International Journal of Research Studies in Science, Engineering and Technology [IJRSSET]
Volume 1, Issue 4, July 2014, PP 18-23
ISSN 2349-4751 (Print) & ISSN 2349-476X (Online)

Authentication of Surveillance Video using Video Fingerprinting

Leena Giri G[1], Varahi J Sirigeri[2]

[1] Associate Professor, Department of Computer Science & Engineering, Dr. Ambedkar Institute of Technology, Bengaluru.
[2] MTech Student, Department of Computer Science & Engineering, Dr. Ambedkar Institute of Technology, Bengaluru.

Abstract: *Installation of video cameras in public facilities for surveillance has becomes more and more popular. The camera data are being used by multiple authorities. There are cases where surveillance video must be transmitted in the clear, which is without encryption. To ensure trust of the received video, it must be authenticated. Video authentication has gained more and more attention in recent years. In this paper, we propose a novel authentication scheme which can distinguish malicious attacks. While lossless video is straightforward to authenticate by cryptographic means, lossy video is more difficult to authenticate. We describe a method that combines an efficiently computed video fingerprint with public key cryptography to enable lossy video authentication.*

Keywords: *Video Surveillance, data privacy, authentication, digital signature, video fingerprint.*

1. Introduction

In previous generation of video surveillance, whose camera data were transmitted over dedicated channels (thus the name, closed circuit TV, CCTV), today’s data are transmitted via Internet protocol (IP) on public, multiuse networks. This brings many benefits, but at least one major challenge: It is difficult to assure that the digital video received at the viewing end is the same as it was actually shot by camera device. Therefore video authentication is a process which ascertains that the content in a given video is authentic and exactly same as when captured. For verifying the originality of received video content, and to detect malicious tampering and preventing various types of forgeries, performed on video data, video authentication techniques are used.

Media authentication methods can be categorized as: stream or packet-based [1], [2], and content-based [3], [4]. Stream based authentication is to directly authenticate at the stream or packet level. The system security can be mathematically proven, as it is based on conventional data security approaches. In Stream based method each block is digitally signed. As long as the number of lost packets is less than a threshold, all received packets can be authenticated. However, this scheme has high computational overhead. It also suffers from a high receiver delay because the receiver has to wait for a minimum number of the received packets for authentication. These methods can offer quantifiably strong security, however they are usually designed for a specific communications protocol or compression scheme, they are not generally robust to video changes other than the changes designed for, and they do not account for transcoding, where authentication means must be transferred across different coding schemes.

In contrast to stream-based methods which authenticate at the data level, content-based methods authenticate at the level of features, and are designed to do so with qualities of persistence and perceptual invariance. Use of an orderless collection of local features. These are calculated by frame, and then assembled into sequences to provide authentication.

In the content-based category are digital watermarks. A watermark is embedded into a video, and when extracted from the received video can indicate distortions, e.g. A tradeoff inherent to watermarking is the boundary between robustness to media channel distortions, and detection of malicious distortions. Some authentication methods use watermarks for transporting a digital signature. Some content-based methods require the authentication to be known by the receiver to authenticate. This is not appropriate for our application where several agencies plus the general public would have to have these keys to authenticate.

System security plays a vital role in an authentication system. Three modules mainly affect system security: feature extraction, ECC, and hashing. Therefore, the security performance of the system may be measured in terms of the probability of the system being cracked, i.e., given an video, the probability of finding another video that can pass the signature verification, under the same parameters.

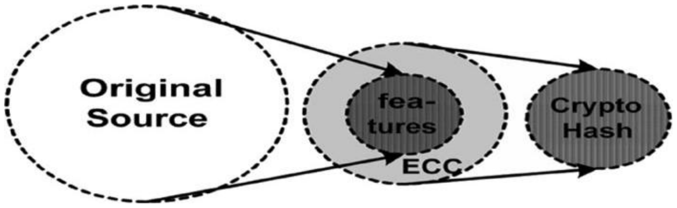


Fig. 1. *System security illustration on content-based authentication.*

Content-based authentication schemes provide the ability to authenticate content that has undergone acceptable manipulations, as long as the content features are preserved. In this paper, we use motion features, which are calculated from surveillance video, to authenticate video from camera to receiver. A sequence of motion of local salient features is calculated from frames, and this sequence is said to be a video fingerprint. Each fingerprint is digitally signed. Video is sent unencrypted, so it can be viewed only by authentic person, and integrity of the signal can also be checked by person who is holding a public key.

Contributions of our work include:

• Combined use of a nonexact video fingerprint with an exact digital signature for lossy authentication.

• Concise video fingerprint that is a concise, but accurate, time series collection of trajectories of salient features.

2. Method

Our method falls under the content-based authentication category for which content is reduced to a video fingerprint.

2.1 Video Fingerprint Generation

Since motion information is important in surveillance video, we extract motion feature from video and use this as a more concise fingerprint. There is much research on motion feature representation. Existing systems can achieve speed through the use of global signatures (e.g., color histograms [5], ordinal signature [6]) but these methods sacrifice video distinctiveness. Our proposed method seeks to find a middle ground: tracking local features, called keypoints, allows us to retain more robustness while aggregating across time; and matching as a global signature allows us to achieve speed efficiency.

The video fingerprint we generate for each video seeks to capture trajectories of motion of the most salient features of the video across time. In Fig. 2, fingerprint generation method accepts an input video V represented as a sequence of sampled frames $V=\{v_0,v_1,v_2 \ldots v_{s-1}\}$. The frames are sampled randomly according to sampling rules agreed between generator and receiver of the fingerprint as shown in fig 3. We generate a histogram of orientations of optical flow for each consecutive pair of frames. Salient features are first detected from sampled frame using a local feature detector, FAST [7] as shown in fig 4. We then calculate the optical flow of these features from one frame to the next by applying the Lucas-Kanade method [8]. This is done by extracting features similar to histograms of orientations of optical flow (HOOF) [9]. We only retain trajectories with magnitudes within a realistic range (e.g., 3 to 50 pixels for 5 frames/sec sampling rate). Orientation of trajectories is less sensitive to lossy video noise than other metrics such as magnitude of trajectories, which can be changed due to rescaling of original video. We distribute the orientations of trajectories of local features motion into a histogram of angular range bins. Each bin records the number of keypoints that have moved in a given orientation for each pair of consecutive sampled frames. In our implementation, we use 8 bins of 45 degree range each as shown in fig 5.

Finally, angular range values are concatenated into B sequences over a time series of sampled frames. Formally, a sequence of fingerprints is presented as $F = \{\{ f_{i,j} : 0 < i < m \} :0 < j < B\}$, where F is the sequence of frame fingerprints, is a fingerprint of a sampled frame, is the length of sampled frame sequence. For our experiment, we choose 8 bins, and thus eight signature time series are generated to comprise a single video fingerprint of a video sequence.

At the end of this step, for each pair of consecutive frames $\{v_i,v_{i+1}\}$ we have collected a histogram: $\{a_{i,j} : 0 < j < B \}$ where records the number of keypoints that moved in a given orientation and is the total number of bins.

Choice of Local Feature Detector: We chose to track local features instead of uniformly sampled points despite the additional computational cost because: the optical flow features are more stable with respect to noise, and this is consistent with intuition that motion of the salient features of each frame is the most distinctive feature. We experimented with different local feature detectors including SIFT, SURF [10] and FAST. We concluded that FAST was the ideal feature detector for our system over SIFT and SURF as it runs significantly faster (it is computed through direct pixel comparisons) and produces more keypoints, which is an advantage as the effect of inaccurate keypoint tracking is mitigated. Although FAST demonstrates less robustness with respect to feature orientation, it is nonetheless sufficient as we are mostly tracking slight changes from frame to frame.

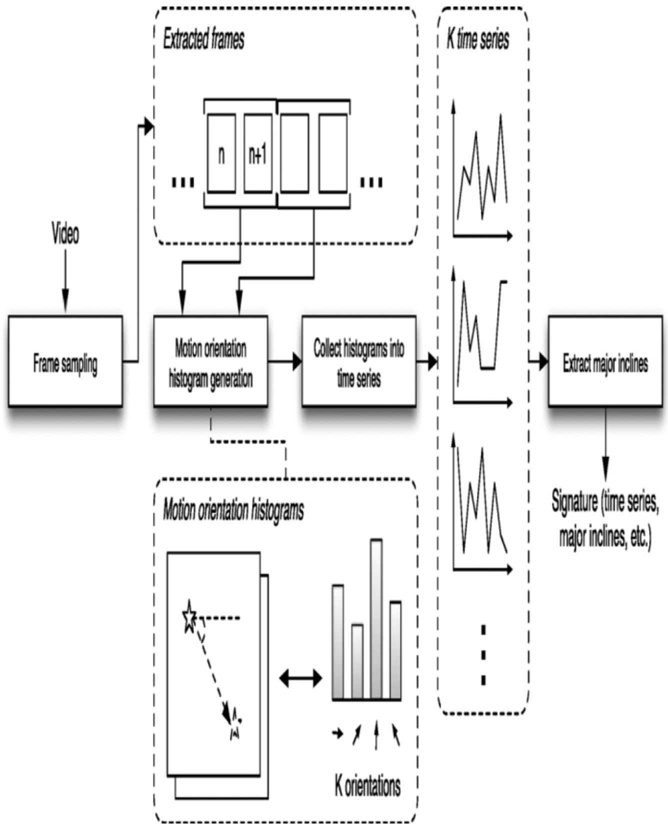

Extracted frames
...
n
n+1
...
K time series
Video
Frame sampling
Motion orientation histogram generation
Collect histograms into time series
Extract major inclines
Signature (time series, major inclines, etc.)
Motion orientation histograms
K orientations

Fig.2. *Video fingerprint generation..*

Fig.3. *Samples frame Selected from the surveillance video.*

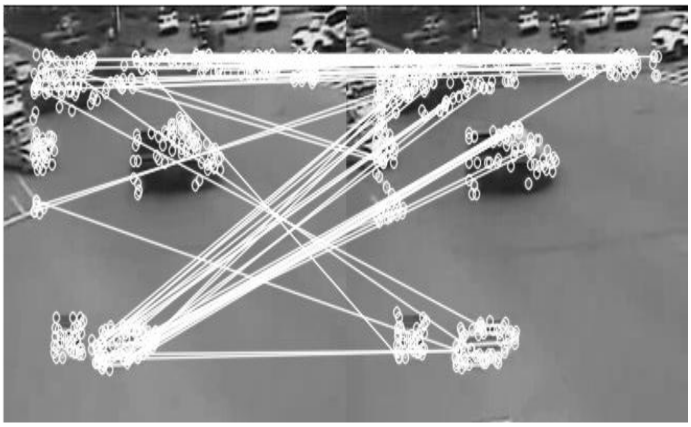


Fig.4. *Local features extracted using FAST.*

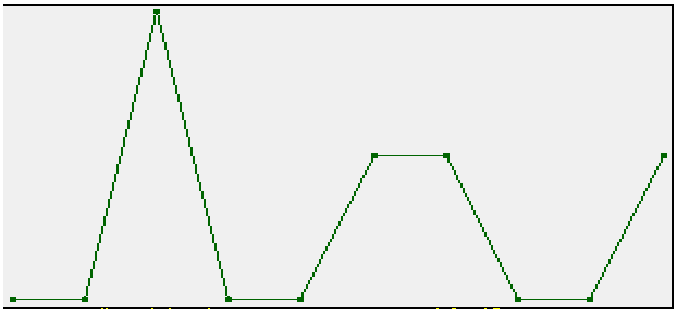
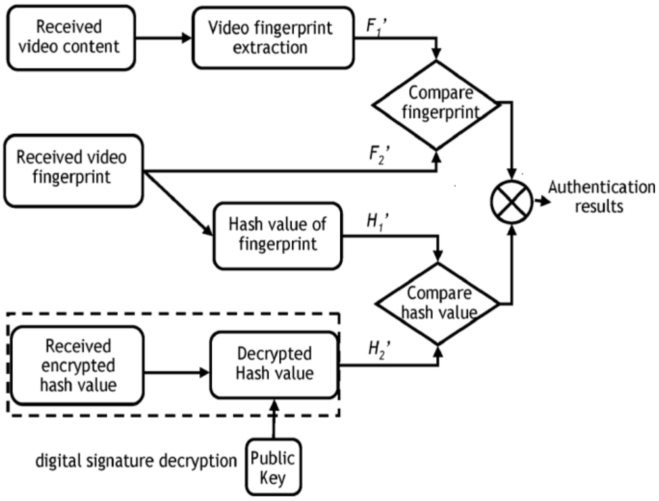


Fig.5. *The histogram shows the values of 8 bins for the orientations of the optical flows.*

These motion features are regarded as incurring too high a computational load; however our system already perform some degree of feature analysis to reduce bandwidth and error rate of false alerts, so such systems have no additional computational cost for this more reliable feature determination.

2.2 Fingerprint Matching

Our authentication scheme does *not* use a robust hash, but instead a robust *method* for hash-matching [11]. Each frame fingerprint is digitally signed. This hash result cannot be used alone for authentication if there are video distortions, however in addition to the digital signature, we include the video fingerprint in the clear (i.e., unencrypted), the combination of which enables authentication of distorted signals. To authenticate, as shown in Fig. 6, the receiver does three operations: 1) The video fingerprint is hashed H_1'(using a public seed), to obtain ; 2) The digital signature is decrypted using the public key and the resulting hash, H_2' , is compared with H_1'; 3) The video fingerprint is calculated of the received video,F_1' and compared with the received video fingerprint F_1' . Then the frame is said to be authenticated if they match.



Received video content
Video fingerprint extraction
F_1'
Compare fingerprint
Received video fingerprint
F_2'
Authentication results
Hash value of fingerprint
H_1'
Compare hash value
Received encrypted hash value
Decrypted Hash value
H_2'
digital signature decryption
Public Key

Fig.6. *Verification of video fingerprint at receiver side.*

Since a video fingerprint is represented as a time series, $D(F'_1, F'_2)$ can be calculated by measuring the distance between time series. However, modifications in video transmission due to scaling, transcoding, and packet loss can cause complications in this distance calculation. To capture the notion of complexity difference between two time series, the method adopts the complexity invariant distance measure. The correction factor is then multiplied by the distance between the two series in order to make it "complexity- invariant". The complexity invariant distance D_{CIV} is computed as:

$$D_{CIV}(F_{1i}, F_{2j}) = \frac{\max\{K(F_{1i}), K(F_{2j})\}}{\min\{K(F_{1i}), K(F_{2j})\}} D_E(F_{1i}, F_{2j})$$

where F_{1i} and F_{2j} are two time series for a histogram bin j, D_E is the Euclidean distance and $K(F_j)$ is a measure of complexity of each time series. Intuitively, $K(F_j)$ measures the RMS of the series derivative, giving more weight to series with greater variance.

After the similarity distances are obtained for B time series, we compute a score $\Delta(F_1,F_2)$ for the compared fingerprint pair, which is a tuple containing the number of time series distances above a certain threshold d_2 and the average of those distances.

That is, for $Dtotal = \{ D_{CIV}(F_{1i}, F_{2j}) : 0 \le j < B, \text{and } D_{CIV}(F_{1i}, F_{2j}) > d_2\}$,

$$\Delta(F_1,F_2) = (|Dtotal|, \sum Dtotal / |Dtotal|)$$

This method is not overly sensitive to d_2, which we determined empirically. Two identical videos should have all bins matched with an average distance of 0.

3. Experiment Results

3.1 Robustness of Video Fingerprint

The video distance metric was computed for the lossy video sequences. Fig. 7.The plot of SSIM (Structural Similarity Measure) versus the distance metric suggests a correlation between the two. It should be noted that even though the video is of poor quality at higher error rates and correspondingly lower SSIM, the video is still authentic.

In the case of video transcoding, Fig.8 suggests that the distance metric is affected more by resolution than by bandwidth changes. Again, the video is authentic in all of these cases.

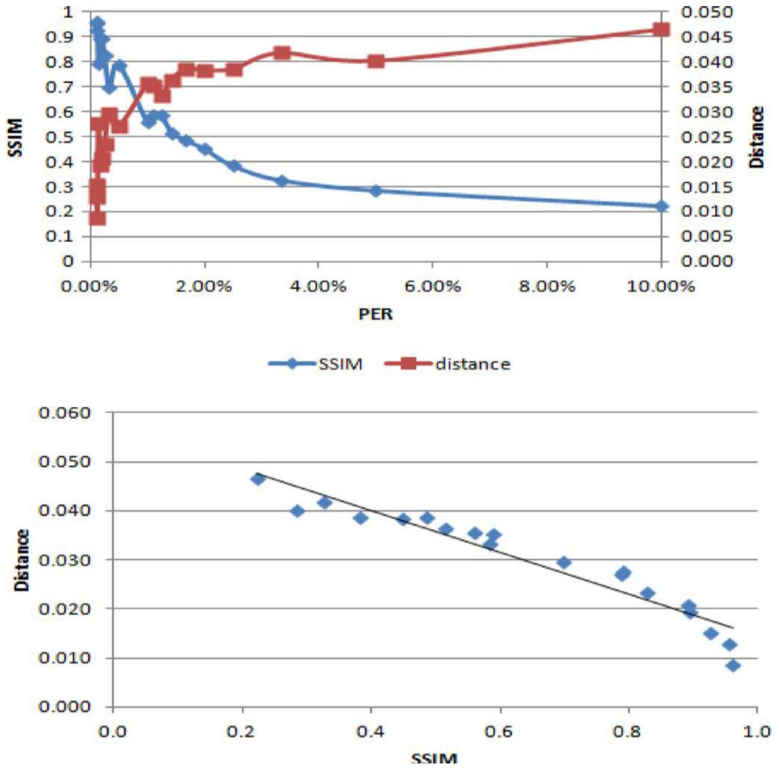


Fig. 7. *(Top) SSIM and distance versus PER; (bottom) SSIM and distance correlation.*

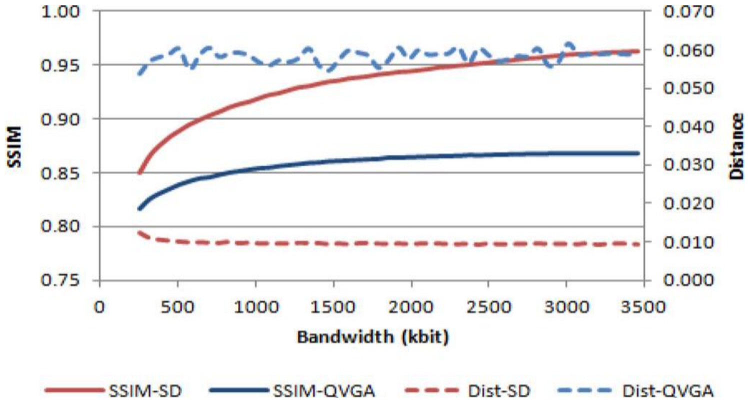


Fig. 8. *Distance metric behavior for transcoded video.*

4. Conclusion

Authentication of lossy video is a challenge that will become increasingly important as agencies and the public share video feeds. We have proposed a method to authenticate video whose authentication rate is high for videos and noise tested. The methods combine a video fingerprint with a cryptographic digital signature. The video fingerprint is sent separately from the video signal, incurring an increase in transmission complexity, and there is a cost of 5–8 seconds latency before authentication results are available.

References

[1] M. Hefeeda and K. Mokhtarian, “Authentication schemes for multimedia streams: Quantitative analysis and comparison,” ACM Trans. Multimedia Comp., Comm., and App., vol. 6, no. 1, pp. 1–10, Feb. 2010.

[2] K. Mokhtarian and M. Hefeeda, “Authentication of scalable video streams with low communication overhead,” IEEE Trans. Multimedia, vol. 12, no. 7, pp. 731–853, Nov. 2010.

[3] X. Yang and M. Lin, “Content-based video: A survey,” in Proc. Int. Conf. Info. Tech: Res. and Edu., Mar. ’04, pp. 50–54.

[4] Q. Sun, J. Aposolopoulos, C. Chan, and S. Chang, “Quality-optimized and secure end-to-end authentication for media delivery,” Proc. IEEE, vol. 96, no. 1, pp. 97–111, Jan. 2008.

[5] J.Huang,S. R.Kumar,M.Mitra,W.Zhu, andR.Zabih, “Image indexing using color correlograms,” in Proc. IEEE Conf. Computer Vision and Pattern Recognition, San Juan, PR,USA, Jun. 1997, pp. 762–768.

[6] D.N. Bhat and S. K. Nayar,“Ordinal measures for image,”IEEE Trans Pattern Anal. Mach. Intell., vol. 20, no. 4, pp. 415–423, Apr. 1998.

[7] E. Rosten and T. Drummond, “Machine learning for high-speed corner detection,” in Proc. Eur. Conf. Comp. Vision, Graz, Austria, 2006, pp. 430–443, LNCS 3951.

[8] B. D. Lucas and T. Kanade, “An iterative image registration technique with an application to stereo vision,” in Proc. 1981 DARPA Imaging Understanding Workshop, Washington, DC, USA, Apr. 1981, pp. 121–130.

[9] R. Chaudhry, A. Ravichandran, G. Hager, and R. Vidal, “Histograms of oriented optical flow and binet-cauchy kernels on nonlinear dynamical systems for the recognition of human actions,” in Proc. IEEE Conf. Computer Vision and Pattern Recognition, Miami, FL, USA, Jun. 2009, pp. 1932–1939.

[10] H. Bay, T. Tuytelaars, and L. V. Gool, “SURF: Speeded up robust features,” Computer Vis. Image Understand., vol. 110, pp. 346–359, 2008.

[11] L. O’Gorman and I. Rabinovich, “Secure identification documents via pattern recognition and public-key cryptography,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 20, no. 10, pp. 1097–1102, Oct. 1998.