RPi- based passive 5-camera system for 3D face acquisition

Diederik van der Valk BSc

Abstract—A Raspberry Pi (RPi-) based 5-camera system was analysed for its potential to create high resolution - up to 0.2mm depth steps - 3D reconstructions of faces. A theoretical framework to determine the limits of the system was proposed, including the input depth resolution, a precision qualifier and maximum time synchronisation offsets between the cameras. The theoretical limits however could not (yet) be verified in practice. The found limits for this specific implementation were a max distance of 14cm from the cameras for a 0.2mm input depth resolution, and a theoretical maximum time offset of 0.016ms. A practical methodology was set up to generate a 3D face reconstruction starting from lens choices and adjustments, via calibration to reconstruction and comparison with other references. This resulted in a reconstruction at 0.4m where $> 90\%$ of the 3D reconstructed points were max 3mm off compared to a reference, set by a medical 3D face reconstruction device - the 3dMD. Various prospects for further improvement were found.

Index Terms—3D face reconstruction, acquisition, multiview imaging, passive cameras, raspberry pi, high resolution, synchronisation

1 Introduction

In many technological fields, such as the medical and security fields, 3D facial acquisitions are being used more frequent. Examples are the 3D facial detection to unlock an IPhone [1] or the usage of detailed pre- and postsurgical analysis of the face with e.g. the 3dMD [2]. These systems however appear to have either a low resolution 3D output, are large - therefore often immobile - expensive or must be used by trained professionals. Additionally according to Berretti et al [3] a need is present for such a system in order to provide enough 3D data that is needed for e.g. learning methods of face recognition.

In 2008 the department of Datamanagement and Biometrics (previously Services and CyberSecurity) already designed a first version of a passive multiview 5-camera system to acquire high resolution facial 3D point clouds [4]. A new hardware setup was since then implemented consisting of five Raspberry Pies with camera modules as visible in Figure 1. The usage of this passive multiview camera imaging technique could form a movable, relative low cost, high output resolution solution as described below. Also some extra advantages and problems to overcome are mentioned.

A passive camera system can be compared to the stereovision of human eyes. The depth perception of an object can be determined by combining 2 separate 2D images, calculated via a method called triangulation. For this triangulation the relationship between the cameras must be known: rotation; translation; and distortions. Using these known relationships each 3D position of an object can be calculated from all corresponding pixel pairs in the images of that object.

The usage of more than 2 cameras can improve the precision of the system and prevent missing information (holes) in the 3D model [5], both resulting in a higher output resolution. Okutomi [6] has shown theoretically that the precision increases by using more images as well as the effect of the basewidth (distance between cameras) on the precision as described later in section 2.2.1. The mentioned holes in 3D models can be caused by occlusion and prevented by using more cameras.

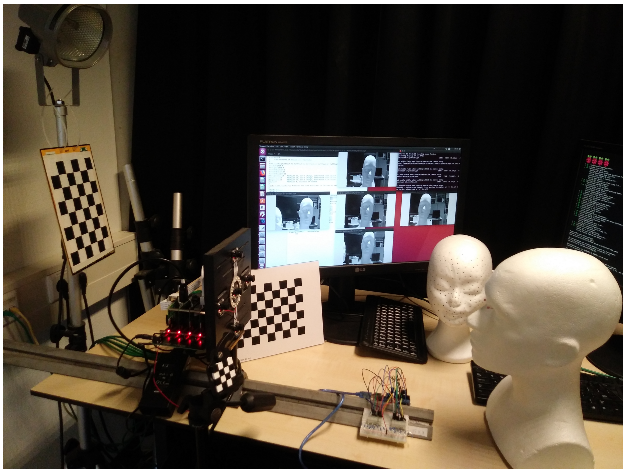

Fig. 1. RPi-based 5-cam setup including acquisition display.

The passive system has the advantage that it does not need a projected light like an active camera system, enabling it to work also outside in bright environments. Furthermore it captures the whole object simultaneous, in contrast to a Time of Flight (ToF) and Structure from Motion (SfM) systems. [7] This enables handling of non-static objects.

Further advantages of using passive systems is that the resolution can be increased by using a higher resolution camera. Another advantage is that depending on the used lenses the object in focus can be chosen to be far away or nearby.

Besides the advantages mentioned above for the use of a multiview passive 5-cam multiview system, there are also

- *D. van der Valk was with the Department of Datamanagement and Biometrics, University of Twente, Enschede, 7522NB NL*

some disadvantages and problems to overcome, such as the resolution-processing power trade-off and the resulting needed lens quality [7], [8]. The department has stated a need for 0.2mm depth resolution 3D facial reconstructions. The above is worked out in in the following question:

”Can a RPi-based passive 5 camera 3D facial acquisition system give a 3D point cloud in real time with an accuracy of 0.2 mm (depth)resolution?”

Further broken down in the following subquestions:

1) Which lenses, angles and corresponding adjustments in focusing are needed to get the (important parts of the) whole face in focus?
2) Can a simple calibration procedure be implemented to acquire the required accurate parameters?
3) Can a synchronisation timing be achieved to capture real time images?
4) Which theoretical resolution can be achieved?
5) Can the found theoretical resolution be approached in practice?
6) How accurate is the system compared to a reference 3D model, created by a medical approved 3D face reconstruction device - the 3dMD.

Further on in this article first a theoretical analysis is given to provide the background and theoretical limitations/requirements of the system. With the theory in mind the used implementation is described. Experiments were performed to verify the basic usage of the system, to determine the timing and resolution properties of the system and to compare the results to another 3D facial acquisition system, being the 3dMD.

2 Theory

In this section first the used mathematical system is described shortly. Secondly resolution calculations based on amongst others this model are given to indicate the theoretical resolutions. The third section deals with the choices of lenses and the resulting limitations on the system. In the fourth part the needed synchronisation timing is discussed.

2.1 Pin hole model

A pinhole camera model can be used to give the mathematical relationship between the 2D image coordinates projected on the image plane and the original world 3D coordinates. A summary is given in Appendix A.

Implementing the pin hole model in the simplified case of 2 cameras, both facing straight forward with only a horizontal translation results in the equations 1, 2, and 3.

$$X = \frac{Z \cdot (x_m - x_0)}{f \cdot m_x} \quad \text{and} \quad x_m = X/Z \cdot f \cdot m_x + x_0 \quad (1)$$

$$Y = \frac{Z \cdot (y_m - y_0)}{f \cdot m_y} \quad \text{and} \quad y_m = Y/Z \cdot f \cdot m_y + y_0 \quad (2)$$

$$Z = \frac{BW_x \cdot f \cdot m_x}{(x_m - x_r)} = \frac{BW_x \cdot f \cdot m_x}{d} \quad \text{and}$$

$$(x_m - x_r) = \frac{BW_x \cdot f \cdot m_x}{Z} \quad (3)$$

where:

- x_0 and y_0 are the camera centre in x and y orientation (set to be equal in this case) indicating the principal point,
- BW_x is the horizontal basewidth or offset between the two cameras, and
- $x_m - x_r$ is the disparity (d) or corresponding pixel distance between the two images.

Using this pin hole model the 3D location can be calculated from 2 2D images and vice versa.

2.2 Resolution

The input (i.e. measured) resolution and the output resolution are different values. Using e.g. interpolation and other techniques the measured resolution can be in-/ or decreased to any wanted value. The quality of the output resolution is of course also dependent on the quality of the measured resolution. The theoretical measurable lateral and depth resolution are worked out below[1].

For the theoretical input resolution the step size per pixel shift is wanted. By taking the derivative with regard to the pixels these lateral and depth resolutions can be determined.

The derivatives with respect to the pixel shift of Equations 1 and 2 can be seen in 4 and 5. The derivative with respect to the disparity of 3 and substituting Equation 3 gives Equation 6. From Equations 4 and 5 it can be seen that the in plane resolution is linear with the depth and independent of the lateral position (In case of Fish eye lenses for instance, the deformation causes a non-linearity breaking this independence). From Equation 6 it can be seen that the depth resolution is quadratic with the depth.

$$\frac{dX}{dx_m} = \frac{Z}{f \cdot m_x} \quad (4)$$

$$\frac{dY}{dy_m} = \frac{Z}{f \cdot m_y} \quad (5)$$

$$\frac{dZ}{dd} = \frac{BW_x \cdot f \cdot m_x}{d^2} = \frac{Z^2}{BW_x \cdot f \cdot m_x} \quad (6)$$

The parameters BW_x and m_x are set by respectively the distance the cameras are apart and the chosen image sensor. The last can be found in the datasheet. The needed focal length depends on the distance to the camera and the size of the 3D object.

2.2.1 *Multiple cameras*

Using multiple cameras can reduce the ambiguity of the correspondence problem and give more certainty. According to Okutomi a longer baseline gives a smaller variance in depth and a larger variation in intensity signal (more contrast) also decreases this variance as shown in Equation 7. Combining multiple baselines resulted in the following equation for the inverse of the variance, i.e. a measure of precision as shown in Equation 8. This equation is given for the situation of all cameras on one line with multiple Basewidth distances [6]:

$$Var(\hat{\zeta}_{r(i)}) = \frac{2\sigma_n^2}{B_i^2 F^2 a(x)} \quad (7)$$

[1] It is assumed the pixel size steps are small enough to differentiate all colours

where:

- $\hat{\zeta}_{r(i)}$ is the estimated inverse depth for Baseline i;
- σ_n is the image noise;
- B_i is the basewidth with index i;
- F is the focal length; and
- $a(x$ intensity signal.

$$\frac{1}{Var(\hat{\zeta}_{r(12\cdots n)})} = \sum_{i=1}^{n} \frac{1}{Var(\hat{\zeta}_{r(i)})} \tag{8}$$

where:

- $\hat{\zeta}_{r(12\cdots n)}$ is the estimated inverse depth of all baselines combined.

Adding more cameras would increase the right hand value of the equation, therefore resulting in a lower variance of the total variance, therefore improving the precision. Assuming this equation also holds for multiple directions and not only on one line as derived, with the variance equal for all stereo camera pairs due to equal distances, this results in the following improvement (reduction) of the standard deviation:

$$std(\hat{\zeta}_{r(12\cdots n)}) = \sqrt{Var(\hat{\zeta}_{r(12\cdots n)})} = \frac{std(\hat{\zeta}_{r(i)})}{\sqrt{\text{\#stereocamerapairs}}} \tag{9}$$

In order to acquire 3D images with a proper high resolution it is important that the measurement device is capable of achieving the high resolution (steps) needed, as described in Equations 4, 5, and 6. Various algorithms exist that can achieve sub-pixel resolution. In order to achieve this increased resolution, it is needed to have a sufficient accuracy and precision of the system. Accurate calibration of the system should ensure that the accuracy of the system is sufficient to provide the proper parameters, e.g. the pin hole model, the set focus of the lenses and the refractive properties of the lens. The calibration steps implemented are described later in 3.5 and 3.6. Furthermore the system must be precise enough such that the deviation of the output around the (estimated) true value should not exceed the chosen resolution. This precision is increased by using multiple cameras as described in Equation 9.

2.3 lens calculations

Using the geometry of the pin hole model the focal length can be calculated, given in Equation 10.

$$f = size_{im} \cdot \frac{Z}{size_{obj}} \tag{10}$$

Combining equations 6 and 10 gives the maximum value for f for a chosen input resolution as seen in equation 11. Maximum since the focal length f is incorporated in the distance Z. A smaller focal length is also possible, but would require the object to come closer to ensure the face is image filling.

$$f_{max} = \frac{dZ}{dd} \cdot BW_x \cdot m \cdot \frac{size_{im}}{size_{obj}} \tag{11}$$

This maximum focal length also results in a optimal distance for a given input resolution, given by equation 12. If the distance is smaller than the Z_{opt} not the whole object fits in the image, causing cropping. For a longer distance the object is not image filling, resulting in less pixels being used and therefore a lower resolution.

$$Z_{opt} = f_{max} \cdot \frac{size_{obj}}{size_{im}} \tag{12}$$

2.3.1 lens focus

If the object distance is slightly different then calculated (or the focus is not set properly), the object will not be be fully focused on the image plane and blurring will occur. The dimensions of this blur are called the Circle of Confusion (CoC). If this CoC is smaller than the pixel size, this blur will be barely noticable. The depth range within which the CoC is as small as deemed acceptable (i.e. smaller than the pixel size) is called the Depth of Field.

The CoC can be calculated for simple lenses by [9], [10]:

$$CoC = A\frac{|S_2 - S_1|}{S_2}\frac{f}{S_1 - f} \tag{13}$$

where

- CoC is the Circle of Confusion [m]
- A is the Aperture diameter [m]
- S_2 is the object distance [m]
- S_1 is the focus distance [m]
- f is the focal length [m]

The aperture diameter can be determined from the f-number and focal length. For static lenses both can be found in the datasheets [11]:

$$A = \frac{f}{\text{F-stop}} \tag{14}$$

Combining equations 13 and 14 and rewriting gives the following boundaries for the object distance:

$$\frac{-S_1}{\frac{CoC}{A}(1 - \frac{S1}{f}) - 1} \leq S_2 \leq \frac{-S_1}{\frac{CoC}{A}(\frac{S1}{f} - 1) - 1} \tag{15}$$

The focus of the system can be set manually or automatic. The system available at the department only has a manual option. The advantage of the manual focus is the limited need for (re)calibration. Automatic focus would require recalibration after each change in focus for the distortion coefficients. After every focus change, due to changed positioning of the lens, impurities in the lens could have shifted, etc.

2.4 Timing

For accurate reconstruction all the multiview images must be acquired at the same time. If there is a delay between the different cameras the corresponding points could have moved in time, resulting in an incorrect reconstruction. An example could be the mouth corner that could be lifted during capture. A general formula for determining the minimal fps needed has been proposed as shown in equation 16. [12]

$$\text{fps} > \frac{1}{K}\frac{v_{max}}{d_{min}} \tag{16}$$

where

- fps is the minimum frames per second,
- K is the tracking efficiency constant, indicating the distance a tracked point can travel in one time step, relative to the d_{min},
- v_{max} is the max speed with which tracked points can move, and
- d_{min} is the global minimum spacing between all tracked points over all frames.

3 The 5-cam system implementation

In this section the used passive 5-camera system is explained. In short the system consisted of 5 Raspberry Pies (RPis) with mounted camera modules. Acquisition software was running on the RPis and controlled from a master workstation. Pin triggering was used to synchronise the acquisition. The resulting multiview images of a checkerboard calibration object were processed offline on the workstation to get the calibration parameters. With the found calibration parameters and the multiview subject images the final reconstruction using a correlation based technique was created.

3.1 Hardware setup

The hardware setup is shown in Figure 1. It consisted of the following parts:

- 5 x interconnected Raspberry Pi 2 Model B's with camera modules [2] mounted on a RPi camera board in a '+' formation with each a distance of 7.5 cm apart.
- External light sources, such as LED strips, construction lights and ceiling LEDs
- 5 x M12 lenses[3] lenses as calculated below (IR filter)
- Tripod and camera fixation device.
- 1 x Ubuntu desktop Intel Core I7 2.93GHz x8 16GB RAM
- A display with a resolution of 1920x1080
- a checker board of 9x7 columns/rows of 20mm squares

3.1.1 lens choices based on head dimensions

As described in subsection 2.2 various lens and resolution calculations can be performed. Using equation 10 and 11 the maximum focal length and resulting optimal distance can be calculated. The parameters used for the calculations: output depth resolution (dZ/dd) $0.2mm$; basewidth (BW_x) $7.5cm$[4]; pixel scaling factor (m) $1/1.4\mu m$; image sensor size (height) $2738.4\mu m$; and max head size 21,6cm [13]. This resulted in a focal length and face distance of respectively $1.7mm$ and $0.14m$.

These above distances were deemed too short for practical applications, so a larger focal length lens and distance were chosen based on availability, at the expense of some depth resolution. This resulted in the following system characteristics: A focus distance of 0.4m, a focal length of 2.91mm, a vertical Field of View of $50°$, a theoretical depth input resolution of 1.0mm and a Depth of Field ranging from 0.35 to 0.46m.

3.2 Acquisition software setup

The acquisition software consisted on the RPi of a C++ compiled program with OpenCV4, raspicam[5] and wiringPi[6] libraries. This program was running on each RPi, continuously acquiring high resolution images, scaling them down and writing these low resolution images to file. After finishing writing the file was renamed to ensure that the output file is always fully written. In case of a GPIO edge detection via interrupt a full resolution image was written to another file.

After initialising this program the RPi created low-res images were continuously downloaded via SCP in a parallel desktop task. The low-res images were displayed on the screen in the same layout as the cameras (a '+') for visual feedback, as shown in Figure 1. User input on the desktop triggered the GPIO pin voltage change via SSH to the master RPi. After a set number of acquisition images all images were downloaded to a server.

3.3 Managing multiple devices and multiple tasks

For the software special attention had to be paid to the fact that multiple devices were used which had to communicate properly and fast enough with each other. Furthermore various functions that were used during acquisition had to share some resources (camera).

3.3.1 Device communication

As an image communication medium it was chosen to use file transfers rather than continuous data streams. The files were easier to implement and the code was interrupt based, potentially causing problems to streaming processes. The steps were as follows: a captured image was written to a temporary non-compressed .pgm file (compressing to e.g. a png file took too long); renamed to ensure completeness of the file; scp transfer on demand; and finished with another renaming of the file to again ensure completeness of the file during transfer.

Command communications were given via SSH[7] between the desktop and the master RPi or all RPis. For this proof of concept the secure connection was each command opened again (about 600ms). Keysharing prevented the need for passwords.

For the synchronisation of all images the capture trigger had to be as synchronised as possible and a GPIO pin was used to connect the RPis. From one RPi the voltage could be changed on the pin which could be used as an interrupt to synchronise the devices.

3.3.2 Shared resources

The RPis had one camera module which could only be accessed by one program at the same time and needed start-up time of a couple of seconds to function properly.

[2] http://www.produktinfo.conrad.com/datenblaetter/1200000-1299999/001214060-da-01-en-RASPBERRY_WEITWINKEL_KAMERA.pdf

[3] https://nl.rs-online.com/web/p/video-modules/1633903/

[4] In the current situation the middle camera is the reference, resulting in 7.5cm the max distance

[5] https://www.uco.es/investiga/grupos/ava/node/40

[6] http://wiringpi.com/download-and-install/

[7] Secure SHell protocol

Switching between programs was therefore not possible and one program was needed to perform all actions with the camera. This start-up also set the resolution and options of the camera and could not be changed during execution without re-initialising the camera.

Capturing an image by the RPi consisted of the following steps: grab, retrieve, process and write. The grab phase was the synchronisation sensitive part that grabbed the frame on a cue and as a result stored the sensor image data in a buffer. The retrieve phase converted the grabbed data to an OpenCV readable format on which pre-processing steps, such as scaling could be performed. The result could be written to file in the write phase.

The grab and retrieve phase both use one buffer, so the following 3 image acquisition options had to share these resources.

- The first function was a continuous high resolution acquisition that downscales the resolution in the processing phase. In order to prevent sharing the resource the continuous capturing loop could be paused using a signal trigger. An interrupt during his process (if not paused) could result in corrupt images and unknown behaviour. It was unknown what would happen in that case, since the buffer (for video use) has the option to store multiple frames in a buffer. It was seen that sometimes the program stopped and corrupt low res images were no issue, since they were directly overwritten with (noncorrupt) images.
- The second function was a pin triggered interrupt high resolution acquisition and had to get the captures (the grab phase) as synchronised as possible.
- The third function was a signal triggered interrupt to get (non-synchronised) high resolution images for visualisation purposes: i.e. checkerboard corner detection.

3.3.3 Parallelism and processing devices

The setup consisted of 5 RPis which could execute various functions in parallel and a desktop with better processing power. For this system it was chosen to only use the RPis for the capturing of the images and keep the desktop involved during development. This allowed faster processing of e.g. the corner detection and prevent one of the RPis to become relatively slower due to the fact that it had to run all the visualisation processes (untested how extensive they were for the processor). Below are some exceptions to this split. Depending on the usage the processing took place on the RPi or on the desktop.

For the preview a framerate of about 10Hz was needed for an acceptable user experience. Low resolution images were for this purpose acceptable. Sending high resolution images took too much time and direct downscaling after capturing on the RPi took an acceptable time of approx 34ms.

3.4 Adjustments and parameter selection

In order to acquire sharp images the lens had to be focused properly. In order to achieve this the checkerboard image was positioned at the calculated object distance (0.4m) and using the default RPi function raspicam with full preview options the focus was set. Using this visual feedback the lenses were adjusted to get a visually as sharp image as possible.

For the acquisition parameters the output was set to be grey scale, therefore skipping the white balance parameters. The brightness and contrast were statically set to the default program value. While the gain was set to 0 the exposure was increased to find visually the shortest shutter speed where for the given light circumstances the image noise did not visually increase more.

3.5 Calibration software setup

The calibration software on the workstation consisted of a C++ compiled program with the OpenCV3 library. It was a slightly adapted version of the calibration by Spreeuwers [4]. It loaded the saved images and checked whether the calibration object (checkerboard) could be detected in each image. In case not all corners were detected that whole image set of all 5 cameras was skipped. The OpenCV calibration functions were used with the following parameters: checkerboard dimensions of 7x9 squares of 20mm, and CALIB_CB_ADAPTIVE_THRESH. The output calibration parameters were saved to a file.

3.6 Calibration Pattern location protocol

In order to get enough rotations and translations: 3 depth, 3 horizontal and 3 vertical levels were chosen, resulting in 27 positions. For all the positions the camera is rotated using tilting and panning towards the middle camera. The angles are not fixed values, but approximately set between 30-45°.

3.7 Reconstruction software setup

The reconstruction software consisted on the workstation of a C++ compiled program with the OpenCV3 library. It was a slightly adapted version of the reconstruction used by Spreeuwers [4]. The code is uses the correlation between all the 5 camera's to determine the most likely 3D points.

3.8 3D output comparison

The freeware Cloudcompare can be used to compare 3D output files. [14] This software can use both meshes and point clouds, but in case of a combination the meshes can be converted to point clouds. The method used to compare 3D face reconstructions was based on Knoops et al. [15]

4 Experiments

4.1 Quality influences

In order to acquire theoretically high quality reconstructions, several topics were known to be of influence. Some experiments were performed to find the significance of these influences. The topics analysed in detail were lens quality, focus, calibration, and illumination.

4.1.1 Quality influence setup

For all aspects the default implementation as described in section 3 was used.

For the lens quality and focus analysis the visual inspection for each RPi was done using the default RPi program "raspicam" with full resolution (colour) preview. These previews were limited to the maximum resolution of the displays, being 1920x1080. The different lenses used were:

- a fisheye lens, f=1.67mm, F-stop=2.35, FOV(V)=89.5°;
- a cheap lens, f=4.0mm, F-stop=2.0, FOV(?)=76°;
- a RPi recommended lens, f=2.91, F-stop=2.72, FOV(V)=50°.

For the effect of the illumination several light setups were aimed at the head. Furthermore the shutter speed was adjusted from the default value to the maximum value. The light setups consisted of: the unchanged indoor situation with LED ceiling lights at an sideways angle of about 30°; alignment of ceiling lights to be frontal; an extra LED strip for frontal illumination; optionally a dark curtain to prevent light from other sources; and a construction light.

4.1.2 Quality influence measure

The lens quality combined with the ability to focus correctly was determined using visual inspection of the 2D images. In case of improper focus or lens quality the images would appear blurry and pixels blocks could be detected. A second element of the focus was the alignment of the cameras to the objects. The relative area of the image that showed the object was measured to determine the object coverage percentage. The maximal values of the relative height times the relative width were used for this object coverage percentage.

The calibration had as input 27 images per camera where overall the calibration object (checkerboard) had to cover an as large as possible section of the sensor area. The calibration object was large enough to ensure overlap between images. The boundaries were determined where the checkerboard was just visible in all cameras for a given distance. Per camera the shortest reachable distances to the edges of the image were determined over all images combined, resulting in another object coverage percentage.

The resulting stereo parameters of the calibration were verified using the known distances and angles of the setup. I.e. the translation and rotation between the cameras and the reference camera. The translations had to be approximately 75mm in only one direction and the maximum angle of a camera was measured to be $< 10°$, which resulted in diagonal values (r_{11}, r_{22}, r_{33}) of > 0.98 and $abs(r_{21}, r_{31}, ...) < 0.17$, using the standard rotation matrix calculations.

The effect of the illumination was analysed using visual inspection of the noise present in the reconstruction.

4.1.3 Quality influence results

Visual inspection of the first series of fisheye lenses showed that only 14% (50% of the height) object coverage percentage was reached at 0.4m. No further inspection of the quality of the lenses was performed. Visual inspection of the second set of lenses showed many pixels blocks (due to blurring) and a text of about textsize 16 at a distance of 70 nor 40 cm could not be read. The object coverage percentage was 24% (71% of the height) at a distance of 0.7m. Visual inspection of the third set of lenses allowed the user to see the text sharp. It resulted in approximately the same object coverage as the second set, but at a distance of 0.4m.

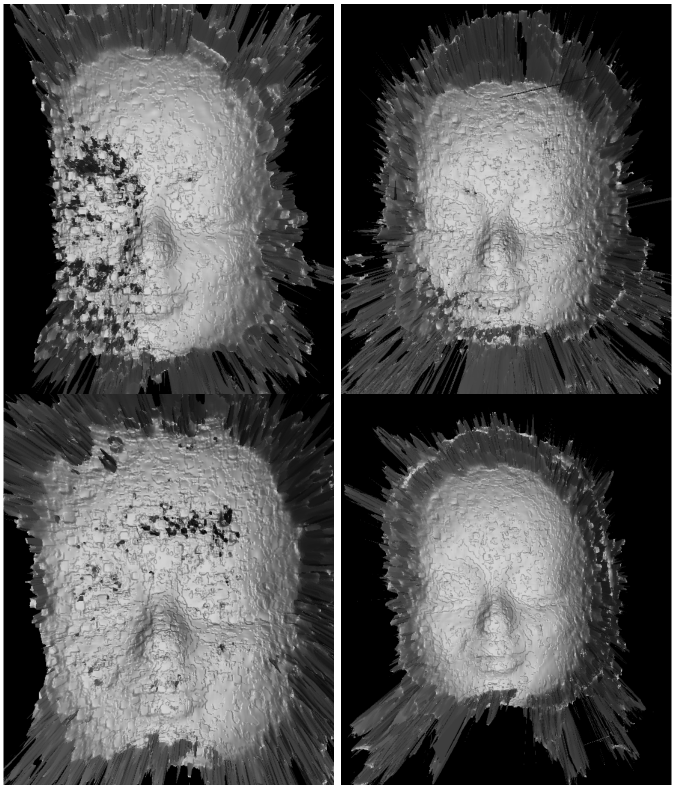

Fig. 2. Effects of light: subj05 top left only ceiling light at 30°; subj01 top right only ceiling light perpendicular; subj06 bottom left only LED strip; subj07 bottom right with a construction light.

A calibration object coverage of 47% at a distance of 40cm and a coverage of 73% at a distance of 60cm was found.

Calculations of the rotation and translation matrix suggested that the cameras on purpose set under an angle (with respect to the middle reference camera) had a max angle of $arccos(0.9947) = 5.9°$ and the cameras without set angle had a max angle of $arccos(0.99986) = 0.96°$. The needed translations were calibrated to be: 73.7; 74.47; -75.09; and -73.9mm. The translations in the other directions ranged from -1.21 to 0.72mm and the cameras placed under an angle had a z translation of 4.34 and 5.15mm.

In Figure 2 four reconstructions can be seen with different light sources. These sources were f.l.t.r.: subj05, the ceiling light coming from a side angle; subj01, ceiling light coming from the front; subj06, a frontal LED; and subj07, a construction light. It can be seen from comparing subj05 and subj01 that the ceiling light under an angle (in 2D showing light shadows) worsens the reconstruction significantly compared to frontal light. Comparing sub01 with subj06 shows more noise with only the LED strip than with using the ceiling light. Subj07 can be seen to contain the least noise and is captured using a construction light.

Subj06 and subj07 were also performed with higher shutter speed values than the default: all options resulted in visually the same image.

4.1.4 Quality influence discussion

The lens choice was seen to be of a significant influence on the quality and object coverage percentage of the 2D images. The last set of lenses gave visually acceptable results, but the result was not measured whether it reached the needed lateral resolution in the 2D images. An objective method is however still needed to verify the quality of the lens and the applied adjustments of the focus of the lens. The Depth of Field was calculated to be 0.35-0.46m with its focus at 0.4m. With the estimated needed face reconstruction depth of about 10cm there is not much margin.

It is recommended to verify this lateral resolution using 2D charts of known resolutions. Many charts exist for this purpose, e.g. the ISO12233 [16], [17]. With this resolution known it can be stated if the quality of the lenses is sufficient in theory or new lenses should be considered, keeping the low-cost principle of the system in mind. Also other quality controls can be verified, like the radiometric properties: is the illumination over the sensor evenly divided (often center is illuminated more), is it wavelength dependent, etc. [18] An option to get a measurement of the radiometric properties is to capture an object with known color/gray scaling. From the known object properties the expected range of pixel values can be determined. Comparing the expected pixel values with the measured values can give an indication of the radiometric similarity over the whole image, possibly allowing compensation of this radiometric distortion.

The object coverage percentage was performed on the original unedited 2D images, but in practice these images were undistorted, potentially changing the percentage. It is recommended to perform these analyses on the undistorted images. The object coverage percentage was found to be really low, mostly caused by the fact that the portrait heads were captured in a landscape image. It is recommended to rotate the setup 90° to capture in portrait mode, resulting in a higher number of pixels that can be used to represent the head (if moved closer).

For the calibration procedure the checkerboard was positioned at different distances around the focus distance. These distances were for this proof of concept study only approximated. It is recommended to specify these distances later based on the DoF once accurate verification of the resolution is implemented. It was seen that the calibration object coverage had a higher percentage at a larger distance. This increase was logical due to the fact that the cameras are for larger distances relatively closer to each other. In case the system is further specialised for faces at known distances it is recommended to adjust the angles of the cameras such that the objects (face and calibration checkerboard) are all focused in the middle of the image, keeping in mind the DoF boundaries. This adjustment allows a larger area of the image to be used for the calibration and reconstruction.

The found calibration parameters were in the order of magnitude as expected. The depth offset of the top and bottom cameras was explained by the rings positioned under the camera to achieve the wanted angle. The calibration parameters however were only analysed in detail once, so it is recommended to verify the consistency of these calibration parameters over multiple measurements and time/temperature. The possible found variation in calibration parameters (if present) could thereafter be used to check the influence on the reconstruction quality. This gives an indication if recalibration is needed and how often.

Regarding the quality of the calibration it is furthermore recommended to investigate the possibility of the reprojection error as an extra objective absolute qualifier of the calibration. On first sight this reprojection error only appears to be a value to be minimised and not a value which can be compared between different reconstructions (at different distances, etc.). An independent relation between a form of the reprojection error and for instance the quality of the reconstruction could provide an objective calibration measure. It is currently undefined how how accurate the calibration must be (for a certain wanted resolution).

The effect of proper illumination is very important as can be seen from Figure 2. The usage of the construction light (providing most illumination) gave the best results. It is recommended to implement an extra illumination source to the camera system, as more light suggests to provide better results. An option would be to connect a camera flash, which could be triggered by the master RPi. For this implementation the difference between a point source and a (homogeneously) spread out light source should be considered. A point source could result in reflective effects that are different per camera.

The use of the shutter speed parameter gave visually less effects than expected. As later seen in section2 about timing the shutter speed parameter behaved differently than the documentation suggested. It is therefore recommended to first verify the behaviour of this parameter before conclusions can be drawn.

4.2 Resolution

In order to verify the found theoretical resolution a 3D test object was 3D printed, captured, reconstructed and compared to the input model of the 3D test object. The dimensions of the created test object were first verified, and afterwards the reconstruction was analysed.

4.2.1 Resolution setup

For the 3D printing a leapfrog model ??? was used with simplify3D as software. The theoretical resolution of the printer was in depth 0.1mm and in width 0.3mm. Experience with this printer suggested that the lateral resolution in practice would not suffice for the verification of the theoretical resolution. This lateral resolution was further due to lack of time not not analysed. The 3D printed object consisted of a black-white plastic checkerboard with different depth steps between the squares, ideally ranging from 0.1mm till 1.5mm as seen in Figure 3. The size of each square was set to be 10x10mm with a base layer of 5mm. These "larger" dimensions were used to provide a stable underground and reduce the chance of imperfections due to starting and stopping of the printing per layer. The test object was placed in a holder and the default implementation as described before in 3 was used to capture the object. The 3D model was then compared to the input CAD model with CloudCompare to see the differences in resolution with respect to the theoretical resolution.

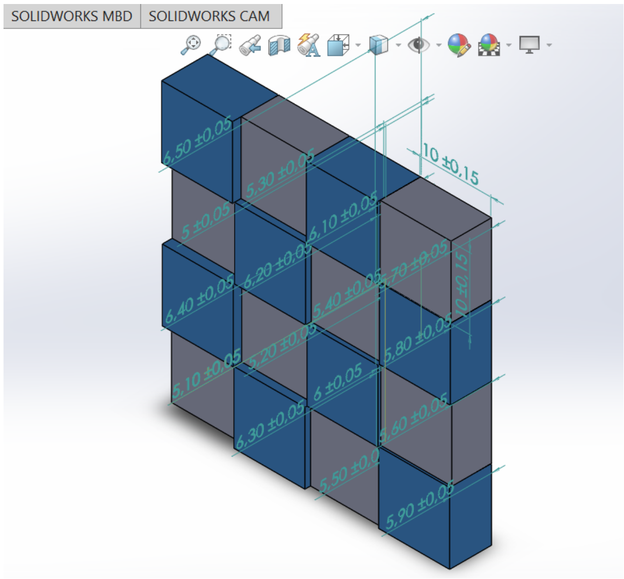

Fig. 3. Resolution test object. All dimensions in mm.

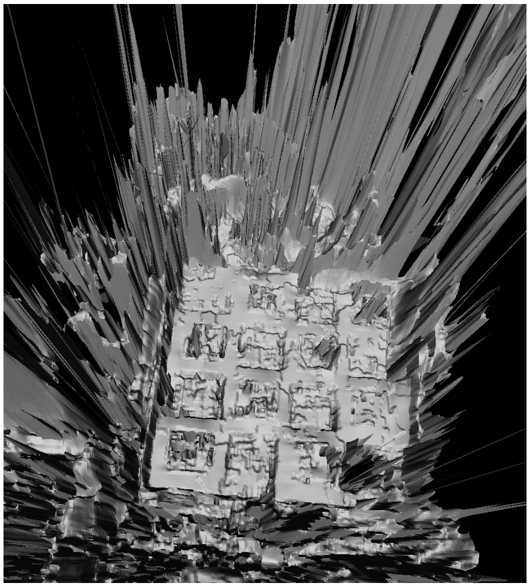

Fig. 4. Reconstruction of 3D testObject.

4.2.2 Resolution measure

The verification measurements of the 3D printed object was done using a digital calipers (Bahco 1150D). The measurements were performed on 3 positions per square (only the outer edge squares) to get an idea of the variation in sizes over the material.

The acquisition of the test object was done with the ceiling light under an angle from the side, a resolution of 1920x1080 and at a distance of about 0.4m.

4.2.3 Resolution results

The theoretical input resolution dimension as described before in 3.1.1 were 1.03mm in depth and 0.19mm in lateral directions.

The measurement of the 3D depth test object showed that the object was higher than the input CAD model: on average $0.25 \pm 0.14mm$ std (n=12) higher. The relative differences were smaller than the input model: avg. $-0.19 \pm 0.09mm$ std (n=6). The maximum average depth step measured was $1.3mm$.

The 3D reconstruction of the test object can be seen in Figure 4. No comparison with the 3d CAD model in CloudCompare was performed due to the visually low resolution in the previous steps.

4.2.4 Resolution discussion

The resolution of the 3D reconstructions depends on the lateral resolution in the 2D images. As described in section 4.1.4 it is recommended to first perform 2D resolution measurements to be able to provide founded conclusions on this topic. The visual quality of the 2D images however should not be 100% leading. It might be possible that lower resolution 2D images result in good reconstruction results, due to the used algorithm. Blurring of some pixels (merging their intensities) might for instance result in smoother results.

The method of creating the test object with this 3D printer was found to be too demanding for the small dimensions needed. It did show possibilities and limitations of the suggested method.

From Figure 4 it could be seen that the touching edges of the checkerboard squares were less noisy (smooth surface in the image). The middle of the squares was more noisy, possibly caused by a lack of texture there. Analysis of multiple reconstructions showed that the white squares had a better reconstruction than the black squares. Possibly caused by their capacity to reflect light better (more illumination) and the holder was also black, resulting in less contrast for the black squares.

The edges with less noise are quite wide, so it is recommended to repeat the experiment with a similar checkerboard method, but with smaller lateral dimensions to reduce the textureless areas and better illumination. Furthermore it is recommended to make the shape non-symmetric, to help find the orientation of the depth-steps. Another recommendation is to replace the black colour with another colour, both in contrast to the other colour and the surrounding of the object.

Another method of creating depth steps could be stacking a known number of A4-papers, but it must then be ensured that the object remains pressed flat together. Yet another method might be using a milling machine which could achieve the needed resolution. This could be done as one block with the same colour or by glueing separately milled blocks together. If properly done the glue layer is said to be thin enough for the given resolution. The result of course has to be verified.

4.3 Timing synchronisation

In order to acquire correct 3D reconstructions (of moving objects), all the image captures had to be as synchronised as possible. The current speed and synchronisation properties of the system were determined.

4.3.1 Timing setup

The initial ”benchmark” timing analysis of the (individual) RPi systems was done using an interrupt based C++ program. It logged the found events with a system clock time stamp. From these measurements it was deduced that this method of logging was in the order of 0.1ms. Secondly the synchronisation differences per communication method between the devices was looked at: pin and signal trigger. The max difference between the minimum and maximum pin trigger detection offset was about 9ms.

In order to get the overall acquisition duration also the time between interrupt and the moment the image was written was recorded using logging (consisting of the shutter time, readout of the sensor and writing to file). This was done for the low resolution (high resolution scaled down) and high resolution images.

The actual (synchronisation of the) image capture timing could be detected, using the previous results for time indications. This metric consisted of the start and finish times of the capture. A (relative) time indicating setup was made with an Arduino Uno and 2 arrays of 10 LEDs. One LED array counted with small steps per LED and the other array counted with 10x larger steps, triggered using the arduino interrupt timer. Based on the earlier results the LED times were respectively 1ms and 10ms per array step. Only one LED was turned on per array at a time to detect the used shuttertime. Due to the limited availability of output pins on the Arduino all 10 LEDs for the small steps were used, but only 8 LEDs for the big step (the RX and TX pins were not used). The last LED was turned on permanently to help locate the LED Arrays in the image. During the acquisition of the images the system was placed in a holder to ensure the LEDs were located at the same location, allowing easy readout as shown in Figure 5.

4.3.2 Timing measure

The time stamps of the logging had a resolution of 1 ns. For the internal tests the system clock was used as reference. For the communication between devices multiple system clocks were however involved. In order to compensate for the potential bias in system clock the measured value was the time relative to the average measured event time. If the system clock was consistently biased the offset would be shown in this measurement.

The readout of the LED arrays was done based on the position and number of LEDs that were captured to be on. The LED speed was verified in code using the function micros() and in practice using a SLR camera with appropriate shutterspeed. During a capture (time length dependent on the shutter speed) the image sensor was exposed to all the LEDs that had been turned on in that period, resulting in an image showing at least one or more LEDs. It was a relative scale, where the LED most left of a series of connected turned on LEDs was the start time and the number of connected LEDs an indication of the duration. Due to the setup it is possible that this set of LEDs was looped over the end to the beginning.

Initial logging analysis as discussed before showed a max trigger difference of 9ms, so the chosen time steps were 1ms and 10ms, resulting in a time resolution of 1ms. For duration times lower than 10ms the resolution was 1ms, being one LED on the small step array. For duration times $>= 10ms$ the resolution was 10ms, because all lower resolution indicating LEDs were turned on, showing no start and stop any more.

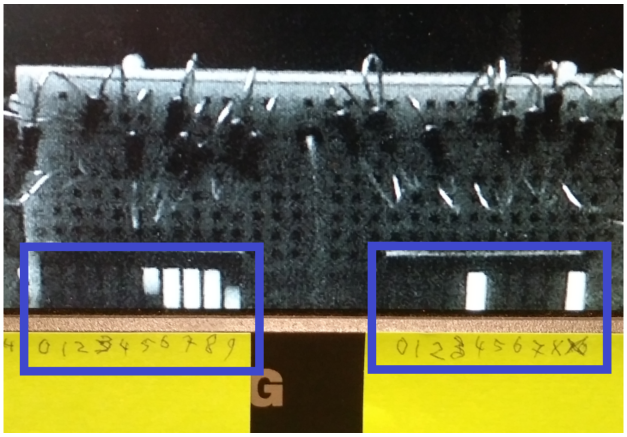

Fig. 5. Time synchronisation readout from screen using annotation, resulting in a starttime of 45ms and a stoptime of 49ms. On the left LEDs indicated by 5 and 9 the rolling shutter behaviour can be seen.

4.3.3 Timing results

The theoretical maximum fps difference allowed was given by equation 16 from section 2.4. The K is 1 for a theoretical perfect system; a theoretical v_{max} of 12.6 m/s was found [19]; and d_{min} is the needed depth resolution of 0.2mm. This resulted in 63000 fps, or a max time difference of $0.016ms$.

The results of the initial speed ”benchmark” are given in the Appendix in Table 1. The trigger synchronisation offset is given in the Appendix in Table 2. The capture time between interrupt and having written the low resolution images took between 19 and 106 ms with an average of 33ms. For the high resolution this was between 24 and 80 ms with an average of 47ms.

The LED array acquisition showed an exposure time (shutter speed) of about 30ms and a max time difference ranging from 10 to 60 ms between the RPis (an exposure time of 6,6ms was given by the documentation). After changing the shutterspeed to the lowest value possible, the experiment was repeated. This gave an exposure time of about 4-5ms and a max time difference ranging from 22 to 66ms when looking at all the RPis.

Comparing the individual RPi start moments to each average start moment showed all averages to be around 0 offset, with a minimum offset of -42.4 and a maximum offset of 38.0ms.

4.3.4 Timing discussion

Comparing the theoretical fps (0.016ms) with for example the found max pin synchronisation difference (8.6ms) showed a factor 500 difference and is therefore not deemed reachable with the current RPi system. Looking further the shutter speed of 30ms (duration) is even more than 1000 times longer than the theoretical fps (offset). It is suggested that also with other systems this theoretical value won’t be reachable with a reasonable amount of effort. This theoretical value was made for the extreme case of maximum movement. In practice the head will probably move a lot slower if at all (measurable). It is recommended to find the (significance of the) impact of a synchronisation offset for non-extreme cases. For this experiment a distinction could be made between within object movements (muscle contractions) and object movements (static or dynamic scenes). An example experiment would be capturing a static object and capturing the same object moving at a known speed, resulting in a relationship between speed and reconstruction error. For now the system is used to capture static heads, therefore the priority of the unknown effect of the synchronisation offset is lower than the other aspects (e.g. the resolution).

From Table 1 it was seen that pin interrupt was the fastest trigger method and therefore chosen. The user input method of starting this more synchronised pin trigger was performed via SSH to the master RPi which took max 523 ms (assuming the master-RPi network communication is equal to master-master via network, otherwise max 1058). The max time between interrupt and the image being written was 80ms, which is lower than this SSH time. Assuming no parallel SSH commands can be processed the max acquisition speed from the desktop is therefore 523ms (2Hz).

As seen in Table 2 the timing with regard to the pin VS signal interrupt is confirmed. The max SSH trigger synchronisation time difference is significantly larger than the max pin trigger time difference as was expected. The comparison of the individual RPi event times with respect to the average event time showed that the 0 offset was within the 95% confidence interval (2x the std) for m1 and s2. The system clocks of s3, s4 and s5 suggested to have a significant bias up to a couple of ms. Taking the minimum and maximum offset values over multiple measurements gave a max overal seen trigger synchronisation offset of 8.5ms. After bias compensation of s4 this would be 6.7ms.

The time between an interrupt and the image actually having been written to file had a variation of about a factor 5 for the low resolution and a factor 4 for the high resolution images. The lowest time taken (19ms) however is lower than the found shutterspeed time (approx 30), suggesting measurement errors. It was however not double checked if the same shutterspeed was used in both experiments. It is therefore recommended to perform extra analysis into the consistency of the timing of the software, shutterspeed and logging. During execution around the interrupts the logging could be influenced by optimisation caching options of the software. E.g. if during the logging test first all time stamps were generated(fast) and only written afterwards to file(slow) the logging time is too fast, resulting in a bigger time resolution step.

Figure 5 showed an example measurement of a capture. It can be seen that the LEDs 5 and 9 are illuminated only half, showing a rolling shutter behaviour. The rolling shutter is also a method which can influence the timing of the system. It is recommended if the timing is found to be in need of extreme synchronisation to replace the sensor with a non rolling shutter behaviour.

The synchronisation time differences found with the LED setup ranged up to 66ms on the setup of max 80ms which looped over. It can therefore not be said with certainty if the difference was 66ms or (80-66=)14ms. It is recommended to first check all timing steps to determine the cause of the large time differences between the RPis (suggested by the time between interrupt and the image being written).

4.4 Test reconstruction

In order to verify the capacity of the system to provide correct reconstructions of faces a reconstruction was made with different systems and compared.

4.4.1 Reconstruction setup

A reconstruction of 2 polyfoam heads was made with different settings. The heads are further referenced to as Ingrid (female) and Henk (male). Ingrid had many pins distributed over her face of about 0.6mm high and 1.6mm wide for feature extraction (in other applications). Henk had 2 pins in the eye corners, extruding 12mm with a diameter of 4mm and 32mm apart. The calibration was performed once. The used parameters were the following: a construction light for extra illumination, the shutter speed parameter set to the system value 20 or 100, and a distance of 0.4m. Using the 3dMD provided by the department RAM within the University of Twente a scan was made half a year earlier of both static heads and with the ArtecEva 4 years earlier. The wrl files of the reconstruction were converted online to stl to be loaded in CloudCompare [20]. The RPi system reconstruction only reconstructed the front part of the face, so the suggested method by Knoops of cropping the face with the tragus as orientation points was not possible. Instead the cropping planes were visually applied parallel to the frontal camera, removing the back of the head. The 3D models were aligned visually and if needed sampling to 100.000 points was done to allow point picking. The final alignment was done with fine registration (ICP) with as settings an overlap of 90% and the scales were allowed to be scaled if comparing different devices.

The (signed) distances were computed using the cloud/mesh setting in order to prevent effects of choosing the sample size manually. The histogram of the calculated distances was exported as csv for further analysis.

The comparisons made were between: 2 RPi system reconstructions of Ingrid; between the RPi reconstructed Henk and the 3dMD reconstruction; and between the 3dMD reconstruction and the ArtecEva reconstruction of Henk. Furthermore a reconstruction of a real head was made to see the visual result.

4.4.2 Reconstruction measure

The scaling was measured by selecting marker points in the point cloud and comparing the found distances with measurements of the same distances on the real object. The final histogram of point to point distance output (if properly scaled) showed the percentage of pixel pairs found per distance.

4.4.3 Reconstruction results

The 3dMD files (.obj) were about 14 MB in size. The RPi reconstructed files were about 42-46MB for the polyfoam heads and 56MB for a real head including hair, etc. Visual inspection in CloudCompare of the wireframe showed that the RPi reconstructed files had smaller resolution steps.

The scaling verification showed for the RPi reconstructed Ingrid that 2 points were 32mm apart and the corresponding points were measured to be 29mm. For the RPi and 3dMD reconstructed Henk after scaling to one other the points were in both model and real time 3cm apart.

Aligning the RPi and 3dMD model required a scaling of 0.93. RPi with RPi was not scaled. 3dMD comparison with

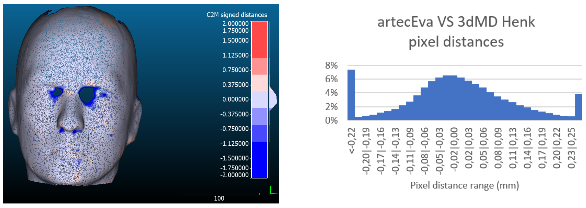

Fig. 6. Reconstruction comparison of Henk with artecEva and 3dMD. Scale limited to -2 till +2mm.

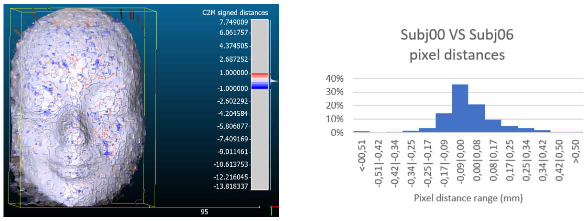

Fig. 7. Comparison of 2 RPi reconstructions of Ingrid.

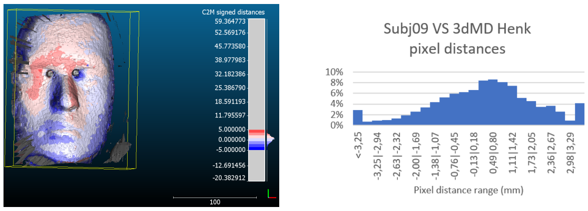

Fig. 8. Comparison of RPi and 3dMD reconstruction of Henk.

ArtecEva required scaling, but it was not recorded which scaling.

The reconstruction comparisons with pixel distance histograms can be seen in Figures 6, 7, and 8. The colour scaling between the images was not equal.

A reconstruction of the authors head can be seen in Figure 9. Around the chin the chosen depth layers can be seen. Point picking along a line crossing one such depth layer boundary showed a depth layer difference of approximately 0.5mm. The steps along the line however were smaller: the lateral steps were on average 0.22mm (n=18) and the depth steps at the edge on average 0.17mm (n=3).

4.4.4 *Reconstruction discussion*

For the scaling measurement specific points had to be selected in the model to compare with the real life objects. The pins on Ingrid were found to be too small to be detected. Also on the 3dMD model the pins could not always be found in the 3D model. They were visual in the texture map. The positions had to be estimated and were chosen to be the inner eye corner where the curvature was strongest. For Henk the pins in the inner eye corners were used. In the RPi reconstruction these points were seen as cylinders sticking out of the head (see Figure 8), probably due to occlusion behind the pin. The middle of the pins was taken as a reference. It is recommended to use at least 2 small (1mm is suggested) contrast pins that stick out far enough to be detected (5mm is suggested). This enables better verification of the scale. The scaling so far appeared to be in the right order of magnitude, but it is recommended to be verified in detail.

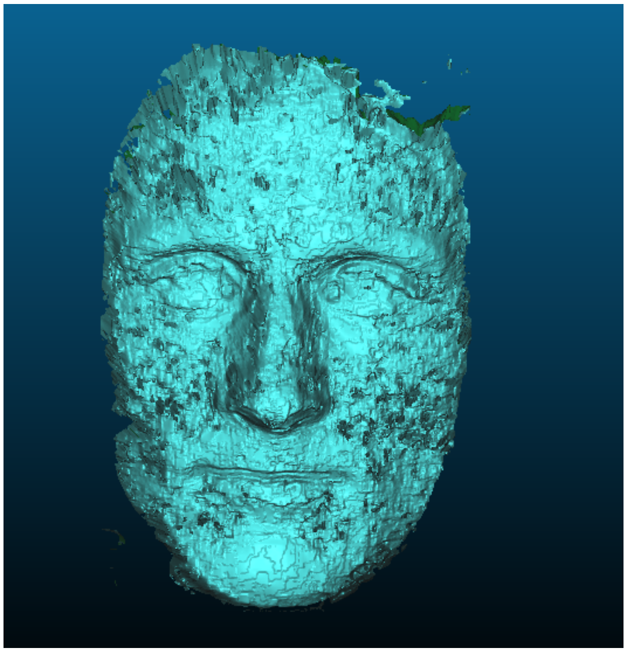

Fig. 9. Reconstruction of authors head

It is recommended to analyse the method of the scaling of the objects by CloudCompare (during alignment): whether it is necessity and how it is implementated. Both the RPi and 3dMD reconstructions are calibrated with known dimensions, so they should be equal, but still got recommended to be scaled. In case the centre of the objects is different it might be possible that the scaling deforms the objects, which is a not wanted effect.

The alignment between the 3dMD and ArtecEva models was accidentally performed without cropping, but with an overlap setting of 80%. To compensate for it the max distances were reduced, excluding large point to point distances from the measurements. This can be seen in Figure 6. The differences larger than —2.0mm— were removed from the model, as for instance can be seen in the eyes. The histogram is therefore only based on a max distance of 2.0mm, not all datapoints.

The alignment option to set an overlap of 90% or lower gave visually good alignment results, but due to the separate cropping of the objects it is undefined which parts are (not) overlapping. It is recommended to crop the heads once more after first alignment and merging of the centres was performed. This ensures that overall the same points are selected and the overlap can be increased to 100%, ensuring that all points are used for the alignment and distance calculations.

Comparing the output of the commercial reconstruction tools showed that the distances are really close with most of the point to point distances ranging from -0.22mm to +0.25mm as seen in Figure 6. In order to see the consistency of the RPi reconstruction method 2 reconstructions were compared as seen in Figure 7, showing that most distances are in an approximately similar range of -0.50 to 0.50mm. Comparing the RPi reconstruction with the 3dMD however as shown in Figure 8 showed that the distances ranged from about -3mm to +3mm.

The point picking along a visual depth level of the chin of Figure 9 showed a depth level difference of 0.5mm, which was the set depth level. The steps to jump from these ”visual” depth levels to one another were smaller than the set resolution. This suggests that besides the theoretical input resolution and the set to be calculated output resolution another even more detailed visualisation resolution was created. It is recommended to see if this visualisation resolution can be reduced to the set calculation output resolution. This could result in a lateral reduction of steps by 5 and a depth reduction of a factor 3, resulting in $5 \times 5 \times 3 = 75$ times smaller output, which in return could also speed the reconstruction up.

5 Conclusion

3D facial reconstruction has been developed over the years and is being used in more applications. These acquisition devices are however often only in limited situations applicable, due to the size, price or need for trained staff members. A Raspberry Pi based system has been proposed as well as analysis of various quality influencing aspects of the system. Preliminary results showed an output which had a point to point difference accuracy of about -3 to +3mm compared to the 3dMD.

The suggested ”RPi model 2 B”-based passive 5-camera acquisition system was designed to acquire 1920x1080 2D images on demand with a used defined trigger of currently max 2Hz due to slow network communications. The 2D images could on a workstation be converted to low resolution 3D images in less than 2 seconds and high resolution images in 7 minutes with an output depth resolution of 0.5mm. Various aspects regarding lens quality and object focus were inspected in theory and applied in practice to get the final result. The angles of the lenses were implemented such a way that the image would be properly acquired by all multiview images in about the same position to maximise the screen filling, while keeping the Depth of Field in mind. A simple calibration procedure was implemented which gave visually good reconstruction results. The synchronisation in timing of the different cameras has been analysed on various aspects, indicating possible improvements (in measurements). A possible a-synchronised timing is not relevant for static object. Theoretical calculations showed that an input depth resolution of 0.2mm could be reached, but only at a distance of about 12cm from the camera. The theoretical resolution for a more applicable distance of 0.4m was determined and practical implementation showed that this could not be approached.

This proof of concept has provided a baseline for the implementation of a passive RPi based multicamera system. Recommendations have been given to further improve this baseline.

Acknowledgments

The author would like to thank Dr.ir. L.J. Spreeuwers for the periodic meetings and useful input as well as Dr. B. Sirmaçek for her ideas. Especially many thanks to ing. G.J. Laanstra for all the support on the practical aspects.

References

[1] “About face id advanced technology,” Nov 2018. [Online]. Available: https://support.apple.com/en-us/HT208108

[2] C. Hong, K. Choi, Y. Kachroo, T. Kwon, A. Nguyen, R. McComb, and W. Moon, “Evaluation of the 3dMDface system as a tool for soft tissue analysis,” *Orthodontics and Craniofacial Research*, vol. 20, no. Suppl 1, pp. 119–124, 2017.

[3] S. Berretti, M. Daoudi, P. Turaga, and A. Basu, “Representation, Analysis, and Recognition of 3D Humans,” *ACM Transactions on Multimedia Computing, Communications, and Applications*, vol. 14, no. 1s, pp. 1–36, 2018.

[4] L. J. Spreeuwers, “Multi-view passive 3D face acquisition device,” {*Biosig*} *2008*, no. February, pp. 13–24, 2008.

[5] R. Fergus, “Lecture 6: Multi-view stereo & structure from motion.” [Online]. Available: https://cs.nyu.edu/~fergus/teaching/vision_2012/6_Multiview_SfM.pdf

[6] M. Okutomi and T. Kanade, “Okutomi_M_1993_1,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 15, no. 4, pp. 253–263, 1993. [Online]. Available: https://www.ri.cmu.edu/pub_files/pub2/okutomi_m_1993_1/okutomi_m_1993_1.pdf

[7] B. Kisa and M. Gelautz, *Advances in Embedded Computer Vision*. Springer, 2014. [Online]. Available: http://link.springer.com/10.1007/978-3-319-09387-1

[8] M. Aboali, N. Abd Manap, A. Majid Darsono, and Z. Mohd Yusof, “Review on three dimensional (3-d) acquisition and range imaging techniques,” *International Journal of Applied Engineering Research*, vol. 12, pp. 2409–2421, 06 2017.

[9] “Chapter 23. depth of field: A survey of techniques,” 2004. [Online]. Available: https://developer.nvidia.com/sites/all/modules/custom/gpugems/books/GPUGems/gpugems_ch23.html

[10] “Circle of confusion,” 2019. [Online]. Available: https://en.wikipedia.org/wiki/Circle_of_confusion

[11] “F-number,” 2019. [Online]. Available: https://en.wikipedia.org/wiki/F-number

[12] M.-H. Song and R. I. Godøy, “How Fast Is Your Body Motion? Determining a Sufficient Frame Rate for an Optical Motion Tracking System Using Passive Markers,” *PLOS ONE*, vol. 11, no. 3, p. e0150993, mar 2016. [Online]. Available: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4788418/

[13] “Human head - wikipedia,” 2019. [Online]. Available: https://en.wikipedia.org/wiki/Human_head#/media/File:AvgHeadSizes.png

[14] CloudCompare, “How to compare two 3d entities.” [Online]. Available: https://www.cloudcompare.org/doc/wiki/index.php?title=How_to_compare_two_3D_entities

[15] P. G. Knoops, C. A. Beaumont, A. Borghi, N. Rodriguez-Florez, R. W. Breakey, W. Rodgers, F. Angullia, N. U. Jeelani, S. Schievano, and D. J. Dunaway, “Comparison of three-dimensional scanner systems for craniomaxillofacial imaging,” *Journal of Plastic, Reconstructive and Aesthetic Surgery*, vol. 70, no. 4, pp. 441–449, 2017. [Online]. Available: http://dx.doi.org/10.1016/j.bjps.2016.12.015

[16] ISO, “Iso 12233:2017 - photography – electronic still picture imaging – resolution and spatial frequency responses.” [Online]. Available: https://www.iso.org/standard/71696.html

[17] S. Westin, “Iso 12233 test chart.” [Online]. Available: https://stephen-westin.com/misc/res-chart.html

[18] S. optics, “Quality criteria of lenses.” [Online]. Available: https://www.schneideroptics.com/pdfs/whitepapers/quality_criteria_of_lenses.pdf

[19] A. Takshi, “How fast a muscle can contract?” 04 2003. [Online]. Available: http://jick.net/p438/phys438/Reports/MuscleContractionSpeed.PDF

[20] makexyz, “Convert wrl files to stl files—makexyz.com.” [Online]. Available: https://www.makexyz.com/convert/wrl-to-stl

[21] N. Pears, Y. Liu, and P. Bunting, *3D imaging, analysis and applications*. London: Springer London, 2014, vol. 9781447140. [Online]. Available: https://link.springer.com/content/pdf/10.1007%2F978-1-4471-4063-4.pdf

[22] K. Kanatani, *Guide to 3D Vision Computation : Geometric Analysis and Implementation*. Springer, 2016.

[23] R. Szeliski, *Computer Vision Algorithms and Applications*. Springer-Verlag London, 2011.

Appendix A
Pin hole model summary

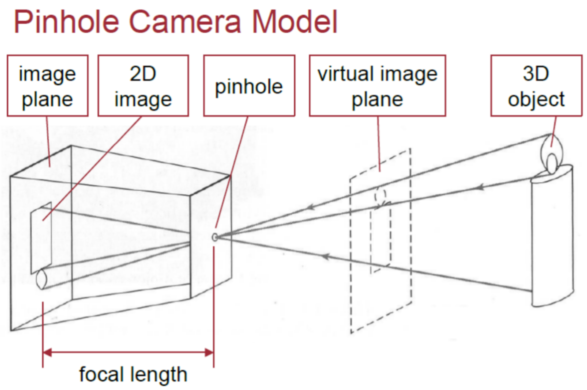

Fig. 10. pinhole camera model, taken from https://www.cs.tau.ac.il/~dcor/Graphics/cg-slides/CameraProjections.pdf

The model can be used to calculate the reconstruction and is used as a basis for many algorithms. As seen in Figure 10 the rays from the 3D object pass through the pinhole and are projected (upside down and left-right switched) on the 2D image plane. The virtual image plane is the theoretical (non inverted) image of which the 2D coordinates will be calculated.

For further reading; the mathematical model is described in detail in amongst others [7], [21], [22], [23]. The resulting homogeneous model can be seen in equation 17 or 18, where the first one is the short notation.

$$\lambda\vec{x} = K[R|\vec{t}]\vec{X} \tag{17}$$

with:

- λ a scaling factor
- $\vec{x}$ the image coordinates (in pixels)
- K the intrinsic parameters,
- $[R|\vec{t}]$ the extrinsic parameters consisting of a rotation and translation matrix, and
- $\vec{X}$ the 3D world coordinates.

$$\lambda \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} f\cdot m_x & s & x_0 \\ 0 & f\cdot m_y & y_0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_x \\ r_{21} & r_{22} & r_{23} & t_y \\ r_{31} & r_{32} & r_{33} & t_z \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \tag{18}$$

where

- λ is a scaling factor [8],
- x and y are the pixel coordinates [px],
- f is the focal length [m],
- m_x and m_y are the Pixels per unit distance scaling [px/m],
- s is the skew correction if the x and y direction are not orthogonal [px],
- x_0 and y_0 are the image coordinates of the principal point per camera, for perfect lenses this if often the middle of the image sensor[9] [px],
- r_{11} till r_{33} are the values of the rotation matrix from world to pixel coordinates [px/m],
- t_x, t_y and t_z form the translation matrix from world to pixel origin [px/m], and
- X, Y and Z the world coordinates [m].

Appendix B
Effect of distortion on a reconstruction

In order to stress the importance of finding correct distortion parameters a reconstruction with and without these parameters was performed and shown in Figure 11.

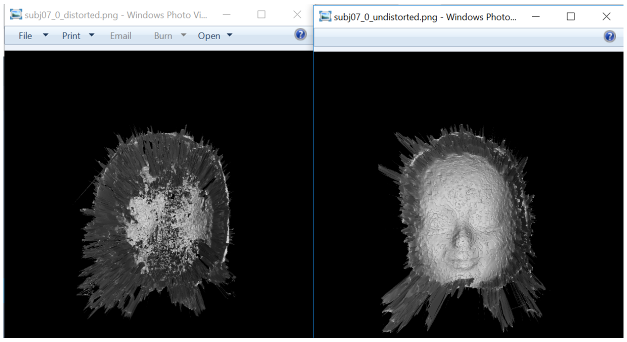
subj07_0_distorted.png - Windows Photo Vi...
File
Print
Email
Burn
Open
subj07_0_undistorted.png - Windows Photo...

Fig. 11. Reconstruction with and without distortion effects

Appendix C
Timing analysis tables

C++ event	time RPi (ms)	freq RPi	time wks (ms)
addition loop	0.00026-0.0004	2.5-3.9MHz	0.000056
Logging time	0.08-0.10	12.5-9.9 kHz	0.02
raise	0.10-0.38	10-2.6 kHz	0.023-0.035
shell signal	17.1-43.9	59-22 Hz	3.9-10.9
SSH shell signal	904-1058	1.1-0.9Hz	362-523
pin interrupt	0.18-4.2*	5.6-0.24kHz	

TABLE 1
Minimal and maximal internal execution times (benchmark) found per activity on the RPis and the workstation (wks). *sometimes a new pin trigger was already given while the previous trigger interrupt hadn't been processed yet

Event	n	Min	Avg	Max	std.s
SSH max sync diff	27	170.6	352.9	630.7	129.7
Pin max sync diff	28	2.2	5.8	8.6	1.3
m1 SSH signal offset	27	-306.9	-42.5	339.4	175.9
s2 SSH signal offset	18	-258.1	-66.4	151.9	119.1
s3 SSH signal offset	27	-186.7	-19.4	244.2	103.1
s4 SSH signal offset	27	-195.1	48.3	386.3	120.1
s5 SSH signal offset	27	-247.8	57.8	296.5	131.6
m1 pin offset	28	-0.263	0.529	3.643	0.742
s2 pin offset	3	0.013	0.291	0.815	0.454
s3 pin offset	28	0.207	2.104	2.778	0.611
s4 pin offset	28	-4.944	-3.415	-1.354	0.789
s5 pin offset	28	-0.086	0.851	1.666	0.355

TABLE 2
Synchronisation time differences, all in ms of one measurement. Determined by system clock time logging during capturing of a static object, using either the pin or SSH signal trigger. RPI s2 stopped during execution, but measurement was continued.

[8] For any homogeneous image point scaled to $\lambda[x, y, 1]^T$, the scale λ is equal to the imaged points depth in the camera centered frame ($\lambda = Z$).

[9] the image coordinates start in a corner of the image, so this is a offset to the principle point