. This could be a function of orthogonality, but also of depth, which is generally higher with higher orthogonality. It could be caused by small disparity estimation errors, amplified by the smoothing constraint of SGM. In the physical setup, the bias could also be caused by errors in the estimation of the ball world coordinates. More research is necessary to draw conclusions.

#### 4.3.2. Influence of camera baseline

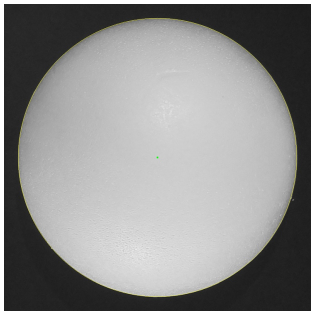
The distance between matching cameras (baseline) has an influence on the reconstruction performance. With a small baseline, a small error in disparity leads to large errors in depth, while a large baseline leads to more apparent dissimilarity of the matched surface. A large baseline will also provide more information on surfaces at high angles to the center camera, because more pixels are available to describe these surfaces.

**Experiment**

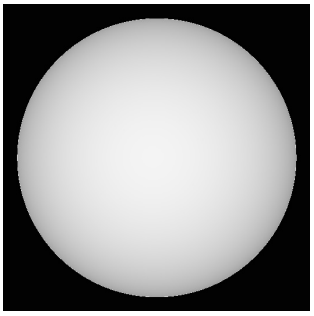
For this experiment, a reconstruction is made with 5 cameras in a '+' arrangement with several different baselines. The AAD of the error is recorded as a function of orthogonality to the center camera. The results will be divided in two categories: high orthogonality ($O > 0.35$) and low orthogonality ($O \leq 0.35$). This will allow for a metric in both flat areas like the forehead, chin, nose bridge and part of the cheeks, and also for high angled areas, like the nose edges and the edges of the cheeks. Figure 13 shows a selection of these high angled areas.

**Results**

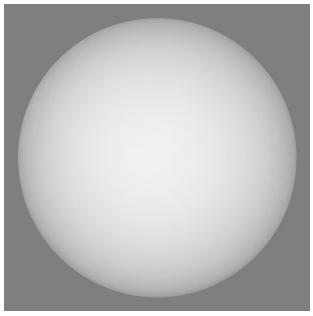
Figures 14 and 15 show the AAD at several baselines between 5 and 100mm. Figure 14 shows that for low angled areas, the best results are achieved for a baseline between 50 and 100mm. This can partially be explained by the decreased sensitivity to a disparity error for a large baseline. Semi-global matching rejects the fronto-parallel assumption in the initial cost function, but in the smoothness constraint it is



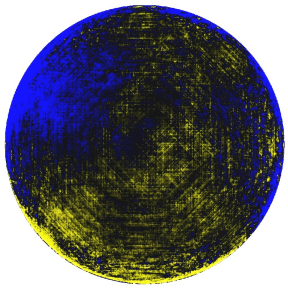
(a) Center image of the styrofoam ball. The yellow ring shows the found circle used for generating the ground truth.



(b) Disparity of the ground truth ball model.



(c) Disparity of the styrofoam ball, generated with 25 cameras at a 5cm baseline.



(d) Difference of reconstructed depth with the ground truth. Blue: reconstruction is too close, yellow: too far away.

Figure 10: Overview of the reconstruction and performance metric on a styrofoam ball.

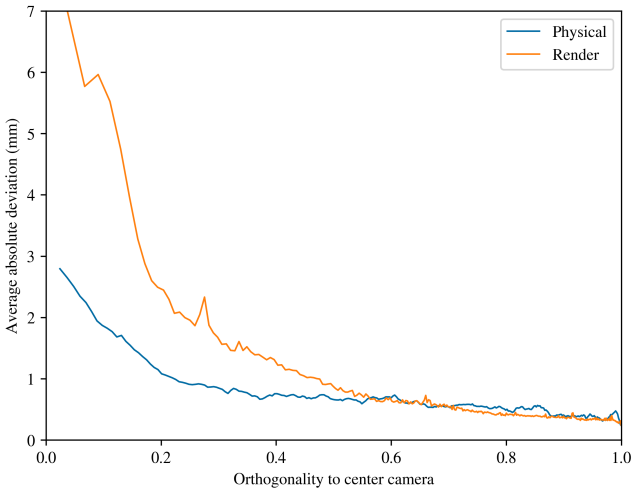



Figure 11: Average absolute deviation versus the surface orthogonality to the center camera. Reconstruction with 25 cameras in a 5 by 5 array with a baseline of 5cm at a distance of 75cm to the subject.

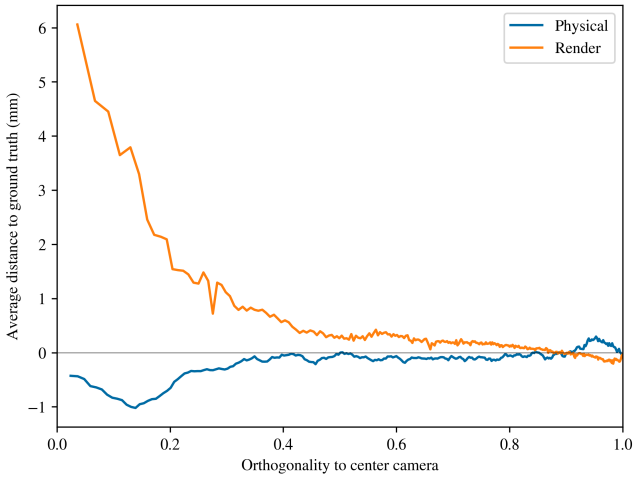



Figure 12: Average distance to the ground truth versus the surface orthogonality to the center camera. Reconstruction with 25 cameras in a 5 by 5 array with a baseline of 5cm at a distance of 75cm to the subject.

still lightly enforced with a penalty for a change in disparity. Because surfaces with high orthogonality satisfy the fronto-parallel assumption, large baselines are not penalized and provide good results. Figure 15 shows a minimum for angled surfaces at baselines between 40 and 60mm. The decreased performance at low baselines can be explained by the lower visibility of the high angled surfaces, while the decreased performance at high baselines can be explained by the broken fronto-parallel assumption, which has stronger implications for higher baselines.

#### 4.3.3. Influence of object surface angle on reconstruction performance

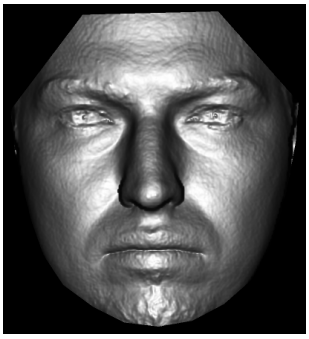
The orthogonality of a surface to the camera intuitively has an influence on the reconstruction performance. A surface that is completely orthogonal has the full amount of pixels available to describe it, while a surface at a 90 degree angle cannot be seen at all. The influence of orthogonality will be examined in this experiment.

**Experiment**

A 5-camera reconstruction is made in a '+' shape for each of the 9 cameras in the center of the 5 by 5 array. The error for each pixel is found by comparing the reconstruction to the ground truth. For each used camera an orthogonality map is generated. The AAD is then plotted against the orthogonality of the surface.

**Results**

Figure 16 shows a clear decrease in performance for orthogonalities under 0.4 and a fairly stable performance for orthogonalities above 0.4. Since the information available on a surface linearly scales with its orthogonality, a theoretical linear relationship could be expected between the AAD and the orthogonality. However, for low orthogonalities this is prevented by SGM's smoothness constrain, which uses in-



(a) Orthogonality of the digital face. white is 1, black is 0.

(b) White marks the selection of the surface with an orthogonality under 0.35

Figure 13: Orthogonality of a central camera with the digital face and a selection of the high-angled surfaces

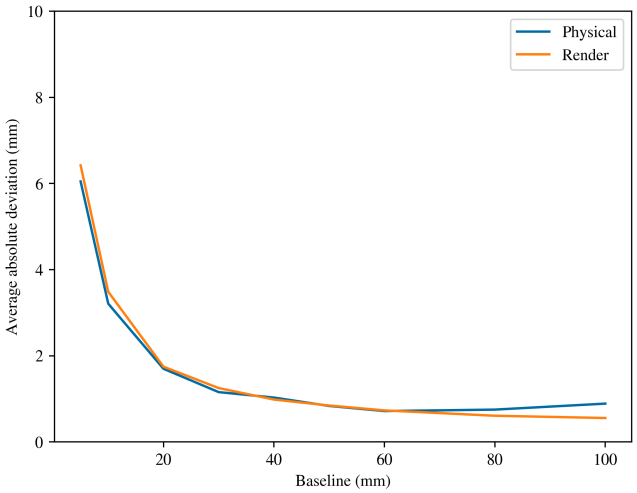



Figure 14: Average absolute deviation for low angled areas ($O > 0.35$) at several camera baselines.

formation of the surrounding pixels to increase performance. For high orthogonalities, the reconstruction method itself limits the performance by not sufficiently making use of the available information.

#### 4.3.4. Influence of camera count on reconstruction performance

The use of more cameras theoretically gives more information of the subject. The noise in reconstruction is averaged and, by providing cameras a different positions, more information on angled surfaces is available.

**Experiment**

A reconstruction is made for an incremental amount of cameras in a 5 by 5, 5cm baseline array. The AAD of the error is again plotted for a high-angled region and a low-angled region.

**Results**

Figures 17 and 18 show the AAD for low angled surface areas and high angled surface areas. As expected, the AAD

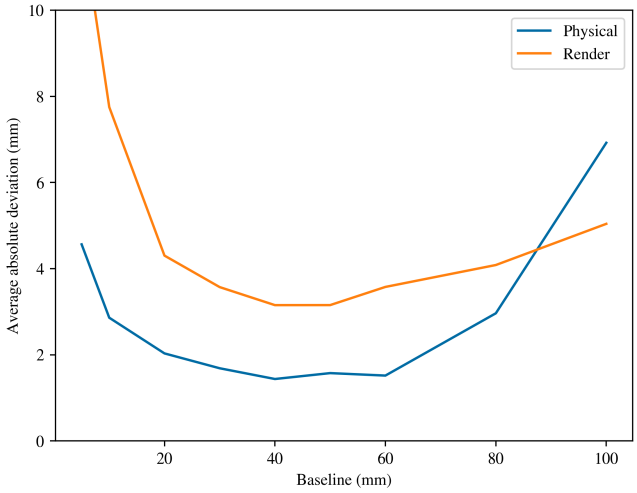



Figure 15: Average absolute deviation for high angled areas ($O \leq 0.35$) at several camera baselines.

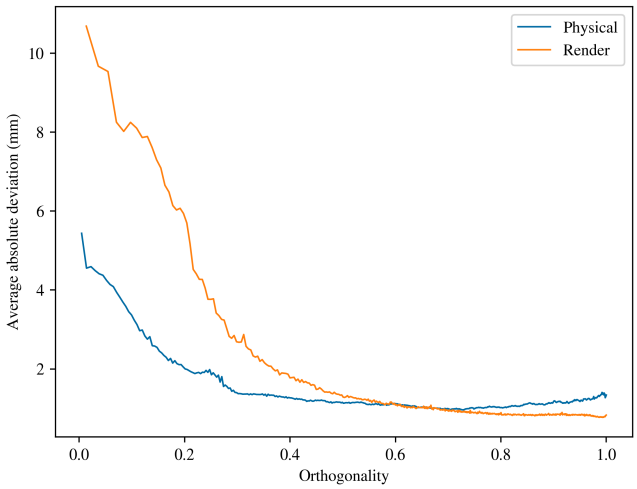



Figure 16: Average absolute deviation against surface orthogonality for the physical setup and the digital render.

decreases with the number of cameras. However, there are clear diminishing returns beyond a camera count of 9. Even for the low angled surfaces, the reached limit is far above the theoretical minimum. An estimation of this theoretical limit, as described in [3] was added to Figure 17. The starting value of the theoretical minimum is chosen between the physical and render starting AAD. The large distance to this minimum suggests that the presented reconstruction method is not utilizing the available information of multiple cameras to its full potential.

#### 4.3.5. Render setup performance

No quantitative results can be given on the ability of the digital render setup to emulate real-world conditions, as the physical results were achieved with a subject of a different shape and texture. However, by comparing the graphs in most

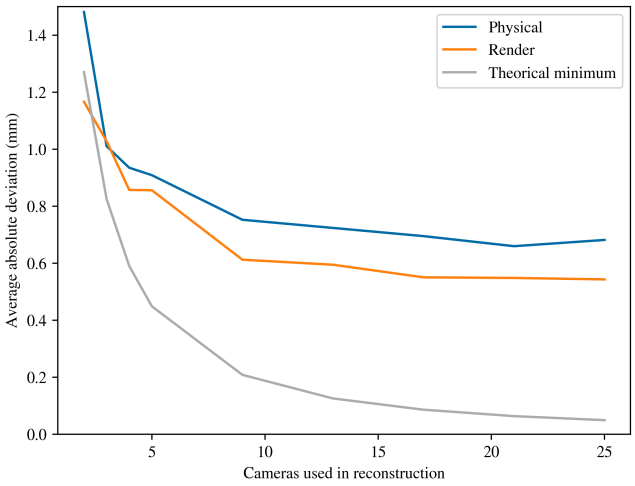



Figure 17: Average absolute deviation against camera count for low angled areas ($O > 0.35$).

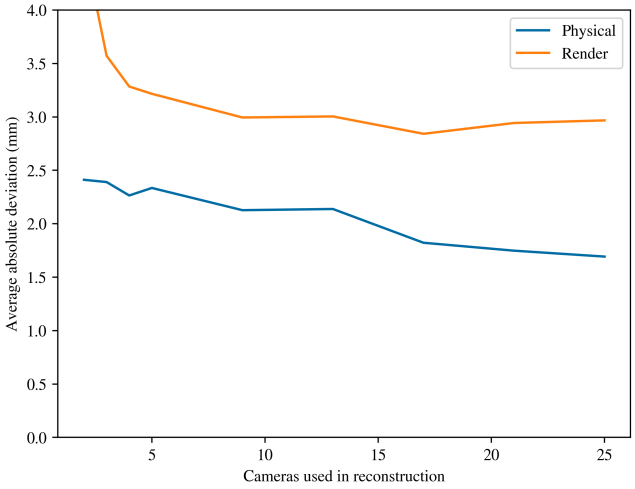



Figure 18: Average absolute deviation against camera count for high angled areas ($O \leq 0.35$).

experiments, one can see that regions with high orthogonality have very similar results between the render and the physical setup, while in regions with low orthogonality, the render strongly under-performs, albeit with similar tendencies in the data. This may partially be caused by the overuse of simulated camera noise, which has more influence on surfaces at low orthogonality, since little information is available for these. Further research is necessary to find one or multiple clear origins of the performance differences.

## 5. Conclusions

A method is presented to generate a 3D face reconstruction using an arbitrary amount of cameras in a single plane. A physical setup and a digital simulation are introduced to produce input images to this method. It is then used to assess the influence of several parameters: the amount of cameras, the distance between the cameras and the orthogonality of the reconstructed surface. The precision and accuracy of the physical setup is measured and the ability of the digital simulation to emulate a real-life scenario is also assessed by qualitatively comparing the results to the physical ones. While the performance of the system itself can be measured and general tendencies can be found, no strong general conclusions can be drawn from the results, as the calibration, the reconstruction method and the performance metric can and should all still be greatly improved.

The experiment on the styrofoam head shows a performance that suffices for 3D face recognition. However, with a sample size of one, no real conclusions can be drawn. The experiments show that the orthogonality of a surface is important to the reconstruction result, but with diminishing results beyond an orthogonality of 0.5. Increasing the amount of cameras leads to better performance, but not to a degree comparable with the theoretical performance. This is deemed a result of the method of reconstruction and assessment. Higher baselines between the cameras improve the results for surfaces angled towards the center camera, but after a baseline of 60mm, this is at the cost of decreasing the performance of the reconstruction of surfaces at an angle to the camera.

All in all, this paper introduces a broad pipeline of methods to research multi-perspective stereoscopy. While improvements are possible, and advised, for most presented methods, I hope that the high potential of multi-perspective stereoscopy for 3D face reconstruction has been conveyed.

## 6. Future research

There are many sections of this paper that require further investigation. Each will be highlighted in a separate paragraph.

**Further research on matching**

The current matching function, local binary pattern, was chosen based on the reported leading performance in noisy environments, which is ideal for the inexpensive camera used in the setup. However, it is not fine-tuned to work with face textures. Choosing a better matching function or altering the current one might lead to improved results. Also, the current choice to match only orthogonally adjacent cameras needs further research. With the current methods it leads to the best results, but this may be the product of lacking methods instead of lacking information.

**Improved test models**

Unlike human skin, styrofoam has deep pores, which cause stronger visual differences from different angles. Using a more realistic human face model will put the test setup closer to real-use performance. In addition, this paper only used a styrofoam ball to assess most parameters because the comparison with a ground truth model for a styrofoam face was technically more complex. A robust method to assess performance on a physical face would be preferable. Lighting of the object is also very important. When describing the effect of surface angle

on reconstruction performance, it is very important that illumination differences do not influence the results. This is something insufficiently researched in this paper and needs further work.

**More elaborate render setup**

If the parameters and circumstances of a rendered setup are very closely matched to a real-world scenario, this might allow for complete emulation of a camera array or video with every desired camera position and orientation without building a setup. This paper insufficiently matches the render setup to the physical setup, which makes it impossible to assess whether conclusions can be drawn from only the render setup.

**More segmented experiments**

The experiments in this paper on surface angle, camera baseline and camera count are not sufficiently independent. For example, by increasing the baseline, the orthogonality of at least one of the cameras to the surface is changing. More independent experiments should be done in order to get a better understanding of the effects of these variables.

**Building a setup with 25 cameras**

A 25 camera setup allows for very precise camera calibration and opens the door to single-shot reconstruction, which can be used to capture real subjects and test facial recognition performance. The disadvantage being that the camera locations can no longer be easily changed, so research on camera distance should be finalized first.

## References


[1] Nick Pears and Ajmal Mian. "3D Face Recognition". In: *3D Imaging, Analysis and Applications*. Ed. by Yonghuai Liu et al. Cham: Springer International Publishing, 2020, pp. 569–630. ISBN: 978-3-030-44070-1. DOI: 10.1007/978-3-030-44070-1_12. URL: https://doi.org/10.1007/978-3-030-44070-1_12.

[2] Luuk Spreeuwers. "Multi-view passive 3D face acquisition device". In: *Journal of Vacuum Science & Technology A - J VAC SCI TECHNOL A* (Jan. 2008).

[3] Wolfgang Forstner. "On the Theoretical Accuracy of Multi Image Matching, Restoration and Triangulation". In: (July 2000).

[4] Omdia Smartphone Intelligence Service. URL: https://omdia.tech.informa.com/pr/2020-dec/quad-camera-is-the-most-popular-camera-setup-for-smartphones.

[5] Xiao Huang et al. "Target enhanced 3D reconstruction based on polarization-coded structured light". In: *Optics express* 25.2 (2017), pp. 1173–1184.

[6] Zhenzhou Wang. "Robust three-dimensional face reconstruction by one-shot structured light line pattern". In: *Optics and Lasers in Engineering* 124 (2020), p. 105798. ISSN: 0143-8166. DOI: https://doi.org/10.1016/j.optlaseng.2019.105798. URL: https://www.sciencedirect.com/science/article/pii/S0143816619305093.

[7] Boulbaba Ben Amor, Mohsen Ardabilian, and Liming Chen. "3D Face Modeling Based on Structured-Light Assisted Stereo Sensor". In: *Image Analysis and Processing – ICIAP 2005*. Ed. by Fabio Roli and Sergio Vitulano. Berlin, Heidelberg: Springer Berlin Heidelberg, 2005, pp. 842–849.

[8] J. Apolinar Muñoz-Rodríguez. "Shape connection by pattern recognition and laser metrology". In: *Appl. Opt.* 47.20 (July 2008), pp. 3590–3608. DOI: 10.1364/AO.47.003590. URL: http://ao.osa.org/abstract.cfm?URI=ao-47-20-3590.

[9] Simon Meers and Koren Ward. "Face Recognition Using a Time-of-Flight Camera". In: *2009 Sixth International Conference on Computer Graphics, Imaging and Visualization*. 2009, pp. 377–382. DOI: 10.1109/CGIV.2009.44.

[10] Araceli Morales, Gemma Piella, and Federico M. Sukno. "Survey on 3D face reconstruction from uncalibrated images". In: *Computer Science Review* 40 (2021), p. 100400. ISSN: 1574-0137. DOI: https://doi.org/10.1016/j.cosrev.2021.100400. URL: https://www.sciencedirect.com/science/article/pii/S157401372100040X.

[11] "Efficient 3D reconstruction for face recognition". In: *Pattern Recognition* 38.6 (2005). Image Understanding for Photographs, pp. 787–798. ISSN: 0031-3203. DOI: https://doi.org/10.1016/j.patcog.2004.11.004. URL: https://www.sciencedirect.com/science/article/pii/S0031320304003991.

[12] Pengfei Dou, Shishir K. Shah, and Ioannis A. Kakadiaris. "End-To-End 3D Face Reconstruction With Deep Neural Networks". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. July 2017.

[13] Baris Gecer et al. "GANFIT: Generative Adversarial Network Fitting for High Fidelity 3D Face Reconstruction". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. June 2019.

[14] Jiaxiang Shang et al. "Self-supervised monocular 3d face reconstruction by occlusion-aware multi-view geometry consistency". In: *arXiv preprint arXiv:2007.12494* (2020).

[15] Federico Pedersini, Augusto Sarti, and Stefano Tubaro. "Multi-Camera Acquisitions for High-Accuracy 3D Reconstruction". In: *3D Structure from Multiple Images of Large-Scale Environments*. Ed. by Reinhard Koch and Luc Van Gool. Berlin, Heidelberg: Springer Berlin Heidelberg, 1998, pp. 124–138. ISBN: 978-3-540-49437-9.

[16] Thabo Beeler et al. "High-Quality Single-Shot Capture of Facial Geometry". In: *ACM Trans. Graph.* 29.4 (July 2010). ISSN: 0730-0301. DOI: 10.1145/1778765.1778777. URL: https://doi.org/10.1145/1778765.1778777.

[17] Heiko Hirschmuller. "Stereo processing by semiglobal matching and mutual information". In: *IEEE TPAMI* 30.2 (2008), pp. 328–341.

[18] P. Viola and W.M. Wells. “Alignment by maximization of mutual information”. In: *Proceedings of IEEE International Conference on Computer Vision*. 1995, pp. 16–23. DOI: 10.1109/ICCV.1995.466930.

[19] S. Birchfield and C. Tomasi. “Depth discontinuities by pixel-to-pixel stereo”. In: *Sixth International Conference on Computer Vision (IEEE Cat. No.98CH36271)*. 1998, pp. 1073–1080. DOI: 10.1109/ICCV.1998.710850.

[20] Heiko Hirschmüller. “Semi-Global Matching: Motivation, Development and Applications”. In: Sept. 2011, pp. 173–184.

[21] Li Wang and Dong-Chen He. “Texture classification using texture spectrum”. In: *Pattern Recognition* 23.8 (1990), pp. 905–910. ISSN: 0031-3203. DOI: https://doi.org/10.1016/0031-3203(90)90135-8. URL: https://www.sciencedirect.com/science/article/pii/0031320390901358.

[22] T. Ojala, M. Pietikäinen, and D. Harwood. “Performance evaluation of texture measures with classification based on Kullback discrimination of distributions”. In: *Proceedings of 12th International Conference on Pattern Recognition* 1 (1994), 582–585 vol.1.

[23] Kliment Yanev. *Pronterface*. URL: https://www.pronterface.com/.

[24] Alexander Duda and Udo Frese. “Accurate Detection and Localization of Checkerboard Corners for Calibration”. In: Sept. 2018.

[25] Z. Zhang. “A flexible new technique for camera calibration”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 22.11 (2000), pp. 1330–1334. DOI: 10.1109/34.888718.

[26] Blender Online Community. *Blender - a 3D modelling and rendering package*. Blender Foundation. Stichting Blender Foundation, Amsterdam, 2018. URL: http://www.blender.org.

[27] Ruslan Vasylev. *Male Scan - Head RAW - zBrush*. URL: https://www.turbosquid.com/FullPreview/Index.cfm/ID/777450.

[28] R. W. Hamming. “Error detecting and error correcting codes”. In: *The Bell System Technical Journal* 29.2 (1950), pp. 147–160. DOI: 10.1002/j.1538-7305.1950.tb00463.x.

[29] Luuk Spreeuwers. “Fast and Accurate 3D Face Recognition”. In: *International Journal of Computer Vision* 93.3 (2011), pp. 389–414. ISSN: 1573-1405. URL: http://dx.doi.org/10.1007/s11263-011-0426-2.

[30] Vahid Kazemi and Josephine Sullivan. “One millisecond face alignment with an ensemble of regression trees”. In: *2014 IEEE Conference on Computer Vision and Pattern Recognition*. 2014, pp. 1867–1874. DOI: 10.1109/CVPR.2014.241.

[31] Davis E. King. “Dlib-ml: A Machine Learning Toolkit”. In: *Journal of Machine Learning Research* 10 (2009), pp. 1755–1758.