

# Exploiting temporal and depth information for multi-frame face anti-spoofing

Zezheng Wang<sup>1</sup>, Chenxu Zhao<sup>1\*</sup>, Yunxiao Qin<sup>1</sup>, Qiusheng Zhou<sup>1</sup>,

Guojun Qi<sup>2</sup>, Jun Wan<sup>3</sup>, Zhen Lei<sup>3</sup>

<sup>1</sup>JD Finance

<sup>2</sup>Huawei Cloud

<sup>3</sup>Chinese Academy of Sciences

{wangzezheng1, zhaochenxu1, qinyunxiao, zhouqiusheng}@jd.com, guojunj@ gmail.com,  
 {jun.wan, zlei}@nlpr.ia.ac.cn

## Abstract

Face anti-spoofing is significant to the security of face recognition systems. Previous works on depth supervised learning have proved the effectiveness for face anti-spoofing. Nevertheless, they only considered the depth as an auxiliary supervision in the single frame. Different from these methods, we develop a new method to estimate depth information from multiple RGB frames and propose a depth-supervised architecture which can efficiently encodes spatiotemporal information for presentation attack detection. It includes two novel modules: optical flow guided feature block (OFFB) and convolution gated recurrent units (ConvGRU) module, which are designed to extract short-term and long-term motion to discriminate living and spoofing faces. Extensive experiments demonstrate that the proposed approach achieves state-of-the-art results on four benchmark datasets, namely OULU-NPU, SiW, CASIA-MFSD, and Replay-Attack.

## 1. Introduction

Face recognition systems have become indispensable in many interactive AI systems for its convenience. However, most of existing face recognition systems are so vulnerable to the face spoofing that attackers can easily deceive face recognition systems by using presentation attacks (PA), e.g., printing a face on paper (print attack), replaying a face on a digital device (replay attack), and bringing a 3D-mask (3D-mask attack). Such PAs can tamper with face recognition systems, placing direct risks to money payment and privacy verification that have high stakes with the public interests. Therefore, face anti-spoofing plays a critically important role in the security of face recognition.

\*denotes corresponding author.

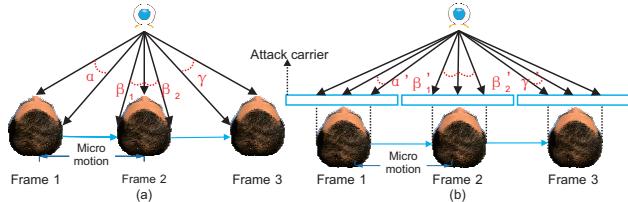


Figure 1. To observe the difference between live and spoof(print attack here) scenes, we represent the micro motion exaggeratedly. The angle of camera viewpoint can reflect the facial motion relative depth among different keypoints. In the living scene (a), the angle  $\alpha$  between nose and right ear is getting smaller with the face moving right ( $\alpha > \beta_2$ ), while the angle  $\beta_2$  between left ear and nose is getting larger ( $\beta_1 < \gamma$ ). However, in the spoofing scene (b),  $\alpha' < \beta'_2$ , and  $\beta'_1 > \gamma'$ .

To defend against face spoofing attacks, a great number of face anti-spoofing methods [5, 14, 21, 23] have been proposed to discriminate the living and fake face. Previous approaches fall into two categories. The first is the traditional methods, which commonly train shallow classifiers with hand-crafted features, i.e., LBP [10], SIFT [26], or SURF [6]. With only texture characteristic considered, these methods are often prone to the attacks, such as replaying mediums and 2D/3D mask. In contrast to texture-based methods, CNN-based methods provide an alternative way to learn discriminative anti-spoofing representations in an end-to-end fashion. Some CNN-based methods [21, 23, 14, 34, 35, 30] only treat face anti-spoofing as a pure binary classification - spoofing as 0 and living as 1, and train the neural network supervised by simple softmax loss. However, these methods fail to explore the nature of spoof patterns [20], which consist of skin detail loss, color distortion, moiré pattern, motion pattern, shape deformation and spoofing artifacts.

Depth supervised face anti-spoofing somehow relieves these issues. Intuitively, images of living faces contain face-like depth information, whereas spoofing face images in

print and replay carriers only have planar depth information. Thus, Atoum *et al.* [2] and Liu *et al.* [20] proposes single-frame depth supervised CNN architecture in their architecture, and improves the performance of presentation attack detection (PAD). However, they only leverage facial depth as an auxiliary strong supervision to detect the static spoofing cues under pixel-wise supervision relatively carefully, neglecting to exploit the *virtual* depth difference between living and spoofing scenes. To compensate with the lack of temporal information, Liu *et al.* [20] takes rPPG signals as extra supervisory cues, and does not use any depth information in temporal domain. However, depth information is more obvious in sequential frames. Indeed, temporal motion is very useful to extract depth information and thus play a great potential role for face anti-spoofing detection.

In this paper, we will explore the temporal depth by combining temporal motion and facial depth, and demonstrate that face anti-spoofing can be significantly benefit from temporal facial depth. No matter whether cameras or subjects move, sequential frames can be represented in a 3D space using the relative motion of objects. Inspired by the work [29] revealing the relationship between relative facial depth and the motion, we believe patterns of motion and depth are distinct between living and spoofing scenes. And the abnormal facial depth will reflect on the unique facial motion pattern in the temporal domain. Fig. 1 illustrates apparently different temporal representations between living and spoofing scenes. To this end, we present a novel depth supervised neural network architecture with optical flow guided feature block (OFFB) and convolution gated recurrent units (ConvGRU) module to incorporate the temporal information from both adjacent and long-sequence frames, respectively. By combining both short-term and long-term motion extractors, the temporal motion can be used effectively to discriminate living and spoofing faces under depth supervision. And to estimate facial depth in face anti-spoofing tasks, we propose a novel contrastive depth loss to learn topography of facial points.

We summarize the main contributions below:

- We analyze the temporal depth in face anti-spoofing and seek the usefulness of motion and depth in face anti-spoofing.
- We propose a novel depth supervised architecture with OFF block (OFFB) and ConvGRU module to uncover facial depths and their unique motion patterns from temporal information of monocular frame sequences.
- We design a contrastive depth loss to learn the topography of facial points for depth supervised face anti-spoofing.
- We demonstrate the superior performances to the state-of-the-art methods on widely used face anti-spoofing benchmarks.

## 2. Related Work

We review related face anti-spoofing works in three categories: binary supervised methods, depth supervised methods, and temporal-based methods.

**Binary supervised Methods** Since face anti-spoofing is essentially a binary classification problem, most of previous anti-spoofing methods purely train a classifier under binary supervision, e.g., spoofing face as 0 and living face as 1. Binary classifiers contain traditional classifiers and neural networks. Prior works usually rely on hand-crafted features, such as LBP [10, 11, 22], SIFT [26], SURF [6], HoG [18, 36], DoG [27, 33] with traditional classifiers, such as SVM and Random Forest. Since these manually-engineered features are often sensitive to varying condition, such as camera devices, lighting conditions and presentation attack instruments (PAIs), traditional methods often perform poorly in generalization.

Since then, CNN has achieved great breakthrough with the help of hardware development and data abundance. Recently, CNN is also widely used in face anti-spoofing tasks [21, 23, 14, 12, 19, 25, 35]. However, most of the deep learning methods simply consider face anti-spoofing as a binary classification problem with softmax loss. Both [19] and [25] fine-tune a pre-trained VGG-face model and take it as a feature extractor for the subsequent classification. Nagpal *et al.* [23] comprehensively study the influence of different network architectures and hyperparameters on face anti-spoofing. Feng [12] and Li [19] feed different kinds of face images into the CNN network to learn discriminative features on living faces and spoofing faces.

**Depth supervised Methods** Compared with binary supervised face anti-spoofing methods, depth supervised methods have a lot of advantages. Atoum [2] utilizes depth map of face as supervisory signals for the first time. They propose a two-stream CNN-based approach for face anti-spoofing, by extracting the local features and holistic depth maps from the face images. In other words, they combine depth-based CNN and patch-based CNN from single frame to obtain discriminative representation to distinguish live vs. spoof. It shows that depth estimation is beneficial for modeling face anti-spoofing to obtain promising results especially on higher-resolution images.

Liu [20] proposes a face anti-spoofing method from a combination of spatial perspective (depth) and temporal perspective (rPPG). They regard facial depth as an auxiliary supervision, along with rPPG signals. For temporal information, they use a simple RNN to learn the corresponding rPPG signals. However, due to the simplicity of sequence-processing, they have a non-rigid registration layer to remove the influence of facial poses and expressions, ignoring that unnatural changes of facial poses or expressions are significant spoofing cues.

**Temporal-based Methods** Temporal information plays a

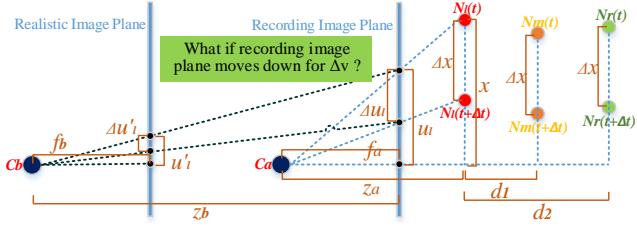


Figure 2. The schematic diagram of motion and depth variation in the presentation attack scene.

vital role in face anti-spoofing tasks. [24, 25, 30] focus on the movement of key parts of the face. For example, [24, 25] make spoofing predictions based on eye-blinking. These methods are vulnerable to replay attack since excessively relying on one aspect. Gan [14] proposes a 3D convolution network to distinguish the live vs. spoof. 3D convolution network is a stacked structure to learn the temporal features in a supervised pattern, but depends on significant amount of data and performs poorly on small database. Xu [34] proposes an architecture combining LSTM units with CNN for binary classification. Feng [12] presents a work that takes optical flow magnitude map and Shearlet feature as inputs to CNN. Their work shows that optical flow map presents obvious difference between living faces and different kinds of spoofing faces. All prior temporal-based methods are incapable of catching valid temporal information with a well-designed structure.

### 3. The Proposed Approach

In this section, firstly, we will introduce the temporal depth in face anti-spoofing and show the useful patterns of motion and depth in face anti-spoofing. Then we will present the proposed temporal PAD method under depth supervision. As shown in Fig. 5, the proposed model mainly consists of two modules. One is the single-frame part, focusing on seeking spoofing cues with the static depth supervision. The other is the multi-frame part, consisting of optical flow guided feature block (OFFB) as the short-term motion module and convolutional gated recurrent units (ConvGRU) modeling long-term motion patterns. The model can explore spoofing cues in both spatial and temporal domains under the depth supervision, successfully.

#### 3.1. Temporal Depth in Face Anti-spoofing

As shown in Fig. 8, there are two image spaces. The right image space is recording image space, where attackers use to record others' faces. The left image space is realistic image space, where real face recognition systems are. In the figure,  $C_a, C_b$  are the focal points of recording image space and realistic image space, respectively,  $f_a, f_b$  are corresponding focal distances, and  $z_b$  is the distance between  $C_b$  and the recording image plane (i.e., the screen of re-

play attack or the paper of print attack), while  $z_a$  is the distance between the focal point and the nearest facial point.  $N_l(t), N_m(t), N_r(t)$  are three points having different distances to the camera at time  $t$ , and  $N_l(t + \Delta t), N_m(t + \Delta t), N_r(t + \Delta t)$  are the corresponding points when they move down vertically by  $\Delta x$  at time  $t + \Delta t$ .  $x$  is the coordinate in the vertical direction, and  $u_l$  is the mapping coordinate in the recording image plane.  $\Delta u_l$  represents the optical flow on the image plane when the point moves down.  $u'_l, \Delta u'_l$  are values mapped on the realistic image plane by  $u_l, \Delta u_l$ , respectively. We have the following two equations revealing the relation between these variables:

$$\begin{aligned} \frac{\Delta u'_l}{\Delta u'_m} &= \frac{d'_1}{z_b} + 1, \\ \frac{\Delta u'_l}{\Delta u'_r} &= \frac{d'_2}{z_b} + 1, \end{aligned} \quad (1)$$

where just like  $\Delta u'_l, \Delta u'_m$  and  $\Delta u'_r$  are optical flow of  $N_m(t)$  and  $N_r(t)$  mapping on the realistic image plane, respectively, which are not shown in the figure for simplicity. And  $d'_1$  and  $d'_2$  are estimated depth difference  $d_1$  and  $d_2$ , that can be calculated by the optical flow. The other is

$$\frac{d'_1}{d'_2} = \frac{d_1}{d_2} \cdot \frac{f_a \Delta x + (z_a + d_2) \Delta v}{f_a \Delta x + (z_a + d_1) \Delta v}, \quad (2)$$

where  $\Delta v$  is the distance of the recording image plane moving down vertically, caused by the shake of attack carrier. Here, we only consider vertical direction of movement. From these relations, we have some important conclusions:

- In real scene, we can obtain the correct estimate of relative depth:  $d'_1/d'_2 = d_1/d_2$ .
- In print attack scene,  $\Delta u'_l = \Delta u'_m, \Delta u'_l = \Delta u'_r$ . Then we can use Eq. 1 to show  $d'_1 = 0, d'_2 = 0$ , which suggests the face is a plane.
- In replay attack scene, if  $\Delta v = 0$ , then  $d'_1/d'_2 = d_1/d_2$ . In this case, there is no abnormality of relative depth. In the meantime, if there are no static spoofing cues, we call this case **Perfect Spoofing Scene(PSS)**. However, making up PSS attacks is costly and approximately impossible in practice. Thus we usually assume  $\Delta v \neq 0$ , where  $d'_1/d'_2 \neq d_1/d_2$ . The estimated relative depth in attacking scenes is different from that in real scenes. And  $\Delta v$  as well as  $d'_1/d'_2$  usually varies in the long-term sequence. Such variations can also be taken as a valid spoofing cue.
- If  $d_2$  denotes the largest depth difference among facial points, then  $d_1/d_2 \in [0, 1]$ , showing that constraining depth label of living face to  $[0, 1]$  is valid. For spoofing scenes, the abnormal relative depth is too complex to be computed directly. Therefore, we merely set depth label of spoofing face to all 0 to distinguish it from living label, making the model learn the abnormality under depth supervision itself.

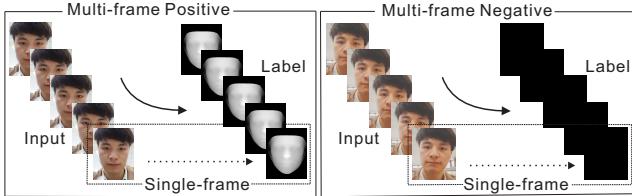


Figure 3. The examples of our input and label for single-frame and multi-frame.

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 0 \end{bmatrix}_0, \begin{bmatrix} 0 & 1 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 0 \end{bmatrix}_1, \begin{bmatrix} 0 & 0 & 1 \\ 0 & -1 & 0 \\ 0 & 0 & 0 \end{bmatrix}_2, \begin{bmatrix} 0 & 0 & 0 \\ 1 & -1 & 0 \\ 0 & 0 & 0 \end{bmatrix}_3, \\ \begin{bmatrix} 0 & 0 & 0 \\ 0 & -1 & 1 \\ 0 & 0 & 0 \end{bmatrix}_4, \begin{bmatrix} 0 & 0 & 0 \\ 0 & -1 & 0 \\ 1 & 0 & 0 \end{bmatrix}_5, \begin{bmatrix} 0 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 1 & 0 \end{bmatrix}_6, \begin{bmatrix} 0 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{bmatrix}_7$$

Figure 4. The kernel  $K_i^{contrast}$  in contrastive depth loss.

More details about the derivation are included in the supplementary material.

### 3.2. Single-frame Part

Single-frame part is important to learn static spoofing information. In this work, we train a simple CNN to regress the depth map instead of traditional classification with softmax loss. In the following, we introduce the proposed depth supervised single-frame architecture in two aspects: depth generation and network structure.

#### 3.2.1 Depth Generation

We train the single-frame network with facial depth map, which reveals face location and 3D shape of the face in the 2D plane image, and extracts useful information in the recognition of real face or not. To distinguish living faces from spoofing faces, we normalize living depth map in a range of  $[0, 1]$ , while setting spoofing depth map to 0 [20].

In this module, we adopt dense face alignment method PRNet [13] to estimate the 3D shape of the living face. PRNet can be used to project the 3D shape of a complete face into UV space. Through this method, we obtain a group of vertices  $V_{n \times 3}$  representing  $n$  3D coordinates of facial keypoints. Since these coordinates are sparse when mapped to the 2D plane image, we interpolate them to obtain dense face coordinates. By mapping and normalizing the interpolated coordinates to a 2D plane image, the generated facial depth map can be represented by  $D \in \mathbb{R}^{32 \times 32}$ . Fig. 3 shows the labels and corresponding inputs of the proposed model.

#### 3.2.2 Network Structure

As shown in Fig. 5, there are three main cascaded blocks connected after one convolution layer in our single-frame part. Each block is composed of three convolution layers

and one pooling layer. We resize pooling features to the pre-defined size of  $32 \times 32$  and concatenate them into one tensor, which is used to regress the depth map followed by three subsequent convolutional groups.

Given an original RGB image  $I^{256 \times 256 \times 3}$ , we can obtain the estimated depth map  $D_{single} \in \mathbb{R}^{32 \times 32}$  from the single-frame part. Supervised by the generated “ground truth” depth  $D \in \mathbb{R}^{32 \times 32}$ , we design a novel depth loss with the following two parts. One is squared Euclidean norm loss between  $D_{single}$  and  $D$  for an absolute depth regression:

$$L_{single}^{absolute} = \|D_{single} - D\|_2^2, \quad (3)$$

and the other is a contrastive depth loss:

$$L_{single}^{contrast} = \sum_i \|K_i^{contrast} \odot D_{single} - K_i^{contrast} \odot D\|_2^2, \quad (4)$$

where  $\odot$  represents depthwise separable convolution operation [16],  $K_i^{contrast}$  represents the contrastive convolution kernel shown in Fig. 4, and  $i$  indexes the location of “1” around “-1”. As shown in Eq. 4 and Fig. 4, the contrastive depth loss aims to learn the topography of each pixel, which gives constraints to the contrast from the pixel to its neighbors.

Putting together, the single-frame loss can be written as:

$$L_{single} = L_{single}^{absolute} + L_{single}^{contrast}, \quad (5)$$

where  $L_{single}$  is the final loss used in our single-frame part.

### 3.3. Multi-frame Part

In this part, we explore the short-term and the long-term motion in temporal domain. Specially, the short-term motion is extracted by optical flow guided feature block (OFFB) and the long-term motion is obtained by ConvGRU module.

#### 3.3.1 Short-term Motion

We use OFF block (OFFB) derived from [32] to extract short-term motion. The famous brightness constant constraint in traditional optical flow can be formulated as:

$$I(x, y, t) = I(x + \Delta x, y + \Delta y, t + \Delta t), \quad (6)$$

where  $I(x, y, t)$  represents the brightness at the location  $(x, y)$  of the frame at time  $t$ . From Eq. 6, we can derive:

$$\frac{\partial I(p)}{\partial x} v_x + \frac{\partial I(p)}{\partial y} v_y + \frac{\partial I(p)}{\partial t} = 0, \quad (7)$$

where  $p = (x, y, t)$ ,  $v_x$  and  $v_y$  represent the two dimensional velocity of the pixel at  $p$ .  $\frac{\partial I(p)}{\partial x}$ ,  $\frac{\partial I(p)}{\partial y}$  and  $\frac{\partial I(p)}{\partial t}$  are

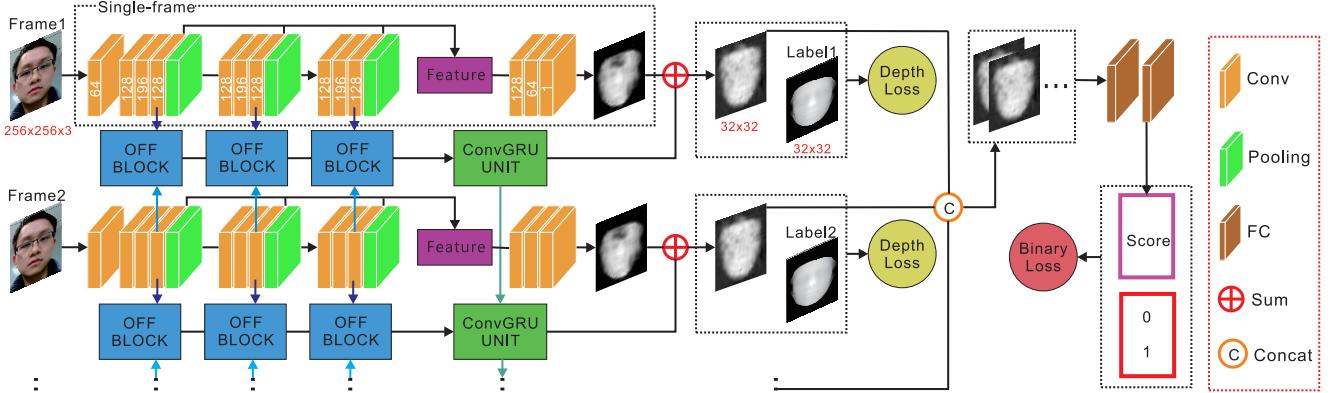


Figure 5. **The pipeline of proposed architecture.** The inputs are consecutive frames in a fixed interval. Our single-frame part aims to extract features at various levels and to output the single-frame estimated facial depth. OFF blocks take single-frame features from two consecutive frames as inputs and calculate short-term motion features. Then the final OFF features are fed into the ConvGRUs to obtain long-term motion information, and output the residual of single-frame facial depth. Finally, the combined estimated multi-frame depth maps are supervised by the depth loss and binary loss in respective manners.

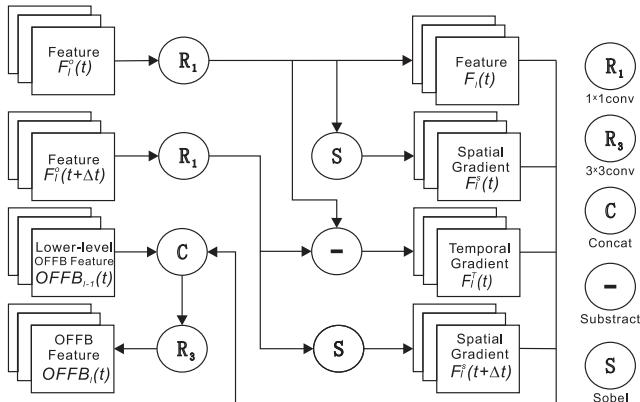


Figure 6. The architecture of our OFF block.

gradients of  $I(p)$  with respects to  $x$ ,  $y$  and the time, respectively.  $(v_x, v_y)$  is exactly the optical flow. In Eq. 7, we can see that  $\vec{I}(p) = [\frac{\partial I(p)}{\partial x}, \frac{\partial I(p)}{\partial y}, \frac{\partial I(p)}{\partial t}]$  is orthogonal to  $[v_x, v_y, 1]$ . Obviously,  $\vec{I}(p)$  is guided by the optical flow. By replacing image  $I$  by features level in equations above,  $\vec{I}(p)$  can be reformulated as:

$$\vec{F}(I; w; t)(p) = \left[ \frac{\partial F(I; w; t)(p)}{\partial x}, \frac{\partial F(I; w; t)(p)}{\partial y}, \frac{\partial F(I; w; t)(p)}{\partial t} \right], \quad (8)$$

where  $F$  is the extracted feature from image  $I$ , and  $w$  is the parameters of the feature extraction function.  $\vec{F}(I; w; t)(p)$  is called optical flow guided features (OFF) [32], which encodes spatial and temporal gradients guided by the feature-level optical flow.

Different from OFF [32], we add extra spatial and temporal gradients and morphological information into our own OFF block (OFFB). As shown in Fig. 6, there are five sub-modules:  $F_l(t)$ ,  $F_l^S(t)$ ,  $F_l^S(t+Δt)$ ,  $F_l^T(t)$ ,  $OFFB_{l-1}(t)$ ,

where  $F_l(t)$  is reduced from the original single-frame feature  $F_l^O(t)$  at feature level  $l$  and time  $t$  to get fundamental morphological information,  $F_l^S(t)$  and  $F_l^S(t+Δt)$  are the spatial gradients at time  $t$  and  $t+Δt$ ,  $F_l^T(t)$  is the temporal gradient between time  $t$  and  $t+Δt$ , and  $OFFB_{l-1}(t)$  is the OFFB feature from the previous feature level  $l-1$ . To reduce the training burden, we introduce a reduction operator –  $1 \times 1$  convolution layer – at the original feature. We concatenate all of the five sub-modules, and feed them into a  $3 \times 3$  convolution layer to reduce the feature dimension and establish  $OFFB_l(t)$ . More details about spatial gradient  $F_l^S(t)$  and temporal gradient  $F_l^T(t)$  can refer to [32]. In the following of this paper, we denote the output of the final OFF block by  $OFFB(t)$ .

### 3.3.2 Long-term Motion

As a short-term motion feature, OFFB feature primarily captures the motion information between two consecutive frames, whereas has difficulty in fusing long-sequence motion. In this regard, we resort to ConvGRU module to gain the long-term motion.

GRU [9] derives from Long Short-term Memory (LSTM) [15], which has simpler structure and fewer trainable parameters. Similar to LSTM, GRU aims to process long sequence information as well.

However, normal GRU neglects the spatial information in its hidden units. Thus, we should take into account the convolution operation in the hidden layers to model the spatiotemporal sequences, and thus propose the Convolutional Gated Recurrent Units (ConvGRU) related to ConvL-

STM [31]. The ConvGRU can be described below:

$$\begin{aligned} R_t &= \sigma(K_r \otimes [H_{t-1}, X_t]), \\ U_t &= \sigma(K_u \otimes [H_{t-1}, X_t]), \\ \hat{H}_t &= \tanh(K_{\hat{h}} \otimes [R_t * H_{t-1}, X_t]), \\ H_t &= (1 - U_t) * H_{t-1} + U_t * \hat{H}_t, \end{aligned} \quad (9)$$

where  $X_t, H_t, U_t$  and  $R_t$  are the matrix of input, output, update gate and reset gate,  $K_r, K_u, K_{\hat{h}}$  are the kernels in the convolution layer,  $\otimes$  is convolution operation,  $*$  denotes element wise product, and  $\sigma$  is the sigmoid activation function. By feeding  $\{\text{OFFB}(t)\}_{N_f-1}^{N_f}$  into  $\{X_t\}_{N_f-1}^{N_f}$ , we set up depth maps  $\{D_{multi}^t\}_{N_f-1}^{N_f}$ , where  $D_{multi}^t = H_t$  and  $N_f$  denotes the number of input frames. Then, based on the residual idea, we integrate the single depth map and multi-depth map:

$$D_{fusion}^t = \alpha \cdot D_{single}^t + (1 - \alpha) \cdot D_{multi}^t, \alpha \in [0, 1] \quad (10)$$

where  $\alpha$  is the weight of  $D_{single}^t$  in  $D_{fusion}^t$ . Finally, we build up the set of multi-frame depth maps  $\{D_{fusion}^t\}_{N_f-1}^{N_f}$ .

### 3.3.3 Multi-frame Loss

We make the final decision to discriminate live vs. spoof in the multi-frame part. Nevertheless, in view of the potentially unclear depth map, we hereby consider a binary loss when looking for the difference between living and spoofing depth map. Note that the depth supervision is decisive, whereas the binary supervision takes an assistant role to discriminate the different kinds of depth maps. In this regard, we establish the following multi-frame loss:

$$L_{multi}^{absolute}(t) = \|D_{fusion}^t - D^t\|_2^2, \quad (11)$$

$$L_{multi}^{contrast}(t) = \sum_i \|K_i^{contrast} \odot D_{fusion}^t - K_i^{contrast} \odot D^t\|_2^2, \quad (12)$$

$$L_{multi}^{depth} = \sum_t^{N_f-1} (L_{multi}^{absolute}(t) + L_{multi}^{contrast}(t)), \quad (13)$$

$$L_{multi}^{binary} = -B^t * \log(fcs(\{D_{fusion}^t\}_{N_f-1}^{N_f})), \quad (14)$$

$$L_{multi} = \beta \cdot L_{multi}^{binary} + (1 - \beta) \cdot L_{multi}^{depth}, \quad (15)$$

where  $D^t$  and  $B^t$  are depth label and binary label at time  $t$ ,  $\{D_{fusion}^t\}_{N_f-1}^{N_f}$  is the concatenated depth maps of  $N_f - 1$  frames,  $fcs$  denotes two fully connected layers and one softmax layer after the concatenated depth maps, which outputs the logits of two classes,  $\beta$  is the weight of binary loss in the final multi-frame loss  $L_{multi}$ . In Eq. 14, we use cross entropy loss to calculate the binary loss. In Eq. 15, we simply take a sum of the binary loss and depth loss.

## 4. Experiment

### 4.1 Databases and Metrics

#### 4.1.1 Databases

Four databases - OULU-NPU [7], SiW [20], CASIA-MFSD [37], Replay-Attack [8] are used in our experiment. OULU-NPU [7] is a high-resolution database, consisting of 4950 real access and spoofing videos. This database contains four protocols to validate the generalization of models. SiW [20] contains more live subjects and three protocols are used for testing. CASIA-MFSD [37] and Replay-Attack [8] are databases which contain low-resolution videos. We use these two databases for cross testing.

#### 4.1.2 Metrics

In OULU-NPU and SiW dataset, we follow the original protocols and metrics for a fair comparison. OULU-NPU and SiW utilize 1) Attack Presentation Classification Error Rate  $APCER$ , which evaluates the highest error among all PAIs(e.g. print or display), 2) Bona Fide Presentation Classification Error Rate  $BPCER$ , which evaluates error of real access data, and 3)  $ACER$  [17], which evaluates the mean of  $APCER$  and  $BPCER$ :

$$ACER = \frac{APCER + BPCER}{2}. \quad (16)$$

HTER is adopted in the cross testing between CASIA-MFSD and Repaly-Attack, which evaluates the mean of False Rejection Rate (FRR) and the False Acceptance Rate (FAR):

$$HTER = \frac{FRR + FAR}{2}. \quad (17)$$

### 4.2 Implementation Details

#### 4.2.1 Training Strategy

The proposed method combines the single-frame part and multi-frame part. Two-stage strategy is applied in the training process. *Stage 1*: We train the single-frame part by the single-frame depth loss, in order to learn a fundamental representation. *Stage 2*: We fix the parameters of the single-frame part, and finetune the parameters of multi-frame part by the depth loss and binary loss. Note that the diverse data should be adequately shuffled for the stability of training and generalization of the learnt model. The network is fed by  $N_f$  frames, which are sampled by a interval of three frames. This sampling interval makes sampled frames maintain enough temporal information in the limitation of GPU memory.

#### 4.2.2 Testing Strategy

For the final classification score, we feed the sequential frames into the network and obtain depth maps

Prot.	Method	APCER(%)	BPCER(%)	ACER(%)
1	CPqD	2.9	10.8	6.9
	GRADIANT	1.3	12.5	6.9
	FAS-BAS [20]	1.6	1.6	1.6
	<b>OURs</b>	2.5	0.0	<b>1.3</b>
2	MixedFASNet	9.7	2.5	6.1
	FAS-BAS [20]	2.7	2.7	2.7
	GRADIANT	3.1	1.9	2.5
	<b>OURs</b>	1.7	2.0	<b>1.9</b>
3	MixedFASNet	5.3±6.7	7.8±5.5	6.5±4.6
	<b>OURs</b>	5.9±1.9	5.9±3.0	<b>5.9±1.0</b>
	GRADIANT	2.6±3.9	5.0±5.3	3.8±2.4
	FAS-BAS [20]	2.7±1.3	3.1±1.7	<b>2.9±1.5</b>
4	Massy_HNU	35.8±35.3	8.3±4.1	22.1±17.6
	GRADIANT	5.0±4.5	15.0±7.1	10.0±5.0
	FAS-BAS [20]	9.3±5.6	10.4±6.0	9.5±6.0
	<b>OURs</b>	14.2±8.7	4.2±3.8	<b>9.2±3.4</b>

Table 1. The results of intra testing on four protocols of OULU-NPU [7].

$\{\mathbf{D}_{fusion}^t\}^{N_f-1}$  and the living logits  $\hat{b}$  in  $fcs(\mathbf{D}_{fusion}^t)$ . The final living score can be obtained by:

$$score = \beta \cdot \hat{b} + (1 - \beta) \cdot \frac{\sum_t^{N_f-1} \|\mathbf{D}_{fusion}^t * \mathbf{M}_{fusion}^t\|_1}{N_f - 1}, \quad (18)$$

where  $\beta$  is the same as that in equation 15,  $\mathbf{M}_{fusion}^t$  is the mask of face at frame  $t$ , which can be generated by the dense face landmarks in PRNet [13], and the second module denotes that we compute the mean of depth values in the facial areas as one part of the score.

#### 4.2.3 Hyperparameter Setting

We implement our proposed method in Tensorflow [1], with a learning rate of 3e-3 for single-frame part and 1e-2 for multi-frame part. The batch size of single-frame part is 10 and that of multi-frame part is 2 with  $N_f$  being 5 in most of our experiment, except that batch size being 4 and  $N_f$  being 3 in protocol 2 of OULU-NPU. Adadelta optimizer is used in our training procedure, with  $\rho$  as 0.95 and  $\epsilon$  as 1e-8. We set  $\alpha$  and  $\beta$  to optimal values by our experimental experience, and according to the analysis of the following section that protocol 4 in OULU-NPU is most challenging to the generalization, we recommend that the parameters  $\alpha = 0.8$  and  $\beta = 0.9$  are suitable for the realistic scenes.

### 4.3. Experimental Comparison

#### 4.3.1 Intra Testing

We compare the performance of intra testing on OULU-NPU and SiW dataset. OULU-NPU proposes four protocols to evaluate the generalization of the developed face presentation attack detection (PAD) methods. Protocol 1 is designed to evaluate the generalization of PAD methods under previously unseen illumination and background scene.

Prot.	Method	APCER(%)	BPCER(%)	ACER(%)
1	FAS-BAS [20]	3.58	3.58	3.58
	<b>OURs</b>	0.96	0.50	<b>0.73</b>
2	FAS-BAS [20]	0.57±0.69	0.57±0.69	0.57±0.69
	<b>OURs</b>	0.08±0.17	0.21±0.16	<b>0.15±0.14</b>
3	FAS-BAS [20]	8.31±3.81	8.31±3.80	8.31±3.81
	<b>OURs</b>	3.10±0.79	3.09±0.83	<b>3.10±0.81</b>

Table 2. The results of intra testing on three protocols of SiW [20].

Method	Train	Test	Train	Test
	CASIA-MFSD	Replay-Attack	Replay-Attack	CASIA-MFSD
Motion [11]	50.2			47.9
LBP-1 [11]	55.9			57.6
LBP-TOP [11]	49.7			60.6
Motion-Mag [3]	50.1			47.0
Spectral cubes [28]	34.4			50.0
CNN [35]	48.5			45.5
LBP-2 [4]	47.0			39.6
Colour Texture [5]	30.3			37.7
FAS-BAS [20]	27.6			28.4
<b>OURs</b>	<b>17.5</b>			<b>24.0</b>

Table 3. The results of cross testing between CASIA-MFSD and Replay-Attack. The evaluation metric is HTER(%).

Protocol 2 is designed to evaluate the generalization of PAD methods under unseen attack medium, such as unseen printers or displays. Protocol 3 utilizes a Leave One Camera Out (LOCO) protocol, in order to study the effect of the input camera variation. Protocol 4 considers all above factors and integrates all the constraints from protocols 1 to 3, so protocol 4 is the most challenging. Table 1 shows that our proposed method ranks first on three protocols - protocol 1,2,4, and ranks third on protocol 3. We can see that our model performs well at the generalization of external environment and attack mediums, and is slightly worse when it comes to the input camera variation. It's worth noting that our proposed method has the lowest mean and std of ACER in protocol 4, which is most suitable for the real-life scenarios.

Tab. 2 compares the performance of our method with FAS-BAS [20] on SiW [20]. According to the purposes of three protocols on SiW [20] and the results in Tab. 2, we can see that our method performs great advantages on the generalization of (1) variations of face pose and expression, (2) variations of different spoof mediums, (3) cross PAI.

#### 4.3.2 Cross Testing

We utilize CASIA-MFSD and Replay-Attack dataset to perform cross testing. This can be regarded as two testing protocols. One is training on the CASIA-MFSD and testing on Replay-Attack, which we name protocol CR; the other is training on the Replay-Attack and testing on CASIA-MFSD, which we name protocol RC. In table 3, we see HTER of our proposed method is 17.5 on protocol CR and 24.0 on protocol RC, reducing 36.6% and 15.5% respectively compared with the previous state of the art. The im-

Module	Single-frame	OFF-block	ConvGRU	D-S	B-S	ACER(%)
Model 1	✓			✓		4.4
Model 2*	✓	✓		✓		3.5
Model 2	✓	✓		✓	✓	2.3
Model 3*	✓		✓	✓		3.3
Model 3	✓		✓	✓	✓	3.1
Model 4*	✓	✓	✓	✓		1.7
Model 4	✓	✓	✓	✓	✓	<b>1.3</b>

Table 4. The results of ablation study on protocol 1 of OULU-NPU. B-S denotes binary-supervision, and D-S denotes depth-supervision.

Module	Depth Loss		ACER(%)
	Euclidean Depth Loss	Contrastive Depth Loss	
Model 1*	✓		5.8
Model 1	✓	✓	<b>4.4</b>

Table 5. The results of ablation study on the influence of contrastive depth loss on protocol 1 of OULU-NPU.

provement of performance on cross testing demonstrates the generalization and superiority of our proposed method.

### 4.3.3 Ablation Study

We implement experiment on seven architectures to demonstrate the advantages of our proposed sequential structure under the supervision of depth. As shown in table 4, Model 1 is the single-frame part of our method. Model 2 combines single-frame CNN with OFFB under binary supervision and depth supervision. Model 3 combines single-frame CNN with ConvGRU under binary supervision and depth supervision. Model 2\* and model 3\* discard the binary supervision. Model 4 is our complete architecture, integrating all modules. Model 4\* discards binary supervision compared with model 4. Comparing ACER of model 2\* and model 3\* with that of model 1, we see that our OFFB module and ConvGRU module both improve the performance of face anti-spoofing. And the ACER of model 4\* shows that the combination of OFFB module and ConvGRU module has more positive effects. Via discarding the binary supervision, we test the effect of binary supervision on our model 2\*, 3\* and 4\*. In this model, we find that multi-frame model with simple depth supervision can also outperform the single-frame model and binary supervision indeed assists the model to distinguish live vs. spoof.

In table 5, we study the influence of contrastive depth loss. Model 1 is our single-frame model supervised by both the euclidiean depth loss and the contrastive depth loss, while model 1\* is supervised only by euclidiean depth loss. Comparing model 1 with model 1\*, we can see that contrastive depth loss can improve the generalization of our model.

Moreover, the inference of the model 1 costs around 18 ms and that of model 5 costs around 96 ms, which indicates that our method is efficient enough to be applied in reality.

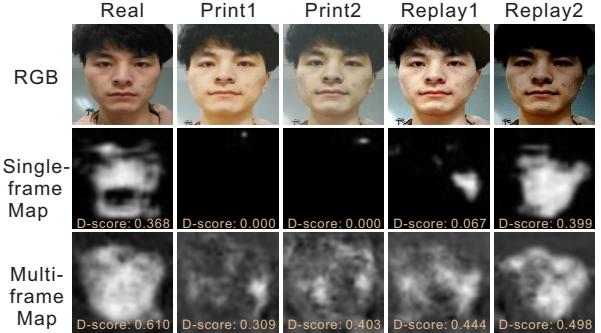


Figure 7. The generated results of a group of hard samples in OULU-NPU. D-score denotes the living score calculated by the mean of depth values in facial area, which is the depth subpart in equation 18.

### 4.3.4 Qualitative Analysis

Figure 7 presents the generated depth maps of a group of hard samples for the same person ID in OULU-NPU. From figure 7, we can see that our multi-frame maps are more complete than single-frame maps in real scenes. Although the multi-frame maps in spoofing scenes are visually noisier than those in the single-frame maps, the discrimination is obvious when only considering the multi-frame maps themselves. Specially, in single-frame maps, the D-score of real scene is 0.368, which is lower than that of replay2 scene. This is probably caused by the illumination condition in replay2 scene. By contrast, in multi-frame maps, the D-score of real scene is higher than the D-scores in all of the attack scenes. We can observe the discrimination among the multi-frame maps obviously.

It's worth noting that this group of images are results of hard samples, which is failure case in single frame but success case in multi frame. We use residual architecture, adding the multi-frame depth to the single-frame depth, which leads to the depth values in multi frame are higher than those of single frame.

## 5. Discussion

We only discuss a simple case of equal motion in vertical dimension in Sec. 3.1. In reality, the facial variation and motion are more complex, including forward/backward motion, rotation, deformation and so on. In these cases, we still assume that there are discriminative patterns of temporal depth and motion between living and spoofing faces in the temporal domain. Extension of theory and application on these aspects is our future research. Face anti-spoofing based on facial motion and depth is indeed promising and valuable.

## 6. Conclusions

In this paper, we analyze the usefulness of motion and depth in presentation attack detection. According to the

analysis, we propose a novel face anti-spoofing method, which is depth supervised and consists of adequate temporal information. To seek the spatiotemporal information, we take OFF block as short-term motion module and ConvGRU as long-term motion module, and then combine them into our architecture. Our proposed method can discover spoof patterns efficiently and accurately under depth-supervision. Extensive experimental results demonstrate the superiority of our method.

## References

- [1] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard, et al. Tensorflow: a system for large-scale machine learning. In *OSDI*, volume 16, pages 265–283, 2016. [7](#)
- [2] Y. Atoum, Y. Liu, A. Jourabloo, and X. Liu. Face anti-spoofing using patch and depth-based cnns. In *IJCB*, pages 319–328, 2017. [2](#)
- [3] S. Bharadwaj, T. I. Dhamecha, M. Vatsa, and R. Singh. Computationally efficient face spoofing detection with motion magnification. In *CVPRW*, pages 105–110, 2013. [7](#)
- [4] Z. Boulkenafet, J. Komulainen, and A. Hadid. Face anti-spoofing based on color texture analysis. In *ICIP*, pages 2636–2640, 2015. [7](#)
- [5] Z. Boulkenafet, J. Komulainen, and A. Hadid. Face spoofing detection using colour texture analysis. *IEEE Transactions on Information Forensics and Security*, 11(8):1818–1830, 2016. [1, 7](#)
- [6] Z. Boulkenafet, J. Komulainen, and A. Hadid. Face anti-spoofing using speeded-up robust features and fisher vector encoding. *IEEE Signal Processing Letters*, 24(2):141–145, 2017. [1, 2](#)
- [7] Z. Boulkenafet, J. Komulainen, L. Li, X. Feng, and A. Hadid. Oulu-npu: A mobile face presentation attack database with real-world variations. In *FGR*, pages 612–618, 2017. [6, 7, 14](#)
- [8] I. Chingovska, A. Anjos, and S. Marcel. On the effectiveness of local binary patterns in face anti-spoofing. In *Biometrics Special Interest Group*, pages 1–7, 2012. [6](#)
- [9] K. Cho, B. Van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio. Learning phrase representations using rnn encoder-decoder for statistical machine translation. *Computer Science*, 2014. [5](#)
- [10] T. de Freitas Pereira, A. Anjos, J. M. De Martino, and S. Marcel. Lbp- top based countermeasure against face spoofing attacks. In *ACCV*, pages 121–132, 2012. [1, 2](#)
- [11] 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*, pages 1–8, 2013. [2, 7](#)
- [12] L. Feng, L.-M. Po, Y. Li, X. Xu, F. Yuan, T. C.-H. Cheung, and K.-W. Cheung. Integration of image quality and motion cues for face anti-spoofing: A neural network approach. *Journal of Visual Communication and Image Representation*, 38:451–460, 2016. [2, 3](#)
- [13] Y. Feng, F. Wu, X. Shao, Y. Wang, and X. Zhou. Joint 3d face reconstruction and dense alignment with position map regression network. In *CVPR*, 2017. [4, 7](#)
- [14] J. Gan, S. Li, Y. Zhai, and C. Liu. 3d convolutional neural network based on face anti-spoofing. In *ICMIP*, pages 1–5, 2017. [1, 2, 3](#)
- [15] S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural Computation*, 9(8):1735–1780, 1997. [5](#)
- [16] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam. Mobilenets: Efficient convolutional neural networks for mobile vision applications. In *CVPR*, 2017. [4](#)
- [17] international organization for standardization. Iso/iec jtc 1/sc 37 biometrics: Information technology biometric presentation attack detection part 1: Framework. In <https://www.iso.org/obp/ui/iso>, 2016. [6](#)
- [18] J. Komulainen, A. Hadid, and M. Pietikäinen. Context based face anti-spoofing. In *BTAS*, pages 1–8, 2013. [2](#)
- [19] 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*, pages 1–6, 2016. [2](#)
- [20] Y. Liu, A. Jourabloo, and X. Liu. Learning deep models for face anti-spoofing: Binary or auxiliary supervision. In *CVPR*, pages 389–398, 2018. [1, 2, 4, 6, 7, 14](#)
- [21] O. Lucena, A. Junior, V. Moia, R. Souza, E. Valle, and R. Lotufo. Transfer learning using convolutional neural networks for face anti-spoofing. In *International Conference Image Analysis and Recognition*, pages 27–34, 2017. [1, 2](#)
- [22] J. Määttä, A. Hadid, and M. Pietikäinen. Face spoofing detection from single images using micro-texture analysis. In *IJCB*, pages 1–7, 2011. [2](#)
- [23] C. Nagpal and S. R. Dubey. A performance evaluation of convolutional neural networks for face anti spoofing. *arXiv preprint arXiv:1805.04176*, 2018. [1, 2](#)
- [24] G. Pan, L. Sun, Z. Wu, and S. Lao. Eyeblink-based anti-spoofing in face recognition from a generic webcam. In *ICCV*, pages 1–8, 2007. [3](#)
- [25] K. Patel, H. Han, and A. K. Jain. Cross-database face anti-spoofing with robust feature representation. In *Chinese Conference on Biometric Recognition*, pages 611–619, 2016. [2, 3](#)
- [26] K. Patel, H. Han, and A. K. Jain. Secure face unlock: Spoof detection on smartphones. *IEEE transactions on information forensics and security*, 11(10):2268–2283, 2016. [1, 2](#)
- [27] B. Peixoto, C. Michelassi, and A. Rocha. Face liveness detection under bad illumination conditions. In *ICIP*, pages 3557–3560. IEEE, 2011. [2](#)
- [28] A. Pinto, H. Pedrini, W. R. Schwartz, and A. Rocha. Face spoofing detection through visual codebooks of spectral temporal cubes. *IEEE Transactions on Image Processing*, 24(12):4726–4740, 2015. [7](#)
- [29] B. Shahraray and M. K. Brown. Robust depth estimation from optical flow. In *ICCV*, 1988. [2](#)
- [30] R. Shao, X. Lan, and P. C. Yuen. Deep convolutional dynamic texture learning with adaptive channel-discriminability for 3d mask face anti-spoofing. In *IJCB*, pages 748–755, 2017. [1, 3](#)
- [31] X. Shi, Z. Chen, H. Wang, W. C. Woo, W. C. Woo, and W. C. Woo. Convolutional lstm network: a machine learning approach for precipitation nowcasting. In *International Conference on Image Processing*, 2017. [1, 2](#)

- ence on Neural Information Processing Systems, pages 802–810, 2015. 6
- [32] S. Sun, Z. Kuang, L. Sheng, W. Ouyang, and W. Zhang. Optical flow guided feature: A fast and robust motion representation for video action recognition. In *CVPR*, pages 1390–1399, 2018. 4, 5
- [33] 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*, pages 504–517, 2010. 2
- [34] Z. Xu, S. Li, and W. Deng. Learning temporal features using lstm-cnn architecture for face anti-spoofing. In *ACPR*, pages 141–145, 2015. 1, 3
- [35] J. Yang, Z. Lei, and S. Z. Li. Learn convolutional neural network for face anti-spoofing. *Computer Science*, 9218:373–384, 2014. 1, 2, 7
- [36] J. Yang, Z. Lei, S. Liao, and S. Z. Li. Face liveness detection with component dependent descriptor. In *ICB*, page 2, 2013. 2
- [37] Z. Zhang, J. Yan, S. Liu, Z. Lei, D. Yi, and S. Z. Li. A face antispoofing database with diverse attacks. In *ICB*, pages 26–31, 2012. 6

## Appendix

### A. Temporal Depth in Face Anti-spoofing

In this section, we use some simple examples to explain that exploiting temporal depth and motion is reasonable in the face anti-spoofing task.

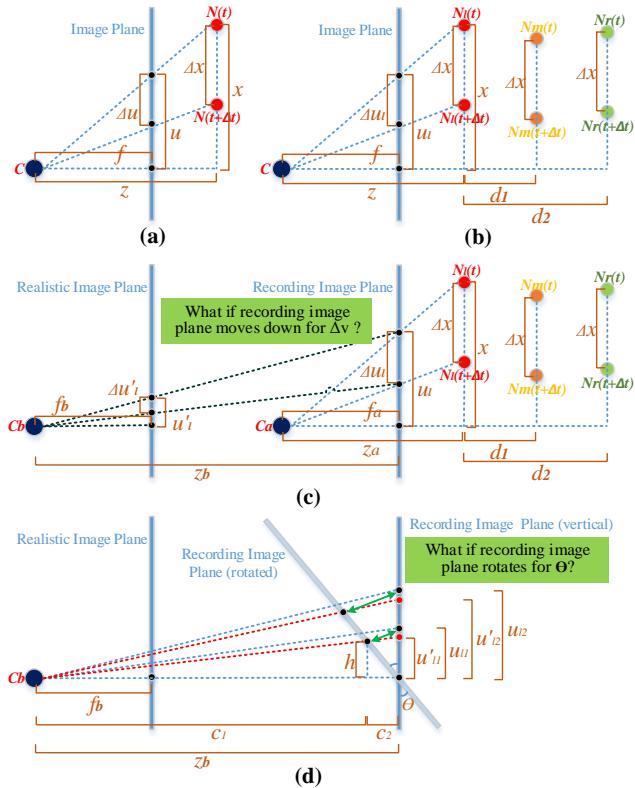


Figure 8. The schematic diagram of motion and depth variation in different scenes.

#### A.1. Basic Scene

As shown in Fig. 8(a), node  $C$  denotes the camera focus. *Image Plane* represents the image plane of camera.  $N(t)$  is one facial point at time  $t$ , and  $N(t + \Delta t)$  is the corresponding point when  $N(t)$  moves down vertically for  $\Delta x$  at time  $t + \Delta t$ . For example,  $N(t)$  can be the point of nose or ear.  $f$  denotes the focal distance, and  $z$  is the horizontal distance from the focal point to the point  $N(t)$ .  $u$  and  $x$  are the corresponding coordinates in vertical dimension. When  $N(t)$  moves down vertically to  $N(t + \Delta t)$  for  $\Delta x$ , the motion can be reflected on the image plane as  $\Delta u$ . According to the camera model, we can obtain:

$$\begin{aligned} \frac{x}{u} &= \frac{z}{f}, \\ \Leftrightarrow u &= \frac{fx}{z}. \end{aligned} \quad (19)$$

When  $N(t)$  moves down vertically for  $\Delta x$  to  $N(t + \Delta t)$ , the  $\Delta u$  can be achieved:

$$\Delta u = \frac{f \Delta x}{z}. \quad (20)$$

As shown in Fig. 8(b), to distinguish points  $N_l$ ,  $N_m$  and  $N_r$ , we transform Eq. 20 and get  $\Delta u_l$ ,  $\Delta u_m$  and  $\Delta u_r$  ( $\Delta u_m$  and  $\Delta u_r$  are not shown in the figure):

$$\begin{aligned} \Delta u_l &= \frac{f \Delta x}{z}, \\ \Delta u_m &= \frac{f \Delta x}{z + d_1}, \\ \Delta u_r &= \frac{f \Delta x}{z + d_2}, \end{aligned} \quad (21)$$

where  $d_1$  and  $d_2$  are the corresponding depth difference. From Eq. 21, there are:

$$\begin{aligned} \frac{\Delta u_l}{\Delta u_m} &= \frac{z + d_1}{z} = \frac{d_1}{z} + 1, \\ \frac{\Delta u_l}{\Delta u_r} &= \frac{z + d_2}{z} = \frac{d_2}{z} + 1. \end{aligned} \quad (22)$$

Removing  $z$  from Eq. 22,  $d_1/d_2$  can be obtained:

$$\frac{d_1}{d_2} = \frac{\frac{\Delta u_l}{\Delta u_m} - 1}{\frac{\Delta u_l}{\Delta u_r} - 1}, \quad (23)$$

In this equation, we can see that the relative depth  $d_1/d_2$  can be estimated by the motion of three points, when  $d_2 \neq 0$ . The equations above are about the real scenes. In the following, we will introduce the derivation of attack scenes.

#### A.2. Attack Scene

##### A.2.1 What if the attack carriers move?

As shown in Fig. 8(c), there are two image spaces in attack scenes: one is recording image space, where we replace  $z$ ,  $f$  by  $z_a$ ,  $f_a$ , and the other is realistic image space, where we replace  $z$ ,  $f$  by  $z_b$ ,  $f_b$ . In the recording image space, it's similar to Eq. 21:

$$\begin{aligned} \Delta u_l &= \frac{f_a \Delta x}{z_a}, \\ \Delta u_m &= \frac{f_a \Delta x}{z_a + d_1}, \\ \Delta u_r &= \frac{f_a \Delta x}{z_a + d_2}, \end{aligned} \quad (24)$$

where  $\Delta u_l$ ,  $\Delta u_m$ ,  $\Delta u_r$  are the magnitude of optical flow when three points  $N_l(t)$ ,  $N_m(t)$ ,  $N_r(t)$  move down vertically for  $\Delta x$ .

In the realistic image space, there are:

$$\begin{aligned}\Delta u'_l &= \frac{f_b \Delta x_l}{z_b}, \\ \Delta u'_m &= \frac{f_b \Delta x_m}{z_b}, \\ \Delta u'_r &= \frac{f_b \Delta x_r}{z_b},\end{aligned}\quad (25)$$

where  $\Delta x_l$ ,  $\Delta x_m$  and  $\Delta x_r$  are the motion of three points on the recording image plane, and  $\Delta u_l$ ,  $\Delta u_m$ ,  $\Delta u_r$  are the corresponding values mapping on the realistic image plane.

Actually, there are  $\Delta x_l = \Delta u_l$ ,  $\Delta x_m = \Delta u_m$ ,  $\Delta x_r = \Delta u_r$ , if the recording screen is static. Now, a vertical motion  $\Delta v$  is given to the recording screen, just as  $\Delta x_l = \Delta u_l + \Delta v$ ,  $\Delta x_m = \Delta u_m + \Delta v$ ,  $\Delta x_r = \Delta u_r + \Delta v$ . By inserting  $\Delta v$ , we transform Eq. 25 into:

$$\begin{aligned}\Delta u'_l &= \frac{f_a f_b \Delta x + z_a f_b \Delta v}{z_a z_b}, \\ \Delta u'_m &= \frac{f_a f_b \Delta x + (z_a + d_1) f_b \Delta v}{(z_a + d_1) z_b}, \\ \Delta u'_r &= \frac{f_a f_b \Delta x + (z_a + d_2) f_b \Delta v}{(z_a + d_2) z_b},\end{aligned}\quad (26)$$

Due to that only  $\Delta u'_l$ ,  $\Delta u'_m$ ,  $\Delta u'_r$  can be observed directly in the sequential images, we can estimate the relative depth via  $\Delta u'_l$ ,  $\Delta u'_m$ ,  $\Delta u'_r$ . So we leverage Eq. 23 to estimate the relative depth  $d'_1/d'_2$ :

$$\frac{d'_1}{d'_2} = \frac{\frac{\Delta u'_l}{\Delta u'_m} - 1}{\frac{\Delta u'_l}{\Delta u'_r} - 1}, \quad (27)$$

and then we can insert Eq. 26 into Eq. 27 to get:

$$\frac{d'_1}{d'_2} = \frac{d_1}{d_2} \cdot \frac{f_a \Delta x + (z_a + d_2) \Delta v}{f_a \Delta x + (z_a + d_1) \Delta v}. \quad (28)$$

According to equations above, some important conclusions can be summarized:

- If  $\Delta x = 0$ , the scene can be recognized as print attack and Eq. 28 will be invalid, for  $\Delta u'_l = \Delta u'_r$ , and the denominator in Eq. 27 will be zero. So here we use Eq. 26 and

$$\begin{aligned}\frac{\Delta u'_l}{\Delta u'_m} &= \frac{d'_1}{z_b} + 1, \\ \frac{\Delta u'_l}{\Delta u'_r} &= \frac{d'_2}{z_b} + 1,\end{aligned}\quad (29)$$

to obtain:

$$d'_1 = d'_2 = 0. \quad (30)$$

In this case, it's obvious that the facial relative depth is abnormal and the face is fake.

- If  $\Delta x \neq 0$ , the scene can be recognized as replay attack.

- If  $\Delta v = 0$ , there is:

$$\frac{d'_1}{d'_2} = \frac{d_1}{d_2}. \quad (31)$$

In this case, if these two image planes are parallel and the single-frame model can not detect the static spoof cues, the model will fail in the task of face anti-spoofing, owing to that the model is hard to find the abnormality of relative depth estimated from the facial motion. We call this scene **Perfect Spoofing Scene(PSS)**. Of course, making up PSS will cost a lot and is approximately impossible in practice.

- If  $\Delta v \neq 0$  and we want to meet Eq. 31, the following equation should be satisfied:

$$\frac{f_a \Delta x + (z_a + d_2) \Delta v}{f_a \Delta x + (z_a + d_1) \Delta v} = 1, \quad (32)$$

then,

$$\begin{aligned}(d_2 - d_1) \Delta v &= 0, \\ \Leftrightarrow d_2 - d_1 &= 0, \text{ if } \Delta v \neq 0.\end{aligned}\quad (33)$$

However, in our assumption,  $d_1 \neq d_2$ , so:

$$\frac{d'_1}{d'_2} \neq \frac{d_1}{d_2}. \quad (34)$$

This equation indicates that relative depth can't be estimated precisely, if the attack carrier moves in the replay attack. And  $\Delta v$  usually varies when attack carrier moves in the long-term sequence, leading to the variation of  $d'_1/d'_2$ . This kind of abnormality is more obvious along with the long-term motion.

- If  $d_2$  denotes the largest depth difference among facial points, then  $d_1/d_2 \in [0, 1]$ , showing that constraining depth label of living face to  $[0, 1]$  is valid. As analyzed above, for spoofing scenes, the abnormal relative depth usually varies over time, so it is too complex to be computed directly. Therefore, we merely set depth label of spoofing face to all 0 to distinguish it from living label, making the model learn the abnormality under depth supervision itself.

### A.2.2 What if the attack carriers rotate?

As shown in Fig. 8(d), we rotate the recording image plane for degree  $\theta$ .  $u_{l2}, u_{l1}$  are the coordinates of  $N_l(t), N_l(t + \Delta t)$  mapping on the recording image plane. The two black

points at the *right* end of green double arrows on recording image plane (vertical) will reach the two black points at the *left* end of green double arrow on recording image plane (rotated), when the recording image plane rotates. And the corresponding values  $u_{l2}, u_{l1}$  will not change after rotation. For convenient computation, we still map the rotated points to the vertical recording image plane. And the coordinates after mapping are  $u'_{l2}, u'_{l1}$ .  $c_1, c_2, h$  are the corresponding distances shown in the figure. According to the relationship of the fundamental variables, we can obtain:

$$\begin{aligned} h &= u_{l1} \cos \theta, \\ \frac{z_b}{c_1} &= \frac{u'_{l1}}{h}, \\ c_2 &= u_{l1} \sin \theta, \\ c_1 + c_2 &= z_b. \end{aligned} \quad (35)$$

Deriving from equations above, we can get  $u'_{l1}$ :

$$u'_{l1} = \frac{z_b u_{l1} \cos \theta}{z_b - u_{l1} \sin \theta}, \quad (36)$$

and  $u'_{l2}$  can also be calculated by imitating Eq. 36:

$$u'_{l2} = \frac{z_b u_{l2} \cos \theta}{z_b - u_{l2} \sin \theta}. \quad (37)$$

Subtract  $u'_{l1}$  from  $u'_{l2}$ , the following is achieved:

$$u'_{l2} - u'_{l1} = (u_{l2} - u_{l1}) \cdot \frac{z_b^2 \cos \theta}{(z_b - u_{l1} \sin \theta)(z_b - u_{l2} \sin \theta)}. \quad (38)$$

Obviously,  $u_{l2} - u_{l1} = \Delta u_l$ . We define  $u'_{l2} - u'_{l1} = \Delta u_l^\theta$ . And then we get the following equation:

$$\begin{aligned} \Delta u_l^\theta &= \Delta u_l \cdot \frac{z_b^2 \cos \theta}{(z_b - u_{l1} \sin \theta)[z_b - (u_{l1} + \Delta u_l) \sin \theta]}, \\ \Delta u_m^\theta &= \Delta u_m \cdot \frac{z_b^2 \cos \theta}{(z_b - u_{m1} \sin \theta)[z_b - (u_{m1} + \Delta u_l) \sin \theta]}, \\ \Delta u_r^\theta &= \Delta u_r \cdot \frac{z_b^2 \cos \theta}{(z_b - u_{r1} \sin \theta)[z_b - (u_{r1} + \Delta u_l) \sin \theta]}, \end{aligned} \quad (39)$$

where the relationship between  $\Delta u_m^\theta, \Delta u_r^\theta$  and  $N_m(t), N_r(t)$  are just like that between  $\Delta u_l^\theta$  and  $N_l(t)$ , as well as  $u_{m1}, u_{r1}$ . Note that for simplification, we only discuss the situation that  $u_{l1}, u_{m1}, u_{r1}$  are all positive.

Reviewing Eq. 25, We can confirm that  $\Delta x_l = \Delta u_l^\theta, \Delta x_m = \Delta u_m^\theta, \Delta x_r = \Delta u_r^\theta$ . According to Eq. 27, the final  $d'_1/d'_2$  can be estimated:

$$\frac{d'_1}{d'_2} = \frac{\frac{\Delta u_l}{\Delta u_m} \cdot \beta_1 - 1}{\frac{\Delta u_l}{\Delta u_r} \cdot \beta_2 - 1} = \frac{\left(\frac{d_1}{z_a} + 1\right) \cdot \beta_1 - 1}{\left(\frac{d_2}{z_a} + 1\right) \cdot \beta_2 - 1}, \quad (40)$$

where  $\beta_1$  and  $\beta_2$  can be represented as:

$$\begin{aligned} \beta_1 &= \frac{(z_b - u_{m1} \sin \theta)(z_b - u_{m2} \sin \theta)}{(z_b - u_{l1} \sin \theta)(z_b - u_{l2} \sin \theta)}, \\ \beta_2 &= \frac{(z_b - u_{r1} \sin \theta)(z_b - u_{r2} \sin \theta)}{(z_b - u_{l1} \sin \theta)(z_b - u_{l2} \sin \theta)}, \end{aligned} \quad (41)$$

where  $u_{l2} = u_{l1} + \Delta u_l, u_{m2} = u_{m1} + \Delta u_m, u_{r2} = u_{r1} + \Delta u_r$ . Observing Eq. 40, we can see if  $\beta_1 < 1, \beta_2 > 1$  or  $\beta_1 > 1, \beta_2 < 1$ , there will be  $d'_1/d'_2 \neq d_1/d_2$ .

Now, we discuss the sufficient condition of  $\beta_1 < 1, \beta_2 > 1$

1. When  $u_{m1} > u_{l1}, u_{m2} > u_{l2}, u_{r1} < u_{l1}, u_{r2} < u_{l2}$ , the  $\beta_1 < 1, \beta_2 > 1$  can be established. Similar to Eq. 21, the relationship of variables can be achieved:

$$\begin{aligned} \frac{f_a x_{l1}}{z_a} &= u_{l1}, \frac{f_a x_{l2}}{z_a} = u_{l2}, \\ \frac{f_a x_{m1}}{z_a + d_1} &= u_{m1}, \frac{f_a x_{m2}}{z_a + d_1} = u_{m2}, \\ \frac{f_a x_{r1}}{z_a + d_2} &= u_{r1}, \frac{f_a x_{r2}}{z_a + d_2} = u_{r2}, \end{aligned} \quad (42)$$

From Eq. 42 and  $u_{m1} > u_{l1}, u_{m2} > u_{l2}$ , we can obtain:

$$\begin{aligned} x_{m1} &> x_{l1} \cdot \frac{z_a + d_1}{z_a}, \\ x_{m1} + \Delta x &> (x_{l1} + \Delta x) \cdot \frac{z_a + d_1}{z_a}, \\ x_{r1} &< x_{l1} \cdot \frac{z_a + d_2}{z_a}, \\ x_{r1} + \Delta x &< (x_{l1} + \Delta x) \cdot \frac{z_a + d_2}{z_a}, \end{aligned} \quad (43)$$

where  $x_{l1}, x_{l2}, x_{m1}, x_{m2}, x_{r1}, x_{r2}$  are corresponding coordinates of  $N_l(t + \Delta t), N_l(t), N_m(t + \Delta t), N_m(t), N_r(t + \Delta t), N_r(t)$  in the dimension of  $\mathbf{x}$  in recording image space. In facial regions, we can easily find corresponding points  $N_l(t), N_m(t)$ , which satisfy that  $d_1 \ll z_a$  (i.e.,  $d_1 = 0$ ) and  $x_{m1} > x_{l1}$ . In this pattern, Eq. 43 can be established. To establish Eq. 44, we only need to find point  $N_r(t)$ , which satisfies that  $x_{r1} < x_{l1}$ . According to the derivation above, we can see that there exists cases that  $d'_1/d'_2 < d_1/d_2$ . And there are also many cases that satisfy  $d'_1/d'_2 > d_1/d_2$ , which we do not elaborate here. When faces move, the absolute coordinates  $x_{l1}, x_{l2}, x_{m1}, x_{m2}, x_{r1}, x_{r2}$  vary, as well as  $\beta_1, \beta_2$ , leading to the variation of estimated relative depth of three facial points at different moments, which will not occur in the *real* scene. That's to say, if the realistic image plane and recording image plane are not parallel, we can seek cases to detect abnormal relative depth with the help of abnormal facial motion.

### A.2.3 Discussion

One of basis of the elaboration above is that the structure of face is similar to that of the hill, which is complex, dense

Prot.	Method	APCER(%)	BPCER(%)	ACER(%)
1	FAS-BAS [20]	–	–	10.0
	<b>OURs</b>	2.9	12.5	<b>7.7</b>
2	FAS-BAS [20]	–	–	14.1
	<b>OURs</b>	13.6	13.3	<b>13.5</b>
3	FAS-BAS [20]	–	–	<b>13.8±5.7</b>
	<b>OURs</b>	18.1±8.7	10.0±6.3	14.0±4.5
4	FAS-BAS [20]	–	–	10.0±8.8
	<b>OURs</b>	2.5±3.8	12.5±11.1	<b>7.5±5.5</b>

Table 6. The results of cross testing from SiW [20] to OULU-NPU [7].

and undulate. This is interesting and worth being exploited in face anti-spoofing.

Even though we only use some special examples to demonstrate our viewpoints and assumption, they can still prove the reasonability of utilizing facial motion to estimate the relative facial depth in face anti-spoofing task. In this way, the learned model can seek the abnormal relative depth and motion in the facial regions. And our extensive experiment demonstrates our assumption and indicates that temporal depth method indeed improves the performance of face anti-spoofing.

## B. Extra Experiments

### B.1. Comparison

In Tab. 6, we show results of cross testing on models trained on SiW [20] and tested on OULU-NPU [7]. We can see that our method outperforms FAS-BAS [20] on three protocols. Our method even decrease 2.3%, 0.6% and 2.4% of ACER on protocol 1, protocol 2 and protocol 4, respectively. In protocol 3, ACER of our method is 14.0±4.9%, which is higher than that of FAS-BAS [20] for 0.2%, which shows that our method is sensitive to the variation of input cameras. This indicates the future research direction.

### B.2. Feature Distribution

As shown in Fig. B, we present distribution of different kinds of features in our method. The left subfigure is about the feature of single-frame part. The middle subfigure is about the feature of OFF block. The right subfigure is about the feature of ConvGRU. These features are all extracted from our model 1. We can see that the features are getting more discriminative from left to right. The intra class distances are getting nearer from left to right. And the inter class distances are getting further from left to right. This indicates that our features of motion are discriminative in face anti-spoofing.

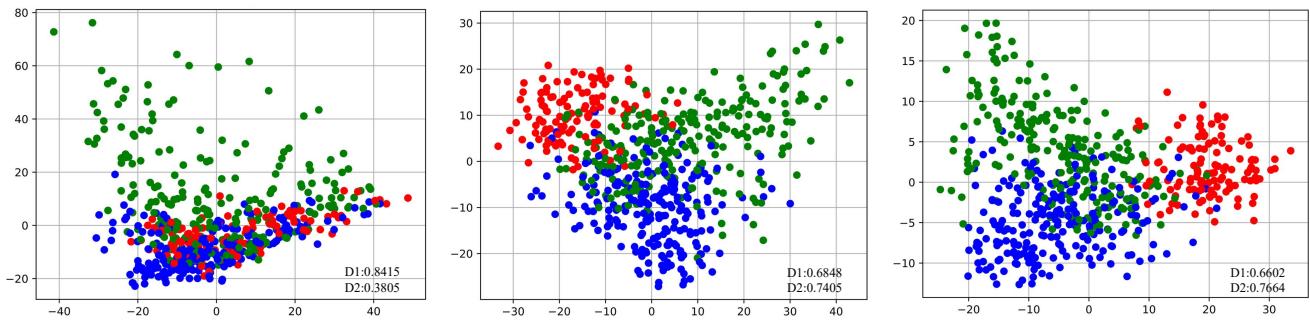


Figure 9. Distribution of features in different modules in our method. We split all samples into three classes. Red, blue, and green points denote samples of real, print attack, and replay attack, respectively. D1 is intra class distance where smaller value indicates better performance, and D2 is inter class distance where larger value indicates better performance. All distances are calculated after normalizing the features.