Classification of Fingerprints Captured Using Optical Coherence Tomography

Ctirad Sousedik[1], Ralph Breithaupt[2], and Patrick Bours[3]

[1,3]Norwegian University of Science and Technology (NTNU), Gjøvik, Norway
{ctirad.sousedik,patrick.bours}@ntnu.no
[2]Federal Office for Information Security (BSI), Bonn, Germany
ralph.breithaupt@bsi.bund.de

Abstract. We propose a technique for analysis of fingerprints scanned free-air (not pressed against a glass) with Optical Coherence Tomography (OCT). Fingerprints from the surface and subdermal parts of the finger are extracted from a 2GB volumetric scan in cca. 2 s using our specialized technique and GPU acceleration on GeForce GTX 980. The technique provides fingerprints that perform with promising error rates that demonstrate the potential of the OCT for improved fingerprint identification, as well as its potential for prevention of biometric spoofing (PAD).

1 Introduction

In the recent years, biometrics are on the rise as a convenient alternative authentication mechanism. Unlike passwords, which can be easily forgotten, and access cards or keys, which can be easily lost, biometrics provide a means of authentication that is always readily available.

Among the many existing biometric modes, such as iris, face, retina and others, fingerprint stands among the best known and most widely applied.

However, despite the three-decade-long history, fingerprint sensing solutions still struggle with a number of challenges, which limit their applicability especially for unsupervised scenarios such as border control:

- Worn out fingers - For persons whose fingertip skin has been subjected to a lot of stress, the fingerprint can be abraded or significantly damaged (guitar players, construction workers, chemists etc.)
- Wet or greasy fingers - liquids on the surface of the fingerprint tend to diffuse into the fingerprint valleys when the finger is pushed against the sensor surface, which makes the acquisition difficult
- Dry fingers - If the fingerprint skin is too dry, it does not come into good contact with the fingerprint sensor surface, which results in low quality
- Infant fingerprints - the fingerprint skin of infants is very soft, and if pressed against a sensor surface, the fingerprint pattern will not be observable - this limits the usage of fingerprinting in fight against child trafficking

Similarly to other biometric modes, a significant challenge comes also with the susceptibility of fingerprinting to spoofing attacks [10, 14].

A large body of research exists addressing this challenge of Presentation Attack Detection (PAD) for 2D fingerprint sensors. However, no single solution as of yet provides for a good level of security, especially considering resistance against novel materials and production techniques regarding the artefact fingerprints [14]. The existing approaches in the industry typically focus on combining a larger number of additional single-purpose sensors and features extracted from the 2D image to take the PAD decision. Considering the variability of properties of genuine human fingers, this requires machine learning approaches, which inherently depend on the training data, and as such are vulnerable against novel approaches not considered before [14].

1.1 OCT fingerprinting

Due to the above mentioned shortcomings of the existing fingerprint sensors, the community has been looking for an alternative solution.

A promising path is offered by the Optical Coherence Tomography (OCT). The OCT is a light-beam-based scanning technology that is capable of penetrating the fingerprint skin up to the depth of 2-3mm and acquiring a 3D volumetric representation of the light-scattering properties of the scanned sample - the surface fingerprint along with the sub-surface structure - Fig. 2, 3. Contact with a surface is not necessary, and as such many of the challenges such as dry, wet, greasy or infant fingers can be easily overcome. In addition, the OCT is able to spot a second representation of the fingerprint in the subsurface data - the master template responsible for the stability of fingerprint during a person's lifetime. In addition to the inner fingerprint, presence of which could readily be used for the PAD purposes, the OCT is also able to spot sweat glands - fine spiral structures which end as sweat pores on the surface on the fingers. And last but not least, the volumetric measurements from the OCT readily provide a general scattering profile of the underlying material.

However, along with the significant promise, a rather significant challenge comes also associated with the step up from 2D to 3D scanning. The amounts of data generated by the OCT are very significant, and require novel scanning and processing approaches in order to achieve the practical speeds of a few seconds as required by many applications, such as border control.

2 Related Work

Some initial research has been carried out regarding the application of the OCT for the fingerprint sensing scenario.

Cheng and Larin [2] performed fingerprint Liveliness Detection by using autocorrelation analysis applied to the OCT. They scanned a small 2D in-depth part of the fingerprint tip in both the x-direction (2.4mm) and in the z-direction (2.2mm). In the depth direction they applied autocorrelation analysis. They

proved that autocorrelation values resulting from living fingerprints differ substantially from the values resulting from fake fingerprints. The same authors also obtained 3D density representations of fake and living fingerprints. They were among the first to use OCT technology to do this. A 3D OCT scan of a fake fingerprint surface on a real finger was published in their paper.

Peterson and Larin [12] classified OCT scans into fake and real samples. For this they used various neural network based approaches. The features that were used were based on first-order image statistics as well as on Gabor filter responses. Self-organizing maps (SOM) were used to reduce the dimentionality of the vector of Gabor filter reponses.

Nasiri-Avanaki et al. have also shown in [11] that OCT scanning can be used to separate between genuine fingerprints and fake ones. In [1], Bossen et al. have shown that standard fingerprint comparison methods can be used for classification when applied on both the inner and outer fingerprint extracted from an OCT scan. Menrath [9] came up with a method to use OCT scans for detection of both sweat glands and extraction of the outer and innter fingerprint. His results are based on rather small fingerprint areas of 4x4 mm.

Khutlang et al. [7] have been capturing a partial fingerprint area using a commercially available OCT scanner and experimenting with the the detection of the inner fingerprint and its extraction.

Despite the fact that a number of initial studies exists regarding the application of the OCT for fingerprint sensing and genuine/fake finger detection, very few studies take the speed of the processing into account. OCT fingerprint scans represent volumetric data of very significant sizes if one aims for standard resolutions of 500dpi or 1000dpi in 2D, which easily exceeds 1 GB per finger (1024x1024x1024 at 8bit). This fact calls for an approach where the speed of the processing techniques is taken seriously into account as an actual research challenge, rather than disregarded as simply a matter of faster processing hardware.

In addition, majority of the studies assume a finger pressed against a flat surface (e.g. glass) during the scanning, which greatly simplifies the processing challenges (since the fingerprint surface is flattened) but introduces a number of disadvantages shared with standard 2D fingerprint sensors, such as difficulty to scan wet, greasy, dry or soft-skinned fingers, which could otherwise be easily overcome by the OCT [3,8].

Last but not least, the free-air scanning approach taken in our work has the potential for touch-less fingerprint sensing applications, where the subjects do not have to come into contact with any potentially unclean surface of the fingerprint sensor, increasing acceptability and convenience.

Regarding the work that satisfies the challenging constraints mentioned above, we are aware only of the promising works by Darlow et al. [3–5].

3   Fingerprint Surface Extraction in 3D

Our approach is to scan the fingerprint free-air, such that it is undistorted by pressing against a flat surface (such as glass). This comes with the challenge of

precise 3D segmentation of the volumetric fingerprint OCT scan, so that further analysis is possible. A well segmented OCT fingerprint scan could be further used for extracting the fingerprints both from the surface and from underneath the skin and it could also serve well for further analysis of the skin regarding PAD (Fig. 2).

3.1 Database

We utilized the following OCT fingerprint scan dataset collected by our earlier work [13]:

- 1408x1408x1024 voxels
- 8bit per voxel
- 72 participants
- all 10 fingers
- 720 OCT fingerprint scans in total
- 2x2cm scanning area

Along with the OCT fingerprints, we also collected standard 2D fingerprints from an identical set of participants, in order to enable testing of compatibility of the OCT fingerprints and the standard 2D fingerprints. This dataset contains 500 2D fingerprints, of all 10 fingers from 50 participants.

3.2 Efficient edge detection

In order to address the challenge of processing 2GB OCT fingerprint scans in a matter of a few seconds, as required by numerous applications including border control, we took a combined approach of GPU acceleration and our specifically developed fast approximate filtering technique. Utilizing the following mathematical equality,

$$\int_{-\infty}^{\infty} f(\tau) \int \cdots \int g(t-\tau)dt \cdots dtd\tau = \int \cdots \int \int_{-\infty}^{\infty} f(\tau)g(t-\tau)d\tau dt \cdots dt, \quad (1)$$

it is possible to perform a convolution of a signal and a convolution core by deriving the convolution core, performing the convolution, and integrating the result. The advantage comes with the fact that a wide filtering core, which would take a significant computational effort to perform convolution with, can in specific cases derive into a very sparse representation that is very efficient to perform convolution with. Following this train of thought, we designed the following filter in order to detect edges in the scan-lines of the OCT fingerprint data - Fig. 1:

$$G(n) = \begin{cases} -height_2 + (n+1) \cdot slope & \text{if } n \in [-\frac{size}{2}, -1] \\ height_2 - n \cdot slope & \text{if } n \in [0, \frac{size}{2} - 1] \\ 0 & \text{otherwise} \end{cases} \quad (2)$$
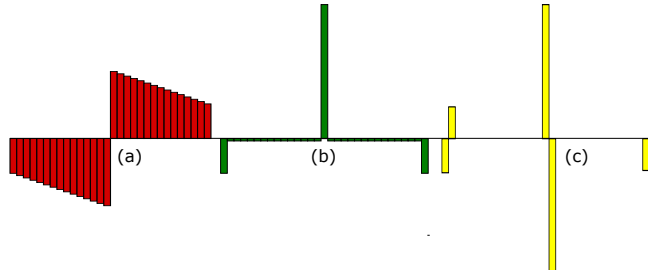
Fig. 1. Fast approximate edge detection filter and its derivatives for $height_1 = 15$, $height_2 = 29$, $size = 30$: (a) filter core $G(x)$; (b) first derivative $G'(x)$; (c) second derivative $G''(x)$ ($slope = \frac{height_2 - height_1}{size/2 - 1}$)

Second derivation of such a filtering core results in the following core:

$$
G''(n) = \begin{cases}
-height_1 & \text{if } n = -\frac{size}{2} \\
height_1 - slope & \text{if } n = -\frac{size}{2} + 1 \\
2 \cdot height_2 + slope & \text{if } n = 0 \\
-(2 \cdot height_2 + slope) & \text{if } n = 1 \\
-height_1 + slope & \text{if } n = \frac{size}{2} \\
height_1 & \text{if } n = \frac{size}{2} + 1 \\
0 & \text{otherwise}
\end{cases}
\tag{3}
$$

Notably, such core has always only 6 non-zero coefficients, which come in 3 pairs of 2 coefficients positioned immediately next to each other - Fig. 1. This allows for a very efficient implementation of the convolution on the GPU, since only 3 memory reads per voxel are necessary to perform the filtering along the scan-lines, as the other 3 reads can be implemented using 3 delay variables reusing the results from a previous voxel. Integrating the results twice along the way allows to obtain the convolution of the scan-lines with the core, $G(n)$, in very efficient manner in a single GPU thread per scan-line. In addition, the technique can calculate the results for each of the scan-lines without using any extra intermediary memory buffer, which supports both the speed and the flexibility of the approach.

The actual edge detection is performed by identifying the position of a maximum during the process of convolution of the OCT scan-lines and the above discussed convolution core.

3.3 Outer Fingerprint Surface

However, it is highly sub-optimal to simply perform the above discussed edge detection for each of the (x, y) in-depth scan lines and treat the result as the detected surface. The OCT scan inherently contains large amounts of noise, and such an approach would result in a highly noisy surface, full of holes, where

the detection failed due to noise. In addition, the strength of the finger's response diminishes with depth, which poses further challenges - Fig. 2. In order
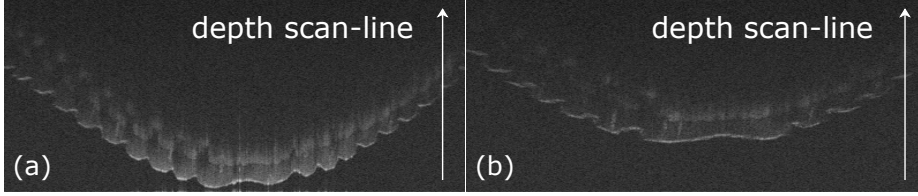

Fig. 2. OCT finger 2D slices - inner and outer fingerprint edges visualization

to address the challenges, we took inspiration from the approach by Darlow et al. [5] regarding performing the detection using a 3D equivalent of the image down-sampling pyramid.

Unlike Darlow et al. [5], who suggest building the pyramid by downsampling the OCT scan in all three dimensions, we have taken an approach where the pyramid is built by downsampling the scan always to 1/2 in width and height, leaving the depth dimension unaltered as illustrated by Fig. 3. The detection of
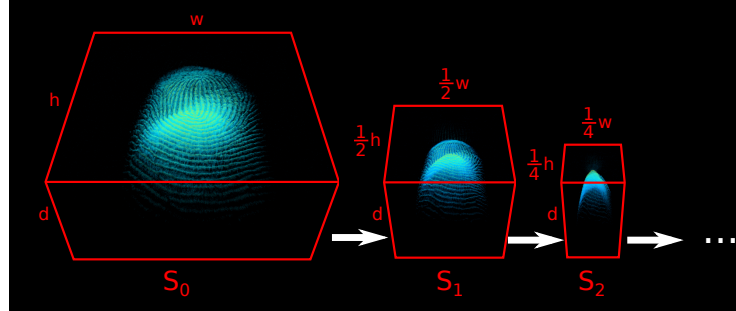

Fig. 3. Downsampling of the OCT scan along the width and height dimensions

the outer fingerprint surface, $outer_0(x, y)$, is then performed using the algorithm 1.1.

The rationale is that it is far more likely to correctly detect the position of the fingerprint surface in a more downsampled scan, since the down-sampling heavily reduces the noise levels. If the surface, $outer_n(x, y)$, is assumed only in proximity of the surface, $outer_{n+1}(x, y)$, detected in a more down-sampled version of the scan, the likelihood of a correct detection is greatly increased.

In addition, detecting the surface, $outer_n(x, y)$, only in a distance, d_n, from a lower resolution surface $outer_{n+1}(x, y)$, allows for a significant speed-up since

the entire scan does not need to be processed on any but the most downsampled level.

Algorithm 1.1: Outer fingerprint detection - 531 ms - GeForce GTX 980

---

1 Copy data from RAM to the GPU;

2 From the original OCT fingerprint scan, $S_0(x, y, z)$, generate N additional versions of the OCT scan, $S_1(x, y, z)...S_N(x, y, z)$, each downsampled to 1/2 of the previous one along width and height dimensions;

3 While filtering the scan-lines using the edge detection filter, detect the maximum position in each $S_N(x, y)$ scan-line and store it to $outer_N(x, y)$;

4 for *each n in N − 1 to 0* **do**

5   Up-sample $outer_{n+1}(x, y)$ by a factor of two into $region_n(x, y)$

6   While filtering the scan-lines using the edge detection filter, detect the maximum position in each $S_n(x, y)$ scan-line and store it to $outer_n(x, y)$ IF at a distance d_n from $region_n(x, y)$;

3.4 Fingerprint Flattening

After the outer fingerprint surface, $outer_0(x, y)$, has been identified at full resolution, the detection of the inner fingerprint surface constitutes the natural next step. The inner fingerprint is more difficult to detect, compared to the outer fingerprint, since the contrast between the inner fingerprint and and surrounding tissue is much lower than the contrast between the outer fingerprint and the empty air - Fig. 2.

In addition, the depth at which the inner fingerprint appears in various people varies widely (range wider than 1-5x in our experience). If one attempts to detect the inner fingerprint simply as a second surface around the outer surface, $outer_n(x, y)$, the outer fingerprint easily interferes with the inner fingerprint on the more downsampled versions of the scan and the detection is highly unreliable, especially for fingers where the distance between the inner and outer layer is rather small.

In order to address this issue, we perform full flattening of the original OCT scan according to the identified outer fingerprint surface. By flattening we mean re-arranging of the data along the z-axis using the following equation:

$$S_0(x, y, z) \leftarrow S_0(x, y, z - outer_0(x, y)) \tag{4}$$

3.5 Inner Fingerprint Surface

The re-arranged data naturally put any remnants of the outer fingerprint close to the bottom of the scan. This prevents interference of the outer fingerprint during the inner fingerprint detection.

Even though the algorithm, used for extracting the inner fingerprint surface, is similar to the algorithm for extracting the outer fingerprint surface, there are two important differences.

The inner fingerprint surface is not searched using a constant filter size, but rather an adaptive filter size. This is necessary to handle the cases of very thin inner layers, where the inner fingerprint is very close to the outer one. It also improves performance for the fingerprint where the the distance between the outer and inner fingerprint surface is large, and as such a larger filter size can perform much more reliably.

The inner fingerprint surfaces, detected even in the most down-sampled versions of the flattened OCT scan, can still contain significant number of errors. This appears to be caused by interference of the inner fingerprint with itself, if a significant amount of down-sampling is considered. The problem is mitigated by a guessing procedure, where any detected points that deviate too much from the average depth range are replaced by a position that gained maximum response in a histogram of the detected positions - by the most likely depth encountered. Although this could seem to damage the inner fingerprint surface, in practice, it provides a precise-enough estimate for the higher-resolution detection steps, and the technique can recover even from severe failures encountered at the most down-sampled level.

The concept is described by algorithm 1.3.

3.6 Surface Conversion to 2D Fingerprint

We extract the fingerprint into 2D representation by utilizing the identified 3D surface, masking out the random noise that appears where the finger was not present - Fig. 6. The concept is described by algorithm 1.2.

Algorithm 1.2: 2D fingerprint extraction - 59 ms - GeForce GTX 980

---

1 Process the 3D fingerprint surface, inner or outer, $s(x, y)$, by a 2D Gaussian core and store to $s_{blurred}(x, y)$;

2 $s_{subtracted} \leftarrow (s - s_{blurred}) \cdot fingerprint\_height\_ratio$;

3 $s_{fingerprint}(x, y) \leftarrow s(x, y)$ IF $s(x, y) \in [0, 1]$ or 1 IF $s(x, y) > 1$ or 0 IF $s(x, y) < 0$;

4 Threshold the confidence map $confidence(x, y)$ into fingerprint and empty area, median filter to remove holes, and utilize to mask out the noisy areas;

---

4 Results

Direct comparison with the method of Darlow et al. [3] is difficult, since we were not able to obtain the codes/executables from the authors and we do not have any opportunity to share the data either due to their sensitive nature. However, we believe our method clearly outperforms the method of Darlow et al. [3] for the following reasons:

− Our approach using adaptive filter size allows processing of the cases where the inner fingerprint is at very small depth - something that is not possible using a simple 3-point edge detector on a scan down-sampled in all 3 dimensions as in [3]

Algorithm 1.3: Inner fingerprint detection - 328 ms - GeForce GTX 980

---

1 Flatten the fingerprint according to Eq. 4;

2 Re-compute the image pyramid;

3 From the original OCT fingerprint scan, $S_0(x, y, z)$, generate N additional versions of the OCT scan, $S_1(x, y, z)...S_N(x, y, z)$, each downsampled to 1/2 of the previous one along width and height dimensions (re-calculate the pyramid for the flattened scan);

4 $current\_maxima = -INF$;

5 **for** *each tested_filter_size in l to h with step s* **do**

6 $\quad$ While filtering the scan-lines using the edge detection filter, detect the maximum position in each $S_N(x, y)$ scan-line and store the maxima positions to $inner_N(x, y)$ and the values of the maxima (the response of the filter at the maxima positions) to $confidence_N(x, y)$;

7 $\quad$ Calculate a weighted histogram $h(d)$ of $inner_N(x, y)$ using $confidence_N(x, y)$ as weights;

8 $\quad$ Gaussian blur the $h(d)$ as a 1D function;

9 $\quad$ Detect maxima m in $h(d)$;

10 $\quad$ **if** $m > current\_maxima$ **then**

11 $\quad\quad$ set $current\_maxima \leftarrow m$;

12 $\quad\quad$ set $optimal\_filter\_size \leftarrow tested\_filter\_size$;

13 set $optimal\_filter\_size$ as the edge detection filtering size for all further operations;

14 While filtering the scan-lines using the edge detection filter, detect the maximum position in each $S_N(x, y)$ scan-line and store the maxima positions to $inner_N(x, y)$ and the value of the maxima (the response of the filter at the maxima positions) to $confidence_N(x, y)$;

15 **for** *each n in N − 1 to 0* **do**

16 $\quad$ **if** $n > ec\_depth\_limit$ **then**

17 $\quad\quad$ Calculate a weighted histogram $h(d)$ of $inner_n(x, y)$ using $confidence_n(x, y)$ as weights;

18 $\quad\quad$ Calculate a histogram $h(d)$ of $inner\_weighted_n$;

19 $\quad\quad$ Gaussian blur the $h(d)$ as a 1D function;

20 $\quad\quad$ Detect the maxima, m, and its position, p, in $h(d)$;

21 $\quad\quad$ From the position, p, search to the right and to the left in $h(d)$ until a number smaller than $m \cdot ratio$ is encountered (intuitively, measure the width of the peak in the histogram);

22 $\quad\quad$ Utilize the identified positions to calculate the acceptable level of deviation, $inner\_layer\_deviation$;

23 $\quad\quad$ Process $inner_N(x, y)$ such that any value further than $inner\_layer\_deviation/2 \cdot d\_ratio$ from m is replaced by m (Replace values too far from the average by the average value)

24 $\quad$ Up-sample $inner_{n+1}(x, y)$ by a factor of two into $region_n(x, y)$;

25 $\quad$ While filtering the scan-lines using the edge detection filter, detect the maximum position in each $S_n(x, y)$ scan-line and store the maxima positions to $inner_n(x, y)$ and the value of the maxima (the response of the filter at the maxima positions) to $confidence_n(x, y)$ IF at a distance d_n from $region_n(x, y)$;

– Our approach uses a filter with a wide support, which allows obtaining a much higher quality surface than in [3]
– Darlow et al. [3] did not follow the path of extracting the 2D fingerprints directly from the 3D surface, and instead utilized the surface simply as an estimate of the fingerprint positions. The fingerprint is read off from the original scan by averaging the data above and below the estimated surface. Our method does not require this, in fact, we were able to obtain complete fingerprints using the layer information alone.

4.1 Compatibility with 2D Fingerprints

To further prove the robustness of our approach, we performed a $N : N$ comparison between the OCT fingerprints and the 2D fingerprints in the above mentioned dataset: The metrics considered are miss-classified comparisons of
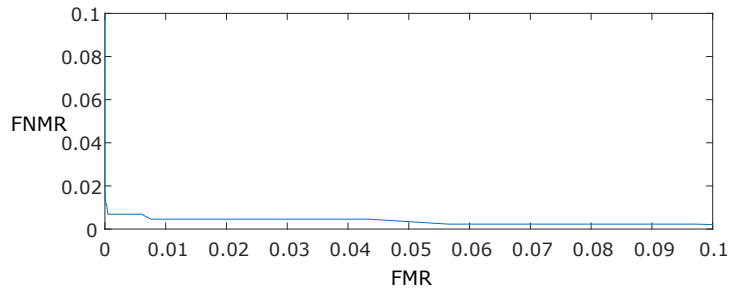

Fig. 4. ROC curve for the outer fingerprints and the 2D fingerprints comparisons
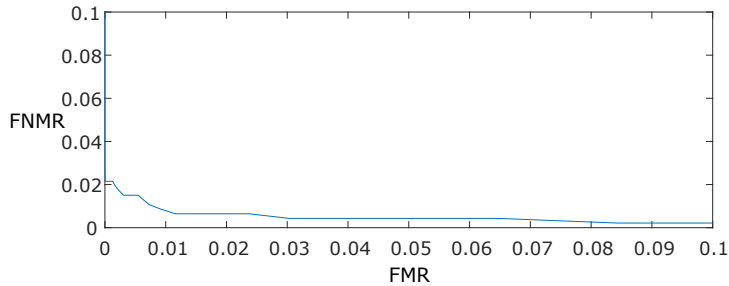

Fig. 5. ROC curve for the inner fingerprints and the 2D fingerprints comparisons

identical fingerprints (FNMR) and miss-classified comparisons of non-identical fingerprints (FMR). The failure-to-extract (FTX) metric expresses the percentage of fingerprints the comparison software failed to extract fingerprint features from [6]. The outer fingerprints compared to the 2D fingerprints with an equal-error-rate (EER) of 0.7% and a FTX rate of 11%. The inner fingerprints compared to the 2D fingerprints with EER of 1% and a FTX rate of 3.5%. The 2D

fingerprints from a commercial sensor resulted in FTX of 3.2%. A commercial fingerprint identification software Verifinger 9.0 was used.

The average durations of the above proposed algorithms are stated in their heading (Al. 1.1, 1.2, 1.3). The speed was measured as the CPU time since the start of the stage until its end, synchronizing with the GPU to make sure the GPU computation has finished as well. In total, our algorithm processes a 2GB scan (including additional data management) in cca. 2 s on GeForce GTX 980.
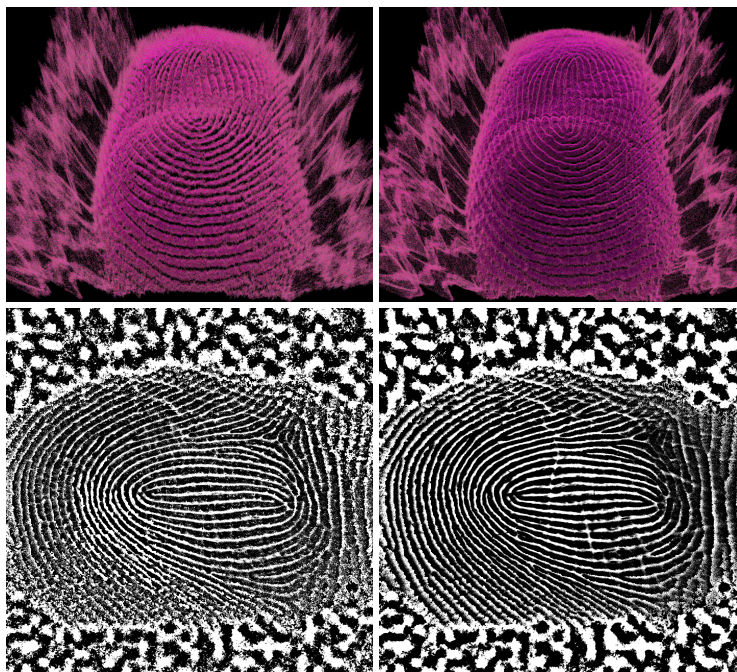
4.2 Fingerprint Surfaces

Fig. 6. Left: inner fingerprint surface 3D surface and extracted 2D fingerprint; Right: outer fingerprint surface 3D surface and extracted 2D fingerprint

5 Conclusion and Future Work

Based on the fact that the inner fingerprint seems to perform better than the outer fingerprint, considering the FTX, we believe it is indeed very promising to expect significant improvements from the OCT. At the same time, the fingerprints already do perform close to practical error rates - in spite of the significant potential of our custom-built OCT scanner for improvements. Our technique can process 2GB in cca. 2 s, which shows promise regarding the speed challenges.

12

The future work would involve further improvements of the underlying scanner design. In addition, the method for the detection of the outer fingerprint could probably benefit from further improvements.

References

1. Bossen, A., Lehmann, R., Meier, C.: Internal Fingerprint Identification With Optical Coherence Tomography. Photonics Technology Letters, IEEE 22(7), 507–509 (2010)
2. Cheng, Y., Larin, K.V.: Artificial fingerprint recognition by using optical coherence tomography with autocorrelation analysis. Appl. Opt. 45(36), 9238–9245 (2006)
3. Darlow, L.N., Connan, J., Singh, A.: Performance analysis of a hybrid fingerprint extracted from optical coherence tomography fingertip scans. In: 2016 International Conference on Biometrics (ICB). pp. 1–8 (June 2016)
4. Darlow, L.N., Connan, J.: Efficient internal and surface fingerprint extraction and blending using optical coherence tomography. Appl. Opt. 54(31), 9258–9268 (Nov 2015)
5. Darlow, L.N., Connan, J., Akhoury, S.S.: Internal fingerprint zone detection in optical coherence tomography fingertip scans. Journal of Electronic Imaging 24(2), 023027 (2015)
6. ISO/IEC 2382-37: Information technology – Vocabulary – Part 37: Biometrics (2012)
7. Khutlang, R., Khanyile, N.P., Makinana, S., Nelwamondo, F.V.: High resolution feature extraction from optical coherence tomography acquired internal fingerprint. In: 2016 17th IEEE/ACIS International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing (SNPD). pp. 637–641 (May 2016)
8. Meissner, S., Breithaupt, R., Koch, E.: Defense of fake fingerprint attacks using a swept source laser optical coherence tomography setup. In: Proc. SPIE, Frontiers in Ultrafast Optics: Biomedical, Scientific, and Industrial Applications. vol. 8611 (2013)
9. Menrath, M.: Fingerprint with OCT. Master's thesis, Fern-Universität Hagen in Cooperation with Bundesamt für Sicherheit in der Informationstechnik (BSI) (2011)
10. Mura, V., Ghiani, L., Marcialis, G.L., Roli, F., Yambay, D.A., Schuckers, S.A.: Livdet 2015 fingerprint liveness detection competition 2015. In: Biometrics Theory, Applications and Systems (BTAS), 2015 IEEE 7th International Conference on. pp. 1–6 (Sept 2015)
11. Nasiri-Avanaki, M.R., Meadway, A., Bradu, A., Khoshki, R.M., Hojjatoleslami, A., Podoleanu, A.G.: Anti-Spoof Reliable Biometry of Fingerprints Using En-Face Optical Coherence Tomography. Optics and Photonics Journal 1(3), 91–96 (2011)
12. Peterson, L.E., Larin, K.V.: Image classification of artificial fingerprints using Gabor wavelet filters, self-organising maps and Hermite/Laguerre neural networks. International Journal of Knowledge Engineering and Soft Data Paradigms 1(3), 239–256 (Oct 2009)
13. Sousedik, C., Breithaupt, R.: Full-fingerprint volumetric subsurface imaging using fourier-domain optical coherence tomography. In: accepted to Biometrics and Forensics (IWBF), 2017 International Workshop on (2017)
14. Sousedik, C., Busch, C.: Presentation attack detection methods for fingerprint recognition systems: a survey. IET Biometrics 3(4), 219–233 (2014)