Video Source Identification in Lossy Wireless Networks

Shaxun Chen[†], Amit Pande[†], Kai Zeng[‡], Prasant Mohapatra[†]
[†]Department of Computer Science, University of California, Davis, CA 95616
[‡]Department of Computer and Information Science, University of Michigan, Dearborn, MI 48128
sxch@ucdavis.edu, pande@ucdavis.edu, kzeng@umich.edu, pmohapatra@ucdavis.edu

***Abstract*— Video source identification is very important in validating video evidence, tracking down video piracy crimes and regulating individual video sources. With the prevalence of wireless communication, wireless video cameras continue to replace their wired counterparts in security / surveillance systems and tactical networks. However, wirelessly streamed videos usually suffer from blocking and blurring due to inevitable packet loss in wireless transmissions. The existing source identification methods experience significant performance degradation or even fail to work when identifying videos with blocking and blurring. In this paper, we propose a method which is effective and efficient in identifying such wirelessly streamed videos. In addition, we also propose to incorporate wireless channel signatures and selective frame processing into source identification, which significantly improve the identification speed.**

I. INTRODUCTION

Source identification is a major component of video forensics. When presenting a video clip as evidence in a court of law, identifying the source (acquisition device) of the video is as important as the video itself. For instance, if a surveillance camera captured the scene of a suspect's alibi, it is necessary to prove that the video was truly recorded by the claimed camera. Otherwise it can be forged or derived from an untrustworthy source, which makes the evidence invalid. In the movie industry, significant revenue loss is caused every year by surreptitious recording in movie theaters and the subsequent illegal distribution. Video source identification is employed to track down such piracy crimes [1] [5]. While Internet enables video sharing at a large scale, it also opens doors for propagation of illegal or inappropriate materials, such as the video including child porn or racial hatred. Video source identification can be used to regulate the individual video sources [2].

Easier access to high-quality digital camcorders and sophisticated video editing tools urges the improvement of video source identification techniques. However, the research on this topic is still in its early stages. The most straight forward way to identify the video source is embedding digital watermarks into video when recording, but this method is computationally expensive and requires modification to recording devices, thus cannot be applied to most off-the-shelf cameras or camcorders. Some researchers [4] proposed to utilize defective pixels (hot or dead pixels) on the camera sensor to distinguish different devices. However, there are many cameras or camcorders do not have defective pixels. The most reliable method reported so far for source identification is based on the sensor pattern noise, which mainly results from the non-uniformity of each sensor pixel's sensitivity to light, and can be treated as the inherent fingerprint of a video capture device [3].

On the other hand, wireless communication has seen a tremendous growth in the recent years. Wireless cameras have also become increasingly popular. In the security camera market, wireless video cameras continue to replace their wired counterparts due to the ease of deployment. In tactical networks, wireless cameras are widely used as video sensors. Such cameras usually do not have local storage; video is captured and wirelessly streamed to a sink. Because of the inevitable packet loss and unpredictable transmission delay in wireless streaming, blocking and blurring frequently appear in the received frames. For such videos, experiment shows that the existing source identification methods suffer from significant performance deterioration or even fail to work. Blocking and blurring caused by the lossy wireless channel severely tamper with the sensor fingerprint recognition.

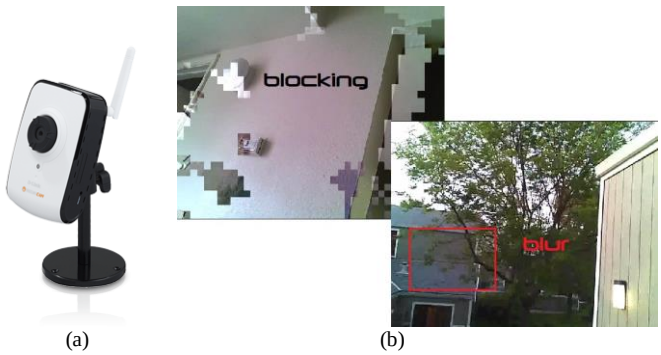
In this paper, we propose a systematic methodology for video source identification which can achieve excellent performance not only for the conventional videos but also for the wirelessly streamed videos with blocking and blurring. In addition, we propose to incorporate both the wireless signature and sensor signature for video source identification. Together with other optimizations, we are able to identify the video source in a near-real-time fashion.

The remainder of this paper is organized as follows. Section II introduces our method for video source identification which tolerates video blocking and blurring. Section III evaluates our work and Section IV concludes the paper.

II. SOURCE IDENTIFICATION OF WIRELESS VIDEOS

A. Background

Wireless video cameras (shown in Figure 1a) are mostly used for security / surveillance purpose. The majority of commercially available products transmit via 802.11 channels, while the video sensors in tactical networks may use dedicated links. Wireless cameras do not have local storage; they capture the scene and stream to the sink in real time. The pixels of a video frame from a wireless camera can be presented as:



(a) (b)

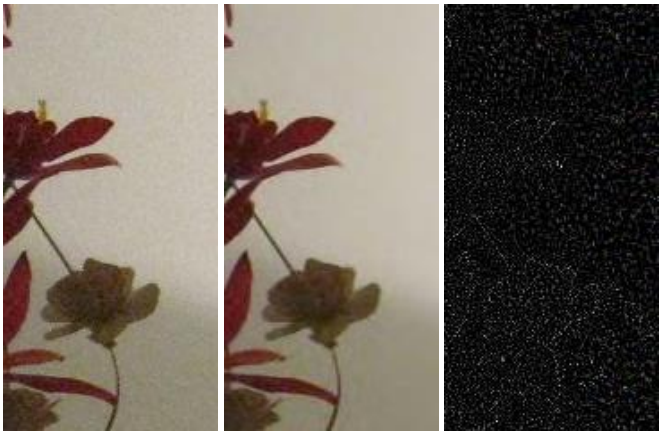
Figure 1. Wireless camera and frames with blocking and blurring

$$z_{ij} = L(y_{ij}) \quad \text{where} \quad y_{ij} = P(x_{ij}) \tag{1}$$

x_{ij} is the incoming light captured by the camera sensor at the pixel (i, j), where i = 1, 2, …, m and j = 1, 2, …, n. $m \times n$ is the sensor resolution. Here we assume the video resolution is equal to the sensor resolution for convenience. P indicates the sensor distortion and onboard processing. L is the distortion caused by wireless transmission. Because of the real-time requirement of security systems, typically UDP is used by wireless cameras for the video streaming and thus no retransmission occurs above the MAC layer. As a result, packet loss is the major factor of L.

B. Sensor Pattern Noise Extraction

Sensor pattern noise is mainly caused by the non-uniformity of each sensor pixel's sensitivity to light. For every frame of the video, various types of noises exist, such as white noise, shot noise and ISO noise. Fortunately, most of them are randomly distributed, which tend to cancel out if we extract from a large number of frames and add them together. However, sensor pattern noise is the same for different frames (taken by the same camera) and going to be strengthened after being added up. Therefore, we can simply extract all the noise as a whole from each frame; the sensor pattern noise is supposed to survive the averaging while other noises would not [6].



(a) F (b) F_{dN} (c) $(F - F_{dN}) \times 10$

Figure 2. Extracting noise from a frame

The basic idea for noise extraction is that for an image (a video frame), correlated signal is compressible and predictable but uncorrelated noise is not. Denote a frame extracted from the video as F. For each color channel of F, data is viewed as a locally stationary i.i.d. signal plus a zero mean and stationary white Gaussian noise; we calculate its fourth-level wavelet decomposition with the 8-tap Daubechies wavelets. For each level, the horizontal, vertical and diagonal subbands are noted as $h(i, j)$, $v(i, j)$ and $d(i, j)$, respectively. Then we calculate the local variance of each subband using MAP estimation, and obtain the denoised frame F_{dN} by using Wiener filter (due to the space limitation, please refer to [3] and [7] for details). Note N as the extracted noise from frame F:

$$N = F - F_{dN} \tag{2}$$

Figure 2 shows an example of an original frame, the denoised frame (using the above method), and the noise extracted from this frame. For visualization, the noise (Figure 2c) is up scaled 10 times. The noise extraction process will repeat for a sequence of frames from the same video. Based on our previous discussion, the sensor pattern noise will survive the averaging while other noises tend to cancel out. Therefore, the sensor pattern noise $\mathcal{N}$ can be expressed as:

$$\mathcal{N} = \frac{1}{k}\sum_{i}^{k} N_i \tag{3}$$

where N_i is the noise extracted from the i^{th} frame and k is the number of the frames processed. Typically, if the video quality is good, the sensor pattern noise can be well established when k is larger than 300 to 500. For source identification, we calculate the sensor pattern noise of the video to be identified (noted as $\mathcal{N}_v$), and compare it with the sensor pattern noise extracted from the camera (camera reference pattern for short, noted as $\mathcal{N}_c$) using the metric of correlation coefficient:

$$corr(\mathcal{N}_v, \mathcal{N}_c) = \frac{(\mathcal{N}_v - \overline{\mathcal{N}}_v)(\mathcal{N}_c - \overline{\mathcal{N}}_c)}{\left\|\mathcal{N}_v - \overline{\mathcal{N}}_v\right\|\left\|\mathcal{N}_c - \overline{\mathcal{N}}_c\right\|} \tag{4}$$

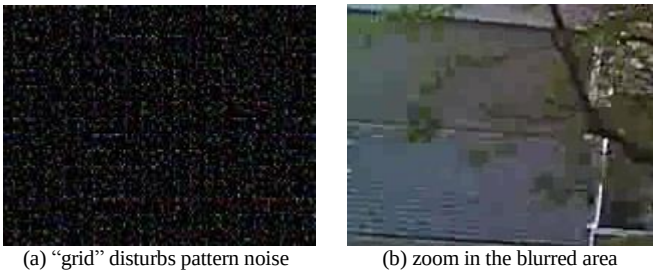
Calculating $\mathcal{N}_c$ is relatively easy because the source camera is typically accessible to law enforcers. We can use it to take a video with sufficient length and high quality, and derive the $\mathcal{N}_c$ accurately. However, as to the video to be identified, the length and quality are pre-given; we have to sit on what we have. Chen et al. reported that generally 40 seconds video ($\geq$450kbps) is good enough for a reliable identification, but 10 minutes of video is required if the video quality is low (150kbps) [1]. Our experiments further show that, the videos contaminated by blocking and blurring are worse: even with decent bit rate (about 500kbps), in some cases, the $\mathcal{N}_v$ simply cannot converge; in other cases, it may require more than 20 minutes of video to get a decent accuracy using the existing method. It means that the wireless video evidence shorter than this length cannon be identified, which is not acceptable.

C. Video Source Identification under Packet Loss

Video blocking affects the extraction of sensor pattern noise in two folds. First, within the blocks, the details as well as the pattern noise are lost. Typically when including more frames, the extracted sensor pattern noise will be strengthened. However, when adding a frame with blocking, within the blocking areas, the pattern noise would in fact be weakened (because in Equation 3, the denominator increases and numerator remains almost unchanged). Second, the borders of the blocking become a strong signal which will survive the extraction and averaging; eventually they form a "grid", which interferes with the real sensor pattern noise. Figure 3a shows a sensor pattern noise contaminated by such "grid" (*blockiness artifacts* [1] have already been removed). It is extracted from a wirelessly streamed video of 360 frames with frequent packet loss, and it is up scaled 50 times for visualization purpose.

Video blurring can result from multiple reasons, such as high compression ratio or fast motions. But in this work, we only consider the blur caused by packet loss. Video is encoded based on blocks. Instead of losing all the data of a block, blurring is the result of losing high frequency component, but this information loss is still block based. That is, one block may look more blurred than another. By zooming in the red area in Figure 1b, we can see that the blur caused by packet loss is

essentially smaller-size blocking (see Figure 3b). Therefore, we treat the blocking and blurring uniformly thereinafter, which makes our method more clear and time efficient.



(a) "grid" disturbs pattern noise (b) zoom in the blurred area

Figure 3. Blockish effects in video

The basic idea of our method is as follows. For the frames with blocking, we rule out the blocking areas but still use the rest of the frame for pattern noise extraction. Such frames are not completely discarded because it would waste useful information and make the identification slower.

First, we introduce our method for blocking detection. Because we do not have the access to the original video without packet loss, a non-reference method is required. The existing work, such as [8] and [9], are based on block boundary detection or Fourier transform. Since we have to perform the wavelet transform for noise extraction (see Section IIB), we propose a blocking detection method which is also wavelet based. In this way, we can largely reduce the computational overhead.

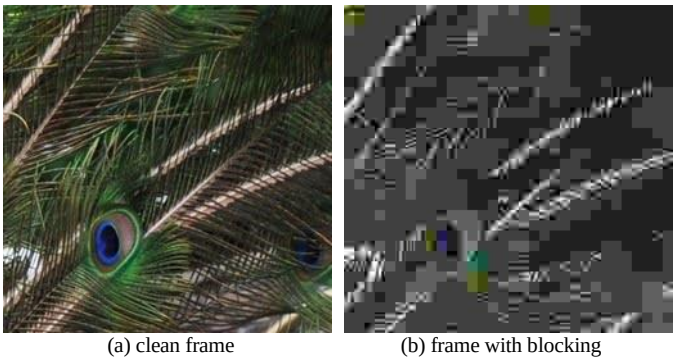
Taking a close look at the first-level wavelet transform result, we have two observations on frames with heavy blocking:

1) More elements in the diagonal subband $d(i, j)$ are zeros or close to zero compared with clean frames.

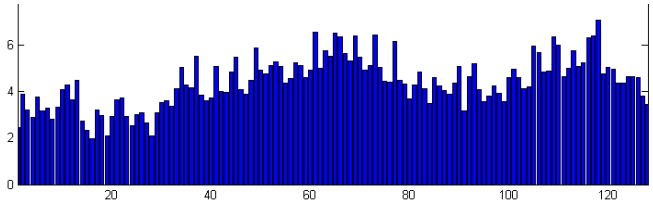
2) If we add the absolute values of $d(i, j)$ by rows (or by columns), the sum demonstrates periodic characteristics like the teeth of a saw.

$$S_i = \frac{1}{p}\sum_{j}^{p} |\, d(i, j)\,| \qquad 1 \le i \le q \tag{5}$$

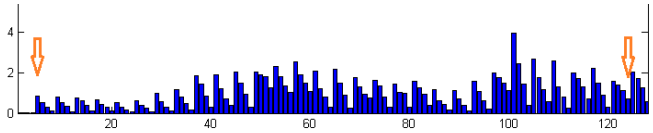
Equation 5 adds absolute values of $d(i, j)$ by rows and then takes an average. Here we assume the size the frame (or part of the frame involved in the blocking detection) is $2p \times 2q$, and still use the 8-tap Daubechies. Figure 4 shows an example. Figure 4a is a clean frame and 4b is the same scene but with blocking. The S_i values of Figure 4a and 4b are shown in Figure 4c and 4d, respectively. We can verify two observations mentioned above. In fact, they are not only true for this example, but generally applicable for the blocking areas of a video frame.



(a) clean frame (b) frame with blocking



(c) S_i of the frame in (a), i = 1, 2, …, 128



(d) S_i of the frame in (b), i = 1, 2, …, 128

Figure 4. Obervations of $d(i, j)$ under video blocking

Apparently, between two observations, the first is easier to exploit. But other than blocking, it is also true for big chunk of objects in plain color, such as a clean sky or white walls, which happen to be the best sources for extracting sensor pattern noise (they have fewer details, so sensor pattern noise is better preserved). Fortunately, observation 2 is unique to video blocking and is our better choice. In order to detect the periodic pattern in S_i efficiently, some tricks are needed. We maintain two cursors: one from the very left (i = 1), looking for the first local maximum, and another from the very right ($i = q$), seeking the first local minimum. Considering Figure 4d as an example, after this step, the left cursor should stop at bar 5 while the right one at bar 124 (marked by arrows in Figure 4d). Then, treating these two cursors as end points, we "fold" the x-axis in half and the values of overlapping bar pairs are added and then divided by 2. We get:

$$S_j' = \frac{1}{2}(S_{v+j-1} + S_{w-j+1}) \qquad 1 \le j \le \frac{w-v+2}{2} \tag{6}$$

Here v is the index where the left cursor stops and w for the right one. j is a positive integer. We then compare the variance of S_j' and S_i:

$$\beta = \frac{\mathrm{var}(S_j')}{\mathrm{var}(S_i) + C_0} \qquad 1 \le j \le \frac{w-v+2}{2}, 1 \le i \le q \tag{7}$$

C_0 is a small positive constant to make sure the denominator is not zero. If β is lower than a threshold, we assume the periodic pattern (observation 2) is found, and thus report video blocking. The rationale behind is that if S_i is periodic, and within each period it is monotonic, S_j' calculated by Equation 7 should be much more evenly distributed than S_i. Here β reflects the degree of video blocking (the smaller, the heavier). The performance and time costs of our blocking detection approach are given in Section III.

Now we briefly describe our source identification method for wireless videos as a whole. A frame is divided into 32×32 pixel squares, where each square is parsed by the blocking detection algorithm presented above (it is equivalent to dividing $d(i, j)$ into 16×16 squares and calculating S_i separately). If the result is positive (video blocking exists), this square is discarded; otherwise, it is adopted for pattern noise extraction (using the method in Section IIB). When accumulating the noise from multiple frames, k in Equation 3 may have different values for each square. We use this method to calculate the sensor pattern

noise for the video to be identified ($\mathcal{N}_v$). For the camera reference pattern ($\mathcal{N}_c$), usually blocking detection is not necessary because in this case video recording condition is under the control.

Next, Equation 4 is employed to judge whether $\mathcal{N}_v$ and $\mathcal{N}_c$ are from the same source or not. The correlation coefficient between $\mathcal{N}_v$ and $\mathcal{N}_c$ is typically from 0.1 to 0.7 when they have the same source. If the videos used to extract $\mathcal{N}_v$ and $\mathcal{N}_c$ are of the same bit rate, the correlation tends to be higher. Otherwise, it is slightly lower. When $\mathcal{N}_v$ and $\mathcal{N}_c$ are not from the same camera, $corr(\mathcal{N}_v, \mathcal{N}_c)$ is almost always lower than 0.01, where the gap is significant enough for us to perform identification (the threshold is noted as ψ). For videos with blocking and blurring, our method exhibits significant higher performance than the existing method (see Section IIIC).

D. Expediting the Identification Process

The existing forensics methods are mainly used for post-mortem analyses. Although we deal with blocking and blurring uniformly, and have developed a fast blocking detection technique, the overall identification time is non-trivial. Processing a 640×480 video frame (including blocking detection and noise extraction) still takes about 10 seconds on an average laptop (with an Intel Core 2 Duo CPU). Being set to the highest video quality (1~1.5 Mbps), we need approximately 200 frames to make a reliable decision. In this subsection, we introduce several measurements to further expedite our method.

1) Parallelization. Simply shifting to a more powerful computer will not help much because the original approach is single-threaded. We parallelized the computationally expensive operations, such as wavelet transform and local variance estimation. With this modification, our method shows good scalability and the speed boosts accordingly in multi-core computers. For example, the modified version is more than 8 times faster when migrating from a 2-core laptop to a 2-CPU 12-core workstation (Xeon X5650×2).

2) Selective frame processing. In a video clip, compared with P- and B-frames, an I-frame contains more fundamental information and details. By extracting noise exclusively from I-frames, we find that only 20~40 I-frames are able to achieve acceptable performance.

3) Combining wireless fingerprints. We propose to incorporate wireless channel signatures with the sensor fingerprint for source identification. Profiles of wireless characteristics are built for each legitimate camera base on their history information. The metrics include: packet loss ratio, jitter, average signal strength, signal strength variance, and the percentage of blocking frames. Using the channel profile, we need even less frames to make a reliable identification.

With these improvements, our method is able to perform the video source identification in a near-real-time fashion, and thus can be used to defend against the wireless camera spoofing attack. Due to the space limitation, the details are omitted.

III. EVALUATIONS

A. Experiment Settings

The wireless cameras we use to evaluate our source identification method are listed in Table I. We choose these models because they are of the most popular brands in the market and have very similar specifications, which puts higher requirement on source identification. The sink is wiredly connected to a Cisco Linksys WRT160N V2 wireless-N router, to which the wireless cameras send the video. The webcam of X301 streams its video using VLC through the laptop's 802.11- n wireless network card. MPEG4 videos are used unless otherwise specified.

TABLE I. CAPTURE DEVICES OF OUR EXPERIMENTS

Model	Amount	Sensor	Format	Net
Linksys WVC80N	4	640×480 CCD	MPEG4/ MJPEG	802.11n
D-Link 942L	1			
Axis M1011-W	1			
webcam of Lenovo X301	1			

B. Performance on Video Blocking Detection

The false alarm and mis-detection rate of our blocking detection method are shown in Figure 5. False alarm is defined as the video blocking reported by our approach while there is actually none; mis-detection refers to the cases where over 50 percent area of the square is occupied by blocking but our approach fails to detect. A total number of 6000 squares (32*32 pixel units in frames) are calculated (about 2300 of them are blockish), which are evenly collected from the cameras listed above. The x-axis shows the video bitrate. The threshold of β is set to 0.6. From the results we can see both the false alarm and mis-detection rates are very low, especially for high bit rate videos, which is sufficient for our succeeding processing.

Figure 6 shows the time cost of our blocking detection approach for ten 640×480 frames using an average laptop. The method provided in [9] is referred as the baseline. The third bar plots the time of our approach without including the time spent on wavelet transform. Since it has been done during the noise extraction, we do not need to calculate the wavelet transform again. By using our own blocking detection approach, we speed up about 15 times.

C. Performance on Source Identification

We first test the $corr(\mathcal{N}_v, \mathcal{N}_c)$ where $\mathcal{N}_v$ and $\mathcal{N}_c$ are from different cameras. Linksys WVC80N is used as the reference device to calculate $\mathcal{N}_c$ (via a 20-minutes-long video, at highest quality). 60 videos are shot by other 6 cameras for $\mathcal{N}_v$ calculation (10 videos each), all of which are 3 minutes long and in high quality (but with heavy blocking). Figure 7 plots the results, from which we can see the correlations are very low (the absolute values are mostly less than 0.003).

In Figure 8, we calculate the $corr(\mathcal{N}_v, \mathcal{N}_c)$ where $\mathcal{N}_v$ and $\mathcal{N}_c$ are of the same camera (also with blocking). The solid curve is an approximation of the bars; the same approximations for lower-bit-rate videos are plotted in the dashed curves. We can see that the correlations are high (greater than 0.2). The huge difference between the results exhibited in Figure 7 and Figure 8 offers us enough room to establish a threshold and differentiate the source with high confidence. We set the threshold ψ = 0.01. That is, if $corr(\mathcal{N}_v, \mathcal{N}_c)$ is larger than 0.01, we assume the video to be identified are from the same source as $\mathcal{N}_c$; otherwise it is not.

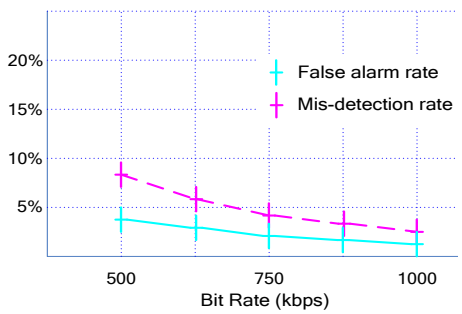


Figure 5. Accuracy of our blocking detection approach

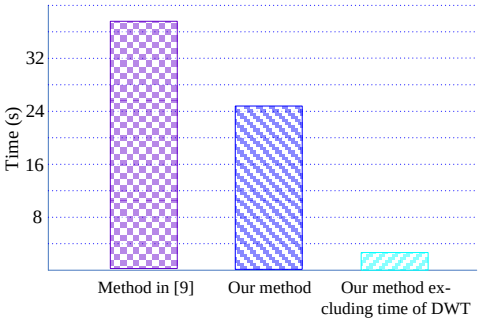


Figure 6. Time efficiency of our blocking detection approach

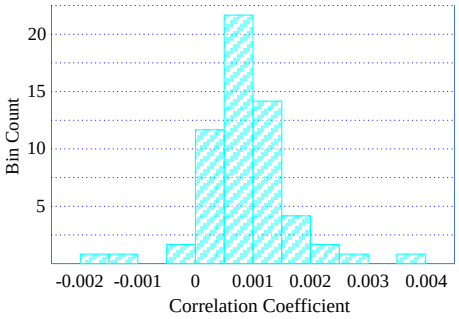


Figure 7. Correlation between blockish videos from different sources

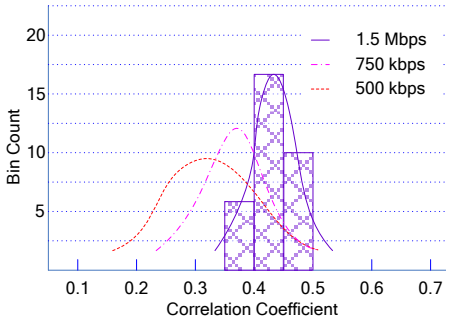
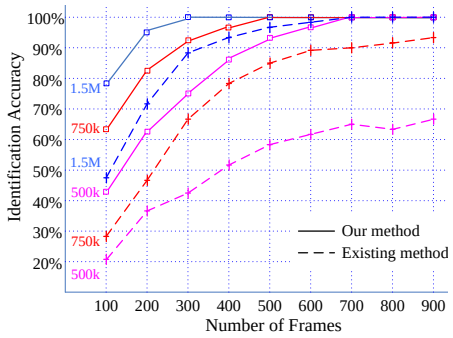
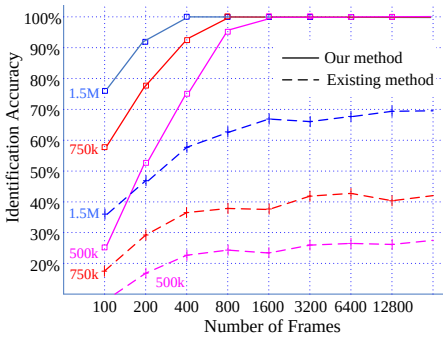


Figure 8. Correlation between blockish videos from the same source



(a) light blocking



(b) heavy blocking

Figure 9. Identification accuracy compared with the existing method

Now we compare the performance of our method with the existing source identification method. Figure 9a gives the comparison result when only light blocking occurs (less than 20% of the frames are contaminated by blocking and blurring). Three bit rates are tested respectively. For each rate, 140 videos are collected (20 from each camera). All videos are 640×480 and 10 fps. Accuracy is defined as the correct identifications divided by the total number. x-axis denotes the number of frames used for $\mathcal{N}_v$ extraction. The solid lines show the performance of our method while the dashed lines are from the existing method (provided in [1]). We can see that under the same bit rate, our method achieves the perfect accuracy with much less frames. For low bit rates, the gap is even larger. Short videos (with blocking) cannot be accurately identified by the existing method.

In Figure 9b, we show the performance comparison under the heavy blocking (about 40% of the frames are contaminated). For 6400 and 12800 frame cases, we test 70 videos; the other settings are the same as the test above. The result clearly shows, under heavy blocking, the existing method cannot achieve acceptable performance even if the video is long enough. We also conduct the experiments on detecting wireless camera spoofing attack using our method, but cannot present the results here due to the space limitation.

IV. Conclusion

In this paper, we introduced a systematic method for source identification of wirelessly streamed videos. Our method exhibits excellent performance even in the presence of video blocking and blurring. To achieve this goal, we developed a novel blocking detection approach. We also proposed to incorporate wireless channel signatures and selective frame processing to accelerate our method, which enables the near-real-time video source identification.

Real-world experiments are conducted to evaluate the effectiveness and efficiency of our method. The results show that it largely outperforms the existing method in both accuracy and time efficiency.

References

[1] M. Chen, J. Fridrich, M. Goljan, J. Lukas, "Source digital camcorder identification using sensor photo response non-uniformity," SPIE International Conference on Security, Steganography, Watermarking of Multimedia Contents, vol. 6505, no. 1, 2007.

[2] Y. Su, J. Xu, B. Dong, "A source video identification algorithm based on motion vectors," International Workshop on Computer Science and Engineering, vol. 2, pp. 312-316, 2009.

[3] J. Lukas, J. Fridrich, M. Goljan, "Digital camera identification from sensor pattern noise," IEEE Transactions on Information Forensics and Security, 1(2): 205-214, 2006.

[4] Z. Geradts, J. Bijhold, M. Kieft, K. Kurosawa, N. Saitoh, "Methods for identification of images acquired with digital cameras," Enabling Technologies for Law Enforcement and Security, pp. 505-512, Feb 2001.

[5] F. Lefebvre, B. Chupeau, A. Massoudi, E. Diehl, "Image and video fingerprinting: forensic applications," SPIE-IS&T Electronic Imaging, vol. 7254, article 725405, pp. 1-9, 2009.

[6] G.C. Holst, CCD Arrays, Cameras and Displays, JCD Publishing, 1998.

[7] M.K. Mihcak, I. Kozintsev, K. Ramchandran, "Spatially adaptive statistical modeling of wavelet image coefficients and its application to denoising," IEEE International Conference on Acoustics, Speech, and Signal Processing, vol. 6, pp. 3253-3256, 1999.

[8] J. Yang, H. Choi, T. Kim, "Noise estimation for blocking artifacts reduction in DCT coded images," IEEE Transactions on Circuits and Systems for Video Technology, 10(7): 1116-1120, 2000.

[9] Z. Wang, A.C. Bovik, B.L. Evan, "Blind measurement of blocking artifacts in images," IEEE International Conference on Image Processing, Vancouver, Canada, 2000.