

# 3D finger vein patterns: acquisition, reconstruction and recognition

Thomas van Zonneveld

*Data-management and biometrics group University of Twente - Enschede  
Faculty of Electrical Engineering, Mathematics and Computer Science (EEMCS)*

**Abstract**—Fingerprint and facial recognition systems are widely used for recognition and identification purposes. However, a drawback of these methods is that they are relatively easy to spoof, since the biometric features are acquired from surface of the human body. A partial solution to this problem is the use of vein patterns from inside the fingers. Typically, used vascular systems are 2D systems and the rotations that occur during the acquisition phase may potentially not be accounted for, resulting in a lower accuracy of these systems. Previous research has attempted to identify and address these issues by developing 3D vein pattern recognition systems, but the quantity of papers written on this topic is severely limited and mostly these works are not well documented. To fill this research gap, this work develops a new finger vein scanner that allows 3D vein patterns to be constructed by combining multiple highly detailed 2D vein images. These images are used to construct 3D reconstructions and to perform recognition experiments. The results indicate that 2D systems can perfectly handle rotations in artificial fingers and achieve 99% accuracy for 2D while 98% accuracy is achieved for 3D. For real fingers 2D outperforms 3D with 95% accuracy for 2D compared to 91% for 3D, and the EER rates are 0.045 and 0.089 for 2D and 3D respectively.

## I. INTRODUCTION

Biometrics are part of everyday life. They are used in many different areas, from border control and personal identification to crime scene investigations and smartphone unlocking. In these situations biometrics rely on individuality, meaning that each person has their own unique biometric features, and consistency, meaning that the patterns are unique over time. Fingerprints or facial recognition methods are often used as these are non-invasive and well researched.

However, a disadvantage of fingerprints and facial recognition is that traces of fingerprints are left almost everywhere and (fake) photos of faces can be easily obtained. Consequently, recognition systems can be relatively easy spoofed; fingerprints can be replicated from clay and facial systems can be duped with photographs. Furthermore, these features are obtained from the surface of the human body, implying that they are affected by external factors such as skin diseases, wounds or wrinkles. Logically, these factors can affect recognition performance.

A solution to these problems is the use of vascular biometrics; making use of the shape and patterns of blood vessels and veins inside the human body. By using near infrared (NIR) light (light with a wavelength ranging from 700nm to 1300nm), veins can be detected as shadows because blood

absorbs NIR light [3], [12], [16] ( see image 1). Research in this area is gaining attention and although vascular patterns can be detected almost anywhere in the body, research often focuses on vein patterns in the fingers because they contain many blood vessels that are convenient to detect.

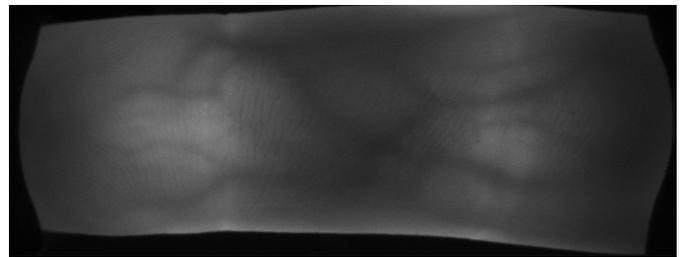


Fig. 1: Raw capture of finger veins using NIR transmission

The advantage of finger vein patterns over fingerprints and facial recognition is that vascular features are protected by the human skin and are thus more difficult to acquire for spoofing purposes. Finger vein patterns are not susceptible to skin deformation [20], they are unique and constant, even between identical twins, and have high accuracy [5]. In addition, finger veins can be detected without physical contact with the sensor, making this the method more hygienic [2].

Unfortunately, current 2D vein recognition systems are potentially prone to translations/rotations/transformations in finger registration, as fingers can easily move during the acquisition process. This increases the complexity of recognition as a small rotation along the longitudinal axis would significantly change the detected vascular pattern [19]. In [9], affine transformations are used to align fingers affected by translations and rotations. It estimates errors and attempts to compute pixel deviations for a plausible range of motion, but this comes at cost of processing time. Similarly, [6] notes that simple normalization and matching methods can avoid some of the transformations, but not all.

By creating a 3D structure of the veins, these problems can potentially be overcome as 3D clouds can be rotated [22]. Also, an attempt can be made to find a rotation/translation vector between two 3D vein patterns that minimizes the error between the two 3D patterns. Additionally, A 3D scan can be used to handle rotations by projecting a 3D scan to a 2D scan for different views/angles. Thus, a 3D scan can improve the quality of 2D scans [30]. Moreover, in [25] it

was demonstrated that even a 2D finger vein recognition system can be fooled with printed patterns of veins. With a scanner that uses multiple perspectives, this becomes much more difficult [21]. Furthermore, a 3D scan contains more spatial information than a 2D scan [31], and if the depth of veins is taken into account in recognition systems, this can potentially improve the recognition performance of finger vein scanners [15], [33].

A schematic drawing of a 3D finger vein recognition system is shown in figure 2 and roughly consists of three steps. First, 2D finger images must be acquired. At least two images are needed for 3D reconstruction and these images may need to be preprocessed to improve quality and contrast. Second, veins must be extracted from these finger images and 3D vein clouds can be reconstructed by correlating these points. Third, recognition experiments can be performed to analyze the potential benefits of the 3D reconstructed veins.

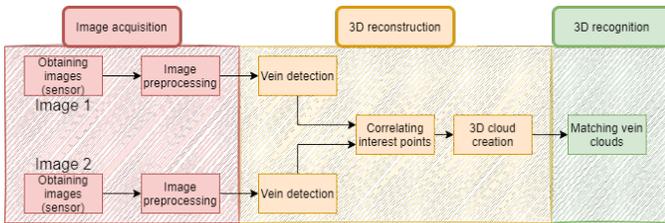


Fig. 2: Reconstruction pipeline for vein images

### A. Contributions

In this study a qualitative experiment is conducted to assess the effects of rotation on recognition performance. As far as the authors are aware, this has not been done before in the existing literature. Furthermore, there is no literature that has combined a 3-camera setup with the proposed approach to produce 3D vein reconstructions.

The main contribution of this work is the development of a new sensor capable of capturing highly detailed vein images from multiple perspectives, suitable for 3D reconstructions. This sensor can not only be used for 3D recognition but has proven its usefulness in a number of other projects at the DMB faculty. In addition, a suitable 3D reconstruction method has been researched and implemented, and 3D recognition experiments have been performed using this method on a number of artificial fingers whose precise rotations are known.

### B. Research questions

The aforementioned contributions stem from the general goal of gaining more insight into the potential problems of 2D vein recognition and the possible improvement of the recognition performance of 3D vein patterns over 2D vein pattern.

- Investigate the potential benefits of 3D finger vein recognition with respect to 2D finger vein recognition.

This implies the development of a system capable of generating 3D reconstructions of veins in a human finger, and

evaluating the advantages in terms of recognition of these 3D patterns over 2D patterns and to achieve this goal, three intermediate steps are required. First, a physical setup capable of acquiring 2D images of finger veins with the goal of 3D reconstruction is needed. Second, 3D reconstructions must be created from these 2D images. Third, the advantages of 3D vein reconstructions compared to 2D recognition should be analysed. Since a physical setup that captures single 2D images has already been developed in a previous project ([23]), it is attempted to improve this sensor instead of creating an entirely new sensor. Summarizing, the following research questions are formulated.

- 1) Develop a physical setup that is suitable for acquiring finger vein patterns in 2D suitable for 3D reconstruction  
How to revise the current available vein scanners to obtain good quality recordings of finger veins from multiple viewing angles.
- 2) Obtain 3D finger vein reconstructions  
What is a suitable method for 3D reconstruction of finger vein images.
- 3) Investigate the benefits of recognition with 3D reconstructed finger veins with respect to 2D finger vein recognition.

### C. Layout of this research

This research begins with a brief discussion of previous research in section II and a analysis of available scanners in section II-A. After a brief mathematical introduction to camera models and 3D reconstruction is given in section III, two 3D reconstruction methods are presented in section IV and in section V recognition performance evaluation methods are discussed. In section VI, the development of a new finger vein scanner is discussed. Section VII discusses the results of the two reconstruction methods and the recognition results are given in section VIII. A discussion is given in section IX and conclusions on the research questions are given in section X.

## II. RELATED WORK

The work done on 3D vein reconstruction to overcome the potential limitations in recognition caused by rotation is limited. However, research has been conducted on fusing multiple vein images by using multiple cameras at different angles in order to increase the recognition performance. Similarly, vein pattern detection in 2D is well researched.

In [10], the authors propose an approach that fuses together two images of finger veins. Based on a database of 6976 images, the results show that equal error rates can be more than halved when features are fused. Also in [20], this view is supported by evaluating a varying number of combined views of finger vein images (with 1° degree increments). By using preprocessing techniques such as ROI detection and contrast enhancement (CLAHE) and the maximum curvature method for vein detection [14] and image enhancement, the authors claim to lower the EER in recognition experiments

from 0.47 for a single image to 0.08 using both dorsal and palmar views simultaneously. In [21], this experiment is repeated, lowering the EER even further from 0.44 to 0.12 and 0.036 by using two and three views, respectively. In general, these results show that fusing the dorsal and palmar views together significantly reduces the EER. However, both [20], [21] indicate that for single views, the best results can be obtained with the palmar views of the fingers.

A first approach to overcome the problems of rotation and lack of depth information using a 3D system in hand-vein images is presented in [32]. Zhangs uses a dual camera system where the cameras have slight variations in their optical axis. Edge detection and CLAHE are used for vein detection. Using SAD and KC to create and correlate point clouds, a correlation matrix is created for 18 images. No numerical results are given. Similarly, in [11], two cameras are used to acquire a 3D point cloud of finger veins. The images are preprocessed using CLAHE and Sobel-detected edges are used as keypoints. These keypoints are correlated with SAD, and triangulation is used to create a 3D point cloud. The iterative closest point algorithm (ICP) is used to match two point clouds together. Depending on the threshold and the number of iterations, the matching times range from five to 37 seconds. Similar to [32], no quantitative matching results are given.

A setup that uses three camera's with an angle of about  $20^\circ$  between them is proposed in [1]. Two pairs of two cameras can be used to generate two 3D point clouds. Veins are detected using the repeated line algorithm [13]. Images are rectified and SAD is used in combination with disparities to obtain depth information for each point of interest. Both clouds are fused together and results show that 3D clouds of a paper model differ less than 2.5 pixels compared to the actual 3D model. In terms of recognition performance, no verification experiments were performed.

A different approach is taken in [7]. The entire finger is projected in 3D on which the veins are mapped. Vein recognition is done using convolutional neural networks, and the authors claim that the EER for their method was halved (2.37) compared to their 2D methods (6.53, 7.00, 6.70).

#### A. Available scanners

Two vein scanners have been developed at the University of Twente. A first version, V1, was developed by Bram Ton in 2012 [27]–[29]. Using a NIR transmission method and a reflective IR mirror, NIR light is reflected into a single monochrome camera (BCi5) equipped with a IR pass filter. IR Leds (SFH4550) with a peak frequency of 860nm are used. The resolution of the camera is  $1280 \times 1024$ px, but the region of interest has a resolution of  $672 \times 380$ px. The accompanying software is MATLAB based. Although the images are of good quality, the setup is impractical due to its size and weight.

To address the drawbacks of Ton's scanner, a new scanner, V2, was developed in 2018 by Sjoerd Rozendal [23]. The design of the V2 scanner (without led cover) is shown in figure 3.

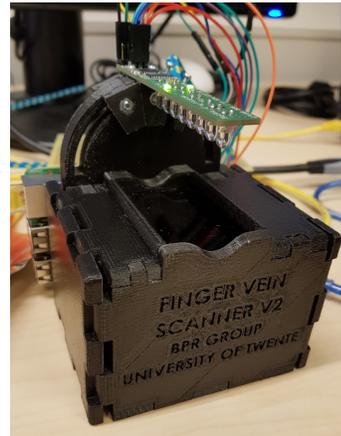


Fig. 3: Finger vein scanner V2

Here, a much smaller setup was designed with the ability to accommodate three cameras for future 3D reconstructions, although only one camera was implemented. The used camera is a wide angle 5MP RGB RB-WW camera from Joy-IT with S-mount lens and the ROI of the images is  $638 \times 340$ . IR filters are placed in the lenses. 8 IR LEDs (SFH4550) are controlled by an I2C interface. Images are of good quality but there is a slight overexposure near the contours of the finger. In 2018, [1] created a first 3D reconstruction with this scanner. Minor software issues and overexposure were partially fixed by Bram Peeters [18]. Also, Bram improved some software features in the same year and added a LED cover that makes the LEDs more directional. The accompanying software is written in C++ and uses Raspicam [17].

### III. THEORETICAL BACKGROUND

This section will introduce a basic knowledge of camera projections. Firstly the mapping of 3D world points to a 2D image plane is explained followed by a short explanation of stereo vision.

#### A. Single camera projections

A camera projects world points from 3D world coordinates  $(U_w, V_w, W_w)^T$  into 2D points (pixels) in a camera coordinate system  $(u_c, v_c)^T$ . This comprises three steps.

First, the camera and world coordinate systems must be aligned by a rotation and translation. The world/camera coordinate systems can be rotated by  $R$  and translated with  $C$  with respect to each other. Assuming a point  $P$  in world coordinates  $(P_w)$  and the similar point in camera coordinates  $(P_c)$ , the relation between these points is  $P_c = R(P_w - C)$ , resulting in a mapping from world points in the world coordinate system to world points in the camera coordinate system:  $(U_w, V_w, W_w)^T \mapsto (X_c, Y_c, Z_c)^T$  (the camera extrinsics).

Second, world points expressed in camera coordinates must be projected into camera coordinates;  $(X_c, Y_c, Z_c)^T \mapsto (x_c, y_c)^T$  (perspective projection). If one models a camera as a pinhole camera and uses triangular relations, one can obtain the projection of world points (represented in camera coordinates) into the image plane via  $(X_c, Y_c, Z_c)^T \mapsto (fX_c/Z_c, fY_c/Z_c)^T$  [4]

Third, the camera coordinates must be scaled and converted to pixel coordinates via the cameras intrinsic parameters:  $(x_c, y_c)^T \mapsto (u_c, v_c)^T$ . On a CMOS sensor or on an image, coordinates are positive integer values  $(u_c, v_c)^T$  and these are represented in pixels, which means scaling from millimeters to pixels.

The entire conversion in homogeneous coordinates is given in equation 1.

$$\begin{pmatrix} u'_c \\ v'_c \\ z'_c \end{pmatrix} = \begin{bmatrix} \frac{f}{S_x} & 0 & O_x & 0 \\ 0 & \frac{f}{S_y} & O_y & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_x \\ r_{21} & r_{22} & r_{23} & t_y \\ r_{31} & r_{32} & r_{33} & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{pmatrix} U_w \\ V_w \\ W_w \\ 1 \end{pmatrix} \quad (1)$$

$$\begin{aligned} u_c &= \frac{u'_c}{z'_c} \\ v_c &= \frac{v'_c}{z'_c} \end{aligned} \Leftrightarrow \begin{pmatrix} u'_c \\ v'_c \\ z'_c \end{pmatrix} \quad (2)$$

### B. Multiple view camera theory

1) *A simple stereo system:* Depth information is not represented in a 2D image. Points along the same light ray are projected onto the same image point. The X and Y coordinates of a point that is K times 'deeper' in space, scales accordingly with K, which means that a point along the same line is a scaling by a factor K:

$$x_c = f \frac{X_c}{Z_c} = f \frac{K X_c}{K Z_c}, \quad y_c = f \frac{Y_c}{Z_c} = f \frac{K Y_c}{K Z_c}$$

Fortunately, 3D points along the same lightray that are projected onto the same 2D image point in one camera (point  $x$ ) are projected onto a line (the epipolar line  $l'$ ) when projected into a camera that is slightly translated/rotated, as shown in figure 4. Each potential depth ( $X?$ ) of a point  $x$  in the left image, is projected onto to a point on the epipolar line in the right image. Using sum of absolute differences, these points can be analysed to find out whether this projected point corresponds to  $x$ . To reduce the search space of projected points, the position of the epipolar line must be known.

2) *A simple rectified stereo system:* In a rectified stereo system, the two epipolar lines are collinear (see figure 5). To obtain such a rectified stereo system or to align the epipolar lines, tools like as MATLAB'S (stereo) calibration, obtain a mapping (the fundamental matrix  $F$ ) from points in the left stereo image, to a line in a right stereo image [4]. This is done by finding a set of corresponding points on checkerboard patterns.

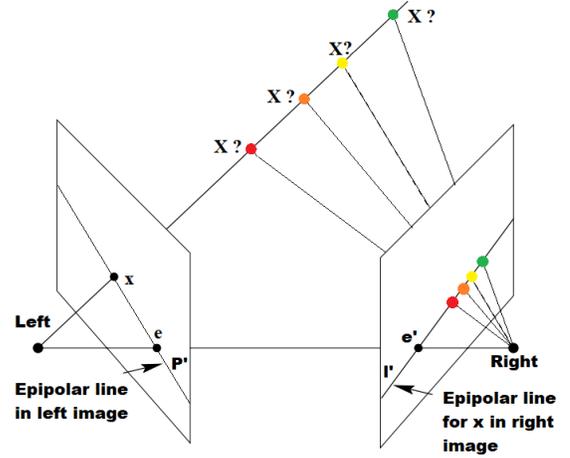


Fig. 4: Stereovision schematics. Adapted from [4]

In such a rectified stereo system, the disparity ( $d$ ) can be calculated as the horizontal distance between a point in the left image ( $x_l$ ) and the same point in the right image  $x_r$ . Or:  $d = x_l - x_r$ . This disparity is inversely proportional to depth, since far objects move very little between the left and right images, while near objects will move a lot. The exact mathematical relation between the disparity  $d$  and the depth  $Z_c$  can be determined via equal triangles and is given in equation 3. Here, the focal length  $f$  and the baseline  $b$  are constant scaling factors.

$$d = \frac{fB}{Z_c} \quad (3)$$

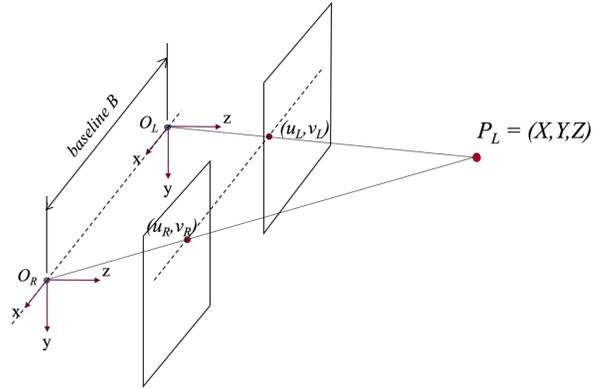


Fig. 5: Simple rectified stereo system [8]

## IV. 3D RECONSTRUCTION METHODS

In this study, we attempt to obtain 3D reconstructions using two different methods. A first method uses rectification and a second skips the rectification step and project points directly, based upon estimated depths [24]. In both cases, veins are used as interest points. Inputs are detected veins with corresponding finger images and the output of the reconstruction should be a 3D pointcloud with XYZ coordinates. For both reconstruction methods, the camera parameters must be known and cameras must therefore be calibrated.

### A. Number of cameras

Although two cameras are suitable for constructing one 3D reconstruction, increasing the number of cameras has some advantages. When three cameras are used, the total overlapping area between all views is larger than when only two cameras are used, resulting in a larger 3D reconstruction. Also, when two sets of two cameras are used, two 3D reconstructions can be created for both sides of the finger. These 3D reconstructions can then be combined into one 3D reconstruction using a weighted average. It is for those reasons that three cameras are preferably used instead of two camera's. Regarding naming conventions, the left camera is denoted by  $C1$ , the middle camera by  $C2$  and the right most camera is denoted as  $C3$ . The accompanying images are called  $I1$ ,  $I2$  and  $I3$ .

### B. Image preprocessing and vein detection

To reduce computation time, images are preferably downscaled. Depending on the image quality, CLAHE or histogram equalisation can be used for image enhancement in combination with ROI detection.

Veins shall be used as interest points for 3D reconstruction. The preferred method for obtaining these vein/interest points is the Miura Maximum curvature method [14]. The reason for this choice is based upon previous results in earlier phases of this research and on personal preferences of DMB faculty members of the EEMCS faculty. An implementation of this method is written by Bram Ton [26].

### C. 3D reconstruction by rectification and Matlab (method 1)

When stereo images are undistorted (adjusting for lens deformations) and rectified, the epipolar lines between the images are collinear and horizontal. This means that the interest points in the first image are in exactly the same row as in image 2. Based on the disparity between these points, the depths can be estimated using equation 3.

To calibrate, rectify, and undistort images, Matlab's stereo calibration tool from the Image Processing and Computer Vision toolbox can be used in combination with `rectifyStereoImages()`. Based on a disparity estimate from a stereo anaglyph with rectified and undistorted images, a valid disparity range  $[d_{min} : d_{max}]$  can be determined for two rectified stereo images. Veins can be detected in the rectified image of the middle camera ( $I2_r$ ) and the coordinates of these interest points are stored. For each of the  $n$  interest points, the reference point ( $R2_r^n$ ) in the corresponding rectified image  $I2_r$  is sought and a reference window ( $W2^n$ ) is created around this point.

Next, for each disparity  $d \in [d_{min} : d_{max}]$ , the corresponding point in the rectified image 1 ( $I1_d^n$ ) (or rectified image 3 ( $I3_d^n$ )), alongside with a window at this point  $W1_d^n$  are obtained. This window is compared with  $W2^n$  via sum of absolute differences (SAD):  $sum(abs(W2^n - W1_d^n))$ . The corresponding SAD score is then stored for each disparity

for this specific interest point. The optimal disparity ( $d_{opt}^n$ ) for this specific interest point is the disparity corresponding to the lowest SAD.  $d_{opt}^n \leftarrow \min([SAD_{d_{min}}^n : SAD_{d_{max}}^n])$ . This disparity can be stored in a matrix at location  $R2_r^n$ .

The above process is repeated for all  $n$  interest points and for all disparities. The result is a disparity map for the rectified image from  $C2$ ,  $I2_r$ . Finally, this disparity map can be converted into a depth map via `reconstructScene()` in combination with the camera intrinsics and extrinsics. Matlab can convert disparity maps into depth maps via `reconstructScene()`.

One disadvantage of this method is the rectification step. Because of this, only two camera's can be used simultaneously for one 3D reconstruction. This means that to use all three cameras, two sets of stereo pairs must be created. For both of these sets, the reference image must be  $I2$ . The result is two disparity/depth maps that both are within their own rectified image frame. If the scaling/rotations is not to severe, iterative closest point (ICP) can be used to combine these two clouds. This function is implemented in Matlab by `pregistericp()`.

### D. SAD as a function of depth (method 2)

Instead of using rectification, if all geometric information of a camera system is known, image points can, for estimated depths, directly be reprojected into space in world coordinates (backwards projection). These world coordinates can in turn be projected into an image plane that is translated/rotated with respect to the original image plane (forward projection). Comparing the projected points (with their associated depths) to the original image points, the depth information can be obtained (see figure 4). To make this mathematically possible, a slight modification is made to equation 1.

$$\begin{pmatrix} u'_c \\ v'_c \\ z_c \\ 1 \end{pmatrix} = \begin{bmatrix} \frac{f}{S_x} & 0 & O_x & 0 \\ 0 & \frac{f}{S_y} & O_y & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_x \\ r_{21} & r_{22} & r_{23} & t_y \\ r_{31} & r_{32} & r_{33} & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{pmatrix} U_w \\ V_w \\ W_w \\ 1 \end{pmatrix} \quad (1 \text{ revisited})$$

Now both the intrinsic and the extrinsic parameters are a  $4 \times 4$  matrix and their product is a  $4 \times 4$  invertable matrix  $H$ .

$$\begin{pmatrix} u'_c \\ v'_c \\ z_c \\ 1 \end{pmatrix} = H \begin{pmatrix} U_w \\ V_w \\ W_w \\ 1 \end{pmatrix} \quad \text{and} \quad \begin{pmatrix} U_w \\ V_w \\ W_w \\ 1 \end{pmatrix} = H^{-1} \begin{pmatrix} u'_c \\ v'_c \\ z_c \\ 1 \end{pmatrix}$$

As a result, it is possible to calculate how points in e.g. reference image  $I2$  are projected in  $I1/I3$  for a variety of depths. For each vein point  $R2^n$  in  $I2$ , projections for different depths  $z_c \in [z_{c_{min}} : z_{c_{max}}]$ , world coordinates can be obtained via a backwards projection ( $H_2^{-1}$ ). Similarly, to obtain the corresponding projected points  $P1_{z_c}^n/P3_{z_c}^n$ , the forward projection ( $H_1, H_3$ ) can be used. For each depth and for each of the projected points  $P1_{z_c}^n/P3_{z_c}^n$ , a reference window in  $I2$  at  $R2^n$  can be compared to a window in  $I1/I3$  at  $P1_{z_c}^n/P3_{z_c}^n$  via SAD. Scores for both  $I1$  and  $I3$  are stored

for each depth. To obtain a final depth for the original image point in  $I2$ , the depth that corresponds to the lowest SAD for both  $I1$ ,  $z1_{c_{opt}}^n$  and  $I3$ ,  $z3_{c_{opt}}^n$  shall be used.

$$\begin{aligned} z1_{c_{opt}}^n &\leftarrow \min([SAD_{z1_{min}}^n : SAD_{z1_{max}}^n]) \\ z3_{c_{opt}}^n &\leftarrow \min([SAD_{z3_{min}}^n : SAD_{z3_{max}}^n]) \end{aligned}$$

This means that for each  $R2^n$  in  $I2$  two depth estimation are obtained; one via  $I1$  and one via  $I3$ . To obtain final 3D projections, the forward projection must be solved with the original image points  $R2^n$  in  $I2$  whilst scaled by  $z1_{c_{opt}}^n$  or  $z3_{c_{opt}}^n$ .

An advantage of this method is that only image undistortion via `UndistortImage()` is required and that no image rectification is needed. Furthermore, as each point in  $I2$  has two associated depths, both obtained by correlation in 2 different cameras, a weighted average of the resulting depthmaps can be easily implemented.

#### E. Potential SAD complications in 3D reconstruction

There may be complications when using SAD on circular fingers and these can potentially affect the final reconstruction results. If a reference window is placed directly in front of camera 2, the visible object will lie flat in this projection. However, if the same object is seen by a rotated camera, e.g.  $C1$ , it will be projected at an angle, resulting in a transformed/compressed object. Even if both cameras were to capture the exact same object, their projections will not be exactly the same due to transformation/compression. As a result, points may not be assigned a correct depth due to an incorrect SAD. At the same time, depth estimations errors are more severe towards the edges of fingers. A mistake in the correct pixel location (the result of SAD correlation) for a point directly in front of the middle camera results in a small error. However, when the point moves more towards the sides of the finger, a small mistake for the pixel location can result in a larger depth error as the tangents of the finger are much steeper.

## V. RECOGNITION METHODS

To investigate the effects of finger rotations and the potential advantages of 3D vein recognition over 2D recognition, both for 2D and for 3D, performance characteristics should be obtained. This requires recognition experiments and error measurements that show how well 2 patterns are similar. For 3D recognition this error measure can be implemented by using the Root Mean Square Error (RMSE) of the Iterative Closest Point (ICP) algorithm. For 2D matching this can be done by 2D correlation as proposed in [13] and implemented by [26]. Based upon the RMSE and the correlation scores, the false accept rate (FAR,  $\frac{FA}{FA+TR}$ ) and the false rejection rate (FRR,  $\frac{FR}{TA+FR}$ ) can be determined. An EER can be calculated and based on the corresponding threshold the accuracy ( $\frac{TA+TR}{TA+FR+FA+TR}$ ), sensitivity ( $\frac{TA}{TA+FR}$ ) and the specificity ( $\frac{TR}{TR+FA}$ ) can be calculated.

#### A. Artificial fingers

To qualitatively analyse the effect of finger rotations on recognition performance for both 2D and 3D, it is useful to know the exact rotation of the rotated fingers. Since it is not possible to accurately rotate real fingers on the proposed scanner, it is preferred to work with artificial fingers that have a representative pattern of human veins. Such fingers can be made by gluing a vein pattern onto a transparent PVC tube. 2 non rotated examples are given in figure 6. For each of these artificial fingers, a non-rotated base image should be captured that resembles a correct entry in a database. To ensure realistic acquisition settings, these images should not be captured at the same time as the test fingers.

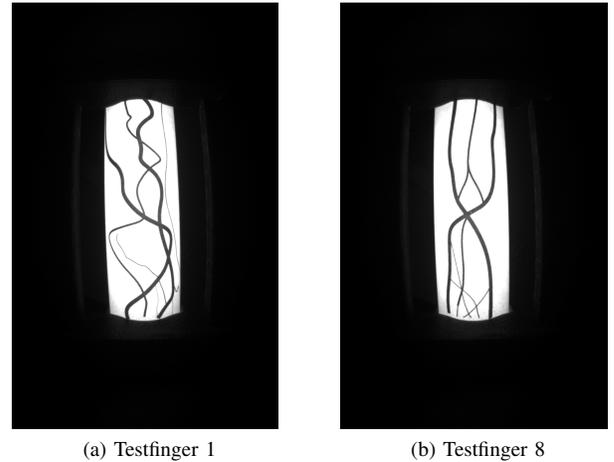


Fig. 6: 2 artificial testfingers

All rotated fingers can then be compared to the original unrotated vein images (the base images). The result is a matrix of size  $XY \times X$  where  $X$  is the number of fingers and  $Y$  is the number of available rotations. Each entry provides a RMSE score or correlation value for how well this (rotated) vein image would align with any of the non rotated base images.

Using these scores, the FAR and FRR rates can be determined and accordingly the EER can be obtained. Based on the threshold value corresponding to the EER, the accuracy, specificity and precision can be determined. At the same time, this threshold can be used to determine the incorrectly aligned fingers, including their associated rotations. In addition, for each rotated finger, the base finger corresponding to the minimum score is obtained. This results in a matrix containing the identified base fingers for each of the (rotated) fingers used.

#### B. Real finger images

To validate the whole system in a real situation, from vein detection to 3D recognition, real vein images should be used. By taking multiple images from a few fingers, while repositioning the fingers on the scanner for each image, slight rotations and translations that represent actual rotations will be present. Performing recognition experiments with these images with a series of 'non rotated base fingers' (resembling

earlier acquired database entries) will determine more realistic system characteristics. In this case the non rotated (data)base images are images from fingers that are attempted not to be rotated while captured at a different moment in time. In these situations, however, the exact rotation cannot be determined.

## VI. DEVELOPING A 3D FINGER VEIN SCANNER

To develop a suitable finger vein scanner, firstly, a suitable camera must be chosen. Based upon the dimensions of this camera, a final physical setup can be developed.

### A. Requirements of a usable scanner

To develop a 3D vein reconstruction, the used images should be of high quality and rich in (vein) detail. Images must be consistent, i.e., images should be similar regardless of the environment in which the images were acquired. Furthermore, it is preferred that no image enhancements are required so as not to alter any vein pixels. In addition, a system should be developed that allows to easily capture images and to directly see the effect of different camera settings. In other words, a live preview is highly desirable. Regarding the physical design, it is of high priority that the camera housing is stable; the cameras should not move after calibration. Also, the housing should be closed to avoid unwanted light scattering inside the housing.

### B. Disadvantages of current available setups

Two setups are available for image acquisition, both versions V1 and V2 are available. Since the V1 scanner contains only one camera, it is not suited for 3D reconstructions. And although the V2 scanner can accommodate three cameras, this scanner has serious drawbacks.

First, the design quality of the physical construction of the sensor is impractical. The housing is unstable and the cameras are wobbly, the cable management causes interferes with the camera signals and the housing of the sensor needs improvement. In addition, the images are inconsistent and the contrast varies significantly between similar captures. The accompanying C++ implementation of Raspicam [17] contains bugs and is impractical to use due to build times and lack of a live preview. In addition, RGB cameras with Bayer filters are used, and although the gain and exposure are theoretically accessible, the accompanying Raspicam software is unable to implement these settings correctly. This makes it difficult to obtain detailed images and makes them inconsistent. The result is a lower level of detail that is unsuitable for 3D recognition experiments (see figure 7).

### C. Approach for finding a suitable camera

In order to find a suitable camera for image acquisition, a selection of cameras was tested and compared with the images from the V1 scanner (BCi5) (see figure 33. For each camera, the best possible images were sought by changing the adjustable parameters in the software. To validate the image quality, all RGB images are converted to grayscale images and they are visually inspected for consistency and a high level of

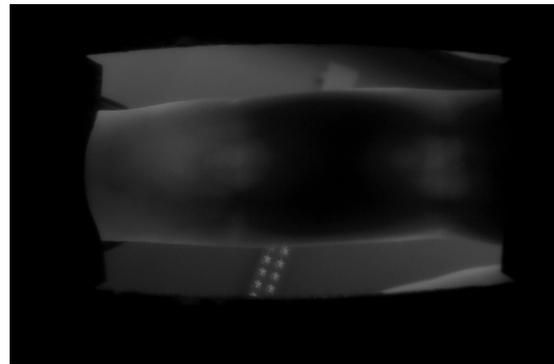


Fig. 7: Unacceptable image capture from the V2 scanner

detail. Most tested cameras are general Pi cameras with an OV5674 chip that is commonly used by hobby enthusiasts. A special camera without infrared filter (NOIR) and a high resolution camera were included in the selection to increase the variety of CMOS sensors. A special monochrome camera was included in the selection to test the effects of increased pixel sizes and not having a Bayer color filter. A summary of all cameras tested and the findings mentioned below can be found in the appendix in table VI.

### D. Camera evaluation results

In general, RGB cameras with an OV5674 sensor, have been found to be sensitive to changes in external light sources (open/closed blinds) and the vein quality of these images is not suitable for vein detection. The filters in the lenses of these OV cameras are not seriously affecting the vein quality. Similarly, the effect of thick filter material in the V1 setup has been found to be negligible, and filters in general Pi cameras are not specified (NS). Furthermore, cameras that contain a pinhole are impractical to use and they are difficult to focus and cannot properly capture the entire finger due to their small aperture. Due to the large field of view of the WW2 lens, these images are also not suitable for vein acquisition. In addition, the quality of the veins of both pinhole cameras is unacceptable, as no veins are even visible at all on the Pi NOIR V2 (see figure 32).

A notable improvement in contrast is obtained when using the high resolution camera (see figures 30, 31). Not only is the pixel size slightly increased as a results of a different CMOS sensor (which may result in an improved spectral response), also the lenses allow for variable aperture. However, despite the improved image quality, a drawback of these cameras is the size of the cameras' big 12mm/16mm lenses in combination with the larger sensor. This would not lend itself to a non bulky and lightweight sensor.

A significant improvement in visible image quality is achieved by using the monochrome camera (see figure 8). Many smaller veins are visible that are not as clearly visible on the images from the V1/V2 scanner and the contrast is much better. Also, veins are visible without any image enhancement techniques. Since the sensor is not by default supported by Raspberry, the

manufacturer provides a software template to manually adjust the gain and exposure while displaying a live preview. Images taken with this sensor are consistent and are not seriously affected by external light sources. One minor drawback is the size of the PCB board, which is a factor of two larger than the originally PCB's used for the cameras in the V2 scanner.

Compared to all other cameras tested, the detailed images from the monochrome camera are significantly better than any other images. At the same time, the image quality and contrast are consistent and the images are not affected by external light sources. Another advantage is that the software template allows to adjust the gain/exposure properly while a live preview shows the direct effects. In addition to that, no image enhancement is required to make veins visible. Considering that this camera seems to meet all earlier stated requirements, the OV9281 camera is the preferred camera for finger vein detection in this research.

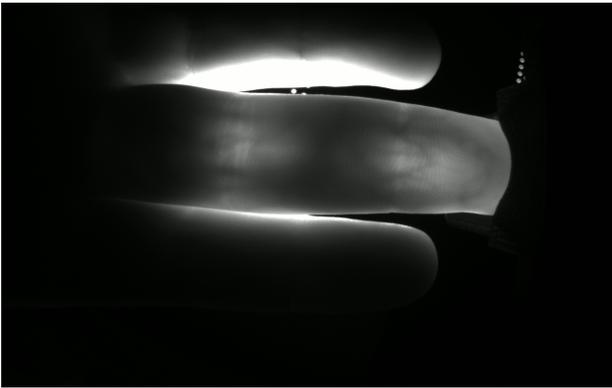


Fig. 8: Raw image captured with the OV9281

### E. Designing a physical build

To design a scanner, the design of the V2 scanner (see [23]) is taken as a starting point. Modifications needed to make this sensor suitable for 3D reconstruction include a more solid and robust design and space for the larger PCB's for the monochrome cameras. This housing will be created by using a 3D printer, and each camera shall be controlled via a dedicated Raspberry Pi 4 with its own power supply, similar to the previous design.

The final design contains two sloping edges of  $21.691^\circ$ , to which the cameras are bolted. The angle is chosen so that the optical axes of the cameras are intersecting at best in the centre location of a finger whilst keeping a reasonable similar angle as used in [1]. The centres of the leftmost and rightmost cameras are horizontally translated by  $\pm 27.33\text{mm}$  and vertically by  $4.81\text{mm}$  with respect to the centre of the middle camera. The PCBs for each camera are stacked below the middle camera, to avoid any interference from cables. The shortest distance between the middle camera and the lowest point of the fingers is  $65\text{mm}$ . A picture of the final setup is shown in figure 9.

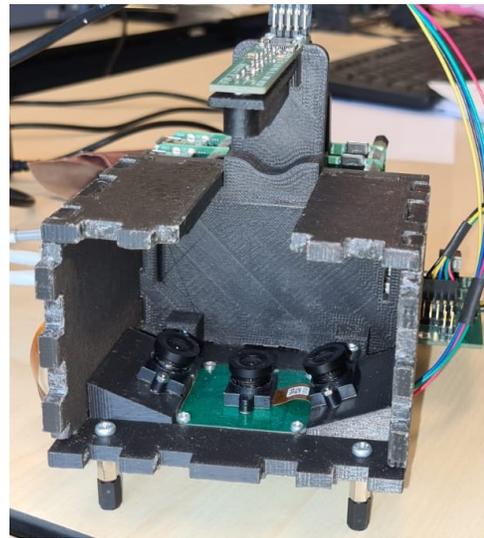


Fig. 9: Final 3D printed setup (without front cover, with LED cover)

### F. Accompanying software GUI

To make the setup easy and practical to use, a graphical user interface (GUI) was developed. This GUI was designed such that it allows to adjust essential parameters whilst a live preview shows the effects of these parameters (see figure 10). These parameters include the gain and exposure of the camera, as well as light intensity control of all individual LEDs. Naturally the GUI includes an option to immediately active a PNG or JPEG encoder to save the current preview/snapshot as an image. When using the GUI for multiple cameras, the user can specify a 'master' that, when taking a snapshot, will control the other Raspberry Pi's (the 'slaves') via the GPIOs, to capture multiple images from different cameras simultaneously.

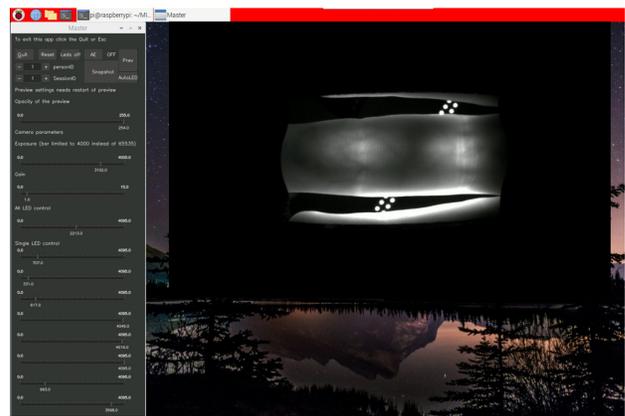


Fig. 10: Finger vein scanner GUI

## VII. 3D RECONSTRUCTION - IMPLEMENTATION AND RESULTS

### A. Camera calibration and lens focusing

To properly calibrate cameras, it is preferred to use a checkerboard that covers as much as possible of the camera.

To ensure an uniformly illuminated checkerboard, a 5mm thick piece of Delrin is placed directly under the LED's to diffuse the light. Based on previous experiments, a checkerboard pattern of 20\*15 with 2mm squares is used and printed on a piece of paper that is glued to a transparent piece of acrylic. This checkerboard pattern is then placed on top of the sensor with the checkerboard facing down. The lenses of the cameras are focused so that the visible details in a linearly scaled calibration pattern are sharp. A total of 30 checkerboard images are captured. To match the images used for reconstruction, the calibration images are rotated 90 degrees and bicubically scaled down to 35% from 800\*1280 to 280\*448 pixels.

The MATLAB tool for stereo calibration with two parameters for radial distortion and no skew or tangential distortion is used to calibrate the cameras. Two sets of cameras are stereo calibrated. Camera 2 + camera 1 ( $C21 = \textit{centre} + \textit{left}$ ) and camera 2 + camera 3 ( $C23 = \textit{centre} + \textit{right}$ ). In both cases, the centre camera  $C2$  is the 'left' camera of the stereo pair whereas  $C1/C3$  is the 'right' camera. For each stereo pair two images are discarded, which means that 28 checkerboard images are used for calibration. The calibration results indicate a mean reprojection error of 0.0930 and 0.0969 for  $C21$  and  $C23$  respectively. Considering the 3D printing accuracy in combination with the calibration results, (see table I for extrinsic parameters), the calibration results seems acceptable. Full details of the calibration results can be found in the appendix in section F.

		X	Y	Z
C1	Rotation ( $^{\circ}$ )	-0.6544	21.6152	-0.6322
	Translation (mm)	23.4267	0.0915	5.3966
C3	Rotation ( $^{\circ}$ )	-0.3958	-21.8958	-0.5389
	Translation (mm)	-22.9393	-0.0862	5.0204

TABLE I: Extrinsic results from camera calibration

Given the camera parameters, an estimate for the 'stepsize' for depths as a function of disparity can be obtained (see equation 3). These estimates are plotted in figure 11.

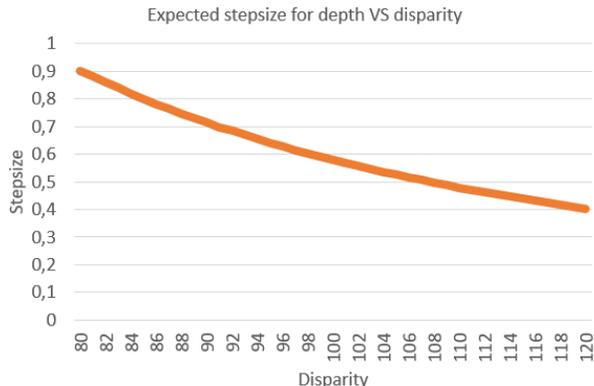


Fig. 11: Depth stepsize estimation for  $B = 23.4$  and  $f = 250$

### B. 3D validation model

To validate the 3D reconstruction methods, a mock up finger is created from a PVC tube around which a printed vein pattern is rolled up. The PVC tube is 20mm in diameter and contains a vein pattern with horizontal, vertical and diagonal lines at a 5mm grid. 2 models are created and shown in figure 12.



Fig. 12: Testfingers with veins from  $C2$  (90 degrees rotated)

### C. Image preprocessing and vein detection

Images are firstly rotated 90 degrees clockwise. Otherwise, it was found that images are incorrectly rotated during rectification. Next, the images are scaled down from 800\*1280px to 280\*448px (bicubic scaling to 35%) to reduce the computation time and images are rectified and undistorted. Next, vein detection is performed by using the maximum curvature implementation by Bram Ton [26] with a  $\sigma = 2$  and a threshold of 0.005. Based on a (manually) selected region of interest (ROI), veins are detected in a specific area of the image and vein detection takes approximately 0.8 seconds. No contrast enhancement or other image preprocessing is performed.

### D. 3D reconstruction by rectification and Matlab (method 1)

1) *Acquiring pointclouds*: Two sets of two images are created.  $C21$  contains images from the middle and left cameras,  $C23$  from the middle and the right cameras. Both sets of images are rectified according to their stereo parameters from calibration. The rectification and lens corrections are performed by `rectifyStereoImages()`. The two resulting images for  $C21$  are 636\*497 pixels whereas for  $C23$  they are 658\*510 pixels. As a result of the rectification process, the rectified images of  $C21$  are horizontally flipped (see appendix figures 37 and 38). Based upon a stereoanlygraph, an estimate for the disparity is obtained, ranging from  $d \in [-120 : -80]$  for  $C21$  and  $d \in [80 : 120]$  for  $C23$ . Next, veins are detected for both pairs in the rectified image from camera 2. The ROI for  $C21$  has a width of 78pixels and a height of 245pixels and for  $C23$  a width of 73pixels and a height of 229pixels are used. For the testpattern 1, 4094 vein points are detected in the rectified image for  $C21$  and 3450 vein points for  $C23$ .

A custom function is written that computes the SAD values for each interest point for each disparity for a given window size (see Appendix section E). For each interest point, the disparity corresponding to the lowest SAD is determined. Additionally, this function checks if this SAD is below a certain percentage of the average SAD for all disparities for each point, to filter out false disparities. The result of this function is a disparity map; each interest point in the rectified image is assigned a disparity. Based upon such a disparity map, a 3D pointcloud is created via `reconstructscene()`.

Disparity maps with a window size of 15 and a SAD threshold of 70% are created for the two images sets *C21* and *C23*. These disparity maps are shown in the appendix in figures 39 and 40. The corresponding 3D pointclouds for *C21* and *C23* are shown in figure 13. As a result of rectification, both 3D pointclouds are in their own 'reference frames', resulting in different positions and sizes for both pointclouds.

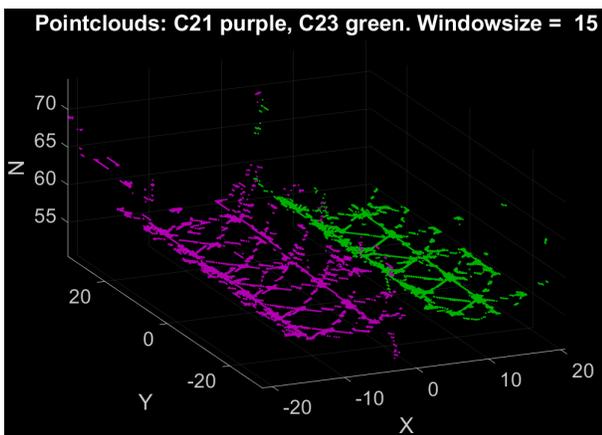


Fig. 13: Pointclouds for rectification method

Looking at a front view of these pointclouds (e.g. via figure 15), for both *C21* and *C23*, the lowest point of the finger is at a height of 55mm. At this depth, the stepsize is approximately 0.51mm near the centre of the finger and the absolute disparity is 106 and 109 for *C21* and *C23* respectively. Moving further away from the centre of the finger to the sides, the stepsize increases to  $\approx 0.69$ mm and disparities are decreasing to  $\approx 95$ . These results seem to be accordance with equation 3 and the plot in figure 11.

2) *Aligning and merging point clouds*: In order to align both pointclouds, ICP is used, implemented via `pcregistericp()`. To avoid a local minima in the ICP algorithm, an initial translation vector is provided to the ICP algorithm. Both pointclouds are translated in the X direction with 22.5mm and 5mm for *C21* and *C23* respectively. Next, ICP is implemented with an inlier of 0.9. Although this number could be decreased for lower RMSE values, a value of 0.9 is chosen as experiments have shown that this implies a maximum number of points used whilst keeping most serious outliers out. Decreasing the number of iterations above 20 has no significant effect on the RMSE and the final RMSE

value to align both pointclouds is 0.2883. The pointclouds after ICP are given in figure 14. As can be observed from the pointclouds, the pointcloud corresponding to *C23* is rotated significantly around the Y axis and this rotation equals  $17.48^\circ$ .

In order to validate the curvature of the 3D pointclouds, a circle with a radius of 20mm is drawn in the pointclouds as shown in figure 15. By visual inspection it is observed that the lowest point of the finger is approximately at a height of 55mm. Furthermore, the accuracy decreases when veins are further away from the centre of the pointcloud. Based upon these results, it is estimated that the region for which the 3D reconstruction is accurate is around the centre at  $x = 13$  from  $x \in (17 : 8)$ .

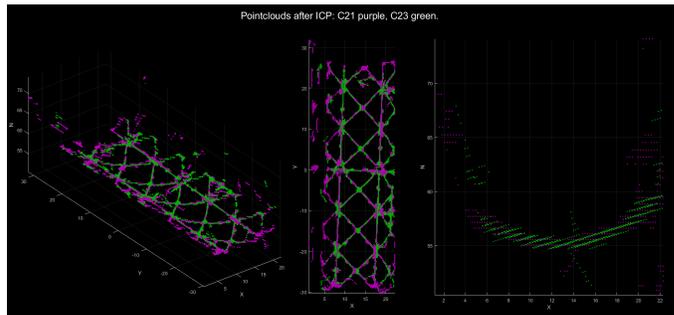


Fig. 14: Pointclouds for *C21* and *C23* after ICP with an inlier ratio of 0.9 and 20 iterations

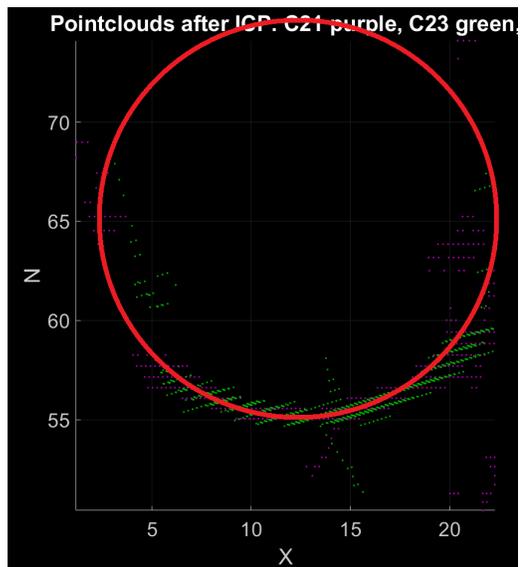


Fig. 15: Curvature evaluation of 3D pointclouds for method 1

#### E. SAD as a function of depth (method 2)

In order to obtain depth information for points via equation 1, the  $H$  matrix for each camera must be obtained. These matrices are obtained by multiplying the camera intrinsics and extrinsics, and to this result a row of  $(0 \ 0 \ 0 \ 1)$  is added.

The intrinsic and extrinsic parameters are taken from the stereo calibration, of which the results are shown in the appendix in section F. For camera 1 and 2, the stereo-calibration from *C21*

is used. For  $C3$  the stereo-calibration from  $C23$  is used. In addition to that, world-coordinates are set equal to the camera coordinates for the centre camera by setting the rotation and translation to 0. Next, scaled (35% bicubic) images are undistorted via `undistortimage()`. Veins/interest points are detected with the same Maximum Curvature implementation used for method 1 and veins are detected in the undistorted image from  $I2$ . The region of interest used for vein detection has a width of 90 pixels and a height of 245 pixels. For testpattern 1 this results in 5347 vein points.

For a range of depths  $d \in [45 - 65]$  obtained via physical measurements, all interest points in  $I2$  are converted to homogeneous coordinates via equation 2 and projected into world coordinates via  $H_2^{-1}$ . The stepsize for depths is set at 0.2mm, as a smaller stepsize tend to ask to much GPU processing power to render pointclouds. Via  $H_1$  and  $H_3$  the world points are projected into the new image planes for both  $I1$  and  $I3$ . These new projected homogeneous coordinates are converted back into pixel coordinates  $u$  and  $v$  and these are stored. This results in two vectors containing all projected points for different depths in  $I1$  and  $I3$ .

For each interest point in  $I2$ , the best correlating points in both  $I1$  and  $I3$  is searched via SAD (with a window size of 15). No filtering implemented regarding SAD scores. The result of this method are two depthmaps for  $I2$ ; one via  $C1$  and one via  $C3$ . For each image point and optimal associated depth, the correct world points can now be calculated by converting each point in the depthmap back into homogeneous coordinates via 2 and again into world coordinates via  $H_2^{-1}$ . Both depthmaps are shown in the appendix in figures 41 and 42 and the resulting pointcloud is given in figure 16. Additionally, a circle with radius of 20mm can be drawn inside the pointclouds as shown in figure 17.

As a result that both pointclouds are in the reference frame of  $I2$ , both pointclouds can be merged together relatively easy. Since camera 1 is better in capturing points for the 'left' side (as these points are closer to the camera and these points are not obstructed by the curvature of the finger) and since camera 3 is better in capturing points at the 'right' side, a weighted average can be created that takes this into account. A function is implemented that combines both depthmaps via  $xz^1 + (1-x)z^3$  where  $x$  is related to the image column and  $z^{1 \vee 3}$  is the depthmap for  $I1$  or  $I3$ . The resulting weighted depthmap is given in the appendix in figure 43. The corresponding pointcloud is given in figure 18.

Similarly as before, a circle with radius of 20mm is drawn inside this pointcloud in figure 19. For both the weighted and the regular pointclouds, all depths vary in steps of 0.2mm over the entire width of the finger. In the weighted pointcloud, a minimum height of  $\approx 56.6$ mm is obtained. Contrary to to method 1, the accuracy of the weighted pointcloud stays approximately similar when moving away further from the centre of the finger. A region around  $x = 0$  of  $x \in (-8 : 8)$  is estimated to correctly represent the 3D pattern. Furthermore,

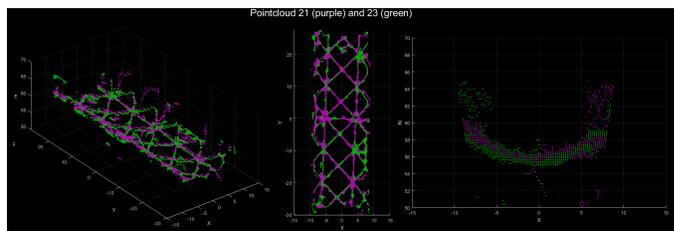


Fig. 16: Pointclouds for SAD as a function of depth

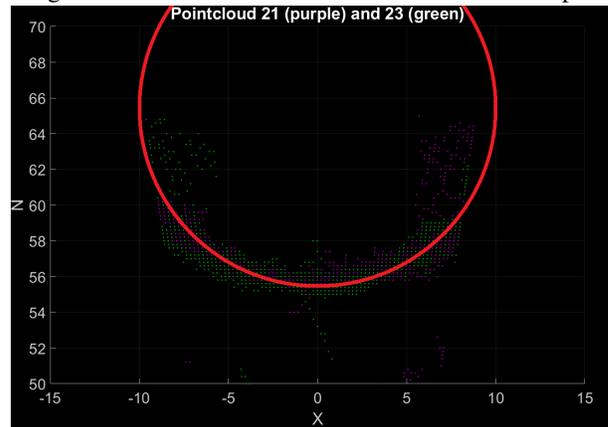


Fig. 17: Pointclouds for SAD as a function of depth

comparing this result with the non weighted version, clearly the weighting factor has a positive effect as seemingly faulty points near the edges are removed and throughout the finger the range of depths is decreased significantly.

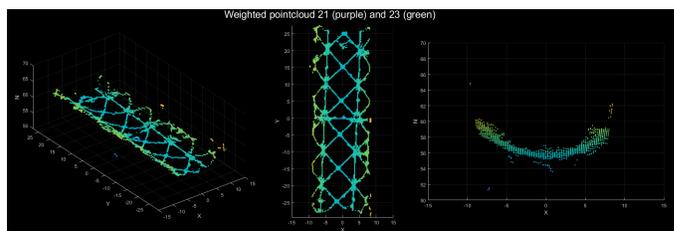


Fig. 18: Weighted pointclouds for SAD as a function of depth

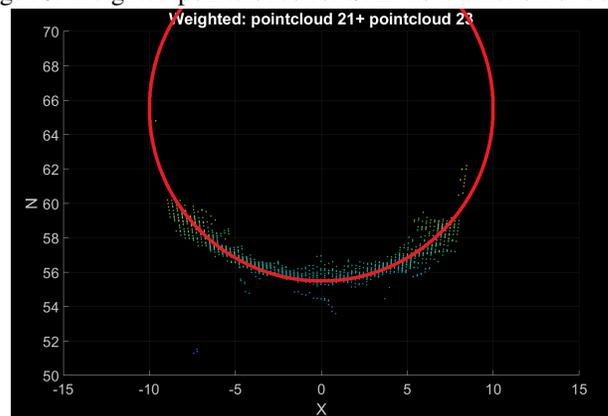


Fig. 19: Weighted pointclouds for SAD as a function of depth

## VIII. RECOGNITION - IMPLEMENTATION AND RESULTS

### A. Implementation

In total 8 artificial fingers were created. For each of these fingers, 21 rotation are applied ranging from  $[-50, -45, \dots, -5, 0, 5, \dots, 45, 50]$  degrees with a stepsize of 5 degrees. This results in  $8 * 21$  images with different rotations for each finger (see table II for naming convention). In total 168 artificial images/pointclouds are created for testing and another 8 non rotated base images/pointclouds are acquired to be references. Besides the artificial fingers, 18 real fingers from 3 different persons are captured (index-, middle- and ring-finger for both left and right hands). For each finger 5 images are created resulting in 90 real images/pointclouds. For each capture, the entire finger is removed and repositioned on the scanner, an exact rotation is thus not known. Additionally 18 base images of the same fingers are obtained, acquired at a different time to more realistically model a recognition system. In all cases, veins are detected using the Maximum Curvature method [14]. All images are captured from 3 cameras as discussed in section VI-E. For 2D recognition images from the middle camera are used. For 3D reconstruction all 3 cameras are used.

For 2D correlation, the correlation method implemented by [26] is used with a search displacement of 100px. The correlation score ( $1 - 2 * correlation$ ) is displayed to evaluate results next to the 3D error scores. Each 2D image is compared via 2D correlation to all corresponding base images, resulting in a  $168 * 8$  correlation matrix for artificial fingers and a matrix with dimensions  $90 * 18$  for real fingers. For each artificial finger, there are  $21 * 8 = 168$  entries for each rotated vein pattern with any of the 8 basefingers.

For 3D recognition, SAD as a function of depth is chosen in combination with ICP with an inlier ratio of 0.4 and 20 iterations. SAD as a function of depth is chosen as results of this method seemed more transparent and depth reconstructions were seemingly more accurate. To avoid ICP falling into a local minimum, each pointcloud is firstly translated by 62mm in the Z direction such that its centre of depth lies at approximately a depth of zero. Next, a series of initial rotations,  $\in [-70, 70]$  with a stepsize of 10 degrees, is applied to each pointcloud before ICP alignment. For each entry in the resulting matrix ( $168 * 8 * 15$  for artificial fingers), the best score amongst all 15 initial rotations is taken, to result in matrix of  $168 * 8$  with minimal RMSE values that correspond to the most optimal initial rotation.

For both 2D and 3D and for both real and artificial fingers, the TA (true accepted), FR (false rejected), FA (false accepted) and TR (true rejected) scores are obtained for a range of thresholds  $\in [0, 1]$  with a stepsize of 0.001. For artificial fingers, for each set of 21 images corresponding to one basefinger, it should correctly assign each of these images to its corresponding base fingers and reject the other  $168 - 21 = 147$  scores for the other 7 base fingers. Furthermore, the FAR and the FRR are obtained. The EER is calculated by taking the average of

FAR and FRR when the difference between these is minimum. The threshold corresponding to this EER is used to calculate the accuracy, the sensitivity and the specificity. Additionally, for each finger, the basefinger corresponding to the minimum score can be obtained. Dividing this by the total number of images provides an identification score.

### B. Results artificial fingers

The overall recognition results for artificial fingers are given in table III. The FAR and FRR (including accuracy, sensitivity and specificity) curves are given in figures 22 and 23. A plot of the 2D correlation scores and the 3D RSME values for all artificial fingers with all basefingers is given in figures 20 and 21 (see appendix figures 44 and 45 for full size graphs). Here vertical lines identify the start of a new set of vein images corresponding to one base finger and the horizontal line corresponds to the EER for the overall system. It can clearly be observed that for smaller rotations the RMSE and correlation values are smaller; the greater the rotation, the larger the error. For 3D, RSME scores of 0 are achieved implying that atleast 40% of the points used for calculating the RMSE are perfectly aligned. The corresponding images, despite taken at different time intervals, indeed have strong visual correspondence.

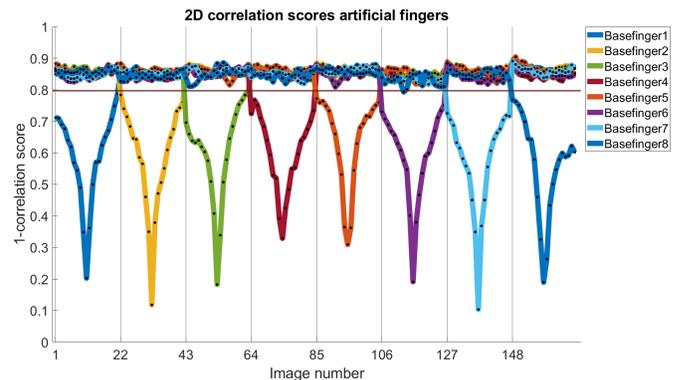


Fig. 20: Correlation scores for 2D recognition with artificial fingers (see table II)

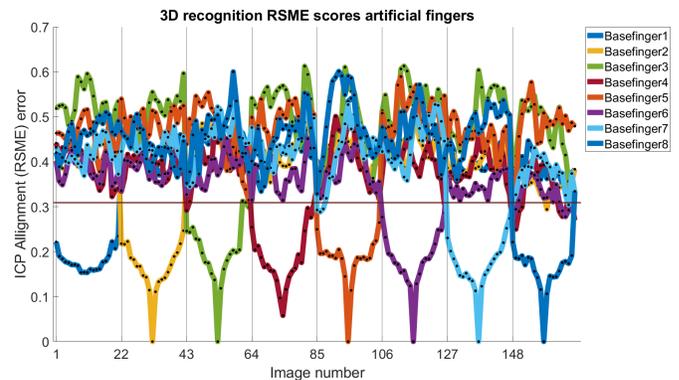


Fig. 21: ICP RMSE scores for 3D recognition with artificial fingers (see table II)

For 2D, at a threshold of 0.797, an EER of 0.000425 and an accuracy of 99.93% are achieved. 1 image is falsely accepted

Finger	1							2							...	8						
Rotation	-50	-45	...	0	...	45	50	-50	-45	...	0	...	45	50	...	-50	-45	...	0	...	45	50
Image number	1	2	...	11	...	20	21	22	23	...	32	...	41	42	...	148	149	...	158	...	167	168

TABLE II: Image numbers corresponding to rotations of artificial fingers

as finger 6. This image has a correlation score of 0.7927 and a rotation of -15 degrees. All 168 fingers are correctly identified. For 3D a threshold of 0.309 is determined, resulting in an EER of 0.018700 and an accuracy of 98.07%. For 3D, 3 images are falsely rejected with rotations and RMSE errors of 40(0.313), 45(0.3093) and 50(0.334) degrees. 23 images are falsely accepted with rotations [-50(5), -45(4), -40(2), -35(1), 5(1), 25(1), 30(2), 35(2), 40(2), 45(1), 50(2)] and scores range from [0.250 : 0.307] (see appendix table XI). 16 of these FA's correspond to finger 8. This finger has limited details compared to other artificial fingers (see figure 12) and in combination with a small inlier ratio for the ICP algorithm, this might result in a small error when aligned with more detailed vein clouds. Regarding identification, all fingers are correctly identified. Furthermore, it is worth to mention that the slope of the FAR in 2D is steep; implying that a slight increase of threshold has significant effects on the accuracy and specificity. Additionally, the RMSE scores for incorrect fingers have a much wider range of values whereas for 2D these values are much more in the same range. More elaborate performance results for all individual fingers are given in the appendix in tables VII and VIII and the identification results are given in the appendix in table IX and X.

Artificial fingers	2D	3D
Threshold (Th)	0.797	0.309
EER	0.000425	0.018700
Accuracy(%)	99.93	98.07
Sensitivity(%)	100	98.21
Specificity(%)	99.91	98.04
TA	168	165
FR	0	3
FA	1	23
TR	1175	1153
Identification(%)	100	99.40
FRR @0%FAR (FA=0)	0.005952 (FR=1, Th=0.792)	0.089286 (FR=15, Th=0.249)
FAR @0%FRR (TA=168)	0.000850 (FA=1, Th=0.797)	0.062075 (FA=73, Th=0.335)

TABLE III: 2D and 3D overall recognition results for artificial fingers

### C. Results real fingers

Results for real fingers images are given in table IV and the FAR and FRR curves are given in figures 24 and 25. The plotted correlation and RMSE scores are given in the appendix in figures 46, 47 and an example of the real fingers used plus its 3D reconstructed cloud are similarly given in the appendix in figures 48 and 49. Results of recognition for individual fingers and identification results are given in the appendix in tables XII, XIII, XIV and XV.

For 2D, a threshold of 0.785 and an accuracy of 95.49% are obtained at an EER of 0.0448. 4 fingers are falsely rejected

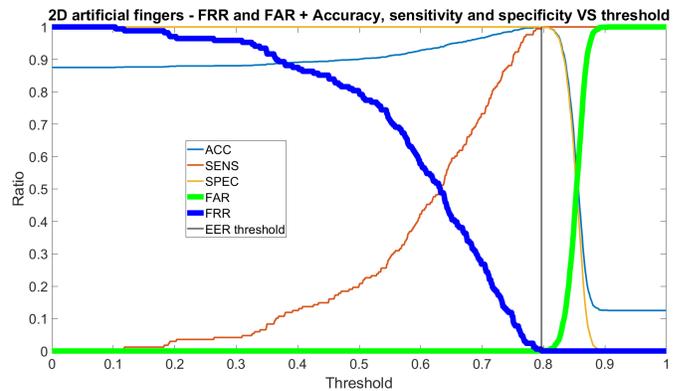


Fig. 22: 2D FAR and FRR curve for artificial fingers

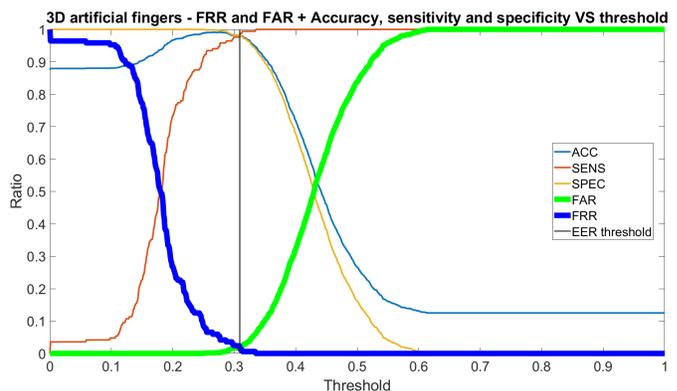


Fig. 23: 3D FAR and FRR curve for artificial fingers

and 69 images are falsely accepted. 5 images are not correctly identified, resulting in an identification score of 94.44%. For 3D, a threshold of 0.509, an EER of 0.0886 and an accuracy of 91.17% are obtained. 8 fingers are falsely rejected and 135 fingers are incorrectly accepted. 11 images are not correctly identified, resulting in an identification score of 87.78%

For both methods, the number of falsely accepted images is large relative to the total correct images, resulting in a lower specificity and accuracy. Since the slope of the FAR in 2D is steep, a slight decrease of the threshold increases the specificity and the accuracy relatively more than it would for 3D. Results for decreasing the threshold such that a maximum accuracy is achieved are shown in table V. For 2D, a maximum accuracy of 99.2% can be achieved for a sensitivity and specificity of 86.7% and 99.9% respectively. For 3D, a maximum accuracy of 98.0% is achieved for a sensitivity and specificity of 67.8% and 99.9%.

Real fingers	2D	3D
Threshold (Th)	0.785	0.509
EER	0.044771	0.088562
Accuracy(%)	95.49	91.17
Sensitivity(%)	95.56	91.11
Specificity(%)	95.49	91.18
TA	86	82
FR	4	8
FA	69	135
TR	1461	1395
Identification(%)	94.44	87.78
FRR @0%FAR (FA=0)	0.144444 (FR=13, Th=0.751)	0.344444 (FR=31, Th=0.399)
FAR @0%FRR (TA=90)	0.513725 (FA=786, Th=0.8140)	0.574510 (FA=879, Th=0.631)

TABLE IV: 2D and 3D overall recognition results for real fingers

Real fingers	%								
	Threshold	EER	Accuracy	Sensitivity	Specificity	TA	FR	FA	TR
2D	0.752	0,066993	99,1975	86,6667	99,9346	78	12	1	1529
3D	0,406	0,161765	98,0864	67,7778	99,8693	61	29	2	1528

TABLE V: Maximum accuracy results for real fingers

## IX. DISCUSSION

In this work a finger vein scanner capable of capturing highly detailed vein images from multiple angles has been developed. These images are used to attempt to reconstruct 3D information of the veins inside the finger and perform recognition experiments with them. Compared to previous scanners, significant improvements have been made in both image quality and 3D reconstructions. Previous work such as [1], [11] or [32] either did not achieve high image quality of the finger veins, or did not use a setup with three cameras simultaneously for 3D finger vein recognition. Therefore, this work present a novelty that, to the best of the authors' knowledge, has not been presented in literature before.

However, the current system is not yet fully deployable for recognition. Image acquisition requires manual light adjustments and visual inspection is required to avoid over-illuminated fingers in the rotated cameras, which are occasionally caused by NIR light reflection from the bones inside the finger. In addition, despite the recognition results, vein reconstructions from real fingers have shown to be quite noisy, and depths estimates do not exactly match real world measurements. In both reconstruction methods, the lowest point of a finger is projected at a depth of approximately 55-56mm, whereas the physical setup has a height of 65mm.

Another point of discussion regarding 3D reconstructions is how the two proposed reconstruction methods should be compared/evaluated. Currently, only the visual inspections are used, and apart from these inspections the two methods are not easy to compare. First, the point clouds of both methods cannot be directly compared because both reconstruction methods use a different set of interest/vein points due to rectification. In order to compare both methods, they should contain the same number of vein points and be in the same coordinate system. Second, an error measure must be

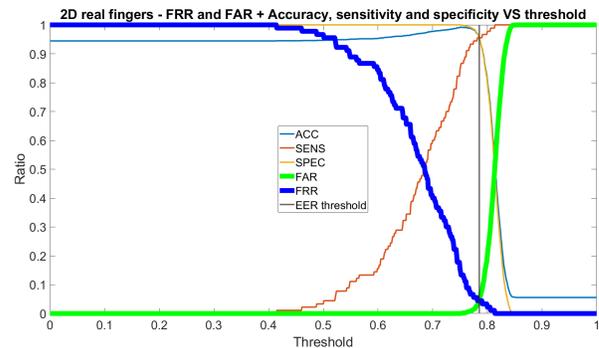


Fig. 24: 2D FAR and FRR curve for real fingers

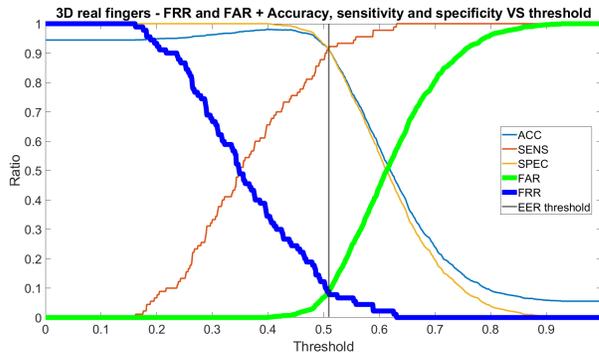


Fig. 25: 3D FAR and FRR curve for real fingers

designed to provide a score for the deviation from a circular object.

Furthermore, there are doubts whether the 3D reconstructions are actual vein reconstructions at all. When NIR light falls on a vein, a shadow is projected onto the skin of the finger, and it is this shadow that is captured on the CMOS sensor. This means that the 3D reconstructions created in this research are reconstructions of vein projections on the outer layers of skin; they are not reconstructions of actual veins. Consequently, the approach using NIR light and CMOS sensors may be inappropriate for 3D reconstructions, and future experiments are needed to confirm or invalidate this method.

Despite the previous work, the results of this research suggest that rotated fingers are not a direct problem for vein recognition (with artificial fingers). Results suggest that recognition is possible even in the presence of severe rotation, a relatively simple 2D correlation can outperform a 3D method. However, 3D detection still has advantages over 2D detection and so it might be advisable to improve 3D reconstruction methods. In addition, it should be mentioned that the 3D method computes an actual error between points whereas the 2D method returns a ratio of overlapping points; both methods differ significantly in the core of their implementation. The 2D correlation scores range from 0 to 1 whereas the RMSE error can range from 0 to infinity. Looking at the RMSE and correlation scores it can for example be observed that the range of incorrect correlation scores is much smaller compared to incorrect RMSE scores. The last point of discussion is the bicubic scaling of the input

images. The images are scaled from 1280\*800 to 280\*448px. It cannot go unmentioned that this removes vein points and vein information from the original image. Also, artefacts may become visible that were not present in the original images and vice versa. To analyse the effect of this scaling in more detail, it would be useful to evaluate both methods without rescaling input images.

## X. CONCLUSION

In this research, the goal was to gain more insights in the benefits of 3D vein reconstructions over 2D vein patterns. To do so, a general research goal was proposed:

- Investigate the potential benefits of 3D finger vein recognition with respect to 2D finger vein recognition.

In order to reach this goal, 3 research questions were proposed and each of these research question will be answered accordingly.

- 1) Develop a physical setup that is suitable for acquiring finger vein patterns in 2D suitable for 3D reconstruction
  - a) How to revise the current available vein scanners to obtain good quality recordings of finger veins from multiple viewing angles.

In order to develop a setup that is able to capture high detailed consistent vein images, a new setup is designed based upon an older design by Sjoerd Rozendal [23]. This new setup consists of 3 new monochrome cameras that are tilted 21.69°. Compared with previous scanners, images are consistent, they have very detailed vein patterns, no image enhancement is required and the setup is convenient to work with. The setup is lightweight and sturdy and can easily be transported. The setup and cameras are accompanied by a graphical user interface and software that allows to adjust LEDs values and to adjust gain and exposure values correctly and easily. Furthermore the software allows for a live preview while capturing images, allowing for easy tweaking of parameters.

- 2) Obtain 3D finger vein reconstructions
  - a) What is a suitable method for 3D reconstruction of finger vein images.

Two methods have been implemented and discussed for 3D vein reconstructions. A first method makes use Matlab's rectification technique whereas a second method skips the step of rectification and projects image points for different depths directly into different images planes via extrinsic and intrinsic parameters. Both methods require SAD for correlating points and both methods are tested on a paper mock up model.

Generally speaking, both methods allow to obtain viable 3D reconstructions. For the rectification method, however,

unexpected rotations are apparent and due to the rectification, multiple 3D images can not be combined as images are scaled/translated and are in their own reference frame. As a results, the reconstruction method directly mapping points into space is more suitable and more transparent in its implementation.

For both methods, these reconstructions are only accurate for a portion of the finger, with a larger region seemingly accurately reconstructed when directly projecting points back into space. Results suggests that SAD as function of depth creates less noise and artefacts in 3D reconstruction and is slightly more accurate compared to the method making use of rectifications. Furthermore, since this method involves no rectification, it holds better cards for combining partial 3D reconstructions. Additionally, although a difference in relative depth between 3D reconstructions and the true depth of the finger is apparent, this would not have major effects on recognition performances since this is consistent throughout all reconstructions.

- 3) Investigate the benefits of recognition with 3D reconstructed finger veins with respect to 2D finger vein recognition.

Results have shown that for artificial fingers 2D recognition works well, even if severe rotations are present in fingers. Results suggest accuracy's up to 99% for 2D correlation and 98% accuracy for ICP RMSE results, while using artificial fingers rotated up to 50°. With real fingers, 2D outperforms 3D as 2D reaches an accuracy's of 95.5% whilst 3D reaches an accuracy of 91.2%. Results when setting the threshold such that a maximum accuracy is achieved, suggest that 2D is more suitable for recognition as sensitivities for 2D are almost 20 percent point higher then those for 3D.

## XI. ACKNOWLEDGEMENTS

This thesis has been performed during the widespread Corona outbreak and as a result it has partially been worked on alone from home. With the help and support from Geert-Jan Laanstra, the DMB lab was however eventually opened and as a results this work has greatly benefited by the social interactions with Geert-Jan, Kevin and others in the lab. Furthermore my friends from *Lusus Naturae* should be thanked. Partially for their theoretical support, but mainly for the weekly 'support meetings' and their acceptance of me working on this thesis while on holiday.

## REFERENCES

- [1] S. Bunda. 3d point cloud reconstruction based on the finger vascular pattern. 2018.
- [2] J. Choi, W. Song, T. Kim, S.-R. Lee, and H. C. Kim. Finger vein extraction using gradient normalization and principal curvature. *SPIE*, 7251, 02 2009.
- [3] R. Fuksis, M. Greitans, O. Nikisins, and M. Pudzs. Infrared imaging system for analysis of blood vessel structure. *Electron Electrical Eng*, 1, 01 2010.
- [4] R. I. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, ISBN: 0521540518, second edition, 2004.

- [5] Hitachi. Finger vein authentication: White paper. 2006.
- [6] B. Huang, Y. Dai, R. Li, D. Tang, and W. Li. Finger-vein authentication based on wide line detector and pattern normalization. In *2010 20th International Conference on Pattern Recognition*, pages 1269–1272, 2010.
- [7] W. Kang, H. Liu, W. Luo, and F. Deng. Study of a full-view 3d finger vein verification technique. *IEEE Transactions on Information Forensics and Security*, 15:1175–1189, 2020.
- [8] J. Lamberts. Stereo and disparity, 2018.
- [9] E. C. Lee, H. Lee, and K. Park. Finger vein recognition using minutia-based alignment and local binary pattern-based feature extraction. *International Journal of Imaging Systems and Technology*, 19:179 – 186, 09 2009.
- [10] Y. Lu, Y. Sook, and D. Park. Finger vein identification system using two cameras. *Electronics Letters*, 50:1591–1593, 10 2014.
- [11] Z. Ma, L. Fang, J. Duan, S. Xie, and Z. Wang. Personal identification based on finger vein and contour point clouds matching. In *2016 IEEE International Conference on Mechatronics and Automation*, pages 1983–1988, 2016.
- [12] D. M. Mancini, L. Bolinger, H. Li, K. Kendrick, B. Chance, and J. R. Wilson. Validation of near-infrared spectroscopy in humans. *Journal of Applied Physiology*, 77(6):2740–2747, 1994. PMID: 7896615.
- [13] N. Miura, A. Nagasaka, and T. Miyatake. Feature extraction of finger-vein patterns based on repeated line tracking and its application to personal identification. *Machine Vision and Applications*, 15:194–203, 10 2004.
- [14] N. Miura, A. Nagasaka, and T. Miyatake. Extraction of finger-vein patterns using maximum curvature points in image profiles. volume E90D, pages 347–350, 01 2005.
- [15] D. Mulyono and S.-J. Horng. A study of finger vein biometric for personal identification. pages 1 – 8, 05 2008.
- [16] J. Murkin and A. Miguel. Near-infrared spectroscopy as an index of brain and tissue oxygenation. *British Journal of Anaesthesia*, 103:i3–i13, 12 2009.
- [17] S. Muñoz. Raspicam: C++ api for using raspberry camera with/without opencv, 2020.
- [18] B. Peeters. Finishing and improving of finger vein system. 2020.
- [19] B. Prommegger, C. Kauba, and A. Uhl. Longitudinal finger rotation in finger-vein recognition. In *2018 International Conference of the Biometrics Special Interest Group (BIOSIG)*, pages 1–8, 2018.
- [20] B. Prommegger, C. Kauba, and A. Uhl. Multi-perspective finger-vein biometrics. In *2018 IEEE 9th International Conference on Biometrics Theory, Applications and Systems (BTAS)*, pages 1–9, 2018.
- [21] B. Prommegger, C. Kauba, A. Uhl, C. Busch, S. Marcel, and R. Veldhuis. *Different Views on the Finger—Score-Level Fusion in Multi-Perspective Finger Vein Recognition*, pages 261–305. Springer International Publishing, Cham, 2020.
- [22] Y. Qi, Y. Zhou, C. Zhou, X. Hu, and X. Hu. Vein point cloud registration algorithm for multi-pose hand vein authentication. In *2016 IEEE International Conference on Identity, Security and Behavior Analysis (ISBA)*, pages 1–6, 2016.
- [23] S. Rozendal. Redesign of a finger vein scanner. 2018.
- [24] L. Spreeuwers. Multi-view passive 3d face acquisition device. In *Proceedings of the Special Interest Group on Biometrics and Electronic Signatures*, number 1 in Lecture Notes in Informatics (LNI) - Proceedings, pages 13–24, Germany, Sept. 2008. Gesellschaft für Informatik.
- [25] P. Tome, M. Vanoni, and S. Marcel. On the vulnerability of finger vein recognition to spoofing. In *2014 International Conference of the Biometrics Special Interest Group (BIOSIG)*, pages 1–10, 2014.
- [26] B. Ton. Miura et al. vein extraction methods.
- [27] B. Ton. Vascular pattern of the finger: Biometric of the future. 2012.
- [28] B. T. Ton and R. N. J. Veldhuis. A high quality finger vascular pattern dataset collected using a custom designed capturing device. In *2013 International Conference on Biometrics (ICB)*, pages 1–5, 2013.
- [29] R. Veldhuis, L. Spreeuwers, B. Ton, and S. Rozendal. *A High-Quality Finger Vein Dataset Collected Using a Custom-Designed Capture Device*, pages 63–75. Springer International Publishing, Cham, 2020.
- [30] Y. Wang, L. G. Hasebrook, and D. L. Lau. Data acquisition and processing of 3-d fingerprints. *IEEE Transactions on Information Forensics and Security*, 5(4):750–760, 2010.
- [31] Y. Zhan, A. S. Rathore, G. Milione, Y. Wang, W. Zheng, W. Xu, and J. Xia. 3d finger vein biometric authentication with photoacoustic tomography. *Appl. Opt.*, 59(28):8751–8758, Oct 2020.
- [32] Q. Zhang, Y. Zhou, D. Wang, and X. Hu. Personal authentication using hand vein and knuckle shape point cloud matching. In *2013 IEEE Sixth International Conference on Biometrics: Theory, Applications and Systems (BTAS)*, pages 1–6, 2013.
- [33] W. Zhou and N. Li. Exploring implementation of three-dimensional skeleton reconstruction of finger vein for pps. *Procedia CIRP*, 83:826–830, 01 2019.

## APPENDIX

## A. Camera testing

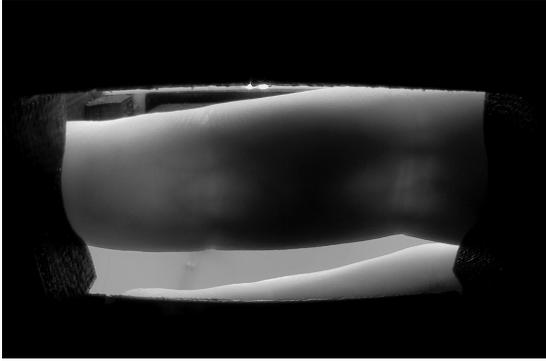


Fig. 26: WW camera without filters

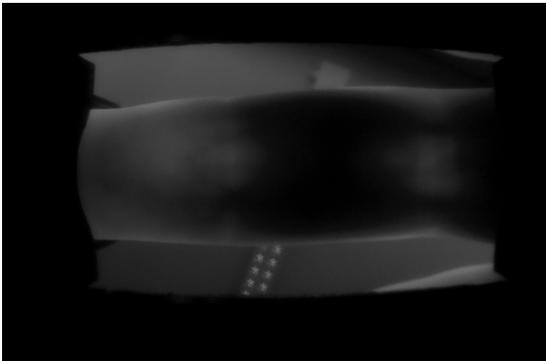


Fig. 27: WW camera with IR pass filter

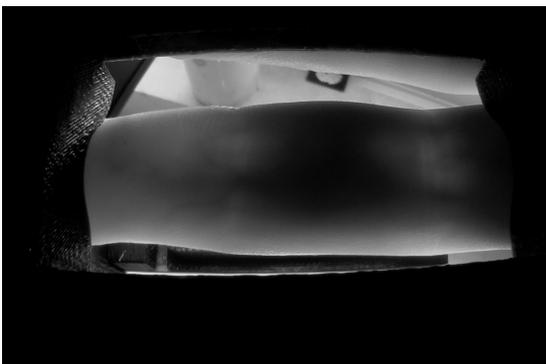


Fig. 28: Night vision camera + no filter



Fig. 29: Night vision camera + no filter + AHE + rescaling

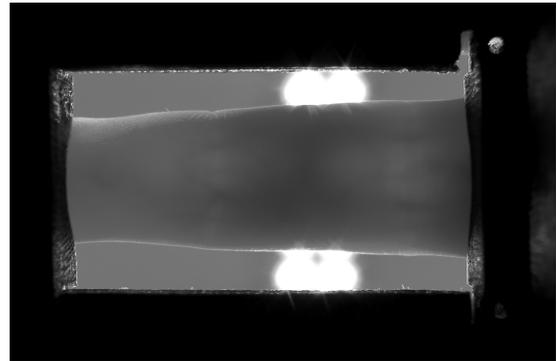


Fig. 30: High resolution camera with 12mm lens

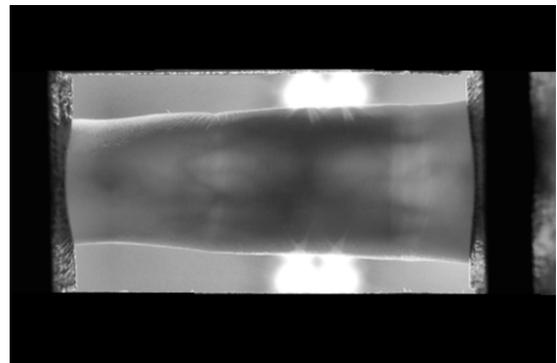


Fig. 31: High resolution camera with 12mm lens + AHE + rescaling

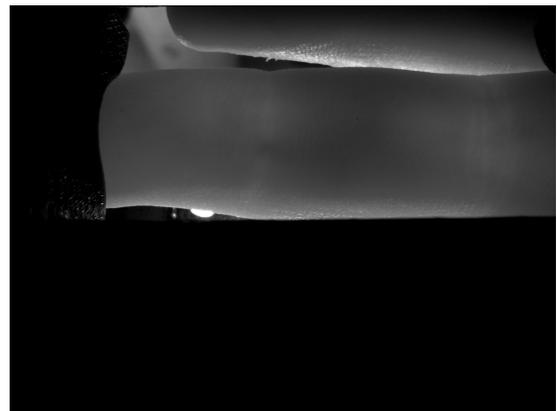


Fig. 32: Pi NOIR V2

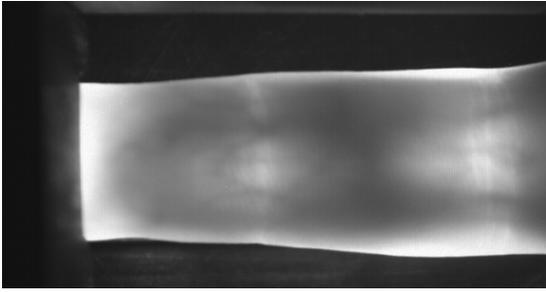


Fig. 33: Vein image from the V1 scanner

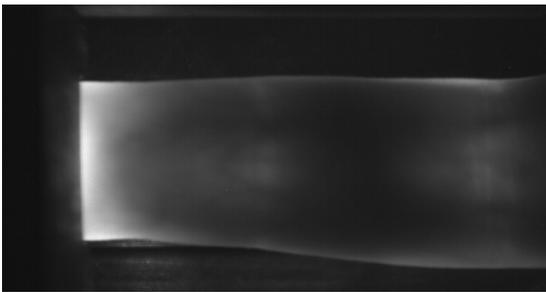


Fig. 34: BCi5 images after shaking a hand vigorously

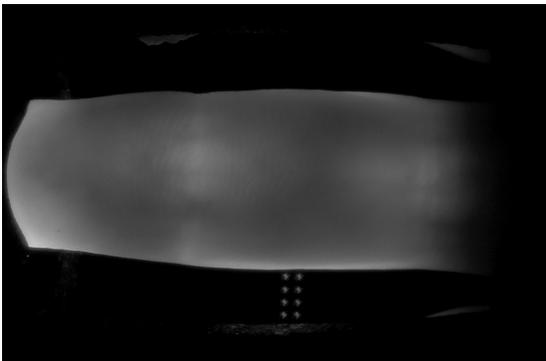


Fig. 35: OV5674 image before shaking a hand vigorously

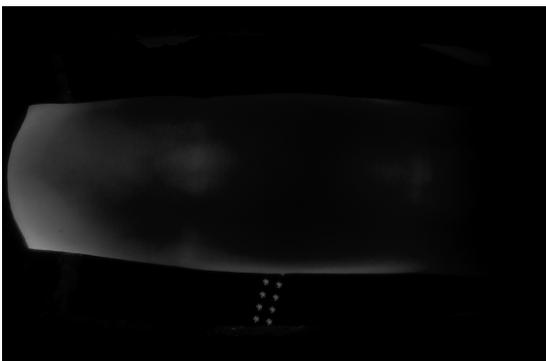


Fig. 36: OV5674 image after shaking a hand vigorously

Camera	BCi5	WW	WW2	Pi Night Vision	Pi JT	Pi NOIR V2	Pi high res	Arducam
Sensor	IBIS5a	OV5674	OV5674	OV5674	OV5674	IMX219	IMX477	OV9281
Resolution	1.3MP	5MP	5MP	5MP	5MP	8MP	12.3MP	1MP
Pixel size (um)	6.7	1.4	1.4	1.4	1.4	1.12	1.55	3
Type	Monochrome	RGB	RGB	RGB	RGB	RGB	RGB	Monochrome
Lens	C mount 12mm	S mount fisheye (160D-120H)	S mount fisheye (200)	S mount (75.7D)	Pinhole	Pinhole	C mount 12/16mm	S mount (130D)
Filter	Custom IR	IR NS	-	-	NS	-	NS	850±50nm Narrow band IR
Software	V1	V2	V2	V2	V2	V2	V2	Arducam
Vein detail & image quality	Acceptable	Moderate	Moderate	Moderate	Unacceptable	Unacceptable	Acceptable	Good
Image consistency	Good	Unacceptable	Unacceptable	Unacceptable	-	-	Moderate	Good
Enhancement required?	Maybe not	Yes	Yes	Yes	-	-	Yes	No
Easy capture + live preview?	Yes	No	No	No	No	No	No	Yes

TABLE VI: Camera evaluation results

*B. Regular 3D reconstruction with Matlab (method 1)*

**Rectified C21: C2 red, C1 blue**



Fig. 37: C21 rectified images

**Disparity for C2 (C21)**

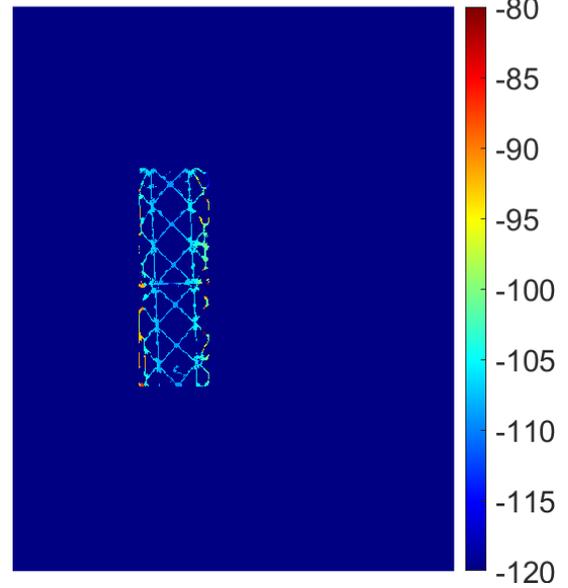


Fig. 39: C21 disparity map (smaller detection width)

**Rectified C23: C2 red, C3 blue**

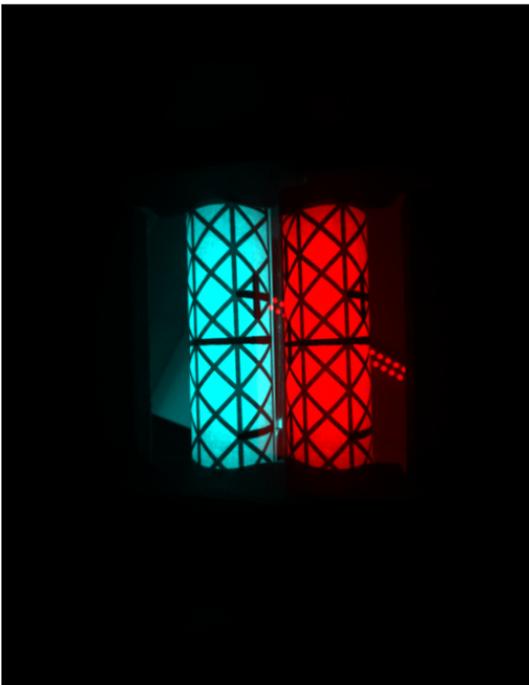


Fig. 38: C23 rectified images

**Disparity for C2 (C23)**

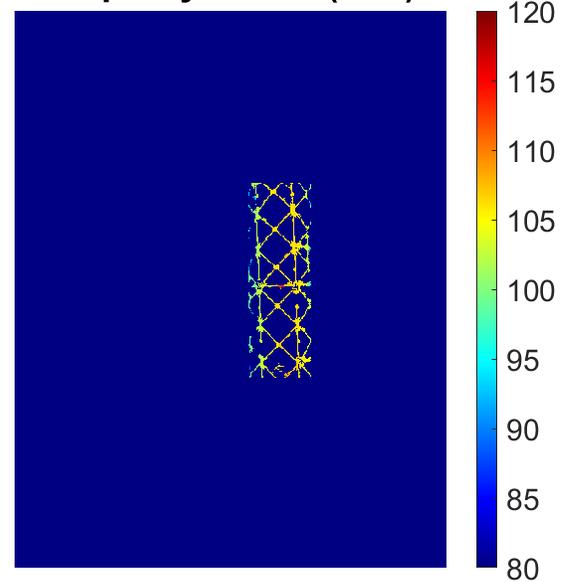


Fig. 40: C23 disparity map (smaller detection width)

C. SAD as a function of depth (method 2)

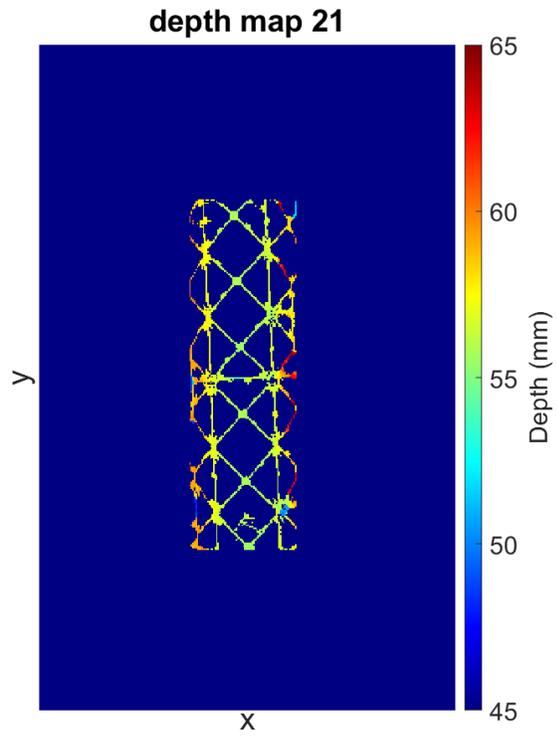


Fig. 41: Depthmap for C21

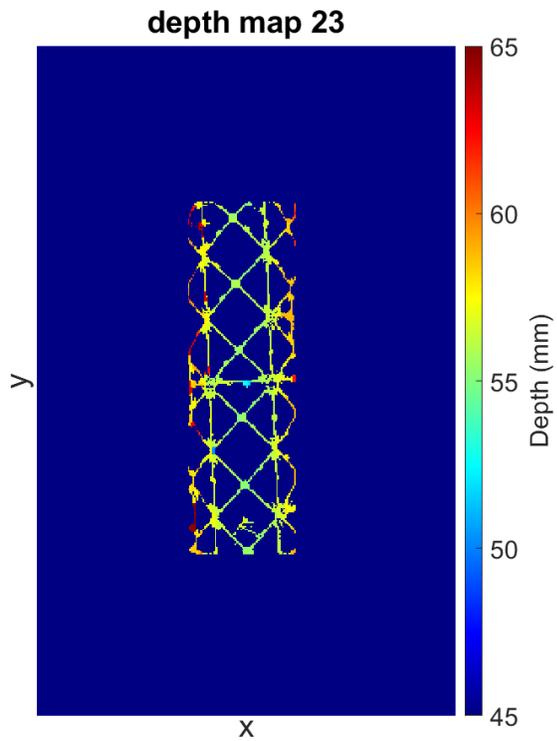


Fig. 42: Depthmap for C23

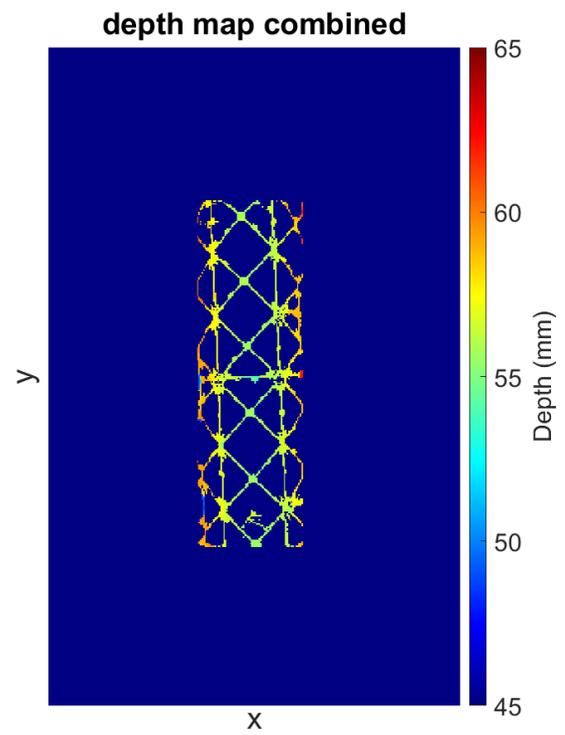


Fig. 43: Combined depthmaps

#### D. Recognition results

2D correlation	Threshold	EER	Accuracy(%)	Sensitivity(%)	Specificity (%)	TA	FR	FA	TR
Finger 1	0.788	0	100	100	100	21	0	0	147
Finger 2	0.784	0	100	100	100	21	0	0	147
Finger 3	0.797	0	100	100	100	21	0	0	147
Finger 4	0.768	0	100	100	100	21	0	0	147
Finger 5	0.772	0	100	100	100	21	0	0	147
Finger 6	0.752	0	100	100	100	21	0	0	147
Finger 7	0.726	0	100	100	100	21	0	0	147
Finger 8	0.690	0	100	100	100	21	0	0	147
Overall	0.797	0.000425	99.93	100	99.91	168	0	1	1175

TABLE VII: 2D correlation characteristics artificial fingers

3D correlation	Threshold	EER	Accuracy(%)	Sensitivity(%)	Specificity(%)	TA	FR	FA	TR
Finger 1	0.225	0	100	100	100	21	0	0	147
Finger 2	0.247	0	100	100	100	21	0	0	147
Finger 3	0.313	0.006800	98.81	100	98.64	21	0	2	145
Finger 4	0.309	0	100	100	100	21	0	0	147
Finger 5	0.265	0	100	100	100	21	0	0	147
Finger 6	0.287	0	100	100	100	21	0	0	147
Finger 7	0.274	0	100	100	100	21	0	0	147
Finger 8	0.283	0.047600	95.24	95.24	95.24	20	1	6	141
Overall	0.309	0.018700	98.07	98.21	98.04	165	3	23	1153

TABLE VIII: 3D ICP characteristics artificial fingers

2D identification	← Number of images identified as basefinger →							
Actual finger ↓	1	2	3	4	5	6	7	8
Finger 1	<b>21</b>	0	0	0	0	0	0	0
Finger 2	0	<b>21</b>	0	0	0	0	0	0
Finger 3	0	0	<b>21</b>	0	0	0	0	0
Finger 4	0	0	0	<b>21</b>	0	0	0	0
Finger 5	0	0	0	0	<b>21</b>	0	0	0
Finger 6	0	0	0	0	0	<b>21</b>	0	0
Finger 7	0	0	0	0	0	0	<b>21</b>	0
Finger 8	0	0	0	0	0	0	0	<b>21</b>

TABLE IX: 2D correlation identification results artificial fingers

3D identification	← Number of images identified as basefinger →							
Actual finger ↓	1	2	3	4	5	6	7	8
Finger 1	<b>21</b>	0	0	0	0	0	0	0
Finger 2	0	<b>21</b>	0	0	0	0	0	0
Finger 3	0	0	<b>21</b>	0	0	0	0	0
Finger 4	0	0	0	<b>21</b>	0	0	0	0
Finger 5	0	0	0	0	<b>21</b>	0	0	0
Finger 6	0	0	0	0	0	<b>21</b>	0	0
Finger 7	0	0	0	0	0	0	<b>21</b>	0
Finger 8	0	0	0	<b>1</b>	0	0	0	<b>20</b>

TABLE X: 3D correlation identification results artificial fingers

Rotation	RSME	Accepted as	Correct finger
-50	0.292	4	3
-50	0.293	7	5
-45	0.288	7	5
-40	0.295	7	5
50	0.307	4	6
-50	0.288	2	8
-45	0.287	2	8
-50	0.253	4	8
-45	0.250	4	8
-40	0.290	4	8
-35	0.299	4	8
-50	0.278	6	8
-45	0.294	6	8
5	0.294	2	8
25	0.289	4	8
30	0.278	4	8
35	0.282	4	8
45	0.306	4	8
50	0.270	4	8
30	0.290	6	8
35	0.278	6	8
40	0.292	6	8
40	0.304	7	8

TABLE XI: 3D falsely accepted images with rotations and scores for artificial fingers

2D Correlation real fingers	Threshold	EER	Accuracy (%)	Sensitiviy (%)	Specificity (%)	TA	FR	FA	TR
Finger 1	0,724	0	100	100	100	5	0	0	85
Finger 2	0,744	0	100	100	100	5	0	0	85
Finger 3	0,686	0	100	100	100	5	0	0	85
Finger 4	0,762	0	100	100	100	5	0	0	85
Finger 5	0,8	0,217647	76,67	80	76,47	4	1	20	65
Finger 6	0,788	0,158824	87,78	80	88,24	4	1	10	75
Finger 7	0,692	0	100	100	100	5	0	0	85
Finger 8	0,759	0	100	100	100	5	0	0	85
Finger 9	0,66	0	100	100	100	5	0	0	85
Finger 10	0,697	0	100	100	100	5	0	0	85
Finger 11	0,767	0,011765	97,78	100	97,65	5	0	2	83
Finger 12	0,7	0	100	100	100	5	0	0	85
Finger 13	0,769	0	100	100	100	5	0	0	85
Finger 14	0,782	0,005882	98,89	100	98,82	5	0	1	84
Finger 15	0,752	0,005882	98,89	100	98,82	5	0	1	84
Finger 16	0,762	0	100	100	100	5	0	0	85
Finger 17	0,805	0,2	80	80	80	4	1	17	68
Finger 18	0,73	0	100	100	100	5	0	0	85
Overall	0,785	0,044771	95,49	95,56	95,49	86	4	69	1461

TABLE XII: 2D correlation characteristics real fingers

3D RSME real fingers	Threshold	EER	Accuracy (%)	Sensitiviy (%)	Specificity (%)	TA	FR	FA	TR
Finger 1	0,297	0	100	100	100	5	0	0	85
Finger 2	0,322	0	100	100	100	5	0	0	85
Finger 3	0,464	0	100	100	100	5	0	0	85
Finger 4	0,603	0,2	80	80	80	4	1	17	68
Finger 5	0,539	0,2	80	80	80	4	1	17	68
Finger 6	0,51	0,005882	98,89	100	98,82	5	0	1	84
Finger 7	0,367	0	100	100	100	5	0	0	85
Finger 8	0,539	0,4	60	60	60	3	2	34	51
Finger 9	0,341	0	100	100	100	5	0	0	85
Finger 10	0,406	0	100	100	100	5	0	0	85
Finger 11	0,377	0	100	100	100	5	0	0	85
Finger 12	0,3	0	100	100	100	5	0	0	85
Finger 13	0,525	0,047059	91,11	100	90,59	5	0	8	77
Finger 14	0,486	0,023529	95,56	100	95,29	5	0	4	81
Finger 15	0,442	0	100	100	100	5	0	0	85
Finger 16	0,318	0	100	100	100	5	0	0	85
Finger 17	0,398	0	100	100	100	5	0	0	85
Finger 18	0,554	0,2	80	80	80	4	1	17	68
Overall	0,509	0,088562	91,17	91,11	91,18	82	8	135	1395

TABLE XIII: 3D ICP characteristics real fingers

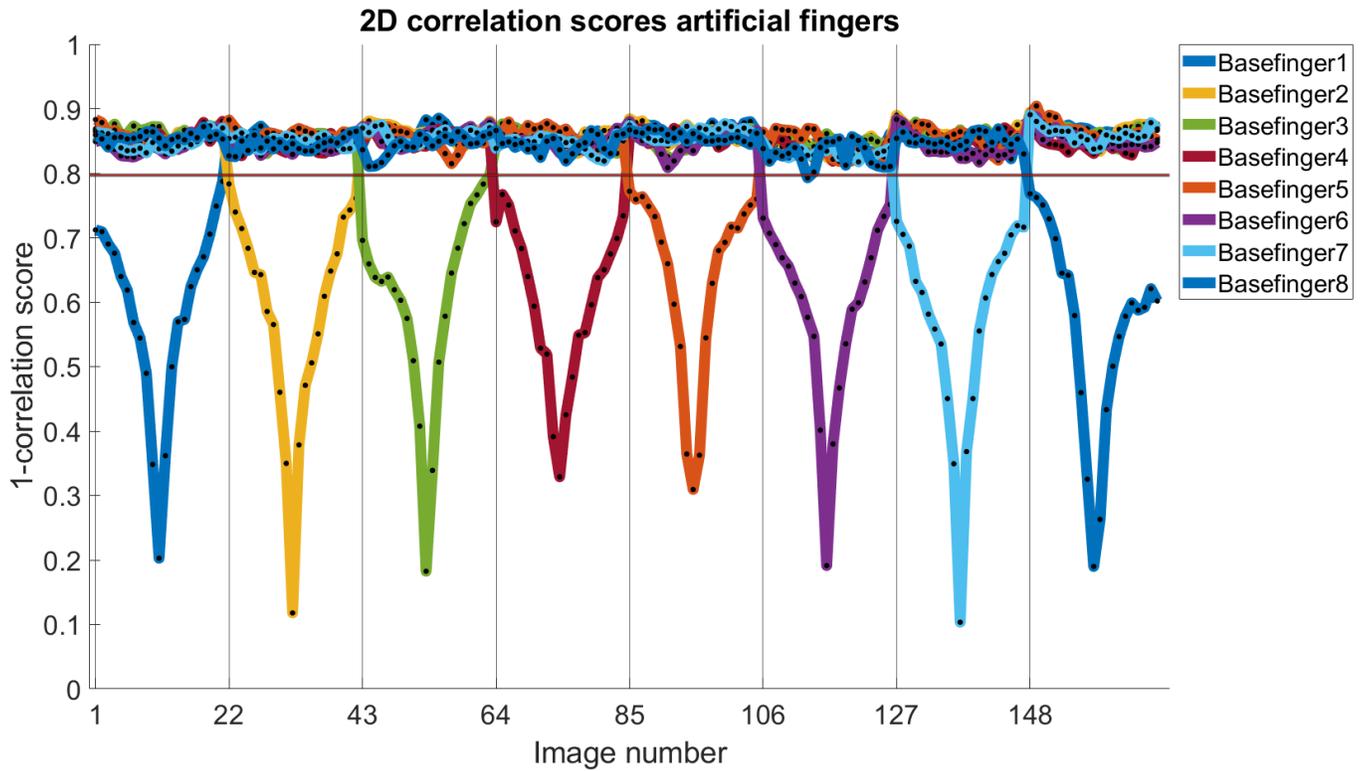


Fig. 44: Correlation scores for 2D recognition with artificial fingers (see table II)

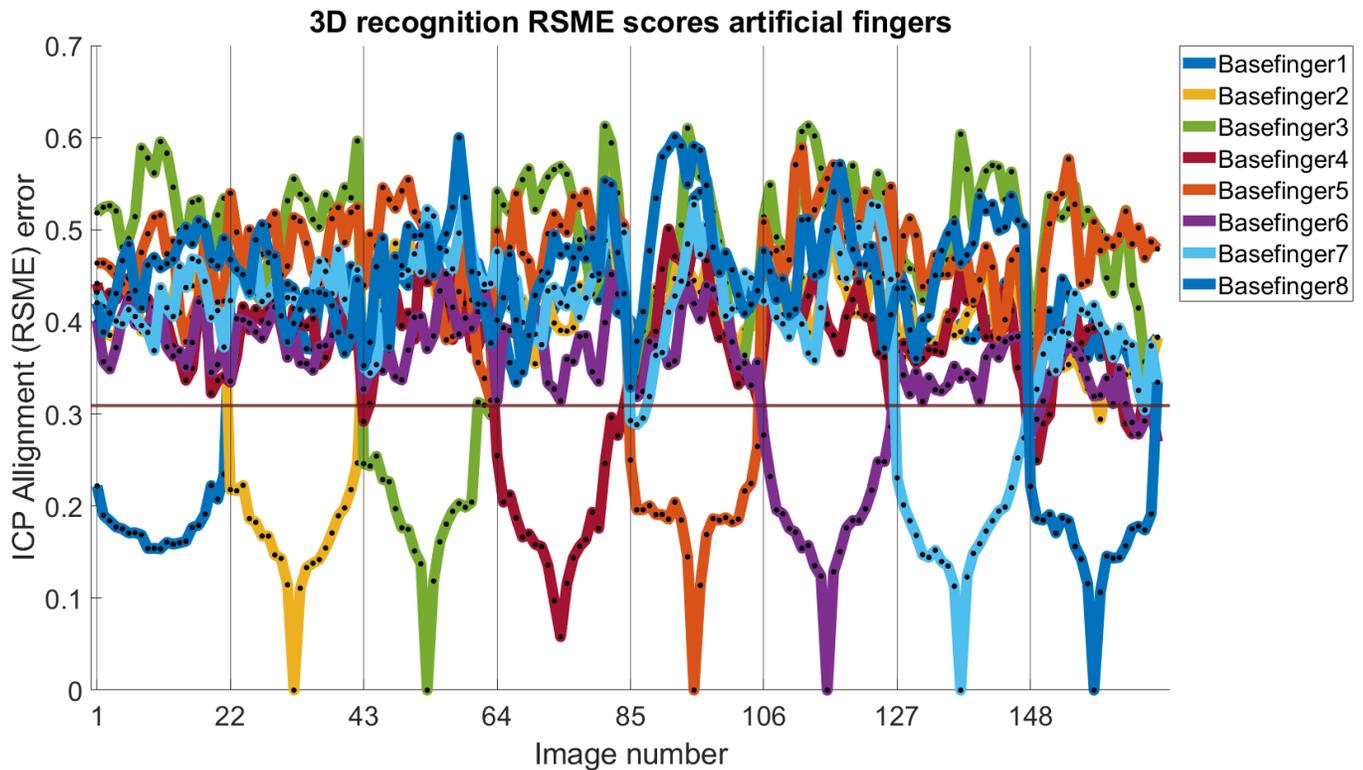


Fig. 45: ICP RMSE scores for 3D recognition with artificial fingers (see table II)

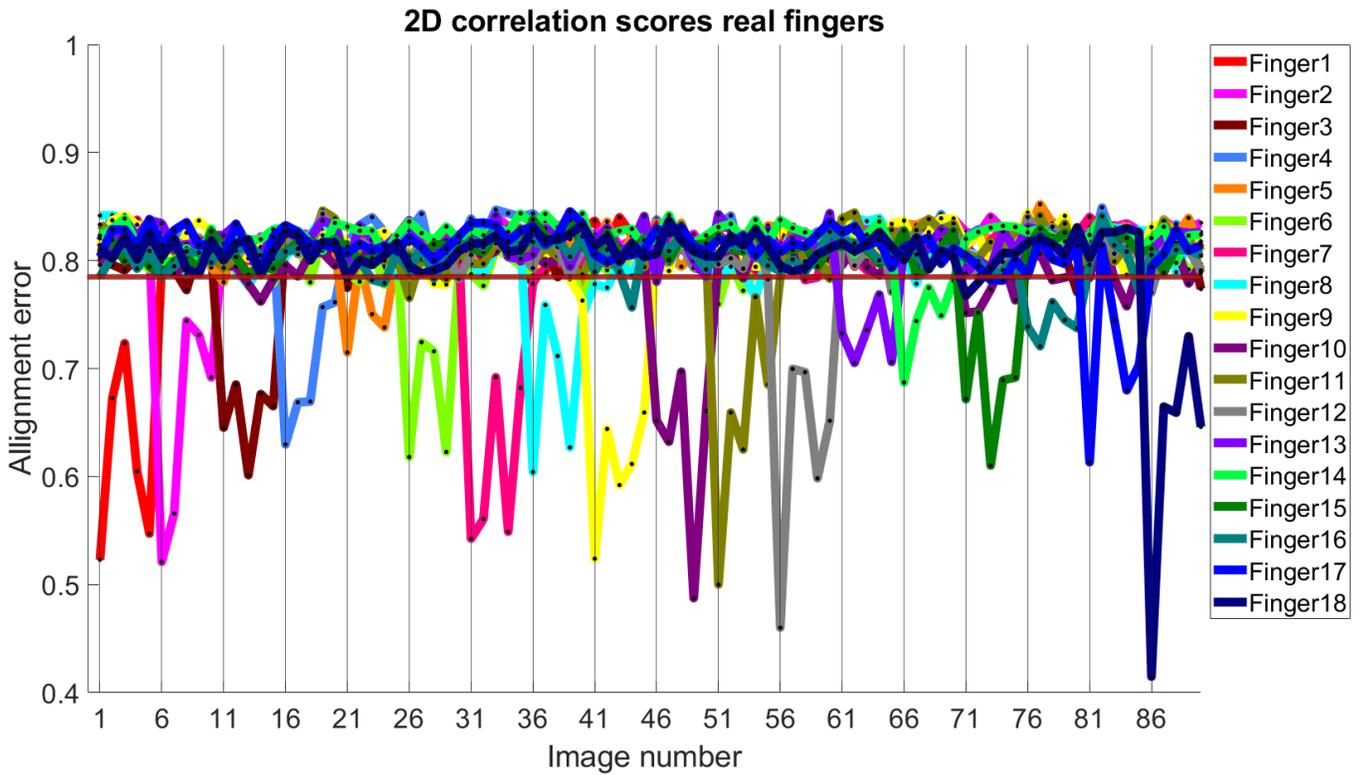


Fig. 46: Correlation scores for 2D recognition with real fingers

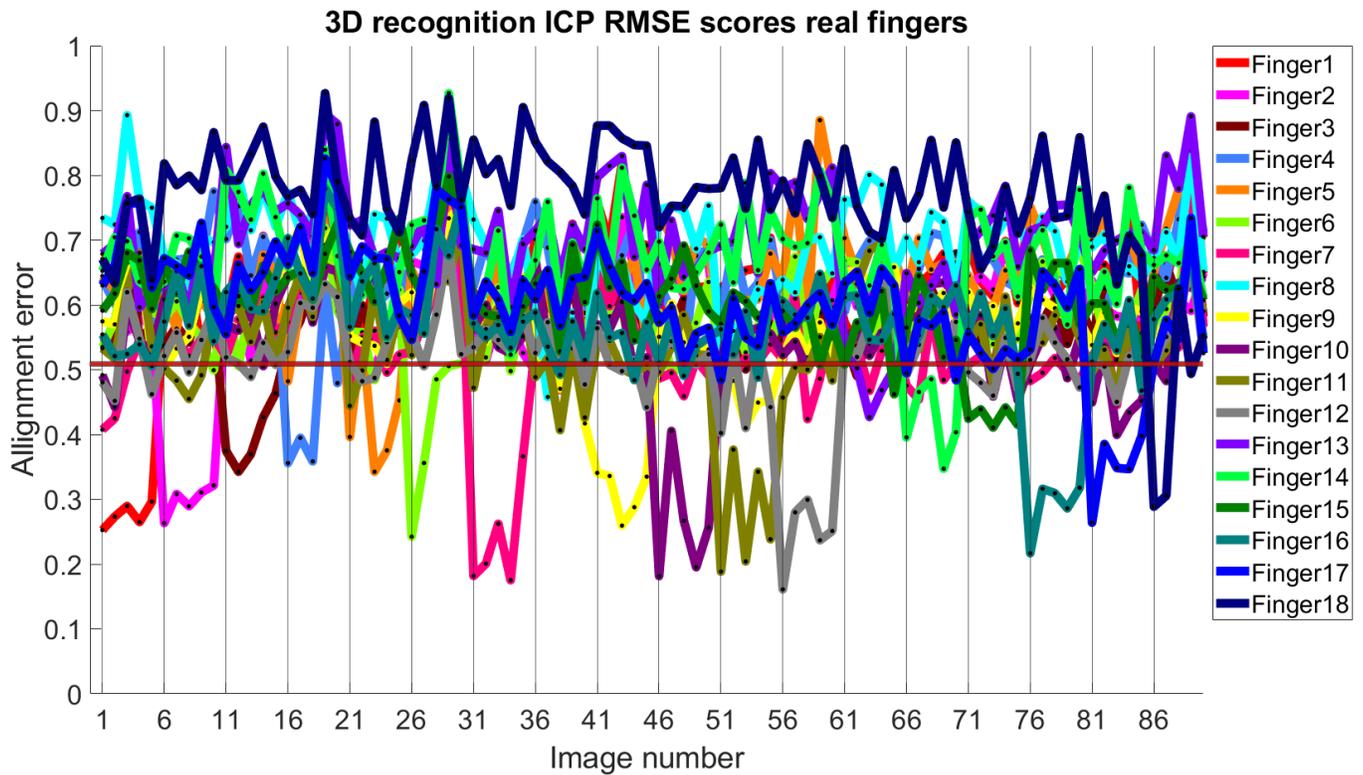


Fig. 47: ICP RMSE scores for 3D recognition with real fingers

2D real correlation	Number of images identified as finger																	
Actual finger	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
Finger 1	<b>5</b>	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Finger 2	0	<b>5</b>	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Finger 3	0	0	<b>5</b>	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Finger 4	0	0	0	<b>5</b>	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Finger 5	0	0	0	0	<b>3</b>	<b>1</b>	0	<b>1</b>	0	0	0	0	0	0	0	0	0	0
Finger 6	0	0	0	0	0	<b>4</b>	0	0	0	0	0	1	0	0	0	0	0	0
Finger 7	0	0	0	0	0	0	<b>5</b>	0	0	0	0	0	0	0	0	0	0	0
Finger 8	0	0	0	0	0	0	0	<b>5</b>	0	0	0	0	0	0	0	0	0	0
Finger 9	0	0	0	0	0	0	0	0	<b>5</b>	0	0	0	0	0	0	0	0	0
Finger 10	0	0	0	0	0	0	0	0	0	<b>5</b>	0	0	0	0	0	0	0	0
Finger 11	0	0	0	0	0	0	0	0	<b>1</b>	0	<b>4</b>	0	0	0	0	0	0	0
Finger 12	0	0	0	0	0	0	0	0	0	0	0	<b>5</b>	0	0	0	0	0	0
Finger 13	0	0	0	0	0	0	0	0	0	0	0	0	<b>5</b>	0	0	0	0	0
Finger 14	0	0	0	0	0	0	0	0	0	0	0	0	0	<b>5</b>	0	0	0	0
Finger 15	0	0	0	0	0	0	0	0	0	0	0	0	0	0	<b>5</b>	0	0	0
Finger 16	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	<b>5</b>	0	0
Finger 17	0	0	0	0	<b>1</b>	0	0	0	0	0	0	0	0	0	0	0	<b>4</b>	0
Finger 18	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	<b>5</b>

TABLE XIV: 2D identification results real fingers

3D ICP RMSE real	Number of images identified as finger																	
Actual finger	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
Finger 1	<b>5</b>	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Finger 2	0	<b>5</b>	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Finger 3	0	0	<b>5</b>	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Finger 4	0	0	0	<b>5</b>	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Finger 5	0	0	0	0	<b>4</b>	0	0	0	0	0	0	<b>1</b>	0	0	0	0	0	0
Finger 6	0	0	0	0	0	<b>5</b>	0	0	0	0	0	0	0	0	0	0	0	0
Finger 7	0	0	0	0	0	0	<b>5</b>	0	0	0	0	0	0	0	0	0	0	0
Finger 8	0	<b>1</b>	0	0	0	0	0	<b>1</b>	<b>1</b>	0	<b>2</b>	0	0	0	0	0	0	0
Finger 9	0	0	0	0	0	0	0	0	<b>5</b>	0	0	0	0	0	0	0	0	0
Finger 10	0	0	0	0	0	0	0	0	0	<b>5</b>	0	0	0	0	0	0	0	0
Finger 11	0	0	0	0	0	0	0	0	0	0	<b>5</b>	0	0	0	0	0	0	0
Finger 12	0	0	0	0	0	0	0	0	0	0	0	<b>5</b>	0	0	0	0	0	0
Finger 13	0	0	<b>1</b>	0	0	0	0	0	0	0	0	0	<b>2</b>	0	<b>2</b>	0	0	0
Finger 14	0	0	0	0	0	0	<b>1</b>	0	0	0	0	0	0	<b>4</b>	0	0	0	0
Finger 15	0	0	0	0	0	0	0	0	0	0	0	0	0	0	<b>5</b>	0	0	0
Finger 16	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	<b>5</b>	0	0
Finger 17	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	<b>5</b>	0
Finger 18	0	0	0	0	0	0	<b>1</b>	0	0	0	0	0	0	0	0	0	<b>1</b>	<b>3</b>

TABLE XV: 3D identification results real fingers

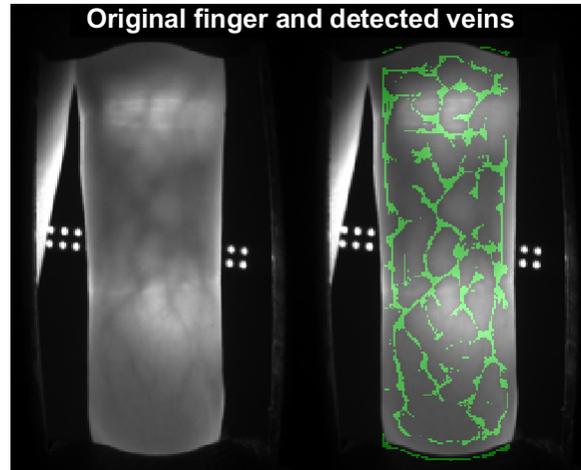


Fig. 48: Real finger and its detected vein/interest points

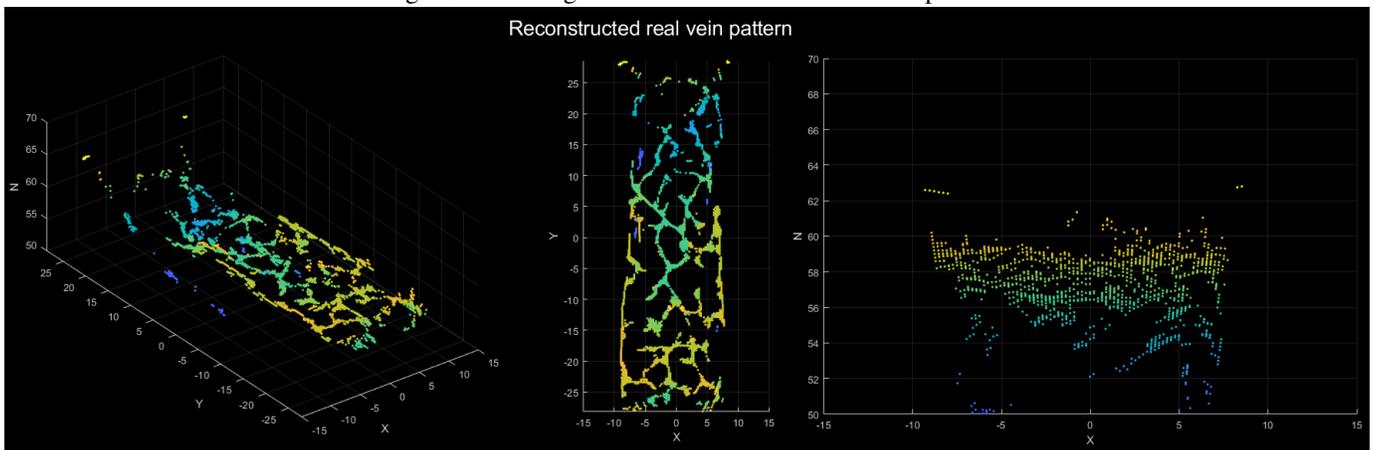


Fig. 49: A 3D reconstructed vein pattern from the finger in figure 48

## E. Disparity algorithm

```

1
2 function [disparity_map, SADs] = vein_disparity(reference, I1, I2, direction, window_size, d_min, ...
   d_max, percentage)
3 % Vein_disparity - Thomas van Zonneveld
4 %   Calculates disparity between two images for specific reference points
5 %
6 %   disparity_map = vein_disparity(reference, I1, I2, direction, window_size, d_min, d_max)
7 %
8 %   disparity_map = disparity mat by looking for interest points from I1 in I2.
9 %   SADs = disparities*width*row image. For each row, at each X point
10 %   the disparity is returned.
11 %   reference = binary vein image (points of interest to be matched) from I1
12 %   I1 = full color original image, from which veins are reference points
13 %   I2 = full color image to find POI in.
14 %   Direction = 1 : searching in image LEFT of reference (positive disparity wrt matlab ...
   orientation)
15 %   Direction = -1 : searching in image right of reference (negative disparity)
16 %   Use imtool(stereoAnaglyph(rectified_I1, rectified_I2)) to estimate disparities.
17 %   D_min = minimum disparity (>0 && < maximum disparity)
18 %   D_max = maximum disparity (> minimum disparity)
19 %   percentage = SAD detected must be < percentage * SAD mean
20
21 % NOTE. Matlab assumes camera 2 is physically left of camera 1. Implying
22 % that camera 1 is the reference. This implies that disparities are positive to the left.
23
24 if ((size(I1) ≠ size(I2)) | size(I1) ≠ size(reference) | size(I2) ≠ size(reference))
25     error('Ensure input images are of equal size')
26 end
27 if ((direction ≠ 1) && (direction ≠ -1) )
28     error('Ensure direction to be 1 or -1')
29 end
30
31 [height, width] = size(I1);
32
33 %create an empty disparity map
34 disparity_map = NaN(size(I1));
35 SADs = NaN(d_max-d_min+1, width, height);
36
37 disp('Window size, d_min, d_max, percentage');
38 disp([window_size, d_min, d_max, percentage])
39
40 for row = window_size:height-ceil(window_size/2) % ensure a window can be obtained later on
41     ref = reference(row, :); % All the columns of the vein image for a specific row
42     [r, y] = find(ref == 1); % find the columns with a vein for this specific row
43     for c = 1 : size(y,2) % for each vein in this specific row
44         % Create a window based on I1
45         window_ref = I1(row-floor(window_size/2):row+floor(window_size/2), ...
   y(c)-floor(window_size/2):y(c)+floor(window_size/2));
46         scores = zeros(d_max - d_min, 2); % Create a storage unit for disparities and SADs in one row
47         % Obtain the the SAD per pixel shift
48         index = 1;
49         for disparity = d_min : d_max % for all disparities
50             scores(index, 1) = (direction * disparity); % write the disparity in a matrix. If ...
   looking in righter image, disparity is negative.
51             col = y(c) - (direction * disparity); % obtain the corresponding column
52             if (col-floor(window_size/2) > 0 && col+floor(window_size/2) ≤ width) % if interest ...
   point + disparity + window_size/2 not exceeding image dimension
53                 % create a window in I2
54                 window = I2(row-floor(window_size/2):row+floor(window_size/2), ...
   col-floor(window_size/2):col+floor(window_size/2));
55                 SAD = sum(abs(window_ref - window), 'all'); % obtain the SAD for the two windows
56                 scores(index,2) = SAD; % write the SAD in the corresponding entry in the score matrix
57             else % if exceeding image boundary, we have no SAD for this disparity.
58                 scores(index, 2) = NaN;
59             end
60             index = index + 1;
61         end
62         SADs(:,y(c), row) = scores(:,2); % for each vein point, store all SADs. For each row, have ...
   a own plane with X = Y = disparities
63         sorted_scores = sortrows(scores,2); % sort scores from low to high SAD's
64
65         temp = mean(sorted_scores(:,2)); % get the mean SAD
66         if (sorted_scores(1,2) < percentage*temp) % check if we have a serious lower SAD

```

```
67         if (sorted_scores(1,1) ≠ direction*d_max) % check if it is not the max value
68             if (sorted_scores(1,1) ≠ direction*d_min) % or minimum
69                 disparity_map(row,y(c)) = sorted_scores(1,1); % write the disparity ...
                    corresponding to the lowest SAD
70             end
71         end
72     end
73 end
74 end
75 end
```

## F. Camera calibration results

```

1 ---- stereo parameters rotated + scaled ----
2
3 ---- MIDDLE + LEFT ----
4 ---- Camera 1 ----
5 Focal length (pixels): [ 249.6161 +/- 0.1714      249.4816 +/- 0.1715 ]
6 Principal point (pixels):[ 139.0771 +/- 0.0648      225.6182 +/- 0.0821 ]
7 Radial distortion: [ -0.3881 +/- 0.0014      0.2493 +/- 0.0070 ]
8 ----Camera 2 ----
9 Focal length (pixels): [ 250.4736 +/- 0.1167      250.7549 +/- 0.1072 ]
10 Principal point (pixels):[ 136.8353 +/- 0.0720      222.6800 +/- 0.0956 ]
11 Radial distortion: [ -0.3789 +/- 0.0017      0.2256 +/- 0.0097 ]
12 ---- Extrinsics ----
13 Rotation of camera 2: [ 0.0087 +/- 0.0005      -0.3773 +/- 0.0004      0.0092 +/- ...
    0.0001 ]
14 Translation of camera 2 (millimeters):[ 23.4267 +/- 0.0142      0.0915 +/- 0.0058      5.3966 ...
    +/- 0.0451 ]
15 Rotation of camera 2 (ZYX) (degrees):
16 -0.6544  21.6152  -0.6322
17
18 mean reprojection error = 0.0930
19
20
21 ---- stereo parameters rotated + scaled ----
22
23 ---- MIDDLE + RIGHT ----
24 ---- Camera 1 ----
25 Focal length (pixels): [ 248.8735 +/- 0.1700      248.7253 +/- 0.1699 ]
26 Principal point (pixels):[ 139.1026 +/- 0.0667      225.7106 +/- 0.0844 ]
27 Radial distortion: [ -0.3854 +/- 0.0015      0.2422 +/- 0.0071 ]
28 ----Camera 2 ----
29 Focal length (pixels): [ 250.4314 +/- 0.1140      250.0559 +/- 0.1040 ]
30 Principal point (pixels):[ 144.0147 +/- 0.0711      225.5221 +/- 0.0929 ]
31 Radial distortion: [ -0.3690 +/- 0.0013      0.1826 +/- 0.0069 ]
32 ---- Extrinsics ----
33 Rotation of camera 2: [ 0.0086 +/- 0.0005      0.3821 +/- 0.0004      0.0106 +/- ...
    0.0001 ]
34 Translation of camera 2 (millimeters):[ -22.9393 +/- 0.0152      -0.0862 +/- 0.0060      5.0204 ...
    +/- 0.0441 ]
35 Rotation of camera 2 (ZYX) (degrees):
36 -0.5389  -21.8958  -0.3958
37
38 mean reprojection error = 0.0969

```