Protecting Real-time Video Chat against Fake Facial Videos Generated by Face Reenactment

Jiacheng Shang and Jie Wu
Center for Networked Computing, Temple University, Philadelphia, PA 19121

***Abstract*—With the rapid popularity of cameras on various devices, video chat has become one of the major ways for communication, such as online meetings. However, the recent progress of face reenactment techniques enables attackers to generate fake facial videos and use others' identities. To protect video chats against fake facial videos, we propose a new defense system to significantly raise the bar for face reenactment-assisted attacks. Compared with existing works, our system has three major strengths. First, our system does not require extra hardware or intense computational resources. Second, it follows the normal video chat process and does not significantly degrade the user experience. Third, our system does not need to collect training data from attackers and new users, which means it can be quickly launched on new devices. We developed a prototype and conducted comprehensive evaluations. Experimental results show that our system can provide an average true acceptance rate of at least 92.5% for legitimate users and reject the attacker with mean accuracy of at least 94.4% for a single detection.**

***Index Terms*—Face forgery, face liveness detection, real-time video chat.**

I. Introduction

In the past a few years, thanks to fast internet speeds and the powerful processing capacity of personal electronic devices, video chat has become a major form of communication. Compared with text-based or audio-based communication, video chat enables users to observe the real emotions and activities of each other without physically being together, which makes the information delivered more accurate and the relationship establishment more efficient. Therefore, many video chat software (e.g. Skype [1] and WebEx [2]) are released for various applications, such as conference meeting, interviewing, and making friends. Based on a report from Statista, the estimated number of Skype users is expected to be 1.67 billion in 2020 [3].

There are two major channels in real-time video chat: image and audio. By default, both channels are regarded as real information since they are generated in real-time, which is why video chat is used as an alternative way to validate the identity of a user in practice. However, since a malicious user can easily get the victim's videos and voice from social networks, both channels can be well counterfeited with the development of AI-assisted techniques. For example, the recorded voice of the victim can be replayed to pass through current voice-based authentication systems. Similarly, recent research in face reenactment shows that the facial expressions on one face can be transferred to any other face in real-time. These facts enable the malicious user to easily use the victim's identity,

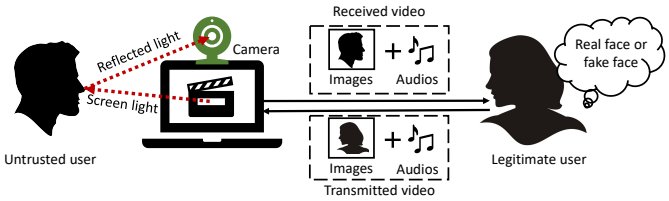


Fig. 1. Face forgery detection in real-time video chat.

which poses a serious threat to legitimate users. Even if the voice replay attack can be efficiently countered by using voice liveness detection techniques [4]–[7], attackers can still fool legitimate users by generating fake image channels.

To defend against fake face videos, various face liveness detection systems are designed using either artifact detection-based methods [8]–[12] or challenge-response-based methods [13], [14]. The basic assumption of the artifact detection-based method is that fake facial images must have imperfect artifact detections. By extracting proper features, the fake facial images can be detected using various classification models. However, in order to gain enough knowledge for building a robust classifier, artifact detection-based methods have to collect fake videos in advance, which usually involves significant training. Moreover, artifact detection usually requires lots of computation resources to achieve better feature extraction and classification, which is not available on resource-limited devices. Challenge-response-based methods are based on the nature of human activities. For example, FaceLive can detect the media-based facial forger by correlating the head movement measured by motion sensors and head pose change recorded in videos [13]. However, the face reenactment attacker can still easily break FaceLive by faking the sensor data since it can have enough knowledge of the target video. Moreover, since the detection is done on the attacker side, the attacker can even send the legitimate user a wrong detection result. Recently, Tang et al. [14] proposed a new liveness detection method by randomly flashing pre-designed pictures (e.g. white and black scenes) on a screen and analyzing the face-reflected light. Nevertheless, their work also relies on a neural network for accurate classification. Moreover, the flashing pictures replace the original video frames, which will degrade the user experience between two legitimate users.

Considering the limitations of existing solutions, we propose a defense system for real-time video chat against fake facial

Fig. 2. An example of face reenactment techniques [16].

videos generated by face reenactment techniques. As shown in Fig. 1, our system requires no extra sensors except the screen and camera that are available on all videotelephony devices. Specifically, the screen is used to emit light signals, and the camera works as a sensor to measure the *relative luminance* (simplified as *luminance* [15] in this paper) of the lights that are reflected from the untrusted user's face. The key insight behind our system is that the luminance of the face-reflected light is proportional to that of the screen light for a legitimate user. Since the face reenactment attacker cannot generate the real-time face reflection in a photo-realistic fashion, the legitimate user can detect the face forgery by: 1) introducing luminance changes in the transmitted video by changing the area of light metering; 2) measuring the correlation between the luminance changes of the screen light and face-reflected light.

To achieve our goal, we solve three major challenges in the design of our system. The first challenge is to robustly extract the luminance information of face reflection from the videos. To address this issue, we leverage the facial landmark detection algorithm to locate the lower part of the nasal bridge as the area of interest and calculate the luminance information using only the color information within this area. Second, the luminance signals are noisy and cannot be directly used for correlation measurement. To solve this problem, we remove the noise components using signal processing techniques and extract the significant light change from filtered signals. The last challenge is to extract useful features from the filtered signal and build a classifier for robust and accurate detection. In our system, we extract four features that describe the luminance change behavior and trend from the filtered signal. A local outlier factor-based classifier is trained on selected features for the final decision.

Compared with existing works, our system has three major strengths: 1) *low-cost*: our system does not require extra hardware or intense computation resources; 2) *good user experience*: since the luminance change in the transmitted video is made by controlling the exposure level, both users can still see each other's faces with only limited loss of video information; and 3) *zero training effort*: our system does not need to collect training data from either a new user or attackers, which means our system can be quickly launched on new devices. We summarize our contributions as follows:

- This is the first work where the luminance of face-reflected light is used to defend against face reenactment attackers.
- We propose robust solutions for extracting luminance signals from videos and finding significant luminance changes from noisy luminance signals.
- We extract four strong features from the filtered signals to describe the luminance change behavior and trend. Moreover, we propose a local outlier factor-based classifier to detect fake faces in videos without collecting training data from either a new user or attackers.
- We develop a prototype and conduct comprehensive evaluations. Experimental results show that our system can provide an average true acceptance rate of at least 92.5% for legitimate users and reject face reenactment attackers with mean accuracy of at least 94.4% for each detection.

II. Preliminary

A. Face Forgery using Face Reenactment

Deepfake (a portmanteau of "deep learning" and "fake") is a type of techniques for human image synthesis based on artificial intelligence. It is used to combine and superimpose existing images and videos onto source images or videos using machine learning techniques . Because of these capabilities, Deepfake techniques have been used to create fake celebrity pornographic videos, fake news, and malicious hoaxes. Face reenactment is an example of Deepfake techniques. The goal of face reenactment techniques is to animate the facial expressions of the target video by a source actor and re-render the manipulated output video in a photo-realistic fashion. Fig. 2 shows an example of the face reenactment technique reported in [16]. We can see that the facial expression in the source video is transferred to the person in the target video with high quality. Compared with other real-time face forgery techniques (e.g. face swapping), face reenactment creates fewer artifacts while achieving high frame rates (up to 47.5 Hz in [17]). For a legitimate user in video chat scenarios, it is hard to detect face reenactment attacks with high accuracy. Although face reenactment techniques have made great success on face forgery, their nature also gives us the insight to defend against them. Since face reenactment techniques only focus on transferring the facial expression, the luminance change of the output video is the same as the target video, which means the attacker cannot have the correlated luminance change of face-reflected light. Even if the face reenactment attacker can use the source actor to observe the luminance change and generate the change in the output video, the extra computational overhead will largely reduce the frame rate make real-time attacks unfeasible.

B. Light Metering of Digital Cameras

To achieve consistent and accurate exposures in the recorded videos, the light meter is essential for current digital cameras. In general, the camera controls the shutter speed and aperture by predicting how much light is actually hitting the subject. Current cameras provide users with various ways to meter light. Among them, spot and multi-zone metering modes are most used and widely available. In multi-zone metering, the camera measures the light intensity at multiple points in the

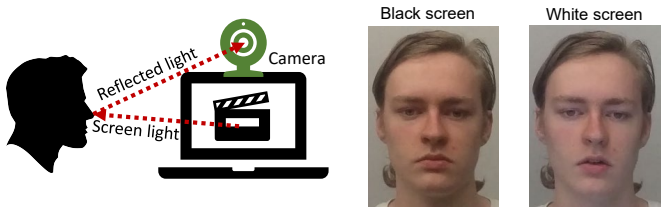


Fig. 3. Luminance of face-reflected light when screen light changes.

scene and then combines the results to find the setting for the ideal exposure. Therefore, multi-zone metering can produce balanced exposure for most scenes and is used as the default mode for most cameras. Alternatively, with spot metering, the camera will measure only a very small area of the scene. By default, this small area is at the center of the scene, but the user or application can easily select a different off-centre spot. If the spot is moving from a relatively low-luminance area to a high-luminance area, the camera will let less light in, which leads to a diminished brightness in the darker area. Similarly, if the spot is moving from a relatively high-luminance area to a low-luminance area, the luminance of the scene rises. Hence, by moving the metering spot between high-luminance and low-luminance areas, the legitimate user can easily control the overall luminance of its video. Since the exposure only changes the brightness of each pixel, this method can reserve partial information (e.g. the face of the legitimate user) in the scene, which ensures a certain level of user experience.

C. Face Reflection of Screen Light

When the untrusted user watches the legitimate user's facial video, the camera can capture the screen light that is reflected by the face of the untrusted user. Here, we model the face reflection of screen light based on the Von Kries coefficient law [18]. For a single type camera, a diagonal model can be described as:

$$I_c(x) = E_c(x) \quad R_c(x), c \; 2 \; fR; G; Bg, \tag{1}$$

where x is a pixel on the face, c is the light with different colors (red, green, and blue), I_c is the luminance of corresponding color, E_c is the illuminant spectral power distribution of the screen light on x, and R_c is the reflectance of pixel x. Therefore, if we focus on a pixel with the same reflectance and change the light luminance, then we have:

$$\frac{I_c(x)^0}{I_c(x)} = \frac{E_c(x)^0}{E_c(x)}, c \; 2 \; fR; G; Bg, \tag{2}$$

where $I_c(x)^0$ and $E_c(x)^0$ are the luminance and illuminant spectral power distribution after the change of screen light. From this equation, we can observe that the luminance of the face reflection is proportional to that of the screen light, which serves as the basic insight of our system.

D. Feasibility Study

To achieve our goal, we first show that the luminance of the face-reflected light is highly correlated to that of the screen light. Specifically, we made a video that flashes between white and black with a frequency of 0.2 Hz and displayed this video on a Dell 27-inch Light-emitting diode (LED) Monitor. We asked a volunteer to sit in front of the monitor while using the front camera of an iPhone 7 to record his facial video. During the recording, the volunteer can freely move the head as long as the whole face can be captured by the camera. Fig. 3 shows the faces when the screen shows black and white colors, respectively. We can clearly observe that the luminance of the face-reflected light increases when the color changes from black to white. As a reference, the luminance value of the nasal bridge increases from around 105 to around 132. Moreover, this fact is true for all types of screens including LED, liquid crystal display (LCD), and organic LED (OLED) since they all reduce the amount of emitted light when displaying darker scenes. This simple case implies that the luminance of the face-reflected light does change proportionally to that of the screen light, which shows the possibility of detecting fake faces using the correlation between two luminance signals.

E. Challenges

Although we can observe the corresponding luminance change of the face-reflected light while the screen's color changes between black and white, it is still challenging to apply this insight to real video chat scenarios for fake forgery detection. First, the face of the untrusted user will likely be moving in the scene and can be partially occluded by other objects (e.g. hair and sunglasses), which introduces extra noise to the luminance signals of the face-reflected light. To address this issue, our system only extracts the luminance information from the lower part of the nasal bridge since this area can be robustly located using the facial landmark detection algorithm and is the least likely part to be obfuscated.

The second challenge is to obtain the luminance change information from the noisy luminance measurements. The raw luminance signals contain various types of noise. For example, dynamic scenes in the video will introduce high-frequency noise to the raw luminance signal of the screen light. Additionally, the luminance change is weaker in practice than in the ideal case in a feasibility study. To remove the noise and robustly locate each luminance change, we designed a series of filters and apply them on the raw luminance signals in order.

The last challenge is to extract useful features from the filtered signal and build a classifier for robust and accurate face forgery detection. To solve this problem, we select four features that describe when and how the luminance signal significantly changes. To reduce the training cost while still ensuring good performance, we build a strong classifier without collecting training data from the attacker and new users using the local outlier factor model.

III. System Design

A. Adversary Model

In our adversary model, the attacker aims to impersonate others using face reenactment while video chatting with vic-

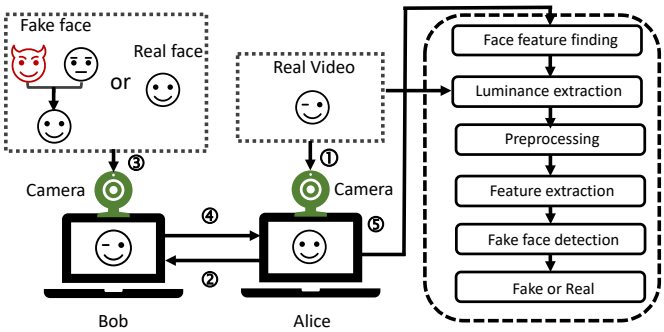


Fig. 4. System architecture.

tims. The capability of the face reenactment attacker is limited in the sense that: 1) the attacker has already or is able to set up a video chat connection with the victim; 2) the attacker can generate fake facial videos with high quality in real-time using any face reenactment technique; and 3) the attacker can redirect the input stream of the current video chat software (e.g. Skype) from the camera to the fake facial videos using a virtual web camera. When these tasks are performed, we suppose the victim cannot visibly identify the fake facial video as a forgery. Note that the adversary model we considered is much stronger than traditional models where attackers replay the fake facial videos using another screen because the facial videos are directly fed to video software without any loss and interference.

The objective of our system is to significantly raise the bar for such face reenactment attacks. To break our defense system, the attacker needs to reconstruct the face-reflected light on the fake face with high quality based on the relative locations of the head, camera, and the screen in real-time. For this, the attacker has to: 1) introduce an extra image processing layer for each frame to reconstruct the face-reflected light; and 2) have enough computation resources to ensure the real-time attack. Therefore, our system is difficult to attack.

B. System Architecture

The key idea underlying our system is to measure the luminance correlation between the screen light and the face-reflected light. When a legitimate user is using videotelephony with an untrusted user, the camera can capture the screen light that is reflected by the untrusted user's face. By comparing the luminance of the screen light and the face-reflected light, we can determine if the face is from a real person or generated by face reenactment techniques. There are two major phases for using our system: a training phase and a detection phase. In the training phase, our system will learn the decision strategy based on the knowledge in the legitimate users' data. After that, our system is ready to be used for detection. Our detection methods can be triggered multiple times during the real-time video chat. If the untrusted user is detected as an attacker, an alert will be sent to the legitimate user to avoid further loss.

Fig. 4 shows the detailed process of our system in five steps. A legitimate user Alice wants to validate whether the facial video sent from the untrusted user Bob is real or fake. To do this, Alice records her own facial videos using a camera in step 1 and sends the real-time facial video via the internet to Bob in step 2. On Bob's side, his device receives Alice's video and displays it on his screen, which means that the luminance of the screen light largely depends on the content in Alice's video in real-time. At the same time, as illustrated in step 3, Bob is recording his facial video whose luminance change should be influenced by not only the ambient light in Bob's environment but also his screen light. By receiving Bob's video in step 4, Alice can get the luminance information of both Bob's screen light and Bob's face-reflected light. In our system running on Alice's device, we first extract the luminance information in both videos and apply filters to the raw signals to extract only significant light changes.

IV. Luminance Extraction

The goal of our system is to detect the liveness of the face in the video by measuring the correlation between luminance signals of the screen light and face-reflected light. Therefore, we first need to robustly extract these two types of luminance information from the two videos. Since we are only interested in the overall luminance of the screen light, we first compress each frame of the transmitted video into a single pixel, and use the luminance value of the compressed pixel to represent the overall luminance of the transmitted video. The luminance of a pixel is defined as:

$$C = 0{:}2126R + 0{:}7152G + 0{:}722B, \tag{3}$$

where C is the luminance value calculated using linear Red Green Blue (RGB) values. The coefficient of each color is assigned based on the human visual perception of brightness.

However, not all facial parts can be used to measure luminance changes. For example, the user may blink the eyes or talk during the recording. Such activities will introduce a lot of variances between neighboring frames. Also, users may wear glasses that reflect lights from other sources, which will introduce much noise to the luminance measurements. Based on our preliminary study, we find that the lower part of the nasal bridge has the most stable images and is hard to be occluded in most cases. Moreover, the luminance changes caused by different screen lights at this area are easy to detect. Therefore, we extract only the lower part of the nasal bridge from each frame of the video for luminance measurement.

When a legitimate user receives the video from the untrusted user, our system extracts frames with a sample rate of 10 Hz. For each frame, we detect the location of the lower part of the bridge by using a facial recognition API for Python [19]. As shown in Fig. 5, the facial recognition API can report four locations on the nasal bridge and five locations on the nasal tip. Since the sampled frames can vary in size depending on camera hardware, we use the locations of the nasal bridge and nasal tip to extract the interested area. As shown in Fig. 5, given the coordinates of the nasal bridge $(a_1; b_1)$ and nasal tip $(a_2; b_2)$, the side length of the interested area is $l = jb_1 \quad b_2j$.

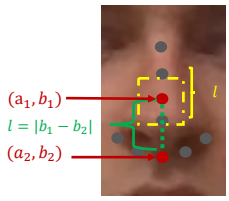


Fig. 5. Facial feature localization and interested area extraction.

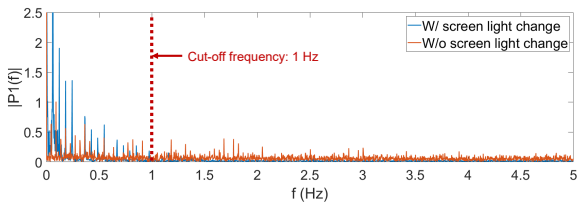


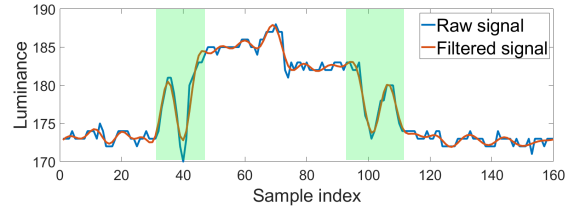
Fig. 6. The spectrum of luminance signals w/ and w/o screen light change.

A square whose center is $(a_1; b_1)$ is extracted from the frame to calculate the luminance. We use the same methods to get the luminance information from the area of interest. Fig. 7(a) shows the luminance signal generated from the lower part of the nasal bridge, and we can see significant rising edge and falling edge appear when the luminance of the screen light significantly changes (green areas).

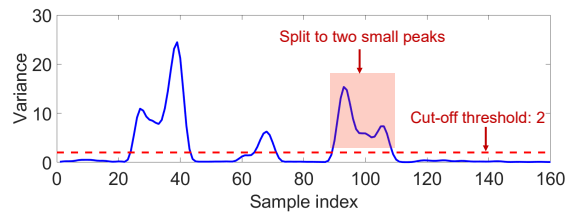
V. Preprocessing

As shown in Fig. 7(a), the raw luminance signals contain various noise. For the transmitted video, the noise is mainly from the object movement in the scene. For the face reflection in the received video, the noise can be introduced by external light sources. Moreover, the inaccurate face localization can lead to jittering in the interested area, which further influences the luminance extraction of the face reflection. Hence, the raw luminance signals need to be filtered before being used for feature extraction.

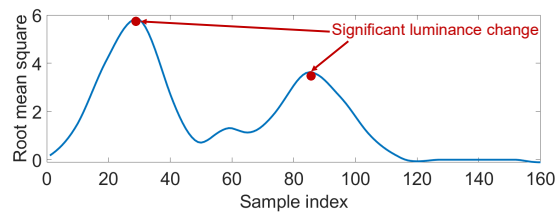
Fig. 6 illustrates the spectrum of the luminance signals of the face-reflected light. It is clear that high-frequency noise exists across whole frequency bands, while screen light changes influence the luminance of the face-reflected light with low frequency under 1 Hz. Based on this observation, we first use a low-pass filter with a cut-off frequency of 1 Hz. As shown in Fig. 7(a), most high-frequency components are removed while the overall trend is reserved. In our system, we only consider significant luminance changes in both luminance signals for two reasons. First, only significant luminance changes in the transmitted video can generate luminance changes in the interested area of the received video. Second, the significant luminance changes in the received video are robust to noise and easier to detect. However, it is hard to locate each significant luminance change in the filtered signal since low-frequency noise still exists. To locate all significant light change in the filtered signal, we leverage a moving window with length of 10 samples and calculate the short-time variance within each window. The basic insight is that the low-frequency noise within a window only generates a low variance. Moreover, the variance value in the moving window can reach a local maxima in two cases: 1) the luminance rapidly increases to a high value; and 2) the luminance drops



(a) The raw and filtered luminance signal



(b) Variance signal



(c) Smoothed variance signal

Fig. 7. Preprocessing of luminance signals.

to a much lower value. Therefore, each significant luminance change can be located by finding the local maxima in the variance signal.

Nevertheless, the variance signal cannot be directly used for locating significant light change. As shown in Fig. 7(b), low-frequency noise can either generate small spikes in the variance signal or split a significant luminance change into multiple lower, neighboring peaks. To remove small spikes, we apply a threshold filter on the variance signal with a cut-off threshold of 2. To group neighboring lower peaks into one significant luminance change, we further smooth the variance signal by applying a moving window with a length of 30 samples and calculating the root-mean-square value in each window. Then, we leverage a Savitzky-Golay filter [20] with a window length of 31 samples using polynomial fitting and a moving average filter with a window length of 10 samples to further smooth the signal, and the result is shown in Fig. 7(c). Finally, the traditional peak finding algorithm is applied on each smoothed variance signal respectively. Since the luminance variation range of the screen light is much larger than that of the face-reflected light, the minimal prominence of the peaks is set to 10 and 0.5 for the screen light and face-reflected light, respectively.

VI. Feature Extraction

In order to detect a fake face in the video, we need to extract proper features that can describe the correlation between two relative luminance signals. In our system, we consider both the similarities of luminance change behaviors and the correlation of luminance change trends. The luminance change behavior is a vector where the value of each element is the time