Creating 3D images for facial recognition using the RealSense SR300

Nahuel Manterola
Electrical Engineering, SCS group, University of Twente
Email: n.i.manterola@student.utwente.nl
Supervisor: Dr. Ir. L.J. Spreeuwers

***Abstract*—3D face recognition is becoming a viable alternative to traditional 2D face recognition. It is more robust in terms of changed lighting and rotation of the face. The RealSense SR300 is a depth camera which makes use of structured light. It acquires the depth data using an infrared projector and camera. This paper presents a method to create a 3D point cloud of a face using this camera.**

Several settings in the software environment are treated to obtain an image with the best quality. Measurements are done on the depth resolution and the per-pixel depth deviation. A Face verification algorithm is used to assess the quality of the generated point cloud and determine a relation between False Accept Rate and Verification Rate using two depth images of 22 subjects. The effects of lighting conditions and the distance between a subject and the camera are tested.

The depth resolution is very dependent on the distance between subject and camera and is optimal at 25 cm. The per-pixel depth deviation is deemed too small to have a large impact at close distances. At larger distances averaging of multiple frames is suggested. Using the verification algorithm, a VR of 0.991 is found at FAR=0. At FAR=0.05 the VR = 1. The camera performs very poorly in outside conditions but can achieve acceptable results when increasing the projector brightness. Inside, the best results are achieved with a lower projector brightness. The maximum distance between the camera and the subject is 40 cm. Beyond this length, the face model becomes unreliable in terms of verification.

This method enables the RealSense SR300 to be used as a reliable, cheap face recognition system for any indoor applications. As this camera comes built into various laptops and tablets, it can be used for various security applications. Several potential recommendations for further research are given to increase the performance of the camera.

I. Introduction

3D face recognition is gaining attention as a reliable way of identification. It offers more robustness against factors like varying lightning and head rotation than traditional 2D facial recognition. Current methods to create the 3D facial scans are expensive or require the camera to be moved around the subject. This project aims to condition a relatively cheap depth camera, the RealSense SR300 (Figure 1), to create images of sufficient quality to use in a verification algorithm [1][2]. The RealSense SR300 uses structured light, projected by an infrared projector and captured by an infrared camera to determine depth. This paper will give insight in the measurements and experiments done to find optimal settings, conditions and limitations of this system.

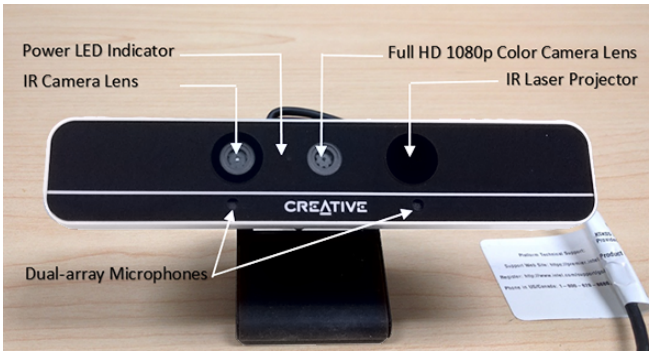


Fig. 1: RealSense SR300 camera [3]

II. Theory and Related Research

A. Structured Light

The depth system of the RealSense SR300 makes use of structured light. This means it projects patterns on the subject and calculates depth based on the deformation of these patterns by the subject (Figure 2). Structured light has an advantage compared to traditional stereo 3D, as it can detect depth on surfaces with little to no texture. However, range cameras using structured light interfere with each other when they use the same projection wavelength, which means they don't work together very well. There are several methods of structured light which have their advantages and disadvantages[4]. The RealSense SR300 uses sequential projection binary encoding. Multiple vertical line patterns are projected in sequence, as shown in Figure 5. In these patterns, the lighter lines represent a 1 and the dark space between the lines a 0. In this manner, every horizontal position gets a binary value assigned to it by the multiple frames. An example of this is shown in Figure 3. The camera captures this pattern, which is deformed by the subject. The projected binary values are detected in the images made by the camera. The angle of projection of each binary value is known, and the angle of each binary value in the images of the camera can also be determined. Figure 4 shows how geometric triangulation can determine the intersection between these two angles (θ_1, θ_2) to determine the depth of point P.

B. Fixed Far Vote Fusion

The registration and identification algorithm used in this paper, FFVF (Fixed Far Vote Fusion)[1][2], was created by Luuk Spreeuwers. FFVF accepts 3D point clouds of faces as

Fig. 2: Face with an IR light pattern projected on it

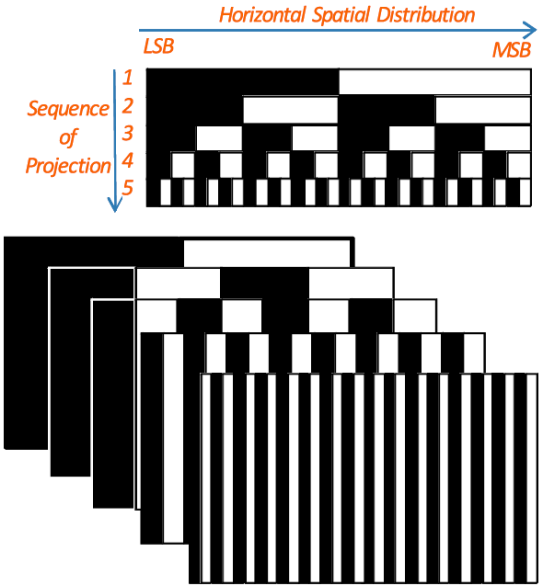


Fig. 3: Binary pattern projected in lines [4]

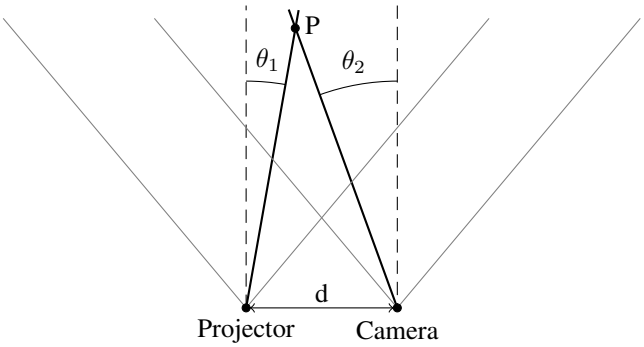


Fig. 4: Geometric representation of the depth acquisition triangulation from a top perspective.

code, which means the binary code of a line only differs with one bit from the adjacent lines[5]. This has an advantage over the standard binary encoding; when a bit of a line is misread, the value has a higher chance to be the same as that of a nearby line. This means the error of this line will be lower since it's projected at a closer location. The used gray code is shown in Figure 6. The range image has a resolution of 640 by 480 pixels and a frame rate of up to 60 frames per second, although converting the depth data to a 3D point cloud at this frame rate is computationally intensive. The official operating range lies between 20 and 150cm.

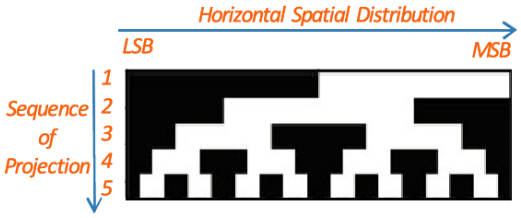


Fig. 6: Binary pattern using with Grey code

inputs and calculates a matching score between them. FFVF registers all face models to an intrinsic coordinate frame. A 2D range image is then created from the front perspective of a registered face model. The image is divided into several regions using masks, which each use PCA-LDA classifiers to compare the section against the same section of another face. A score between 0 and 60 is then determined by using majority voting over all the regions. The system returns higher scores for more similar faces. The algorithm has achieved a verification rate of 99.3% on the Face Recognition Grand Challenge (FRGC) v2 data at false accept rate = 0.1%, and an identification rate of 99.4%. This level of precision means it can be used as an accurate measurement of the quality of the point clouds generated by the RealSense SR300.

III. Camera

A. Depth acquisition method

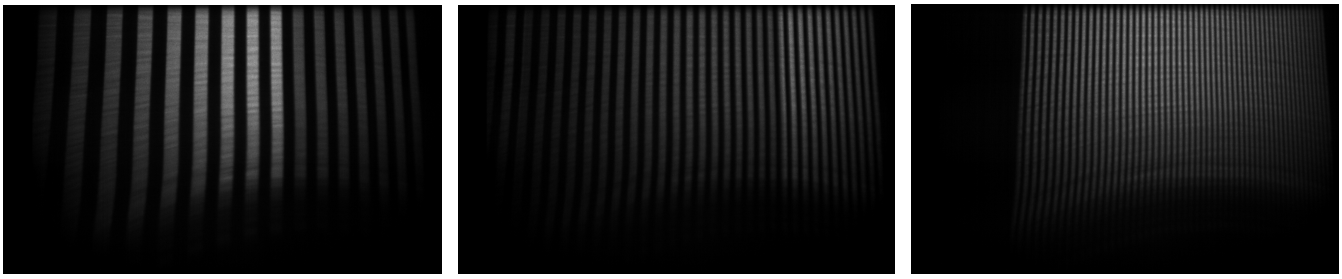
The RealSense SR300 uses 10 sequential binary line patterns per frame to create its depth image. It makes use of Gray

B. Development and code

The RealSense SDK can be used in the languages C++ and C#. The latter is used since it handles a variety of things like garbage collection and allows for an easy user interface, while the development can be focused on determining the right settings. The relevant parts of the code are explained below.

```
//Setup code, single run
sm = PXCMSenseManager.CreateInstance( );
sm.EnableStream(
    PXCMCapture.StreamType.STREAM_TYPE_DEPTH,
    640, 480, 60 );
device =
    sm.QueryCaptureManager().QueryDevice();
projection = device.CreateProjection();
```

The code above is run once to start the data stream. It creates a SenseManager object, which is an interface to access functions of the RealSense SDK. A depth stream is enabled at the max resolution of 640x480 at 60fps. Then the device is queried, which gives, among others, the intrinsic parameters of camera.



(a) (b) (c)

Fig. 5: IR photos of the projected patterns on a planar surface. The line frequency of the pattern changes, all images have the same scale

These parameters can be used to make a projection, which allows mapping depth values info to world coordinates, which is the final goal.

```
//Code run on every frame
sample.depth.AcquireAccess(
    PXCMImage.Access.ACCESS_READ,
    PXCMImage.PixelFormat.PIXEL_FORMAT_RGB32,
    out _depthImgData );
// Code run on user request
projection.QueryVertices(sample.depth,
    vertices);
Thread thread = new Thread(() =>
    WriteWRL(verticesCopy));
```

This code is run on every frame. It requests access to the depth data in a RGB format to render to a window for the user. (Figure 7)

On request of the user, the last two lines of code are executed, in which the projection is used to convert the depth image to real world coordinates. The vertices are then exported to a .wrl file, a common file format for 3D models. This is done in a new thread since it takes more than 1/60 of a second and would delay the streaming process. A skinned and rendered example of how the 3D model looks like is shown in Figure 8.

C. Settings

The RealSense SDK comes with various options to tweak the system to perform optimally for various situations. Several tests were done on multiple configurations, and the following choices have been made:

1) Filter Option: The filter option determines the filter applied to the image. The setting has eight options: 'skeleton', 'raw' and smoothing options, optimized for several distances to the subject. Both 'skeleton' and 'raw' provide a very rough model which delivers very poor results. Of the range options, the 'Very close range' setting is chosen. This provides very low smoothing and low noise.

2) Projector Power: The projector power sets the strength of the laser projector used to create the light patterns. This setting has three options: On, Off and Auto. Setting the power to 'On' sets the power to the maximum value, which leads to overexposure of the camera at close distances, resulting in artifacts on the face in the form of vertical lines. Figure 8 shows a small-scale example of this artifact. The 'Auto' setting varies the power based on the distance to the subject, which results in higher quality images In an indoor environment the 'Auto' setting performs a lot better, as shown in Table I. However, when outdoors or close to a large window, the 'Auto' setting is not bright enough, and better results are achieved with the 'On' setting.

Fig. 7: Depth image of a face. Lighter parts are closer to the camera

3) Motion Range: The motion range option determines the exposure time of the camera. it accepts an input value between 0 and 100, with 0 being the shortest exposure time and 100 being the longest. By setting the projector power to 'Auto' this value is regulated automatically by the SDK. When setting the projector power to 'On', the Motion Range option should be set to 0 to reduce the overexposure to the largest extent.

4) Confidence: The confidence sets the threshold for which points are reliable enough to use. A value between 0-15 determines whether almost all points or only the most reliable points are passed through. The default of 3 is used, as it accepts the whole face of the subject and rejects most background elements.

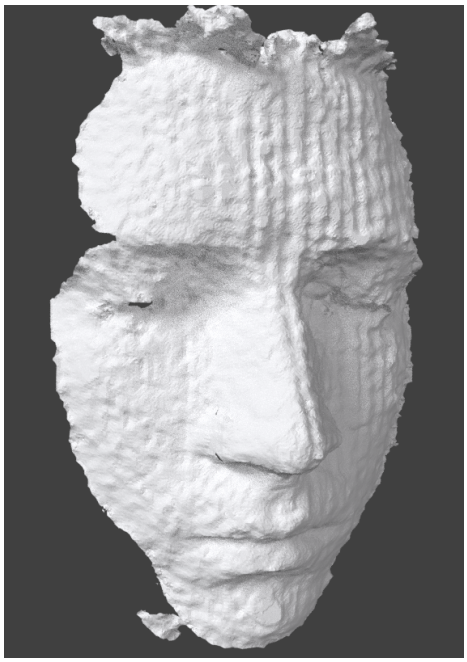


Fig. 8: Skinned and rendered model of a 3D point cloud

It's important to note this is the current setup in version '2016 R2'. This is subject to change in further versions of the SDK.

IV. Experiments and Results

Several measurements were done to determine under which circumstances the camera performs the best. A measurement for resolution was done to find the smallest step in depth the camera can reliably measure. The per pixel depth deviation was measured to find the consistency of each pixel. An experiment on real faces was used to find a relation between threshold scores, verification rate and false accept rate. Two additional tests give an indication of the effect of external light and distance between subject and camera. As different experiments did not take place at the same location or time of day, values obtained should not be compared between different experiments.

A. Resolution

The performance of the system is heavily influenced by its resolution. This is the smallest step in depth the system is able to detect. The theoretical limit of the system's resolution is 0.125mm, due to the way the depth data is stored. An experiment is set up to test the resolution of the system. A range image is made of a vertical plane at several distances. A formula for the plane is calculated using a 'least squares' fitting method with all the vertices of the depth image. Then the difference between the calculated plane and the position of the corresponding points in the depth direction is taken. This gives the depth error in every pixel. A horizontal differential is taken of the depth error, which results in a matrix with relative errors. If this relative error is greater than half the distance in depth between two points, the distance becomes indiscernible. This means that resolution $R = 2\Delta_r$, where Δ_r is the relative error. The average of the absolute resolution of every pixel has been plotted against distance in Figure 9. The resolution at 150 mm is significantly higher in value due to large peaks in depth in the middle of the image, caused by the close distance.

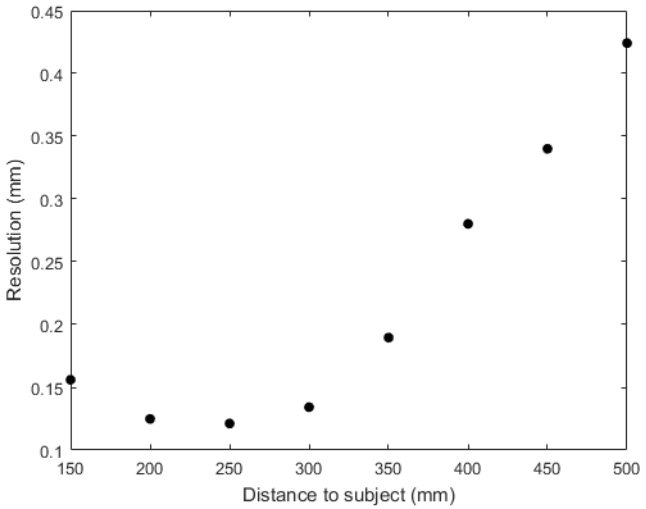


Fig. 9: Average resolution plotted against the distance to a subject

B. Pixel depth deviation

To assess whether each pixel in the depth image is consistent enough to provide reliable data without being averaged or filtered otherwise, measurements are done on the Average Absolute Deviation (AAD) (see Equation 1) of each individual pixel at several distances from a horizontal plane. Figure 10 shows a clear increase of deviation for larger distances between the camera and the plane. This means the camera is more reliable at close distances.

$$AAD = \frac{1}{n}\sum_{n=1}^{\infty}|x_i - mean(X)| \tag{1}$$

Where n is the amount of samples and x_i is a sample of the set X.

C. Effects of external light

A test was performed, where sets of 10 face images were made of a single subject with the projector at 'On' and 'Auto' brightness, in an indoor and outdoor location. All images were made at a distance of around 30 cm. Every set of 10 images was compared with itself in the FFVF algorithm, which produced an average score, shown in Table I.

	Auto power	Full power
Inside	19.1	12.8
Outside	0.8	4.5

TABLE I: Average score of auto comparison of 10 images in outside and inside conditions with the projector at full power and auto power

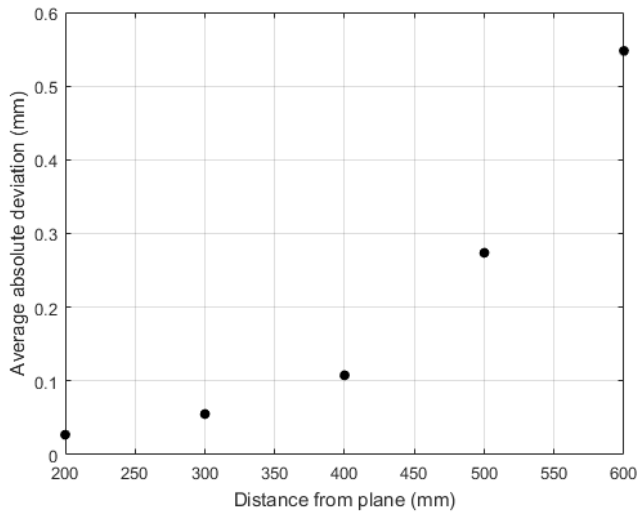


Fig. 10: Average over all pixels of the Average Absolute Deviation of each pixel

D. Distance of subject

An experiment was done with sets of 10 images of the same subject at 20, 30, 40 and 50 cm from the camera. Each set was auto compared in the FFVF algorithm. The averages of the score of these images are shown in Table II. This clearly shows the score to rise at closer distances.

Distance (cm)	20	30	40	50
Average Score	54.7	25.3	13.6	1.1

TABLE II: Average score of auto comparison of 10 images at several distances to the subject

V. Face comparison

To test the 3D point cloud generated by the camera, range images are made of 22 subjects. Each subject has 2 3D-scans made with the face in different positions on screen with a very slight change in head orientation, at approximately 30 cm distance to the camera. The 44 generated point clouds are split into two groups of 22, corresponding to the first and second image made of every subject. Each point cloud of the first group is compared to each one of the second group using the FFVF algorithm. This produces a 22 by 22 matrix of scores, in which the diagonal is the score of 2 point clouds of the same subject. A 5 by 5 sample of this matrix is shown in Table III. Now a fitting threshold score can be determined which gives a combination of FAR (False Accept Rate) and VR (Verification Rate), which is 1-False Reject Rate. Figure 11 shows the FAR versus the Verification Rate. The relevant threshold scores are labeled in the image. The comparison score between two different subjects only gave a nonzero answer in 3 cases, giving 1, 1 and 2 instead. Except for 1 comparison, all comparisons of two point clouds of the same user gave a score of 4 or higher. The system, therefore, reaches a VR of 99,1% with an FAR of 0%.

	Subj1 Img1	Subj2 Img1	Subj3 Img1	Subj4 Img1	Subj5 Img1
Subj1 Img2	5	0	0	0	0
Subj2 Img2	0	56	0	0	0
Subj3 Img2	0	0	11	0	0
Subj4 Img2	0	0	0	15	0
Subj5 Img2	0	0	0	0	39

TABLE III: Sample of face comparison matrix

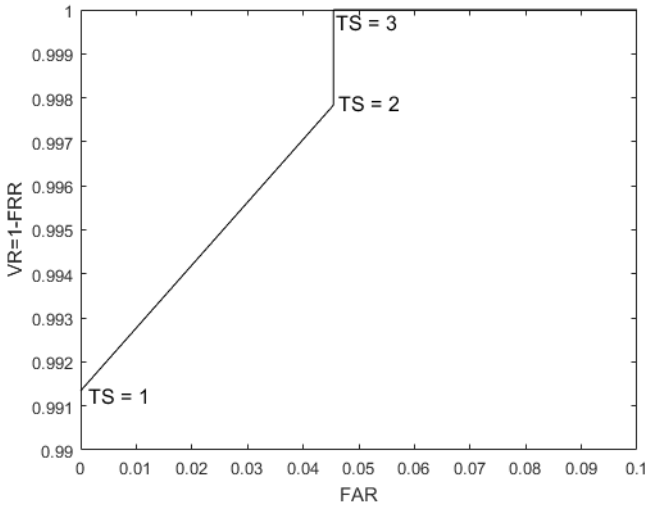


Fig. 11: False Accept Rate vs Verification Rate with Threshold Score (TS) labeled at relevant points

VI. Discussion

The goal of this project was to create 3D point clouds with the RealSense which have sufficient quality to use for 3D face recognition. The results of the experiments show that this is possible. However, quite some requirements must be met to have a consistently reliable result.

The degradation of the depth quality is very noticeable when close to bright IR light sources, like a window. The system has trouble detecting the binary values because the contrast of its projection is reduced due to the external light source. Setting the Projector Power setting to 'On' instead of 'Auto' improves the image in this case due to the increased brightness of the projected pattern. The downside to this is the increase in the aforementioned artifacts. While the range of the RealSense camera is up to 2 meters, the subject should be within 40 cm to get a reliable result. This means the camera can only be used by a user who steps right in front of it.

The FFVF score for comparison between different faces is 0 in the great majority of cases. This means the threshold score can be set very low, without having a high FAR. The average pixel depth deviation is shown to be quite low at the distances at which the camera works well. At further distances it might increase the performance to use averages of multiple range images.

The experiments done with real subjects have their limitations. The sample size is very small, which means the FAR

and VR have a large margin of error. The tests for distance and external light impact would ideally have been done with multiple test subjects, but time did not allow for this.

VII. Conclusion and Recommendations

Based on the obtained results, this camera is able to function as a reliable verification system in an indoor environment. In this, it would be one of the first cheap face recognition systems for personal use. The RealSense SR300 is the current flagship of the RealSense product line and comes built-in with some of the newest laptops and tablets on the market, which means it can be used as a security feature for applications. However, many improvements can still be made increase the performance of the system.

To increase the performance at farther distances, averaging of multiple point clouds can be an outcome. The registration algorithm of FFVF can be used to overlay the multiple faces over each other, increasing the resolution. Since the frame rate of the range data is much higher than is required, there is a lot of headroom for averaging.

Furthermore, the RealSense has both a depth and a color camera. The latter can be used for 2D face recognition to increase the reliability of the system. The software package contains functions to map the color image to the point cloud of the depth system. Further research might enable the creation of 3D point clouds with color information. To use this model the FFVF algorithm would have to be altered.

The quality of the existing system can also be increased by removing the vertical ridges of the face models, belonging to the structured light of the projector. These lines are stronger when using a higher projector brightness and when the subject is closer to the camera. This suggests the lines are caused by overexposure of the camera. Removing these lines is not straightforward, since they have a different position depending on the depth. However, the removal of these lines would greatly increase the performance of the system.

References

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

[2] Luuk Spreeuwers. "Breaking the 99% barrier: optimisation of 3D face recognition". In: *IET biometrics* 4.3 (2015), pp. 169–178. URL: http://doc.utwente.nl/95850/.

[3] Intel. "Introducing the Intel RealSense Camera SR300". In: (2016). URL: https://software.intel.com/en-us/articles/introducing-the-intel-realsense-camera-sr300.

[4] Jason Geng. "Structured-light 3D surface imaging: a tutorial". In: *Adv. Opt. Photon.* 3.2 (2011), pp. 128–160. DOI: 10.1364/AOP.3.000128. URL: http://aop.osa.org/abstract.cfm?URI=aop-3-2-128.

[5] P. Zanuttigh et al. "Time-of-Flight and Structured Light Depth Cameras". In: (2016), pp. 43–78. DOI: 10.1007/978-3-319-30973-6_2.