Detecting Sensor Level Spoof Attacks Using Joint Encoding of Temporal and Spatial Features

Jun Liu, Ajay Kumar
Department of Computing, The Hong Kong Polytechnic University, Hong Kong

***Abstract*— Automated detection of sensor level spoof attacks is critical for success of surveillance technologies and the detection of crimes. This paper presents a new approach to more accurately detect such spoof face presentation attempts during the surveillance. The image data from the spoof samples not only illustrates the subtle texture difference in the spatial domain but is also accompanied by temporal differences as compared to those from the live/real human samples. Therefore, we investigate a new approach to encode the sparsity of these two different categories while combining these two cues from the acquired image sequences. The spatial and temporal information in the acquired image data and the sparse dictionary approach are used to encode such information in effectively separating the spoof and the live categories. This approach does not require large number of training samples, such as for the deep learning based methods, while achieving very good performance. The experimental results presented in this paper on publicly available database achieve outperforming results and illustrate the effectiveness of the proposed approach.**

Keywords—spoof attacks, sparse coding, biometrics, face mask detection, antispoofing techniques

1. INTRODUCTION

Development of effective anti-spoofing techniques is the key to safeguard integrity of biometrics systems that are increasingly deployed to detect and prevent crimes. Conventional biometrics research has been largely directed on the development of highly accurate algorithms that can efficiently identify the matched identities. Such methods essentially require the identification and recovery of discriminative features that are unique for different subjects for the recognition and classification problems, such as face recognition[1], iris recognition, and fingerprint recognition. The similarity of these research is to extract individual features that are unique to different subjects, since different individuals possess different genetic information that often translate into different physiological features. However, when it comes to the development of effective anti-spoofing techniques to thwart sensor-level attacks, the research efforts in a different direction is required. This is largely due to the fact that intruders are always expected to imitate these unique biometric details that would lead to positive but undesirable identification by the biometrics system.

Successful examples of anti-spoofing results [2] are available for a range of deployed biometrics systems with different modalities: fingerprint spoofing detection [3]-[4], face spoofing detection [5], [12], facial disguise and makeup detection [6], [15], *etc*. The development of effective anti-spoofing techniques is critical to safeguard the integrity of deployed biometrics systems. This paper focuses on such problem and its scope has been limited to the development of accurate method to detect 2D face spoofing attempts.

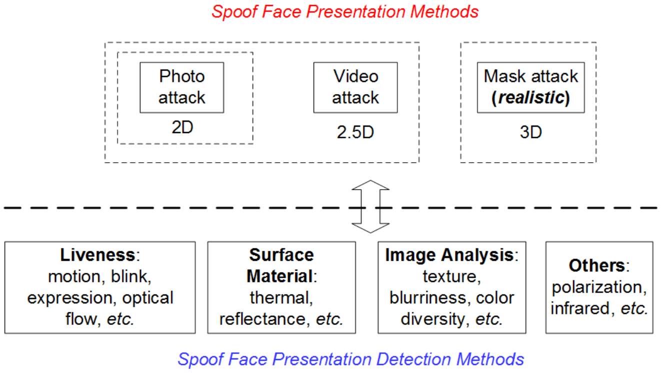
Spoof Face Presentation Methods
Photo attack
2D
Video attack
2.5D
Mask attack (realistic)
3D
Liveness: motion, blink, expression, optical flow, etc.
Surface Material: thermal, reflectance, etc.
Image Analysis: texture, blurriness, color diversity, etc.
Others: polarization, infrared, etc.
Spoof Face Presentation Detection Methods

Figure 1: Classification of popular spoof face presentation attacks and the spoof face detection methods attempted in the literature.

Reference [7], provides an insightful survey on available methods for the detecting sensor level attacks for face recognition systems. Depending on the materials that are used to develop spoof replica of biometric impressions, spoofing can be classified into three categories: 2D spoofing (printed face image attack, image displayed by the screen, *etc*.), 2.5D spoofing (replay of real person's video that is sensed by the face recognition camera), and 3D spoofing (wearable 3D face prototype of real person). The methods for detecting spoofing can be classified as liveness based approaches, material based approaches, image based approaches, *etc*. This classification is summarized in Figure 1. Among all types of spoof face presentation ng attacks, the attack using spoof 3D face mask is believed to be more realistic and difficult to detect. A sample of more realistic 3D face mask image and corresponding real face image from the same person is shown in Figure 2. The types of 3D masks can be significantly different and each of these masks can have different kinds of shape, material composition, texture, albedo, *etc*. Therefore, in the best of our knowledge there is no universal or effective method that can be used to detect spoof face attacks from large variety of 3D face masks.

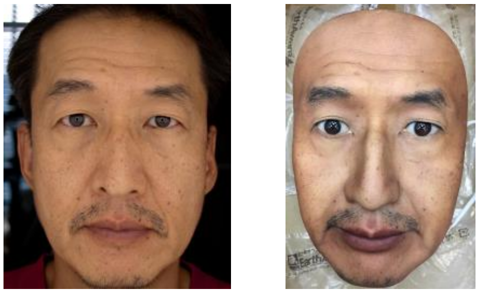

Figure 2: Image samples from a typical face mask with corresponding real face sample (on left is the image from real face while on right is the artificial mask image corresponding to real person shown on the left)

This paper focuses on the detection of sensor level 2D spoof face attacks and uses publicly available database that can allow us to make useful comparisons and conclusions.

Automated detection of spoof attempts for face recognition systems is increasingly inviting attention of researchers, and several promising attempts are available in the literature. Reference [8], utilize different kinds of Local Binary Patterns (LBP) [9] for the detection of spoof face samples. However the error rate achieved from the experiments in this reference is quite high, *i.e*., similarity between the real face samples and the spoof samples, the HTER (Half Total Error Rate) is in the range of 13.87% to 17.17% for the replay-attack database [8]. Using a similar but multi-scale approach, the authors in [10] report superior performance and demonstrate outperforming results. The image sequence acquired from the presentation attempts also encode temporal relationship among the image frames. In order to investigate usage of such temporal information, in addition to the spatial information, reference [13] recovers features using Local Binary Patterns on Three Orthogonal Planes (LBP-TOP) [14] to discriminate the spoof faces images from the real faces. This reference achieves promising results with HTER performance of 7.6% when the Support Vector Machine (SVM) classifier is employed. The error rates achieved from this approach using publicly available database are still quite high and significant improvement is required to for its usage in real applications.

In addition, there are some other spoof face detection methods that measure liveness during the presentation. Biological signals, such as eye blink, lip/mouth movement, can also provide useful characteristics of real faces. In [5], the authors use relative difference between the subject and the background to capture the difference between spoof face and real face. The HTER of this method ranges from 7.27% to 11.03% under different protocols. There are also some other interesting methods that attempt to explore image reflection, blurriness, color diversity, *etc*. In [16], the authors use Fourier spectra analysis to detect the liveness of face. In [17], the authors combine these information together to achieve spoofing detection. However, the effectiveness of these methods are quite limited, particularly when higher resolution or quality of spoof face samples are presented.

This paper develops a more accurate approach for the automated detection of 2D spoof face attacks. The main contribution of this paper lies in effectively utilizing the sparsity of spoof samples and the real face samples by simultaneously encoding these two categories, which can incorporate the spatial and temporal domain information at the same time. Our experimental results on publicly available database achieve outperforming results and demonstrate effectiveness of such an approach for the detection of spoof face presentation attacks.

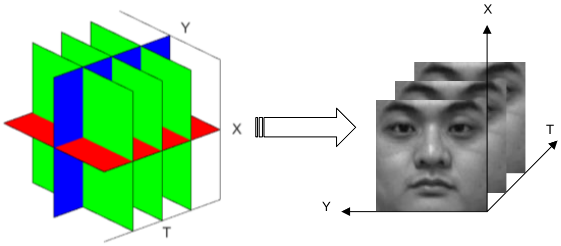

Figure 3: Constructing three planes (XY, XT, YT) using the image sequences.

The rest of this paper is organized as follows: section 2 provides details on the extraction of spatial and temporal information and section 3 describes the usage of sparse encoding method to classify spoof face attack images from the real face images. The experimental results and comparison with other methods appear in section 4 while conclusions from this work is summarized in section 5.

2. Feature Extraction

Since the observed textured-like details on images acquired from the real faces and those from the printed face attack images are different, the local binary patterns can be used to encode such spatial domain information. In addition to such information which is recovered from the spatial domain, we can also analyze temporal domain information to capture the difference between these two categories. The location of key facial points in a printed fake face is expected to be always constrained in the *same* 2D plane, while in case of real face these locations are located at different points in 3D space. Therefore, the temporal domain information, which is recovered from the pixel difference along the time axis, is expected to be different when comparing these two categories. The volume local binary pattern (VLBP) [14] is a promising method for effectively encoding the consecutive frames in an image sequence. However, with the increase in number of sampling points, the number of observed patterns also grows exponentially.

Therefore, in order to utilize the spatial and temporal information at the same time with an acceptable computational time, we investigate dynamic features in three orthogonal planes. This type of feature is expected to recover the information from the following three orthogonal different planes:

XY plane: the texture details of the current image (spatial domain);

XT plane: the dynamic texture in time domain along *x* direction (temporal domain), where T stands for time domain;

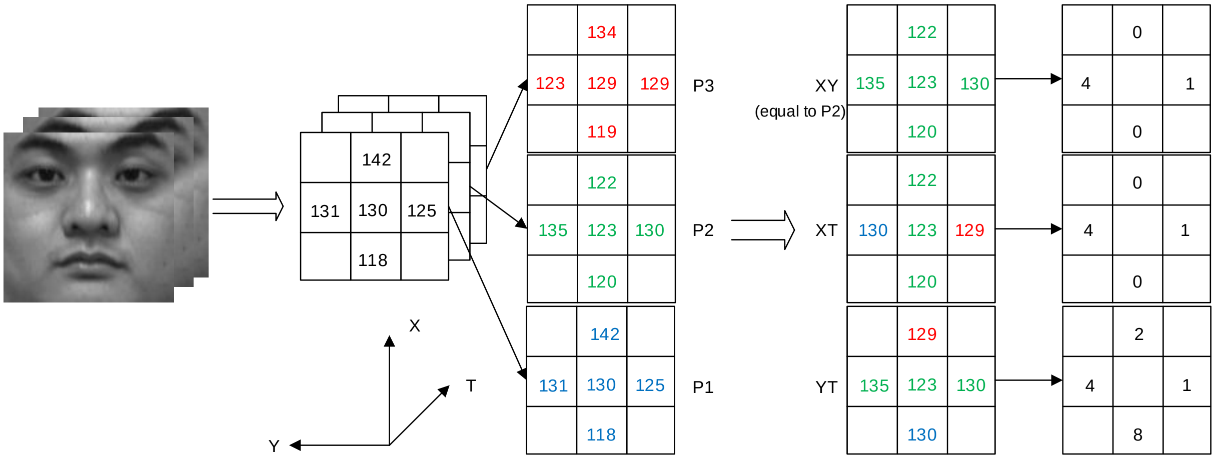

Figure 4: Dynamic feature computation using three consecutive frames (L = 1, P = 4, R = 1).

YT plane: the dynamic texture in time domain along y direction (temporal domain).

In order to ensure the use of features from the entire video clip or the image sequence, we firstly construct a group, which consists of consecutive images with the same time interval. Figure 3 illustrates the definition of time axis T, and spatial axis X and Y in a given image sequence. P1 (blue plane), P2 (green plane), and P3 (red plane) are three independent frames with same time interval. For every input image sequence, three or more consecutive images can be combined into a group to compute the dynamic binary feature. In the first step, these three frames in spatial domain need to be re-ordered into another three different planes: XY plane, XT plane, and YT plane, where T is used here to represent the time axis for the time domain, X and Y are two spatial directions of images in the spatial domain. Figure 4 provides more detailed information on this construction process. The pixel composition in XY plane is equal to the pixel composition in P2 plane (the middle frame of the original image sequence). Then, the XT plane is a combination of pixels in x direction and T direction of original image sequence. In the same manner, YT plane is a combination of pixels in y and T directions. In this manner, spatial domain and temporal domain information can be encoded at the same time. For these three planes (XY, XT and YT), three LBP descriptors can be computed separately for these three reordered planes. The process can be expressed as follows:

$$x_{\phi} = \sum_{p=1}^{P} f(v_p - v_0)2^p, \phi \in \{\mathrm{XY}, \mathrm{XT}, \mathrm{YT}\} \qquad (1)$$

where ϕ represents three orthogonal planes XY, XT and YT rather than three independent planes P1, P2, and P3, p is the index of sampling point, v_p and v_0 are the gray level values of surrounding points and center point respectively while the function $f(v)$ is used to encode the gray level difference between the surrounding neighbors to the center of the sampling circle as follows:

$$f(v) = \begin{cases} 1, & v \geq 0 \\ 0, & v < 0 \end{cases} \qquad (2)$$

In our experiments, time interval between the consecutive frames L is set to 1, the number of points in each plane P is set to 8, and the radius of sampling circle R is 1.

As detailed above, three respective vectors (histogram of LBP features) can be computed to encode the given image sequence or the *video* data. Therefore, if the size on axis ‘A’ (where A refers to X, Y, or T) is a, then the number of vectors in the plane ‘BC’ (BC is ‘YT’ or ‘XT’ or ‘XY’, respectively) is a-1. If we average all the vectors and concatenate their result, a F (F = 3f, f is the dimension of the histogram of LBP) dimension feature vector can be extracted to represent the whole image sequence or *video* data in the database. Consequently, this feature is expected to robustly define the spatial and temporal information from the given video clip or the image sequence in a 3D space.

3. Sparse Encoding of Recovered Features

In order to utilize the similarity within categories, sparse coding approach [18] is used for encoding this similarity. All the inputs or available training samples can be used to firstly construct a dictionary, which consists of the features extracted from these samples. Any unknown test sample can be reconstructed by the atoms in the dictionary with a residual error. Figure 5 illustrated the schematic diagram of such sparse coding process. The dictionary is constructed by real faces features and spoof face attack images features, from the training set. This training set consists of M samples of fake faces and N samples of the real faces, the number of atoms is K (K = M + N). Therefore, the resulting sparse dictionary $x \in \Re^{F*K}$ can be expressed as follows:

$$\begin{aligned} X &= [x_{a,1}, x_{a,2}, \ldots, x_{a,m}, \ldots, x_{a,M}; \\ &\quad x_{r,1}, x_{r,2}, \ldots, x_{r,n}, \ldots x_{r,N}] \qquad (3) \\ &= [x_1, x_2, \ldots, x_k, \ldots, x_K] \end{aligned}$$

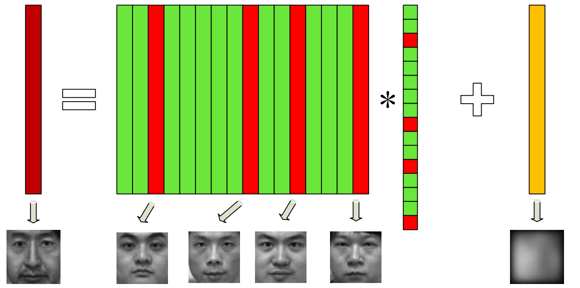

Figure 5: Illustration on reconstructing the features for the unknown test sample using atoms with a residual error (Note that this figure is just for the illustration and not generated from the real data).

where $x_{a,m} \in \Re^F$ represents the m^{th} atom in the fake face part and $x_{r,n} \in \Re^F$ represents the n^{th} atom in the real face part. In order to simplify the notations, the above two parts are merged together as x_k, which simply represents the k^{th} atom in the full dictionary.

Once the training dictionary is constructed, any new feature y generated from the unknown test image sequence or video clip, can be considered as the linear combination that is spanned by atoms in the dictionary with residual error. Therefore, the unknown *spoof* test sample is expected to generate high response from the spoof part of this dictionary, while the real face test sample is expected to generate higher response from the real face part of this dictionary. The linear combination of features from the unknown test sample can be expressed as follows:

$$\begin{aligned} y &= \sum_{k=1}^{K} x_k w_k + \varepsilon \\ &= Xw + \varepsilon \end{aligned} \tag{4}$$

where k is the index of each atom in the dictionary, x_k is the atom in the dictionary, ε is the residual term, and w is the vector representation for the corresponding weights recovered from dictionary. Figure 5 schematically illustrates the reconstruction of feature from the unknown test sample using sparse atoms with the additional residual term.

In order to capture the sparsity of training samples to represent the unknown test sample, l_1 constraint is added to the weight of each atom:

$$l_1 : w' = \arg\min \|w\|_1, \ \textit{subject to } y = Xw \tag{5}$$

Therefore, the sparse encoding representation, which ensures constraint on the training dictionary, can be formulated as follows:

$$\min \sum_{k=1}^{K} \|y - x_k w_k\|_2^2 + \lambda \sum_{k=1}^{K} \|w_k\|_1 \tag{6}$$

where λ is the regularization parameter. In this manner, the spoof face sample detection problem can be considered as a solution to the two-class classification problem. Several methods can be used to solve such optimization problem, *e.g.* greedy methods (orthogonal matching pursuit (OMP)), constrained optimization methods (gradient projection sparse reconstruction (GPSR), *etc.*), proximity optimization methods (primal augmented Lagrangian method (PALM), dual augmented Lagrangian method (DALM), *etc.*). In order to achieve an acceptable computational time, Accelerated Proximal Gradient (APG) [11] approach is employed for the solution. The final decision for the classification of unknown test sample is made from the reconstruction error:

$$k^* = \arg\min_{k} \sum_{k=1}^{K} \|y - x_k w_k\|_2^2 \tag{7}$$

Figure 6 schematically illustrates the process of spoof face sample detection using such sparse encoding based approach.

4. Experiments and Results

In order to evaluate the effectiveness of the proposed approach for the detection of spoof faces and to comparatively evaluate the performance with other competing method, we used publicly available database - Replay-Attack Database [5]. All the experiments were performed on a laptop with Core 2 Duo 2.26GHz Intel CPU, Windows OS under MATLAB environment. In the following sections a brief description of the employed database is provided, which is followed by the experimental comparison with the performance from another competing method in the literature.

A. The Database

The experimental results reported in this paper were utilized publicly available Replay-Attack Database. The

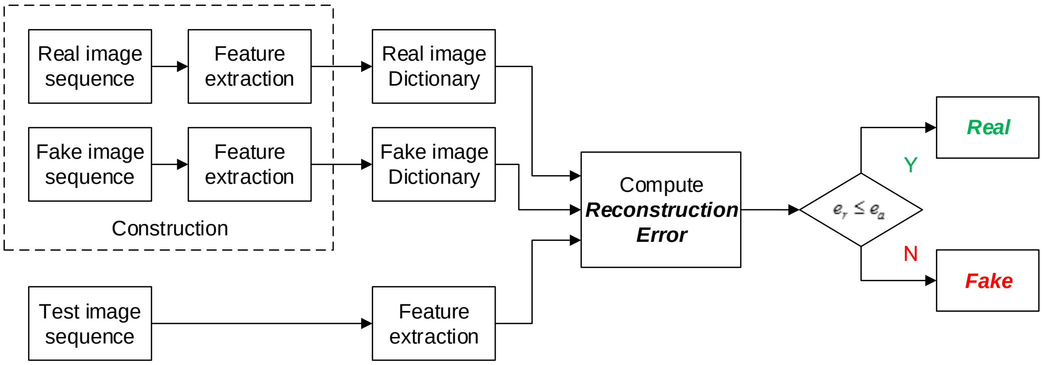

Figure 6: Block diagram representing key steps employed for anti-spoofing approach using the sparse encoding.

Replay-Attack Database consists of 1300 video clips under different settings, which is probably the *larger database* than other public databases, *e.g.* CASIA Face Anti-spoofing Database [19] or MSU database. The videos are captured using different lighting conditions. The data from each subject (real subjects and attack subjects) in the database are captured under two different settings. One set is acquired under controlled lighting conditions, while the other set is acquired under more complex lighting conditions. Besides, each of these images were acquired under different imaging setups. One setup uses fixed equipment, and the other setup uses hand held imaging equipment. Different protocols are clearly detailed in this database. Since none of these advantages are available from other databases, we selected this database as the benchmark for making performance comparisons. Our work in this paper is focused for the detection of 2D spoof attacks and therefore the 2D attacks data in this database is utilized for the performance evaluation. The training and testing protocols, along with the number of respective image samples from this database, are summarized in TABLE I and TABLE II.

TABLE I. Summary of 2D fixed print attack database.

	Attack Subjects	Attack Frames/ subject	Real Subjects	Real Frames/ subject
Train	30	240	60	375
Test	40	240	80	375

TABLE II. Summary of 2D hand print attack database.

	Attack Subjects	Attack Frames/ subject	Real Subjects	Real Frames/ subject
Train	30	230	60	375
Test	40	230	80	375

B. Experimental Results

In this part, three different protocols, which are *well defined* in this publicly available database, are used to evaluate the effectiveness of method detailed in previous sections of this paper. The details of these protocols are summarized as follows:

A. In the first set of experiments, “*fixed 2D print attack”* protocol is used to evaluate the effectiveness of the algorithm. The training and testing images under this protocol are both fixed.

B. In the second set of experiments, the “*hand 2D print attack”* is used to evaluate the effectiveness of the same algorithm. The training and testing parts are both “*hand 2D images*” in this protocol.

C. In the third experiment, in order to ascertain the effectiveness when different kinds of spoofing, both “*fixed*” and “*hand*” version of images are used in this experiment. The protocol of this setting is “*allsupports*”.

The real face samples used for the training are the same for these three different experiments, and the protocol is “*real-train*”. In the same manner, the real parts of testing images are also the same for these three experiments, and therefore the protocol is “*real-test*”.

In order to evaluate the performance for detecting spoof attacks using the *Presentation Attack Detection* (PAD) method, ISO/IEC 30107-3 [20] recommends the following parameters for evaluation: (1) *Attack Presentation Classification Error Rate* (APCER), which means the rate of attacks classified as real presentations and (2) *Normal Presentation Classification Error Rate* (NPCER), which indicate the rate of real faces classified as attack faces. Besides, *Average Classification Error Rate* (ACER), which can be computed from the average of previous two errors, is also presented in the following TABLE III.

TABLE III: The performance (%) from classification results using the method in this paper and in [13].

Method	APCER	NPCER	ACER
$LBPTOP_{3\times3}^{u2}$ + Sparse Encoding (*fixed*)	2.5	0	1.25
$LBPTOP_{3\times3}^{u2}$ + Sparse Encoding (*hand*)	7.5	0	3.75
$LBPTOP_{3\times3}^{u2}$ + Sparse Encoding (*allsupports*)	1.25	1.25	1.25

The subscript of LBPTOP in TABLE III designates that 3 × 3 region is used to extract the feature. In reference [13], the authors also utilize spatial and temporal domain information to accomplish spoofing detection, and SVM is adopted as a classification method. The performance of that experiment is based on Half Total Error Rate (HTER), which has the same meaning as the ACER in this paper. Even though reference [13] does not specify which protocol was exactly used for the performance evaluation, the ‘best’ performance illustrated in this reference is 7.6%. However, the method described in this paper (Table III) can achieve outperforming results under *all* available protocols. The best performance achievable from the method detailed in this paper is 1.25%, which is a great improvement when compared with the state-of-the-art method, as illustrated in Table IV.

TABLE IV. Comparative performance (%) between our method and method in the reference [13].

Accuracy (ACER/HTER)	Sparse encoding	SVM-based method
Protocol: “*fixed*”	1.25	7.6
Protocol: “*hand*”	3.75	7.6
Protocol: “*allsupports*”	1.25	7.6

More recent work in reference [21] details more promising approach to detect spoof face samples using the advanced learning capabilities from the deep neural networks. It can however be noted that in comparison with the deep learning based method in [21], our method is expected to be much more time efficient since its computationally simpler. After feature extraction, the computational times for pure classification optimization process under three protocols are: 0.173 s, 0.170 s, and 0.266 s. Importantly, the deep learning methods, such as in [21], generally require a large numbers of images for the training and therefore the method investigated in this paper can be more attractive when large number of images are not available (particularly difficult to acquire large number of spoof face samples for the training the convolutional neural networks).

5. Conclusions and Further Work

This paper has investigated a new approach to utilize the sparsity of features to automatically detect the sensor level spoof face attacks during the surveillance or during the face recognition scenarios. The strength of this method lies in effectively combining the *spatial domain* and *temporal domain* information that are available at the same time. Therefore, any incoming or unknown test face image sample can be represented as a linear combination of atoms in the training dictionary. The experimental results presented in section 4 validate the effectiveness of investigated approach using the publicly available database from [5]. The robustness of the methods detailed in this paper needs to be further evaluated on other spoof face image samples, particularly those from very real look-like face masks with large variations in the material type, variety and appearances. In view of increasing sophistication in the generation of face masks, there is pressing need to develop large scale databases of such representative real look-like face images. The development of such database with a large scale sophisticated 3D face masks images, under more realistic imaging variations, is the part of our ongoing work.

Acknowledgment

This work is partially supported by project grant number K-ZP40 and PolyU 5169/13E from the Research Grant Council of Hong Kong.

References

[1] W. Zhao, R. Chellappa, P. J. Phillips, and A. Rosenfeld, "Face recognition: A literature survey," *ACM Computing Surveys (CSUR)*, vol. 35, no. 4, pp. 399-458, 2003.

[2] S. Marcel, M. S. Nixon, and S. Z. Li, *Handbook of Biometric Anti-Spoofing*, Springer, 2014.

[3] P. V. Reddy, A. Kumar, S. M. K. Rahman, and T. S. Mundra, "A new antispoofing approach for biometric devices," *IEEE Trans. Biomedical Circuits & Sys*., vol. 2, no. 4, pp. 284-293, Dec. 2008.

[4] B. Tan and S. Schuckers, "Spoofing protection for fingerprint scanner by fusing ridge signal and valley noise," *Pattern Recognition*, vol. 43, no. 8, pp. 2845-2857, 2010.

[5] A. Anjos and S. Marcel, "Counter-measures to photo attacks in face recognition: a public database and a baseline," *Proc. IJCB 2011*, pp. 1-7, 2011.

[6] T.-Y. Wang and A. Kumar, "Recognizing human faces under disguise and makeup," *Proc. ISBA 2016*, pp. 1-6, Sendai, Japan, March 2016.

[7] J. Galbally, S. Marcel, and J. Fierrez, "Biometric Antispoofing Methods: A Survey in Face Recognition," *IEEE Access*, vol. 2, pp. 1530-1552, 2014.

[8] I. Chingovska, A. Anjos, and S. Marcel, "On the effectiveness of local binary patterns in face anti-spoofing," *Proc. BIOSIG 2012,* Darmstadt, Germany, pp. 1-7, Sep. 2012,

[9] T. Ojala, M. Pietikäinen, and D. Harwood, "A comparative study of texture measures with classification based on featured distributions," *Pattern Recognition*, vol. 29, no. 1, pp. 51-59, 1996.

[10] J. Määttä, A. Hadid, and M. Pietikainen, "Face spoofing detection from single images using micro-texture analysis," *Proc. IJCB 2011*, Washington DC, pp. 1-7, 2011.

[11] Y. Nesterov, *Introductory Lectures on Convex Optimization*, Springer, 2004.

[12] E. M. Rudd, M. Günther, and T. E. Boult, "PARAPH: presentation attack rejection by analyzing polarization hypotheses," *Proc. CVPR 2016 Biometrics Workshop*, Las Vegas, Nevada, Jun. 2016.

[13] T. de Freitas Pereira, A. Anjos, J. M. De Martino, and S. Marcel, "LBP - TOP based countermeasure against face spoofing attacks," *Proc. Computer Vision-ACCV 2012 Workshops*, pp. 121-132, 2013,

[14] G. Zhao and M. Pietikainen, "Dynamic texture recognition using local binary patterns with an application to facial expressions," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 29, no. 6, pp. 915-928, 2007.

[15] C. Chen, A. Dantcheva, and A. Ross, "Automatic facial makeup detection with application in face recognition," *Proc. ICB 2013*, Madrid, pp. 1-8, 2013.

[16] J. Li, Y. Wang, T. Tan, and A. K. Jain, "Live face detection based on the analysis of fourier spectra," *Proc. SPIE Symp. Defense and Security*, pp. 296-303, 2004

[17] D. Wen, H. Han, and A. K. Jain, "Face spoof detection with image distortion analysis," *IEEE Trans. Info. Forensics & Security*, vol. 10, no. 4, pp. 746-761, 2015.

[18] J. Wright, A. Y. Yang, A. Ganesh, S. S. Sastry, and Y. Ma, "Robust face recognition via sparse representation," *IEEE Trans. Pattern Anal. Mach. Intell.,* vol. 31, no. 2, pp. 210-227, 2009.

[19] Z. Zhang, J. Yan, S. Liu, Z. Lei, D. Yi, and S. Z. Li, "A face antispoofing database with diverse attacks," *Proc. ICB 2012*, New Delhi, pp. 26-31. 2012.

[20] ISO/IEC CD 30107-3, *Information technology -- Biometric presentation attack detection -Part 3: Testing and reporting*, https://www.iso.org/obp/ui/#iso:std:iso-iec:30107:-1:ed-1:v1:en

[21] D. Menotti, G. Chiachia, A. Pinto, W. R. Schwartz, H. Pedrini, A. X. Falcão, and A. Rocha, "Deep representations for iris, face, and fingerprint spoofing detection," *IEEE Trans. Info. Forensics & Security*, vol. 10, no. 4, pp. 864-879, 2015.