3DPC-Net: 3D Point Cloud Network for Face Anti-spoofing

Xuan Li[1], Jun Wan[2,3], Yi Jin[1*], Ajian Liu[4], Guodong Guo[5], Stan Z. Li[6]
[1]Beijing Jiaotong University, Beijing, China
[2]NLPR, Institute of Automation, Chinese Academy of Sciences, Beijing, China
[3]University of Chinese Academy of Sciences, Beijing, China
[4]Macau University of Science and Technology, Macau, China
[5]Baidu Research, Beijing, China
[6]Westlake University, Hangzhou, China
{18120387, yjin}@bjtu.edu.cn, jun.wan@ia.ac.cn, ajianliu92@gmail.com,
guoguodong01@baidu.com, stan.zq.li@westlake.edu.cn

Abstract

*Face anti-spoofing plays a vital role in face recognition systems. Most deep learning-based methods directly use 2D images assisted with temporal information (*i.e.*, motion, rPPG) or pseudo-3D information (*i.e.*, Depth). The main drawback of the mentioned methods is that another extra network is needed to generate the depth/rPPG information to assist the backbone network for face anti-spoofing. Different from these methods, we propose a novel method named 3D Point Cloud Network (3DPC-Net). It is an encoder-decoder network that can predict the 3DPC maps to discriminate live faces from spoofing ones. The main traits of the proposed method are: 1) It is the first time that 3DPC is used for face anti-spoofing; 2) 3DPC-Net is simple and effective and it only relies on 3DPC supervision. Extensive experiments on four databases (i.e., Oulu-NPU, SiW, CASIA-FASD, Replay Attack) have demonstrated that the 3DPC-Net is comparative to the state-of-the-art methods.*

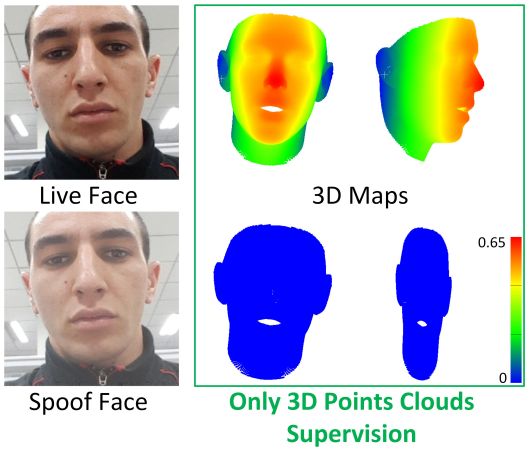


Figure 1. In our work, the point clouds of live faces have the 3D structure information while the point clouds of spoof face are fall into the same plane. Here, the value of Z axis for spoof faces is simply set zero. This work designs a simple but effective encoder-decoder network (3DPC-Net), using only 3D point cloud (3DPC) as supervisory information.

1. Introduction

Face recognition system is widely used in daily life, such as mobile payment, face unlocking and access control systems [29]. However, it is vulnerable to be attacked from printed face(i.e., print attack) [38, 37], replaying a face image or video on a digital device(i.e., replay attack) [10, 8, 23], or wearing a 3D or silicone masks (i.e., 3D attack) [13]. Therefore, face anti-spoofing is an essential prerequisite for the security of face recognition systems.

In early works, texture-based approaches [6, 12, 26] are proposed by using handcrafted features. Recently, deep learning-based methods [33, 15, 24] usually adopt binary cross-entropy loss as supervision directly for face anti-spoofing. However, these methods easy to learn the unfaithful cues, such as screen bezel, instead of the nature spoofing patterns (e.g., skin detail loss, color distortion, moiré patterns and spoof artifacts). Inspired by this, some recent works leverage auxiliary information instead of binary softmax loss as supervision. For example, the depth map [2, 23, 36, 32, 28] is often used as auxiliary supervision

*: Corresponding author; This work was done by Xuan Li who visited the lab of Center for Biometrics and Security Research, NLPR, CASIA.

978-1-7281-9186-7/20/$31.00 ©2020 IEEE

information to improve the accuracy of presentation attack detection (PDA). However, the depth map reconstructed using PRNet [14] or 3DDFA [39] is essentially a kind of pseudo-3D information that does not contain structural features. Furthermore, the design of the face anti-spoofing model tends to be complicated. They need to use additional branches to assist binary supervision to improve the generalization performance of the model. And they are accompanied by multiple loss functions and high computational complexity. In practical applications, it would consider the trade-off between the relative high performances and a few floating point of operations (FLOPs) or few parameters.

To address these issues, 3D point cloud (3DPC) is the first time used as the supervision information, which can make full use of the spatial differences between live and spoof faces. As shown in Figure 1, the point cloud of live faces have the 3D structure information while the point cloud of spoof faces are fall into the same plane. Here, the value of Z axis for spoof faces is set zero. Besides, a simple but effective encoder-decoder network, namely 3DPC-Net, is proposed to reconstruct the face 3D point clouds. The encoder learns a potential shape feature vector, and the decoder learns a 3DPC map. 3DPC-Net only relies on 3DPC supervision with Chamfer loss and its FLOPs is about 26 times smaller than the original Auxiliary [23] (Depth). To validate the effectiveness of the proposed method, we design sufficient experiments in two aspects: intra-database testing and inter-database testing. For intra testing, we use OULU-NPU [8] and SiW [23] for fair comparisons with other state-of-the-art methods. CASIA-MFSD [38], Replay-Attack [10], OULU-NPU and SiW are used for inter-database testing.

The main contributions of this paper include:

- 3D point cloud (3DPC) is the first time used as supervision information for face anti-spoofing. It can make full use of the 3D structural differences between live and spoof faces.
- A simple and effective encoder-decoder network (3DPC-Net) is proposed, which uses chamfer loss and only relies on 3DPC supervision.
- Our proposed method achieves comparative performances in term of ACER, HTER, and FLOPs.

2. Related Works

In this section, we mainly review some recent advances related to the face anti-spoofing from two aspects: 2D-based methods and 3D-based methods.

2.1. 2D-based Methods

Binary Supervision Methods. Since face anti-spoof is essentially a binary classification problem, most of previous methods use binary supervision to learn the difference between live and spoof faces. Traditional face anti-spoofing methods rely on hand-crafted descriptors to extract static features, such as LBP [12], SIFT [26], SURF [7], HOG [21, 34], DOG [30] and traditional classifiers such as SVM. Because hand-crafted descriptors are easily affected by environmental conditions such as camera devices, lighting conditions and presentation attack instruments (PAIs), traditional methods usually generalize poorly. With the help of hardware advancement and data abundance, deep learning networks have made great progress. In these works [22, 24, 25], deep learning networks are used to extract pixel-wise features and fine-tune from pretrained model. However, these face anti-spoofing methods as binary supervision problem with softmax loss might find arbitrary cues, such as screen bezel, instead of the faithful spoof patterns.

Depth Supervision Methods. Whether in a single-domain or cross-domain approach, depth maps are often used to assist the backbone network, which proves that depth information is effective as auxiliary supervision. In work [2], the depth map of a face is the first time utilized as an auxiliary information. They are based on two-streams CNNs, extracting features from local patches and holistic depth maps, respectively. In another work [23], the author proposes a method by combining spatial and temporal features from depth maps and rPPG signals. In the last two years, works [36, 32] utilize depth maps with contrastive depth loss to offer extra-strong supervision. The work [28] aim to learn invariant features between different domains with depth supervision. However, the reconstructed depth map is essentially a kind of pseudo-3D information that does not contain abundant 3D spatial features, and it inevitably cause the performance drop.

2.2. 3D-based Methods

In face anti-spoofing, a 3D virtual synthesis method [18] is used to synthesize virtual spoof faces with bending and out-of-plane rotation by rendering the transformed mesh in 3D space. However, the direct mesh representation is costly in terms of memory and is usually limited to lower resolutions. In addition, the synthesized images from the virtual 3D space do not have real image information, and it still has unfaithful cues. In addition, 3D point clouds are widely used in computer vision tasks, such as object recognition, segmentation and 3D reconstruction. Several representative works such as PointNet [9] utilizes deep learning networks to focus on the overall features of a point cloud for the first time, PointNet++ [27] pays more attention to the local features and DGCNN [31] adopts dynamic graph convolution to enhance the performance of classification and segmentation tasks. AtlasNet [17] generate fine 3D point clouds by inputting 2D images or coarse 3D point clouds. These

works represent the superiority of 3D point clouds for classification tasks and the implementability of generating point cloud maps from images.

3. The Proposed Method

Figure 2 shows the pipeline of our proposed method. It includes two parts: pre-processing and the 3DPC-Net. In the pre-processing stage, we introduce the ground truth (3DPC) generation and the 3DPC processing. Then, the processed 3DPC is used as the label of 3DPC-Net.

3.1. Pre-processing

3.1.1 3D Point Cloud Generation

In this paper, we adopt the 3DDFA [39] method to generate the original facial 3DPC labels. As shown in p_0 of Figure 2, the number of reconstructed facial point clouds has 53,215 points. We assume that the live faces has the 3D spatial information while the spoof faces is a plane (as shown in the second row of Figure 1, the coordinates of Z axis are equal to zero) for the samples from 2D print or video-replay attacks.

3.1.2 Random sampling

Owing to the limitation of hardware configuration, it is hard to used the dense point clouds. Therefore, it is necessary to sample points from the dense point clouds. Specifically, we first randomly sample 10,000 points from the 53,215 points. Then, in the data loading before model training and testing, the 3DPC labels is randomly sampled from 10,000 points to 2,500 points online, which ensure the diversity of label features.

3.1.3 Normalization

The 3DPC after random sampling is not normalized, which could lead to slower convergence speed and lower accuracy. $P(x, y, z)$ is used to represent the original coordinates with 2,500 points.

First, we calculate the maximum distance d of the X, Y, Z axes, d can be formulated as

$$d = max\{l_x - s_x, l_y - s_y, l_z - s_z\}, \tag{1}$$

where l and s are the extreme value of each coordinate axis. For example, l_x is the maximum value and s_x is the minimum value of the X axis.

Second, we get the medium coordinate $p_m(x, y, z)$ of the X, Y, Z axes, $p_m(x, y, z)$ can be formulated as

$$p_m(x, y, z) = (\frac{l_x + s_x}{2}, \frac{l_y + s_y}{2}, \frac{l_z + s_z}{2}). \tag{2}$$

Finally, we normalize 3DPC by mapping the original points to the unit sphere$[0, 1]^2$, the normalized 3DPC coordinates can be obtained by

$$P_n(x, y, z) = \frac{P(x, y, z) - p_m(x, y, z)}{2d} + \frac{1}{2}. \tag{3}$$

3.1.4 Uniform cropping

Generally, the 2D images correspond 0,1 labels. We only need to pre-process 2D images and we can choose a variety of data enhancement methods (e.g., rotation, clipping, affine transformation, shading transformation). However, when the label is the 3DPC, we need to pre-process input image and 3DPC consistently. In the whole training and testing process, we only adopt rotation and cropping without other additional data enhancement tricks.

To crop the input image and 3DPC label consistently, the cropped point cloud coordinates are represented as $P_c(x, y, z)$. The extreme values of $P_c(x, y, z)$ on the X axis can be represented as

$$s_{x_{crop}} = s_x + (r_x / i_w \cdot d_x), \tag{4}$$

$$l_{x_{crop}} = l_x + ((r_x + t_w) / i_w \cdot d_x), \tag{5}$$

where r_x is a random integer from 0 to $i_w - t_w$, i_w, t_w are width of input image and target image, respectively. d_x represents maximum distance on X axis. Similarly, we can also get the Y axis extreme value($s_{y_{crop}}, l_{y_{crop}}$).

The cropped point clouds can be obtained by

$$\begin{aligned} P_c(x, y, z) &= P(x^{'}, y^{'}, z^{'}), \\ where, &\begin{cases} s_{x_{crop}} < x^{'} < l_{x_{crop}} \\ s_{y_{crop}} < y^{'} < l_{y_{crop}} \end{cases} \end{aligned} \tag{6}$$

For rotation, we ensure the input images and 3DPC labels have the same rotation angle. The 3DPC only rotate on the Z axis. The rotation coordinates $P_r(x, y, z)$ can be represented as

$$P_r(x, y, z) = P(x, y, z) \cdot \begin{bmatrix} cos\theta & -sin\theta & 0 \\ sin\theta & -cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}. \tag{7}$$

3.2. 3DPC-Net

3.2.1 Architecture

Our goal is, given an RGB face image, to generate a point cloud with the 3D facial structural features, or an approximate plane with the facial shape. The details of the 3DPC-Net are shown in Table 1. For auto-encoder, we use an encoder based on ResNet18 [19], which has proved its superiority in recognition and classification tasks. The encoder

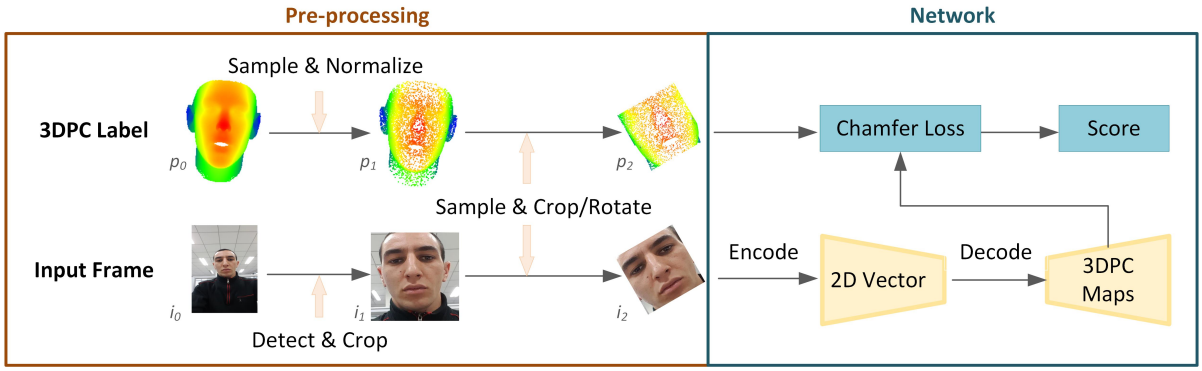


Figure 2. The overview of our proposed 3DPC-Net.

maps the input image to a potential shape feature vector x of size 256. Let $\mathcal{A}$ be a point set sampled in the unit square$[0,1]^2$ and S^* a 3DPC label. Next, We concatenate x with sampled point set $p\epsilon\mathcal{A}$ as input of the decoder. The decoder contains two 1D convolutional layers of size 258, 129 with ReLU non-linearities on the first layer, and tanh on the second output layer. The decoder with parameters θ can generate a surface by learning to map points in R^2 to surface points in R^3. We consider learnable parameterization ϕ_θ and adopt Chamfer loss to minimize the difference between the learned 3DPC map and the generated 3DPC label,

$$\mathcal{L}(\theta) = \sum_{p\epsilon\mathcal{A}} \min_{q\epsilon\mathcal{S}^*} |\phi_\theta(p;x) - q|^2 + \sum_{q\epsilon\mathcal{S}^*} \min_{p\epsilon\mathcal{A}} |\phi_\theta(p;x) - q|^2. \tag{8}$$

3.2.2 Implementation Details

Our proposed method is implemented with the Pytorch framework. In the training stage, models are trained with Adam optimizer, and the initial learning rate and weight decay are 1e-3 and 0, respectively. We train models with maximum 300 epochs and adopt step learning rate strategy, which decays every 5 training steps by a factor of 0.8. The batch size is 256 on one GeForce RTX 2080ti GPU. In the testing stage, we calculate the mean value of predicted 3DPC map as the final score.

4. Experiments

We evaluate the performance of 3DPC-Net from two aspects: OULU-NPU [8] and SiW [23] for intra-database testing and CASIA-MFSD [38], Replay-Attack [10], SiW and OULU-NPU for inter testing. We explore the influence of different point numbers, loss function, supervision information and fusion type. Performance evaluations on all datasets were performmed for fair comparisons with other state-of-the-art methods.

Table 1. Our 3DPC-Net network structure. The encoder network is the same as Resnet18 [19]. The decoder network consists of two 1D convolutional layers with outputs size of 129×2500 and 3×2500, respectively.

	Layers	**Output Size**	**3DPC-Net**
Input	3×224×224 (RGB Image)		
Encoder	Conv1	112×112	7×7, 64
	Conv2_x	56×56	$\begin{bmatrix} 3\times3, 64 \\ 3\times3, 64 \end{bmatrix} \times 2$
	Conv3_x	28×28	$\begin{bmatrix} 3\times3, 128 \\ 3\times3, 128 \end{bmatrix} \times 2$
	Conv4_x	14×14	$\begin{bmatrix} 3\times3, 256 \\ 3\times3, 256 \end{bmatrix} \times 2$
	Conv5_x	7×7	$\begin{bmatrix} 3\times3, 512 \\ 3\times3, 512 \end{bmatrix} \times 2$
	Avg Pooling &FC	256	
Decoder	Conv6	129×2500	1×1, 129
	Conv7	3×2500	1×1, 3
Output	**3×2500 (3DPC Map)**		

4.1. Datasets and Metrics

OULU-NPU. This dataset contains 55 subjects with 4,950 live and spoof videos that are recorded using the front-facing cameras of six different phones. The videos are collected in three sessions with different acquisition conditions. The spoof attacks are designed using two printers and two replay attacks.

SiW. This dataset contains 165 subjects, and 8 live and up to 20 spoof videos for each subject under 1080P HD resolution. The live videos are collected in four sessions with variations of distance, pose, illumination and expression. Two mainly attacks are created using printing papers and replay attacks.

CAISA-MFSD. This dataset is low-resolution datasets and contains 50 subjects, and 12 videos for each subject that are recorded under different resolutions and light conditions. Three different attacks are collected: warp print, cut print attacks and replay attacks.

Replay-Attack. This dataset is low-resolution datasets and

Table 2. The results of the different number of points on OULU-NPU Protocol 1.

Number	APCER(%)	BPCER(%)	ACER(%)
256	2.5	2.8	1.7
512	2.1	0.6	1.5
1024	2.5	0.8	1.7
2500	2.4	0.0	**1.2**

Table 3. The results of ablation study on OULU-NPU Protocol 1.

Module	L_1	L_2	L_c	L_s	3DPC	ACER(%)
Model1				✓		4.2
Model2			✓	✓	✓	3.2
Model3	✓				✓	1.9
Model4		✓			✓	1.5
Model5			✓		✓	**1.2**

contains 50 subjects with 1,300 videos. The videos are collected under controlled and adverse conditions.

In OULU-NPU and SiW dataset, we use Average Classification Error Rate (ACER) [1], Attack Presentation Classification Error Rate (APCER) and Bona Fide Presentation Classification Error Rate (BPCER) for intra-database. Half Total Error Rate(HTER) is adopted in the inter-database testing between CASIA-MFSD and Replay-Attack and from SiW to OULU-NPU, which evaluates the mean of False Rejection Rate (FRR) and the False Acceptance Rate (FAR).

4.2. Ablation Study

We evaluate the performances of different number of points, loss function and supervision information on the protocol 1 of OULU-NPU. As shown in Table 3, L_1, L_2, L_c, L_s represent L1, L2, Chamfer and Softmax loss, respectively. Model 1 only has an architecture similar to the encoder part in our method, except that it is extended with additional softmax loss for binary supervision. Model 2 minimizes the Chamfer and Softmax loss with 3DPC and binary supervision. Model 3 and Model 4 learn 3DPC maps with L1 loss and L2 loss, respectively. Model 5 is our proposed architecture.

Impact of the Number of Points. We experiments with different numbers of points. As can be seen from Table 2, when the number of points is equal to 256, 512, 2014 and 2500, the ACER is 1.7%, 1.5%, 1.7% and 1.2%, respectively. When the number of points is 2500, ACER is the lowest. Due to the limitation of computing performance, we do not conduct more experiments, and finally the number of training and testing output points is 2500.

Impact of Different Loss Function. We minimize the L1, L2, and Chamfer loss between 3DPC maps and 3DPC labels. It can be seen from Table 3 that Model 5 outperform Model 3 and Model 4, which means that the Chamfer loss can better increase the map difference between live faces and spoof faces.

Table 4. The results of intra-database testing on four protocols of Oulu-NPU.

Prot	Method	APCER(%)	BPCER(%)	ACER(%)
1	CPqD [4]	2.9	10.8	6.9
	GRADIANT [4]	1.3	12.5	6.9
	STASN [35]	1.2	2.5	1.9
	Auxiliary [23]	1.6	1.6	1.6
	FaceDs [20]	1.2	1.7	1.5
	DeepPixBis [16]	0.8	0.0	**0.4**
	3DPC-Net	2.3	0.0	1.2
2	MixedFASNet [20]	9.7	2.5	6.1
	DeepPixBis [16]	11.4	0.6	6.0
	FaceDs [20]	4.2	4.4	4.3
	Auxiliary [23]	2.7	2.7	2.7
	GRADIANT [4]	3.1	1.9	2.5
	STASN [35]	4.2	0.3	**2.2**
	3DPC-Net	3.1	2.8	3.0
3	DeepPixBis [16]	11.7± 19.6	10.6± 14.1	11.1 ± 9.4
	MixedFASNet [20]	5.3±6.7	7.8±5.5	6.5±4.6
	GRADIANT [4]	2.6±3.9	5.0±5.3	3.8±2.4
	FaceDs [20]	4.0±1.8	3.8±1.2	3.6±1.6
	Auxiliary [23]	2.7±1.3	3.1±1.7	2.9±1.5
	STASN [35]	4.7±3.9	0.9±1.2	2.8±1.6
	3DPC-Net	2.8±2.1	2.8±1.5	**2.8±0.5**
4	DeepPixBis [16]	36.7± 29.7	13.3±14.1	25.0±12.7
	Massy HNU [4]	35.8±35.3	8.3±4.1	22.1±17.6
	GRADIANT [4]	5.0±4.5	15.0± 7.1	10.0± 5.0
	Auxiliary [23]	9.3± 5.6	10.4±6.0	9.5±6.0
	STASN [35]	6.7±10.6	8.3±8.4	7.5±4.7
	FaceDs [20]	1.2±6.3	6.1±5.1	5.6±5.7
	3DPC-Net	2.1±9.2	5.0±16.7	**3.5±5.4**

Table 5. The results of intra-database testing on three protocols of SiW.

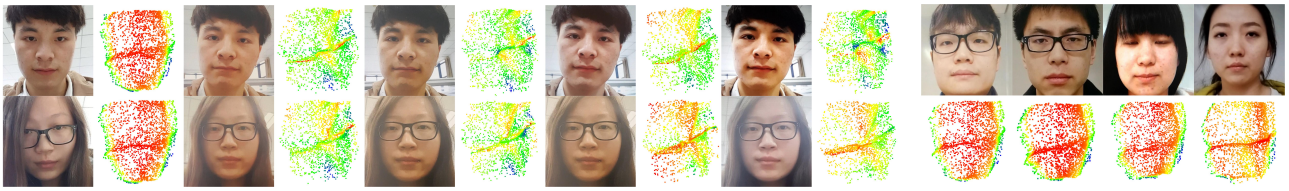
Prot	Method	APCER(%)	BPCER(%)	ACER(%)
1	Auxiliary [23]	3.58	3.58	3.58
	STASN [35]	-	-	1.00
	3DPC-Net	0.69	0.92	**0.80**
2	Auxiliary [23]	0.57±0.69	0.57±0.69	0.57±0.69
	STASN [35]	-	-	**0.28±0.05**
	3DPC-Net	0.46±0.28	0.43±0.06	0.45±0.14
3	STASN [35]	-	-	12.10±1.50
	Auxiliary [23]	8.31± 3.81	8.31 ± 3.80	8.31 ± 3.81
	3DPC-Net	7.50± 38.81	7.85 ±38.13	**7.68** ± 38.50

Impact of Different Supervision. As shown in Table 3, we can see that Model 5 is lowest ACER compared with Model 1 and Model 2. For Model 2, the loss is equal to α times Chamfer loss plus β times Softmax loss. In Table 3, α is 0.9 and β is 0.1. Furthermore, we also conducted corresponding experiments on α equal to 0.7, 0.5, 0.3, and β equal to 0.3, 0.5, 0.7, and obtain results with ACER of 3.7, 3.1 and 7.1, respectively.

4.3. Intra-database Testing

OULU-NPU and SiW datasets are used for intra-database testing. We follow the four protocols on OULU-NPU and three protocols on SiW for evaluation and report their APCER, BPCER, and ACER.

Results on Oulu-NPU. Table 4 shows the comparisons



(a) (b)

Figure 3. (a) The 3DPC maps estimation on OULU-NPU protocol 1 testing subjects. The first two columns are the live faces and their corresponding 3DPC maps, the rest eight columns are four different types of spoof (print1, print2, replay1, replay2) and their corresponding 3DPC maps. (b) Four failed samples on OULU-NPU protocol 1.

Table 6. The results of inter-database testing between CASIA-MFSD and Replay-Attack. The evaluation metric is HTER(%).

Method	Train	Test	Train	Test
	CASIA-MFSD	Replay-Attack	CASIA-MFSD	Replay-Attack
Motion [12]	50.2		47.9	
LBP-1 [12]	55.9		57.6	
LBP-TOP [12]	49.7		60.6	
Motion-Mag [3]	50.1		47.0	
Spectral cubes [11]	34.4		50.0	
LBP-2 [5]	47.0		39.6	
Color Texture [6]	30.3		37.7	
CNN [33]	34.4		50.0	
STASN [35]	31.5		30.9	
Auxiliary [23]	27.6		28.4	
FaceDs [20]	28.5		41.1	
3DPC-Net	**23.4**		**25.7**	

Table 7. The results of inter-database testing from SiW to OULU-NPU dataset.

Prot.	Method	ACER(%)
1	Auxiliary [23]	10.0
	3DPC-Net	**5.5**
2	Auxiliary [23]	14.1
	3DPC-Net	**5.4**
3	Auxiliary [23]	**13.8±5.7**
	3DPC-Net	15.2±3.6
4	Auxiliary [23]	10.0±8.8
	3DPC-Net	**5.6±9.8**

Table 8. The efficiency comparison about different model.

Method	Param(Mb)	FLOPs(G)	Memory(Mb)
Auxiliary(Depth) [23]	2.20	47.41	504.98
3DPC-Net	11.34	1.82	29.43

of our proposed 3DPC-Net with state-of-the-art methods. We can see that the proposed method has excellent results on all four protocols (1.6%, 3.3%, 2.8%, 3.5% ACER, respectively). Our method achieves the lowest mean of ACER on protocols 3 and 4. These indicate that 3DPC-Net performs well at generalization on (a) unseen environmental conditions, (b) unseen print and video-replay attack mediums, (c) unseen input camera variations.

Results on SiW. As show in Table 5, our proposed 3DPC-Net compared with Auxiliary [23] and STASN [35] on SiW dataset. We can see that 3DPC-Net has the lowest mean

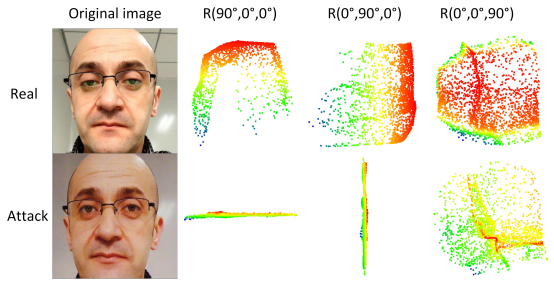


Figure 4. Visualization of images and 3DPC maps at different rotation angles.

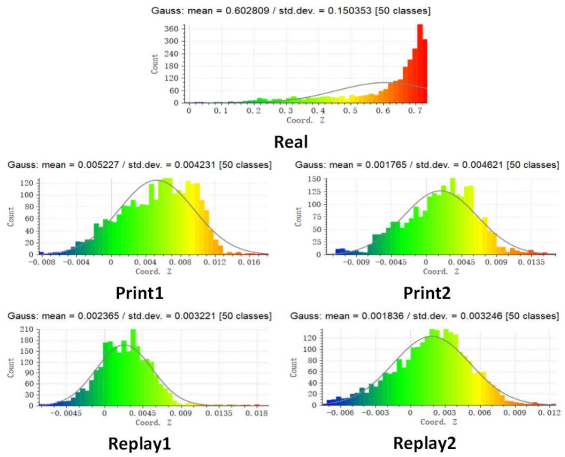


Figure 5. Gaussian distribution corresponding to different maps of the first row in Figure 3 (a).

ACER in protocol 1 and protocol 3. Because ACER is 13.88% in the first sub-protocol of protocol 3 and 1.47% in the second sub-protocol, the result of protocol 3 has a large standard deviation. Overall, 3DPC-Net has good generalization capabilities and can be used for attacks with changes in facial pose and expressions, spoofing media, and attack presentation types.

4.4. Inter-database Testing

To demonstrate generalization of our model, CASIA-MFSD, Replay-Attack, SiW and OULU-NPU datasets are used for inter-database testing.

Results on CASIA-MFSD and Replay-Attack. We conduct experiments on two protocols. The first protocol called C2R is trained on CASIA-MFSD and tested on Replay-Attack. The second protocol called R2C is trained on Replay-Attack and tested on CASIA-MFSD. It can be seen from Table 6 that the HTER of our proposed 3DPC-Net is 23.4% on C2R and 25.7% on R2C, which outperform the previous state-of-the-art methods.

Results from SiW to OULU-NPU. Table 7 shows inter-database testing results trained on SiW and tested on OULU-NPU. We can see that our method surpasses Auxiliary [23] on three protocols (decrease 4.5%, 8.7%, and 4.6% on protocol 1, protocol 2 and protocol 4, respectively). Our ACER on protocol 3 is slightly worse, and it may also be a good idea to combine our method with temporal information.

4.5. Effective Analysis

An excellent model should be a trade-off between the relatively high performances and a few floating point of operations (FLOPs) or few parameters. As shown in Table 8, we compare the parameters, FLOPs and memory of 3DPC-Net with Auxiliary (Depth) [23]. The Auxiliary (Depth) uses only the depth map as the label and uses the original input image of size $3\times256\times256$. As can be seen in Table 8, the auxiliary (depth) parameters are the least. 3DPC-Net has 1.82G FLOP and 29.42Mb memory, which are about 26 and 17 times smaller than the Auxiliary (Depth), respectively.

4.6. Visualization

For visualize our experiments, we adopt the results of OULU-NPU protocol 1 to show 3D structure information and Gaussian distribution of the mapped features through different colors. Examples of successful in estimating 3DPC maps on protocol 1 of OULU-NPU are shown in Figure 3 (a). The first two columns are the live faces and their corresponding 3DPC maps. The rest eight columns are four different types of spoof (print1, print2, replay1, replay2) and their corresponding 3DPC maps. Figure 4 shows 3DPC map with different rotation angles. We noticed that live face map has sufficient 3D information, but the spoof face map is similar to a plane. Figure 5 shows five Gauss distributions of live and different spoof maps. We can see that the 3DPC map of the live face has a larger Z-axis value, average value and standard deviation as a whole, and the 3DPC map of the spoof face is the opposite. Figure 3 (b) shows examples of failure (from left to right are print1, print2, replay1, replay2). We identify 13 failure cases (1.2% ACER). Since all live faces are correctly classified, all cases are from spoof faces.

5. Conclusions and Future work

In this paper, 3D point cloud (3DPC) is the first time used as the supervision information for face anti-spoofing. We also propose a simple but effective encoder-decoder network (3DPC-Net) that only relies on 3DPC supervision. Extensive experiments are conducted to verify the performance of our proposed 3DPC-Net. Further directions include: 1) using 3DPC to explore lighter and practical models for deployment in mobile devices; 2) considering the temporal information used in our proposed method.

Acknowledgement

This work was supported by the Chinese National Natural Science Foundation Projects #61972030, #61961160704, #61876179, the Key Project of the General Logistics Department Grant No.ASW17C001, Science and Technology Development Fund of Macau (No. 0025/2018/A1, 0008/2019/A1, 0019/2018/ASC, 0010/2019/AFJ, 0025/2019/AKP).

References

[1] ISO/IEC JTC 1/SC 37 Biometrics. information technology biometric presentation attack detection part 1: Framework. international organization for standardization. 2016. https://www.iso.org/obp/ui/iso.

[2] Y. Atoum, Y. Liu, A. Jourabloo, and X. Liu. Face anti-spoofing using patch and depth-based cnns. In *IJCB*, 2017.

[3] S. Bharadwaj, T. I. Dhamecha, M. Vatsa, and R. Singh. Computationally efficient face spoofing detection with motion magnification. In *CVPR*, 2013.

[4] Z. Boulkenafet, J. Komulainen, Z. Akhtar, A. Benlamoudi, and A. Hadid. A competition on generalized software-based face presentation attack detection in mobile scenarios. In *IJCB*, 2017.

[5] Z. Boulkenafet, J. Komulainen, and A. Hadid. face anti-spoofing based on color texture analysis. In *ICIP*, 2015.

[6] Z. Boulkenafet, J. Komulainen, and A. Hadid. Face spoofing detection using colour texture analysis. *TIFS*, 2016.

[7] Z. Boulkenafet, J. Komulainen, and A. Hadid. Face anti-spoofing using speeded-up robust features and fisher vector encoding. *SPL*, 2017.

[8] Z. Boulkenafet, J. Komulainen, L. Li, X. Feng, and A. Hadid. Oulu-npu: A mobile face presentation attack database with real-world variations. In *FG*, 2017.

[9] R. Q. Charles, H. Su, M. Kaichun, and L. J. Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *CVPR*, 2017.

[10] I. Chingovska, A. Anjos, and S. Marcel. On the effectiveness of local binary patterns in face anti-spoofing. In *BIOSIG*, 2012.

[11] A. da Silva Pinto, H. Pedrini, W. Schwartz, and A. Rocha. Video-based face spoofing detection through visual rhythm analysis. In *SIBGRAPI*, 2012.

[12] T. de Freitas Pereira, A. Anjos, J. M. De Martino, and S. Marcel. Can face anti-spoofing countermeasures work in a real world scenario? In *ICB*, 2013.

[13] N. Erdogmus and S. Marcel. Spoofing in 2d face recognition with 3d masks and anti-spoofing with kinect. In *BTAS*, 2014.

[14] Y. Feng, F. Wu, X. Shao, Y. Wang, and X. Zhou. Joint 3d face reconstruction and dense alignment with position map regression network. In *ECCV*, 2018.

[15] J. Gan, S. Li, Y. Zhai, and C. Liu. 3d convolutional neural network based on face anti-spoofing. In *international conference multimedia and image processing*, pages 1–5, 2017.

[16] A. George and S. Marcel. Deep pixel-wise binary supervision for face presentation attack detection. In *ICB*, 2019.

[17] T. Groueix, M. Fisher, V. G. Kim, B. C. Russell, and M. Aubry. Atlasnet: A papier-mâché approach to learning 3d surface generation. In *CVPR*, 2018.

[18] J. Guo, X. Zhu, J. Xiao, Z. Lei, G. Wan, and S. Z. Li. Improving face anti-spoofing by 3d virtual synthesis. In *ICB*, 2019.

[19] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *CVPR*, 2016.

[20] A. Jourabloo, Y. Liu, and X. Liu. Face de-spoofing: Anti-spoofing via noise modeling. In *CVPR*, 2018.

[21] J. Komulainen, A. Hadid, and M. Pietikainen. Context based face anti-spoofing. In *BTAS*, 2013.

[22] L. Li, X. Feng, Z. Boulkenafet, Z. Xia, M. Li, and A. Hadid. An original face anti-spoofing approach using partial convolutional neural network. In *IPTA*, 2016.

[23] Y. Liu, A. Jourabloo, and X. Liu. Learning deep models for face anti-spoofing: Binary or auxiliary supervision. In *CVPR*, 2018.

[24] O. Nikisins, A. George, and S. Marcel. Domain adaptation in multi-channel autoencoder based features for robust face anti-spoofing. In *ICB*, 2019.

[25] K. Patel, H. Han, and A. K. Jain. Cross-database face anti-spoofing with robust feature representation. In *CCBR*, 2016.

[26] K. Patel, H. Han, and A. K. Jain. Secure face unlock: Spoof detection on smartphones. *TIFS*, 2016.

[27] C. R. Qi, L. Yi, H. Su, and L. J. Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. In *NIPS*, 2017.

[28] R. Shao, X. Lan, J. Li, and P. C. Yuen. Multi-adversarial discriminative deep domain generalization for face presentation attack detection. In *CVPR*, 2019.

[29] Y. Sun, X. Wang, and X. Tang. *Deep Learning Face Representation from Predicting 10, 000 Classes.* Springer International Publishing, 2016.

[30] X. Tan, Y. Li, J. Liu, and L. Jiang. Face liveness detection from a single image with sparse low rank bilinear discriminative model. In *ECCV*, 2010.

[31] Y. Wang, Y. Sun, Z. Liu, S. E. Sarma, M. M. Bronstein, and J. Solomon. Dynamic graph cnn for learning on point clouds. *ACM Transactions on Graphics*, 38(5):146, 2019.

[32] Z. Wang, Z. Yu, C. Zhao, X. Zhu, Y. Qin, Q. Zhou, F. Zhou, and Z. Lei. Deep spatial gradient and temporal depth learning for face anti-spoofing. In *CVPR*, 2020.

[33] J. Yang, Z. Lei, and S. Z. Li. Learn convolutional neural network for face anti-spoofing. In *CVPR*, 2014.

[34] J. Yang, Z. Lei, S. Liao, and S. Z. Li. Face liveness detection with component dependent descriptor. In *ICB*, 2013.

[35] X. Yang, W. Luo, L. Bao, Y. Gao, D. Gong, S. Zheng, Z. Li, and W. Liu. Face anti-spoofing: Model matters, so does data. In *CVPR*, 2019.

[36] Z. Yu, C. Zhao, Z. Wang, Y. Qin, Z. Su, X. Li, F. Zhou, and G. Zhao. Searching central difference convolutional networks for face anti-spoofing. In *CVPR*, 2020.

[37] S. Zhang, A. Liu, J. Wan, Y. Liang, G. Guo, S. Escalera, H. J. Escalante, and S. Z. Li. Casia-surf: A large-scale multi-modal benchmark for face anti-spoofing. In *CVPR*, 2019.

[38] Z. Zhang, J. Yan, S. Liu, Z. Lei, D. Yi, and S. Z. Li. A face antispoofing database with diverse attacks. In *ICB*, 2012.

[39] X. Zhu, X. Liu, Z. Lei, and S. Z. Li. Face alignment in full pose range: A 3d total solution. *TPAMI*, 41(1):78–92, 2017.