Continuous 3D Face Authentication using RGB-D Cameras

Maurício Pamplona Segundo*[1,2], Sudeep Sarkar[1], Dmitry Goldgof[1], Luciano Silva[2], and Olga Bellon[2]

[1]Computer Science and Engineering, University of South Florida, USA.
[2]IMAGO Research Group, Universidade Federal do Paraná, Brazil ,
mauricio@inf.ufpr.br,{sarkar,goldgof}@cse.usf.edu,{luciano,olga}@ufpr.br

Abstract

We present a continuous 3D face authentication system that uses a RGB-D camera to monitor the accessing user and ensure that only the allowed user uses a protected system. At the best of our knowledge, this is the first system that uses 3D face images to accomplish such objective. By using depth images, we reduce the amount of user cooperation that is required by the previous continuous authentication works in the literature. We evaluated our system on four 40 minutes long videos with variations in facial expressions, occlusions and pose, and an equal error rate of 0.8% was achieved.

1. Introduction

For many years biometrics have been proposed as a substitute for common authentication methods, such as passwords and tokens [4]. However, in most authentication systems, once someone gets access to the desired resource no further verification is performed. Although these systems stop an unauthorized individual from getting access, they cannot ensure that the accessing user is the allowed one, which is not acceptable in high security environments. The continuous authentication addresses this issue by constantly monitoring accessing users to make sure no unauthorized access occurs after the initial verification. Its major advantage is to provide a more secure session, which may be used in computer access control [15] and online examinations [10], and only requires biometric samples to be captured continuously.

In this context, keystrokes appeared as the most straightforward feature for continuous authentication and were the first biometric trait to be used for this purpose [11, 14, 15]. Although the use of keystrokes for continuous authentication does not require additional hardware in a traditional computer configuration, according to Monaco *et al.* [15] it requires more than 200 keystrokes to identify an impostor (*i.e.* at least one minute considering an average computer user). However, as pointed out by Sim *et al.* [19], impostors can damage a protected system with much less effort (*e.g.* the command line **"rm -rf *"** in a Linux console can be typed in a few seconds). To overcome this problem, different biometric features with a higher discriminant power were employed, such as electrocardiograms (ECG) [1], faces [13, 16] and fingerprints [19], as well as multimodal systems [2, 9, 19]. Despite the advantages in accuracy, fingerprint-based systems cannot obtain samples continuously without user cooperation making the continuous authentication too inconvenient for the user, and ECG biometrics require users to wear body sensors and can reveal other information than the identity (*e.g.* health conditions such as arrhythmia [1] and stress).

Facial images can be captured without any user cooperation by low-cost cameras, which are built-in in most of today's computers. However, face recognition based on 2D images is substantially affected by pose, illumination and facial expression variations [21]. To avoid these variations Niinuma *et al.* [16] introduced the concept of soft biometrics, which are color distributions of faces and clothes. This type of description is, however, less discriminant and easier to mimic.

In this work we propose using a RGB-D camera to perform 3D face authentication continuously, since 3D outperforms 2D face recognition in many aspects [5]. First, pose robustness is better achieved when 3D data is available. Second, the Kinect is able to capture 3D images in a wide range of lighting conditions. Finally, the 3D data allows a better classification of foreground and background objects, which facilitates tasks like object detection and tracking. The major drawback of 3D face authentication is the computational cost. However, if the cost of an unauthorized access is too high, then continuous 3D face authentication will

*This work was performed while the author was at the Computer Science and Engineering, University of South Florida, USA.

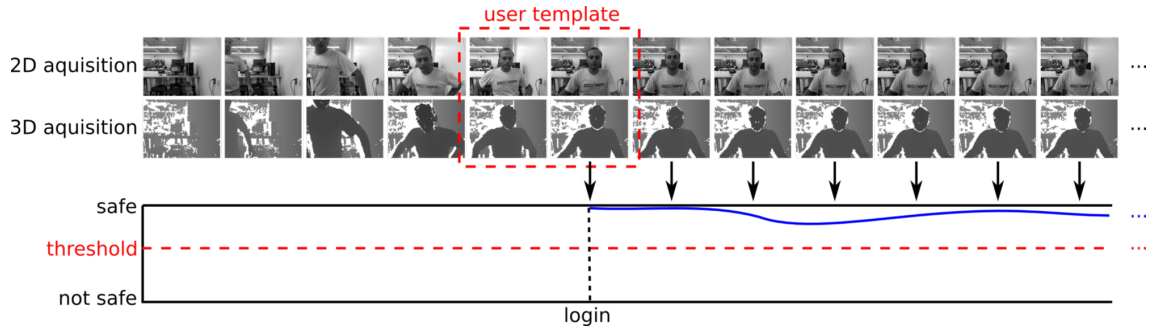


Figure 1. Illustration of the operation of a continuous face authentication system using the Kinect for data acquisition.

be worth being used.

A Microsoft Kinect sensor[1] was used for acquisition, but other RGB-D cameras could be used. The proposed system uses a 3D face detector [17] based on boosted cascade classifiers to locate faces under pose variation. Then, faces are then normalized to a standard pose through the Iterative Closest Points algorithm [3], and Histogram of Oriented Gradients (HOG) features [8] are extracted from three different facial regions. For each frame, only the region least affected by noise is used for matching, which is automatically defined based on facial pose information. Finally, the obtained scores are fused over time to take a decision on the safety of the system.

This paper is organized as follows: Section 2 describes our proposed approach for continuous 3D face authentication. Section 3 shows our experimental results using four 40 minutes long videos acquired by a Microsoft Kinect sensor. Finally, Section 4 presents our conclusions followed by acknowledgment and references.

2. Continuous 3D face authentication

The proposed system uses a RGB-D camera to continuously capture RGB-D images, which contain both color and depth information. However, we ignore the color information in order to avoid its limitations concerning pose and illumination variations. Figure 1 illustrates the operation of the proposed system. The system is assumed to be safe at login, so we take N frames at this point to be used as the user template, with $N = 3$. Then, each following frame is processed and matched against the template, and the resulting score is used to update the probability of the system being safe. If this probability is below a threshold value, the system is considered unsafe and the user loses access immediately.

Each frame is processed through five stages after acquisition, as shown in Figure 2: (1) face detection and pose estimation; (2) face normalization; (3) region of interest (ROI) extraction; (4) HOG feature computation; and (5) matching and score fusion. More details about each stage are given in the following subsections.

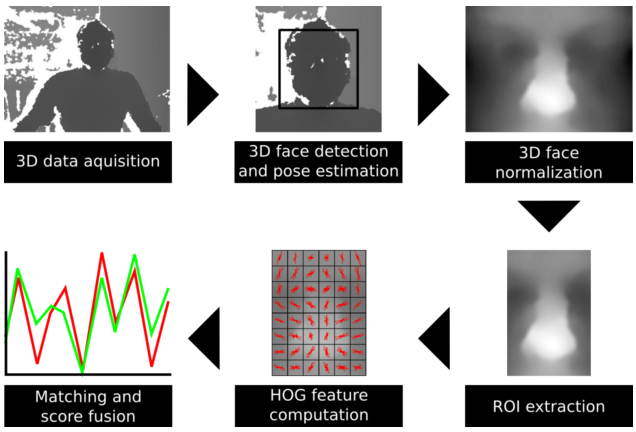


Figure 2. Diagram of the stages used in this work to process an input depth image.

2.1. 3D data acquisition

The Microsoft Kinect RGB-D sensor is based on structured light and captures up to 30 frames per second (fps) of color and depth information. Depth values range from 500 to 4,000 millimeters (mm). Although the Kinect presents a good relation between accuracy, speed and cost, the accuracy depends on the distance between object and sensor [12]. As may be seen in Figure 3, there are more disparity values to represent small distances than large distances. This causes the error of depth measurements to grow with increasing distance, as also shown in Figure 3. Due to this problem, in this work we only use faces up to $1500mm$ away from the acquisition device for recognition purposes.

2.2. 3D face detection and pose estimation

The detection stage is performed by applying a boosted cascade classifier of Haar features to classify image regions as face or non-face [20]. However, instead of using color images for this task, depth images are used because they are invariant to pose and illumination variations. To this

[1] www.xbox.com/kinect

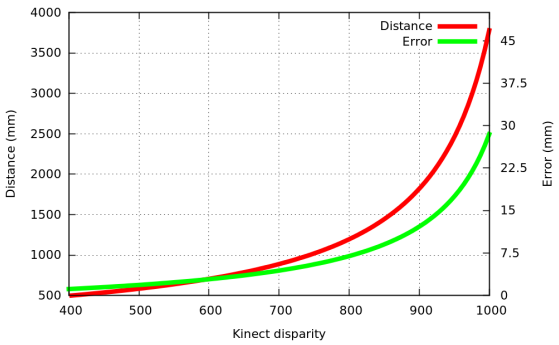
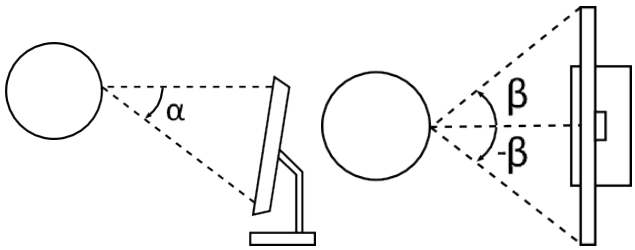


Figure 3. Kinect disparity versus distance/error in millimeters.

end, we use the approach proposed by Pamplona Segundo *et al.* [17] in which a scale-invariant projection image of the depth information is used to represent faces with size proportional to their real size. Thus, the search is limited to a predefined face size and the detection process is significantly speeded up. Also, since there are less non-face candidates (*i.e.* only regions with size equal to the face size are tested) the probability of having a false alarm is reduced.

This idea can also be used to detect faces under pose variation using only a frontal face classifier. To this end, multiple projection images are created from different viewpoints in order to represent rotated faces as frontal faces. In this work, we only considered viewpoint changes around x- and y-axes because pitch (see Figure 4(a)) and yaw (see Figure 4(b)) rotations are the most common pose variations of a regular computer user, as illustrated in Figure 4. The parameters α and β are the maximum values for pitch and yaw rotations, respectively. In this work, $\alpha = 40$ and $\beta = 20$. Projection images were created for all viewpoints within the range specified by α and β at 10 degrees steps, and the detection result is also used to obtain a rough estimation of the head pose. This estimation is given by the rotation values of the viewpoint in which the face was detected.



(a) Side view (b) Top view

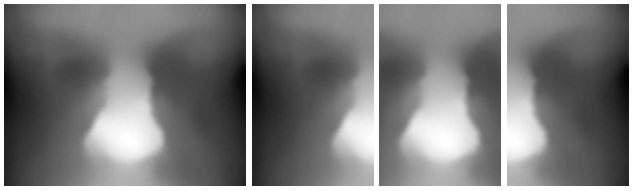
Figure 4. Common pose variations of a regular computer user: (a) pitch and (b) yaw.

2.3. 3D face normalization and ROI extraction

In the normalization stage, the detected face is aligned to an average face image using the ICP algorithm [3] to standardize pose and resolution. A noise-free average face $\mathbf{\Psi}$ is computed using the images of the Face Recognition Grand Challenge (FRGC) v1.0 database [18], which contains 943 images from 275 different subjects. To this end, first all FRGC v1.0 images $\mathbf{\Gamma}_1, \mathbf{\Gamma}_2, \ldots, \mathbf{\Gamma}_M$ are aligned to an initial estimation $\mathbf{\Psi}'$ using ICP. After the alignment, we have the images $\mathbf{\Gamma}'_1, \mathbf{\Gamma}'_2, \ldots, \mathbf{\Gamma}'_M$ in the same coordinate system of $\mathbf{\Psi}'$. With these images we compute the residual vectors $\mathbf{\Phi}'_i = \mathbf{\Gamma}'_i - \mathbf{\Psi}'$, where each value in $\mathbf{\Phi}'_i$ is the distance in the Z-axis between one point in $\mathbf{\Psi}'$ and its closest point in $\mathbf{\Gamma}'_i$. Then, we recompute $\mathbf{\Psi}'$ using the Equation 1 and repeat the entire process until convergence. Finally, the last $\mathbf{\Psi}'$ is assigned to $\mathbf{\Psi}$.

$$\mathbf{\Psi}' = \mathbf{\Psi}' + \frac{1}{M}\sum_{i=1}^{M}\mathbf{\Phi}'_i \qquad (1)$$

The initial value of $\mathbf{\Psi}'$ is given by the first training image resampled on a uniform grid with resolution of $1mm$. The grid is centered in the nose and eyes area and has size of $96\times72mm$, totaling 97×73 points. The result of the normalization stage for a Kinect image is shown in Figure 5(a).



(a) (b) (c) (d)

Figure 5. (a) Example of resulting face image after normalization, and its different ROIs: (b) left region, (c) nose region and (d) right region.

It is not always possible to use the entire face image obtained through the normalization stage because pose variations can substantially affect one side of the face. When this happens, the affected side may present holes and excessive noise due to self-occlusions in the face. To solve this problem, we divide each image in three different ROIs: the left half of the face, the nose region and the right half of the face, respectively shown in Figures 5(b), 5(c) and 5(d). We only use one of these regions for each frame according to its pose, which is obtained in the detection stage. The nose ROI is used for frontal faces, while we use the left ROI when the user is looking to the right and the right ROI when the user is looking to the left. This way we avoid using too noisy image parts and also use the most invariant facial region, according to Chang *et al.* [6], when frontal faces are available.

2.4. HOG feature computation and matching

We use HOG features to describe ROIs because they showed themselves more invariant than the ROI image itself

in our experiments. Each ROI images is scaled to 64×64 pixels, and then the HOG feature is extracted considering a 16×16 block size, 8×8 cell size, 9 orientation bins and a step size of 8 pixels, resulting in a feature vector with 1764 elements. The matching score between a probe image and the user template is equal to the L_1 between their correspondent feature vectors. If the user template has more than one image, the probe image is matched against all of them and the median score is returned instead.

2.5. Score fusion

The objective of this stage is to determine the probability of the system being safe at time t from the history of observations $\mathcal{Z}_t$, called P_{safe}. Each observation $z_i \in \mathcal{Z}_t$ corresponds to the matching score between a probe image and the user template at time i. The fusion of continuous scores is based on the Temporal-First integration proposed by Sim *et al.* [19], which keeps track of P_{safe} over time with a weighted sum of $\mathcal{Z}_t$. In this fusion scheme, older observations are "forgotten" to ensure the current user is still the allowed one and the probability of the system being safe can be computed at any time, even when there is no observation. Equation 2 is used to compute P_{safe}:

$$P_{safe} = \frac{e^{\frac{-\Delta t \times \ln 2}{k}} \times P(safe \mid \mathcal{Z}_t)}{\sum_{x \in X} P(x \mid \mathcal{Z}_t)} \quad (2)$$

where k is the decay rate that defines how fast the system "forgets" older observations (*i.e.* P_{safe} drop to half every k seconds without observations, $k = 15$), Δt is the elapsed time since the last observation z_t, and $X = \{safe, \neg safe\}$. For every $x \in X$, $P(x \mid \mathcal{Z}_t)$ is given by Equation 3:

$$P(x \mid \mathcal{Z}_t) = P(z_t \mid x) \times e^{\frac{(u-t) \times \ln 2}{k}} \times P(x \mid Z_u) \quad (3)$$

where u is the time of the last observation before t, z_u. The system is assumed to be safe at the login time, so $P(safe \mid \mathcal{Z}_0) = 1$ and $P(\neg safe \mid \mathcal{Z}_0) = 0$. $P(safe \mid z_i)$ and $P(\neg safe \mid z_i)$ are respectively given by intraclass and interclass cumulative distribution functions (CDF), presented in Equations 4 and 5.

$$P(z_i \mid safe) = 1 - \frac{1}{2}\left[1 + erf\left(\frac{score - \mu_{safe}}{\sigma_{safe}\sqrt{2}}\right)\right] \quad (4)$$

$$P(z_i \mid \neg safe) = \frac{1}{2}\left[1 + erf\left(\frac{score - \mu_{\neg safe}}{\sigma_{\neg safe}\sqrt{2}}\right)\right] \quad (5)$$

The parameters $(\mu_{safe}, \sigma_{safe}, \mu_{\neg safe}, \sigma_{\neg safe})$ were obtained for each ROI, and the resulting CDFs are shown in Figure 6. An exhaustive search was performed to obtain the set of parameters that minimizes the error in our experiments. The respective values of $(\mu_{safe}, \sigma_{safe}, \mu_{\neg safe}, \sigma_{\neg safe})$ for the left ROI, the nose ROI and the right ROI are $(89.0, 14.5, 128.3, 17.8)$, $(82.5, 13.2, 122.4, 16.2)$, and $(88.8, 12.9, 129.2, 17.4)$. Since only one ROI is used per frame, only its repective CDFs are used in Equation 3.

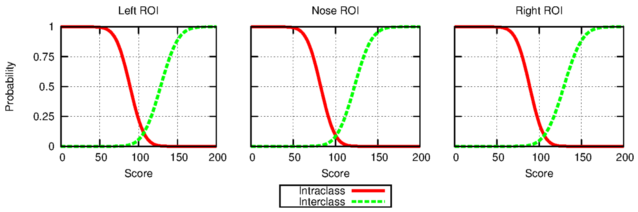


Figure 6. CDFs for the three ROIs employed in this work: left ROI, nose ROI and right ROI.

Our changes in Sim *et al.*'s formulation eliminate the need for keeping a history of observations and also avoids a continuous decrease in the P_{safe} value in the first k seconds after login.

3. Experimental results

For our experiments, we use four 40 minutes long videos acquired by a Kinect sensor. In these videos, the user appears in the scene, logs in the system, uses the computer for approximately 40 minutes and then leaves the scene. The videos were cut so that the first frame shows the user entering the scene and the last picture shows the user leaving the scene. No restrictions were imposed on how the user should use the computer and how the user should behave in front of the computer, but users were not allowed to leave the computer before 40 minutes have passed. Each video sequence has more than 70,000 frames and contains faces with different artifacts that may affect the authentication performance: facial expressions, occlusions, pose and noise. Some examples of these artifacts are shown in Figure 7.

(a) (b) (c) (d)

Figure 7. Examples of artifacts present in Kinect videos: (a) facial expressions, (b)-(c) occlusions and (d) pose.

Each video was used as input for the proposed continuous authentication system, and the results are shown as solid lines in Figure 8. About 2 hours and 40 minutes of authorized access were analyzed, and the system was able to keep the users with high P_{safe} values (*i.e.* above 0.8 in 95% of the frames). After that, we concatenated each video to the

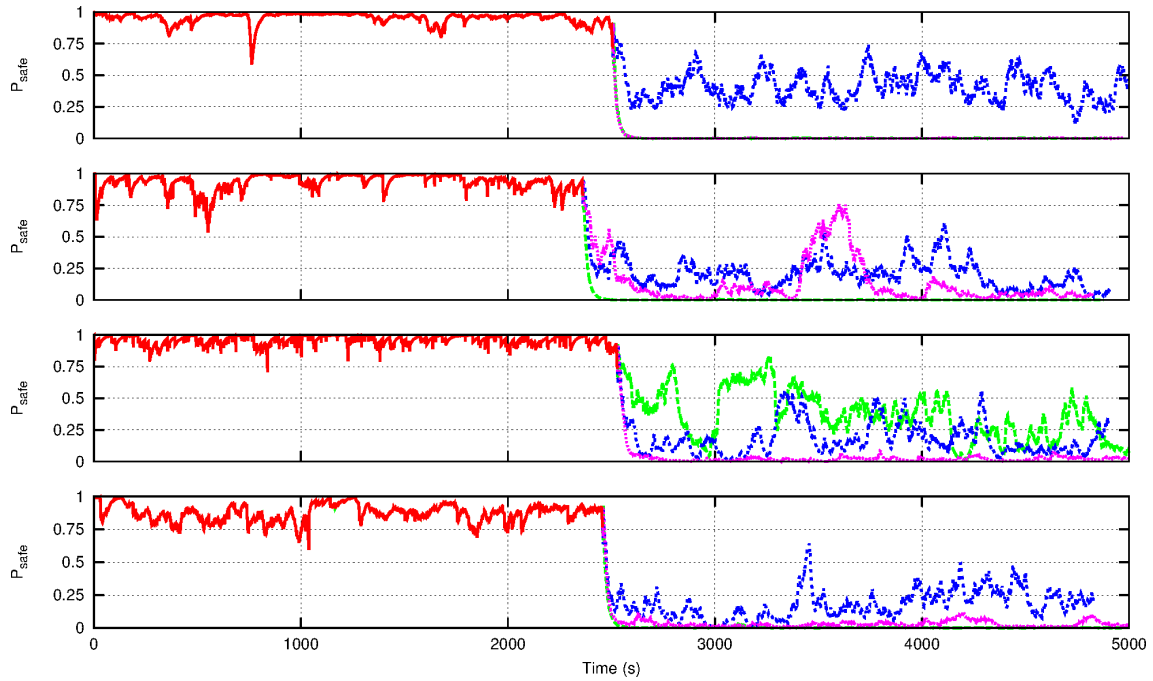


Figure 8. Each plot presents the results for the proposed continuous authentication system for a different subject. The solid line represents the authorized user accessing the computer in the initial 40 minutes, and the dashed lines represent the attacks by other subjects starting around 2500s time interval.

end of the remaining videos to simulate attacks and make sure the proposed system is able to detect intruders right after the authorized user leaves the scene. The 12 attacks were then performed (*i.e.* three for each video), and the results are also shown in Figure 8 as dashed lines. A total of 8 hours of intruder trying to get access were considered, and, as may be observed, the P_{safe} value for the authorized user is constantly higher than the P_{safe} value for intruders. This result is corroborated by the receiver operating characteristic (ROC) curve of the P_{safe} values shown in Figure 9, in which an equal error rate (EER) of 0.8% is achieved.

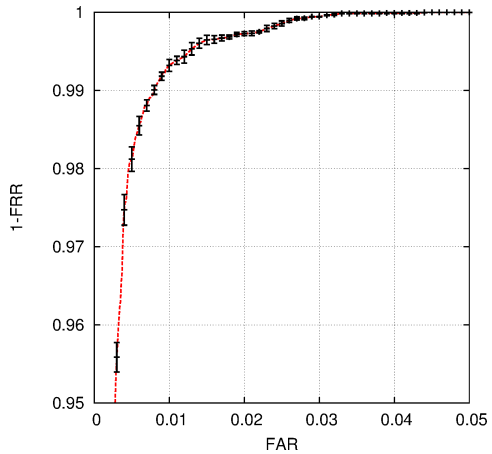


Figure 9. ROC curve of the P_{safe} values obtained by our continuous authentication system (see Figure 8).

Finally, we present an intuitive way to analyze the potential of the system to detect intruders. We consider the initial frame of each video that was concatenated to another video as the beginning of the attack. Then, for a given threshold value, we can see how long the system takes to identify the threat (*i.e.* how many seconds P_{safe} takes to go below the threshold) as presented in Figure 10. The solid line was obtained using the EER threshold, which is equal to 0.715. In this experiment, 75% of the attacks are detected in the first second. However, in one cases the system takes 19 seconds to detect the intruder. This time can be reduced by increasing the threshold, at the cost of increasing the FRR. Figure 10 shows in dashed lines an example of the results for a higher threshold (0.758). Although 91.7% of the attacks are detected in the first second and the worst case only takes 8 seconds to be detected, the EER grows from 0.8% to 2%.

Our experiments were performed at a frame rate of 1 fps in an Intel Core i3 processor, and the remaining frames were discarded by the system. No parallelism was employed to achieve real-time continuous authentication.

4. Conclusion

At the best of our knowledge, this is the first continuous authentication system that uses 3D face images to monitor and ensure that the accessing user is the allowed one. The acquisition was performed by a Kinect sensor, but the system can be used with other RGB-D cameras. The proposed

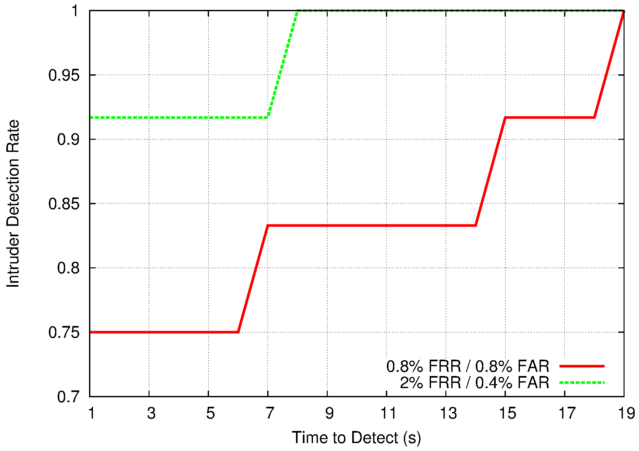


Figure 10. Intruder detection rate versus time to detect an intruder: as the time to detect increases, so does the intruder detection rate.

approach automatically detects, normalizes, describes and matches depth images in real-time. Although depth images are invariant to pose, such variations may cause holes and noise due to facial self-occlusions. To solve this problem, in this work we match different regions of the face depending on which facial parts are clearly visible. In the fusion stage, we present an updated version of Sim *et al.*'s Temporal-First integration [19] that does not require to keep a history of observations and better controls P_{safe} in the initial part of the continuous authentication process.

More than 2 hours and 40 minutes of genuine accesses and over 8 hours of impostors trying to get access to the system were evaluated in our experiments. The proposed approach obtained a 0.8% EER and was able to detect most of the intruders within a one-second window. We also present a more intuitive way to evaluate the security of the system (see Figure 10) by plotting the intruder detection rate along time for different FRR/FAR values.

As a future work, we intend to combine both color and depth images in a way that it does not decrease the performance of the system when the color image is being affected by changes in illumination and/or pose. We also intend to replace the CDFs in the fusion stage with a more robust classification method, such as Support Vector Machines [7].

Acknowledgment

The authors would like to thank Drs. P. J. Phillips, K. W. Bowyer and P. J. Flynn for allowing them to use the FRGC images. This work was partially supported by CNPq (202042/2012-0 SWE), CAPES, and UFPR.

References

[1] F. Agrafioti and D. Hatzinakos. ECG biometric analysis in cardiac irregularity conditions. *Signal, Image and Video Processing*, 3(4):329–343, 2009.

[2] A. Altinok and M. Turk. Temporal integration for continuous multimodal biometrics. In *Workshop on Multimodal User Authentication*, pages 131–137, 2003.

[3] P. J. Besl and H. D. Mckay. A method for registration of 3-d shapes. *IEEE PAMI*, 14(2):239–256, 1992.

[4] R. Bolle, J. Connell, S. Pankanti, N. Ratha, and A. Senior. *Guide to Biometrics*. SpringerVerlag, 2003.

[5] K. W. Bowyer, K. Chang, and P. Flynn. A survey of approaches and challenges in 3D and multi-modal 3D + 2D face recognition. *CVIU*, 101(1):1–15, 2006.

[6] K. I. Chang, K. W. Bowyer, and P. J. Flynn. Multiple nose region matching for 3d face recognition under varying facial expression. *IEEE PAMI*, 28(10):1695–1700, 2006.

[7] C. Cortes and V. Vapnik. Support-vector networks. *Machine Learning*, 20(3):273–297, 1995.

[8] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *IEEE CVPR*, pages 886–893, 2005.

[9] I. G. Damousis, D. Tzovaras, and E. Bekiaris. Unobtrusive multimodal biometric authentication: the HUMABIO project concept. *EURASIP Journal on Advances in Signal Processing*, 2008:110:1–110:11, 2008.

[10] E. Flior and K. Kowalski. Continuous biometric user authentication in online examinations. In *7th Int'l Conf. on Info. Technology: New Generations*, pages 488–492, 2010.

[11] D. Gunetti and C. Picardi. Keystroke analysis of free text. *ACM Trans. Info. and System Security*, 8(3):312–347, 2005.

[12] D. Herrera C., J. Kannala, and J. Heikkilä. Joint depth and color camera calibration with distortion correction. *IEEE PAMI*, 34(10):2058–2064, 2012.

[13] R. Janakiraman, S. Kumar, S. Zhang, and T. Sim. Using continuous face verification to improve desktop security. In *7th IEEE WACV*, pages 501–507, 2005.

[14] J. Leggett, G. Williams, M. Usnick, and M. Longnecker. Dynamic identity verification via keystroke characteristics. *Int'l Jrnl. of Man-Machine Studies*, 35(6):859–870, 1991.

[15] J. V. Monaco, N. Bakelman, S.-H. Cha, and C. C. Tappert. Developing a keystroke biometric system for continual authentication of computer users. In *European Intelligence and Security Informatics Conference*, pages 210–216, 2012.

[16] K. Niinuma, U. Park, and A. K. Jain. Soft biometric traits for continuous user authentication. *IEEE Trans. Information Forensics and Security*, 5(4):771–780, 2010.

[17] M. Pamplona Segundo, L. Silva, and O. R. P. Bellon. Real-time scale-invariant face detection on range images. In *IEEE SMC*, pages 914–919, 2011.

[18] P. J. Phillips, P. J. Flynn, T. Scruggs, K. W. Bowyer, J. Chang, K. Hoffman, J. Marques, J. Min, and W. Worek. Overview of the face recognition grand challenge. In *IEEE CVPR*, pages 947–954, 2005.

[19] T. Sim, S. Zhang, R. Janakiraman, and S. Kumar. Continuous verification using multimodal biometrics. *IEEE PAMI*, 29(4):687–700, 2007.

[20] P. Viola and M. J. Jones. Robust real-time face detection. *Int. Jrnl. of Computer Vision*, 57(2):137–154, 2004.

[21] W. Zhao, R. Chellappa, P. J. Phillips, and A. Rosenfeld. Face recognition: A literature survey. *ACM Computing Surveys*, 35(4):399–458, 2003.