

# Robust Track-and-Trace Video Watermarking

Lei Yang, Qian Chen  
University of Florida  
Gainesville, Florida 32611  
Email: leiyang@ufl.edu  
qiantrue@ufl.edu

Jun Tian  
Futurewei Technologies Inc.  
Hazlet, NJ, 07730  
Email: jtian@huawei.com

Dapeng Wu  
University of Florida  
Gainesville, Florida 32611  
Email: wu@ece.ufl.edu  
Telephone: (352) 392-4954  
Fax: (352) 392-0044

## Abstract

With the development of computers and the Internet, digital multimedia can be distributed and pirated easily. Watermarking is a useful technique for multimedia copyright protection. In this paper, we develop a robust video watermarking system. It consists of two components, i.e., watermarking embedder and watermarking detector. In the embedder, we insert a watermark pattern into video frames according to a watermark payload. The watermark pattern is generated from a Pseudo-random Noise (PN) sequence generator using the spread spectrum technique. User and copyright information are mapped to a binary sequence and then encrypted with Advance Encryption Standard (AES) and encoded/protected by convolutional error correction code (ECC) to produce watermark payload. The watermark pattern is weighted and embedded to each frame to meet perceptual requirements. In addition, the video is slightly geometrically manipulated in order to defend possible collusion attacks. The detector extracts the watermark from the candidate video. Kanade-Lucas-Tomasi feature tracker (KLT) is used to register the candidate video with respect to the original video, to enhance the correlation with the reference. The cross-correlation sequence is binarized, ECC decoded and decrypted. The experimental results show that the proposed video watermark system is very robust to not only geometric attack, but also collusion attacks, and that it is perceptually invisible to human vision system.

## I. INTRODUCTION

Nowadays, computers, interconnected via the Internet, make the distribution of the digital media fast and easy. However, it also requires less effort to obtain the exact copies. Therefore, it poses great challenges to copyright protection for digital media. Digital watermark embedding is a process of integrating the user and copyright information into the carrier media in a way invisible to human vision system (HVS). Its purpose is to protect the digital works from the unauthorized duplication or distribution.

Video watermarking system is desired to embed watermark in such a way that the watermark can be detected later for authentication, copyright protection, and track-and-trace illegal distribution. Videos, composed of multiple frames, can utilize image watermarking techniques in a frame-wise manner [1]. Although the watermarking embedding capacity of video is much larger than that of image, the attacks the video watermarking suffers are more complicated than image watermarking. The attacks include not only spatial attacks, but also temporal attacks and hybrid spatial-temporal attacks.

In the literature of track-and-trace video watermarking, the algebra-based anti-collusion code is investigated [2]–[8]. Its ability to trace one or multiple colluders depends on the assumption that the code is always available and error-free, which may not be true in practice. Besides, the length of anti-collusion code hinder the system user capacity. Hence, practical and multi-functional watermarking systems based on algebra-based anti-collusion code are very limited.

To this end, we propose a robust track-and-trace watermarking system for digital video copyright protection. It consists of two independent bodies, watermarking embedder and watermarking detector. At embedder, user and product copyright information, e.g. a string of length  $L_s$ , is first encrypted with Advanced Encryption Standard (AES) [9] to form a binary sequence. We then apply error correction code (ECC) [10] to the sequence to generate a binary sequence with error-correction ability of length  $L$ , called watermark payload. Meanwhile, a frame-size watermark pattern arises from a pseudo-random noise (PN) sequence [11], [12]. Each binary bit in watermark payload is associated with one video frame and determines how watermark is embedded to this frame. Bit 0 indicates subtracting the watermark pattern from the current frame, while bit 1 indicates adding the watermark pattern to the current frame. We will repeatedly embed  $L$  bits if the video is longer than  $L$  frames, and sync the watermark payload at the beginning of dramatic video scene changes to resist temporal attacks. Furthermore, in order to meet the perceptual quality, we build a perceptual model to determine the signal strength that can be embedded to each frame pixel. Note that the stronger the embedded signal, and hence the easier the watermark can be correctly detected. However, watermark pattern is like random noise, and too strong of the noise signal can cause noticeable distortion to the picture. The randomness of PN sequences also make the embedded watermark information blind to the attackers. To make a trade-off between capacity and visual quality, we build a perceptual model to determine the signal strength that can be embedded to each pixel. Finally, since distributed videos are prone to collusion attacks, we propose to apply geometric transforms to the watermarked videos. This is called geometric anti-collusion coding in this paper. These transforms include rotation, resizing and translation, and should be moderate enough to cause no defect to HVS, but also enhance the capability to resist collusion attacks.

The watermarking detector just carries out the reverse process of the embedder. In this system, we assume the detector can always have access to the original video as the prototype of the candidate video. Because of the geometric anti-collusion

coding at embedder, watermark usually cannot be correctly extracted without any pre-processing even the candidate video is an error-free copy. Additionally, spatial attacks such as further geometric manipulations and temporal attacks may occur to distributed videos. In this paper, we propose to register the candidate video to the original video spatially and temporally. An iterative KLT based scheme is applied for spatial registration, whereas temporal registration is to match frames that minimize the mean-square-error (MSE). We then compute cross-correlation coefficients between re-generated watermark pattern and frame difference of the registered frame and its corresponding original frame, demodulate the coefficient sequence to recover the watermark payload. It is then ECC decoded (convolutional coding for specific) and AES decrypted to derive the original user or copyright information. Successful detection indicates the user or copyright information is correctly extracted, otherwise we say the detector fails to detect the watermark.

The paper is organized as follows: Section II describes the overall architecture of the proposed track-and-trace video watermarking system. The watermarking embedder techniques are discussed in Section III. Section IV introduces watermarking detector techniques. The experimental results presented in Section V verify robustness of the proposed video watermarking. Finally, the conclusion and future work are given in Section VI.

## II. ARCHITECTURE OF ROBUST VIDEO WATERMARKING SYSTEM

The architecture of the track-and-trace video watermarking system includes two independent components, i.e., watermarking embedder (Fig. 1) and watermarking detector (Fig. 2). It is an additive watermarking system. Watermarking embedder consists of two functional components, watermark generator to generate watermark payload (Fig. 1(a)), and watermark embedder to embed the payload to video frames (Fig. 1(b)). Watermarking detector extracts payload from candidate video (Fig. 2(a)), and then recover user or copyright information from the payload (Fig. 2(b)).

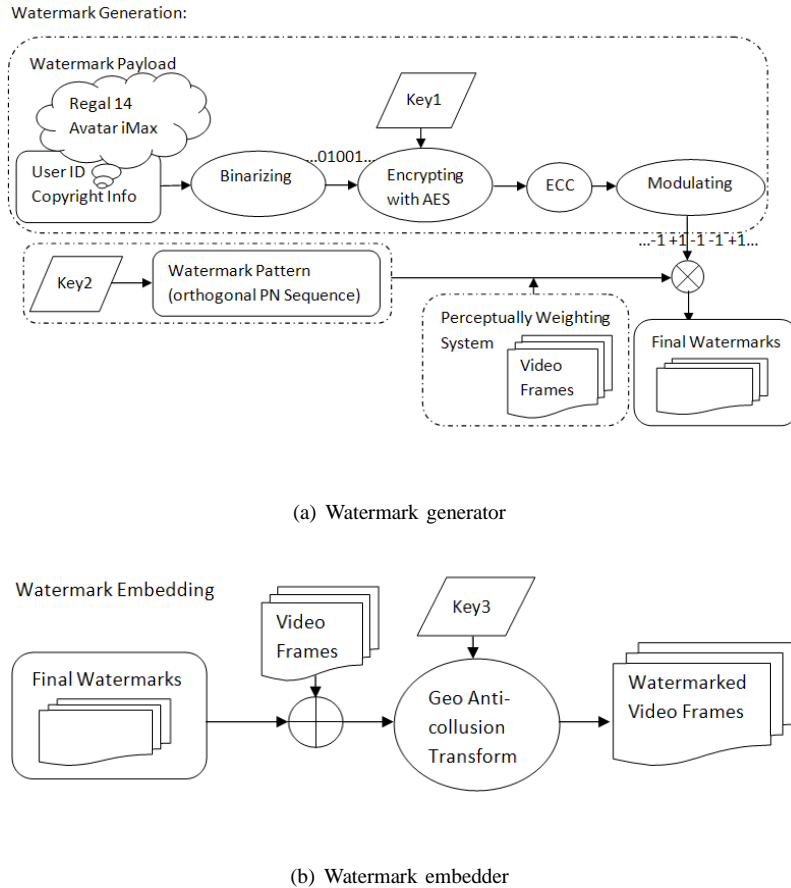
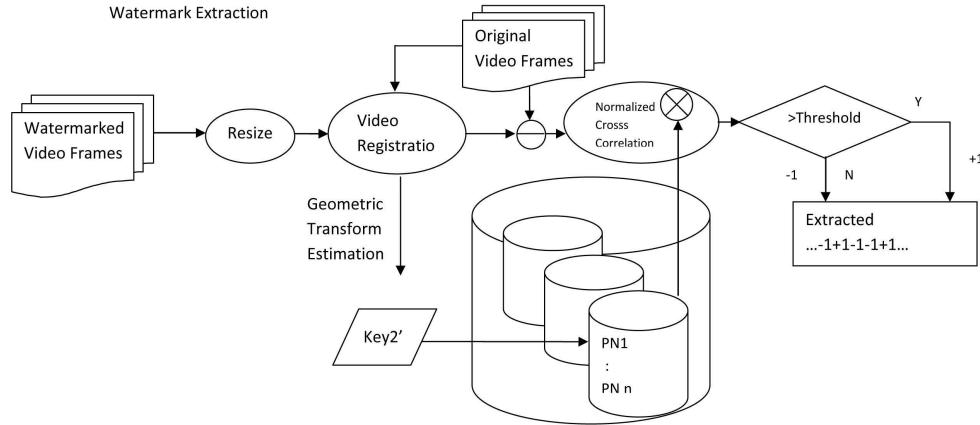


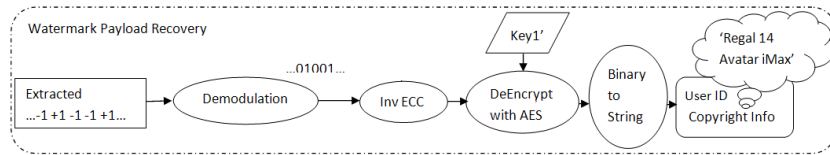
Fig. 1. Track-and-trace video watermarking embedder

### A. Watermarking Embedder

The proposed video watermarking system is an additive system, i.e. adding watermark signal to the original video. The inputs of embedder are the original video, user ID and copyright information. The key configure parameters are the frame



(a) Watermark extractor



(b) Payload recovery

Fig. 2. Track-and-trace video watermarking detector

size (widthxheight) of the input video, AES encryption key ( $Key1$  in Fig.1(b)), pattern ID ( $Key2$  in Fig.1(b)) to generate watermark pattern and  $Key3$  to generate geo-transform parameters for film distributors.

String-type user/copyright information are binarized, encrypted, and ECC coded into watermark payload. If convolution code rate =  $1/2$  is used, the information of  $L_s$  characters is transformed into  $L = 16L_s$  bits. Watermark pattern by using orthogonal PN sequences can resist frame-averaging collusion. The pseudo random watermark patterns weighted by perceptual modeling of each frame are embedded with the largest strength under imperceivable constraint. The length of PN sequence  $N$  is frame size (widthxheight). The number of the orthogonal sequences of length  $N$  is exactly  $N$ . For geometry transform, the bicubic interpolation is used to keep original video information as much as possible. There are  $\pm 5^\circ$  rotation,  $\pm 5$  pixel translation and  $\pm 5\%$  resizing. Thus, the proposed system could accommodate  $1000N$  distributors ideally.

After embedding, the watermarked videos are distributed. They may suffer from intentional manipulations or unintentional degradations later. These attacks include but not restrict to geometric manipulations, erroneous transmission, and collusion.

### B. Watermarking Detector

The inputs of detector are the candidate video and its original copy. The goal is to extract watermark payload from the candidate video and recover the user/copyright information with reference to the original copy. Some of the key configure parameters are the size (width and height) of the two input videos, AES decryption key ( $Key1'$  in Fig.2(a)) and pattern ID ( $Key2'$  in Fig.2(a)) to re-generate watermark pattern. Usually, we set  $Key1' = Key1$ ,  $Key2' = Key2$  for consistency of symmetric AES and PN-sequence generation at both ends.

The distributed video may be enlarged or cropped in size, referring to as resizing attack. Hence, the candidate video may differ in frame size with the original video. The detector employs a resize algorithm to the candidate video to match them

in size wherever necessary. The algorithm is bicubic interpolation (expanding) or decimation (shrinking). Note that we apply geometric anti-collusion coding at embedder. Also, malicious attacks may impose spatial and temporal transforms attempting to remove the watermark information. On the other hand, the detector is very sensitive to these transforms and often fails in detection without any pre-processing to the candidate video. Accordingly, we first register the candidate video to the reference video, both spatially and temporally. Normalized cross-correlation coefficients are computed between each pair of the registered candidate frame and the reference frame. The anti-collusion geometric transform information brought by  $Key3'$  is used to trace possible illegal distributors.

Then we do binary hard decision to get +1/-1 sequence from the coefficients based on a threshold, and demodulate it to a binary 0/1 sequence, which is the extracted watermark payload. Finally, the payload is ECC decoded and AES decrypted to recover the user/copyright information of string type, as illustrated in Fig.2(b). Here Viterbi algorithm is used for ECC decoding regarding convolutional code for ECC encoding at embedder.

The proposed video watermarking system is integrated with various techniques, include spectrum spreading, AES encryption and decryption, ECC encoding and decoding, perceptual weighting model, geometric anti-collusion coding and frame registration. The following section will introduce these techniques in detail respectively.

### III. WATERMARK EMBEDDING TECHNIQUES

#### A. Watermark Pattern Generation

A seed denoted as  $Key2$  in Fig.1(a) is required to generate a PN-sequence as watermark pattern using spectrum spreading. It should be of the same size with the video frame in order to do matrix addition. The PN-sequence can be m-sequence, Walsh codes or Kasami sequence with optimal cross-correlation values. The orthogonal PN-sequences are desired between different videos to resist averaging collusion, and desired between different watermark payload (+1/-1) to resist temporal attacks. Orthogonal m-sequence is used in our system. For frame-size  $N$  (widthxheight), the length of m-sequences is  $N$ , hence, there are  $N$  orthogonal m-sequences.

#### B. Watermark Payload Generation

Product copyright and user information require encryption to keep it from attackers who want to detect or tamper the content. After encryption, the information appears as noise to the attackers. The encryption technique could be Rivest-Shamir-Adleman cryptography (RSA), Data Encryption Standard (DES), Advance Encryption Standard(AES) and so on. The encryption key denoted as  $Key1$  in Fig.1(a) could be the choice from the watermark creator following certain rules. In our system, we choose AES for encryption and set the length of standard key to be 128 bits.  $Key1$  could be both user and video related. We assume that it is a common key to both embedder and detector, known as a symmetrical encryption system. If unsymmetrical encryption system is used, the embedder has private key  $Key1$ , and the detector has public key  $Key1'$ .

Moreover, video distribution process can be viewed as transmission in channel, and the attacks to the media is regarded as channel noise. Therefore, 1/2 convolution code is adopted in our system for error correction coding (ECC). After encryption and encoding, a binary sequence of length  $L$  is generated as watermark payload.

The binary payload is further modulated into +1/-1 sequences as:

$$X' = 2X - 1, \quad (1)$$

where  $\{X\} \in \{0, 1\}$  is the binary payload,  $X'$  is the modulated sequence.

#### C. Perceptual Weighting Model

As mentioned in section I, there is a trade-off between watermark capacity and visual quality in determining the signal strength that can be embedded into video frames. We build a perceptual model in both temporal domain and spatial domain. The objective is to maximize the watermark capacity without causing noticeable degradation to visual quality of videos. The model diagram is shown in Fig. 3. Embedding strength is determined in a pixel-wise manner for every frame. Hence, it is formulated as a heightxwidth mask matrix  $M$ , with each entry describing the weight for the collocated watermark pattern signal. Then for length  $L$  watermark payload, the watermark corresponding to the  $i$ th payload bit in frame  $kL + i$ ,  $k \in Z^+$  is:

$$W = \text{sign}(X'(i)) \cdot M \odot P \quad (2)$$

where  $\odot$  is element-wise product,  $P$  is watermark pattern,  $X'(i)$  is the  $i$ th watermark payload. The pixel values in the watermarked frame  $W$  should be clipped to  $[0,255]$ .

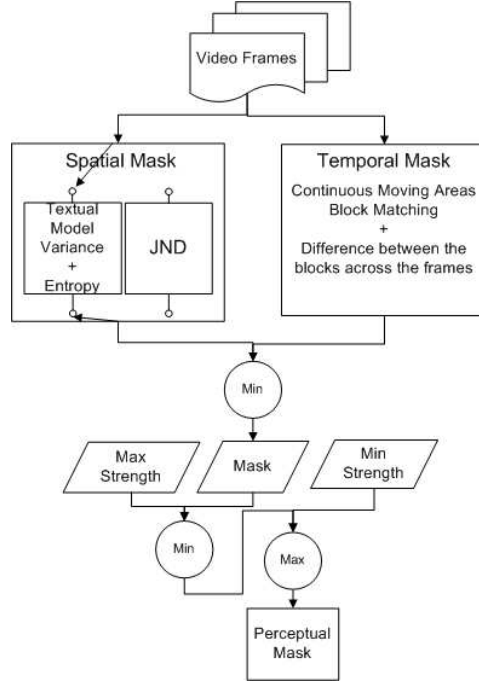


Fig. 3. Perceptual Modeling Diagram

1) *Temporal Perceptual Modeling*: The perceptual model in temporal domain is based on the fact that human eyes are sensitive to changes with slow motion, but not to fast moving changes. Generally, the larger the difference between the current frame and the previous frame, the stronger the embedded signal strength could be. But the simple difference between adjacent frames is not good enough to describe object moving. For example, if an object is moving leaving a smooth background, we cannot embed high strength watermark into the smooth background. Therefore, we propose a block motion matching algorithm to find the difference between blocks in current frame and previous frame with the least sum of absolute differences (SAD), which is defined as:

$$SAD(\Omega, \Omega') = \sum_{(i,j) \in \Omega, (i',j') \in \Omega'} |I_c(i, j) - I_p(i', j')| \quad (3)$$

where  $\Omega$  is the block in the current frame,  $\Omega'$  is the block in the previous frame,  $(i, j), (i', j')$  are the pixel coordinates,  $I_c$  is the current frame,  $I_p$  is the previous frame.

The algorithm for perceptual model in temporal domain is summarized as follows:

```

for each block  $\Omega$  in the current frame do
  for each block  $\Omega'$  in  $N_B(\Omega)$  in the previous frame do
    if  $SAD(\Omega, \Omega') < minSAD$  then
       $minSAD = SAD;$ 
       $Diff = |I_c(\Omega) - I_p(\Omega')|;$ 
    end if
  end for
end for

```

where  $N_B(\Omega)$  is the neighborhood of  $\Omega$  in the range  $B$ .

$$N_B(\Omega) = \{z \in I_p | B_z \cap \Omega \neq \emptyset\} \quad (4)$$

where  $B_z$  is the translation of  $B$  by the vector  $z$ , i.e.,  $B_z = \{b + z | b \in B\}, \forall z \in I_p$ .

Temporal model first perform block matching between two adjacent frames, and calculate the difference between these

matching blocks. Then the differences are scaled as temporal mask.

$$TemporalMask = \alpha \cdot Diff \quad (5)$$

2) *Spatial Perceptual Modeling*: We propose two perceptual models in spatial domain. They can either be used independently or combined together to generate one spatial mask.

Model 1 is based on edge and texture. The underlying principle is that rough area like texture rich area and edges, could be embedded with higher strength watermark, since human eyes are insensitive to changes in these areas. To accurately identify these areas, we use a combination of three metrics to describe such area.

- The first metric  $Map1$  is the difference between current pixel and its surrounding pixels. If the difference is large, it means the current pixel is located in area that can tolerate relative large changes, so that large embedded signal strength is possible.  $Map1$  is calculated by a convolution of the original frame and high pass filter  $H$ , which is defined below:

$$H = \begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix} \quad (6)$$

- The second metric  $Map2$  is the variance of the block which is centered at the current pixel. The larger the variance, the higher the embedded signal strength can be.
- The third metric  $Map3$  is the entropy of the block which is centered at the current pixel. The higher the entropy is, the richer texture the area has, and the higher embedded signal strength could be.

Each of the three metrics can describe how rich the texture is of the local area around pixel, but none of them is sufficient by its own. Therefore, we define the spatial as the product of the three metrics:

$$SpatialMask1 = \beta \cdot Map1 \cdot Map2 \cdot Map3 \quad (7)$$

where  $\beta$  is a scaling factor.

Model 2 is based on saliency map [13] and Just Noticeable Difference (JND) model [14], [15] of video frames. The saliency map highlights salient texture areas in a image, where could be imperceivable embedding locations. To obtain saliency map, the frame is down-sampled and low pass filtered in the fourier transform domain, and then up-sampled to the original frame size. The magnitudes of the saliency map describe the frequency of frame information. Visual just noticable difference reflects nonlinear response of human eyes to spatial frequency. Based on JND human perceptual model, the saliency map is further mapped into spatial mask by:

$$SpatialMask2 = \frac{\eta}{(SaliencyMap + \delta)} \quad (8)$$

To guarantee good visual quality of videos, we choose the minimal value between the spatial mask and temporal mask for each pixel. And the final embedded signal strength is also bounded by a minimum and a maximum value. The perceptual weighting map (PWM) is defined as:

$$PWM = \min(maxStrength, \max(minStrength, \min(TemporalMask, SpatialMask))) \quad (9)$$

#### D. Geometric Anti-collusion Coding

After embedding the watermark payload into the carrier video, we apply geometrical transforms to each copy of the video. The transform is a combination of shifting, resizing and rotation, and varies among different video distributors. For each video copy, its specific transform index is a random variable generated by  $Key3$ , related to user and copyright information. The extent of the transform should be moderate enough in order not to be aware by HVS, but can still be detected by computers. If colluders try to linearly or nonlinearly combine the multiple video copies to eliminate the embedded watermark, the resulted video will usually be blurred and become unaccepted by human eyes. Thus, the geometrical transform protects video watermark from inter-video collusion attacks. This process is called geometrical anti-collusion coding. To preserve as much information as possible, bi-cubic spline interpolation [16] is used to fill the blank area after transform.

## IV. WATERMARK DETECTION TECHNIQUES

### A. Video Frame Registration

Apart from the geometric anti-collusion coding, the input candidate video at detector may go through many changes, either accidental manipulations or malicious attacks. Two major categories among the changes are affine transform in spatial domain, and frame add/drop in temporal domain. Since detector has access to original video, we can use original copy as reference and register candidate video to the reference in both spatial domain and temporal domain.

1) *Spatial Registration*: The spatial registration is based on Kanade-Lucas-Thomasi (KLT) feature tracker. Affine model is used in spatial registration [17], [18]. The affine transform model to any pixel  $(x, y)$  is:

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} a & b \\ c & d \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} e \\ f \end{bmatrix} \quad (10)$$

The objective of spatial registration is to find the 6 affine transform parameters  $a - f$  in the model, so as to do inverse affine transform before detection. KLT achieves this by matching the corresponding feature points in the candidate frame and the original frame, and get the solution to the parameter set. We call the rectified frame  $F^{(1)}$ . For each pixel  $(x, y)$  of  $F^{(1)}$ , we compute its pixel position  $(x', y')$  in candidate frame  $F^{(0)}$ . Take  $F^{(0)}(x', y')$  as the match in  $F^{(1)}(x, y)$  if  $x', y'$  are integers; otherwise, we interpolate  $F^{(0)}(x', y')$  at  $(x', y')$ . However, due to the complexity of the transform and the imperfectness of KLT algorithm, the rectified frame after one-time inverse affine transform is often not good enough to extract watermark from. Therefore, we propose to refine the estimate  $F^{(1)}$  by applying KLT iteratively. Specifically, we have affine transform displacement expressed as:

$$\begin{bmatrix} \delta x \\ \delta y \end{bmatrix} = \begin{bmatrix} x' - x \\ y' - y \end{bmatrix} = \begin{bmatrix} a - 1 & b \\ c & d - 1 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} e \\ f \end{bmatrix} \quad (11)$$

When the  $i$ th KLT iteration gets  $F^{(i)}$  and the corresponding affine parameter set  $a^i - f^i$ , we compute the displacement of each pixel. We keep doing this until the convergence condition is satisfied or we reach the maximum number of iteration. In this system, we check the maximum pixel displacement between two consecutive rectifications:

$$\max_{x^i, y^i \in F^{(i)}} \{|\delta x^i|, |\delta y^i|\} < \epsilon \quad (12)$$

where

$$\delta x^i = x^i - x^{i-1} \quad (13)$$

$$\delta y^i = y^i - y^{i-1} \quad (14)$$

and  $\epsilon$  is a pre-defined threshold.

In most cases, we expect spatial registration based on KLT to improve the detection performance if the candidate video actually experiences certain affine transform. However, the detector is unaware of the exact manipulation to the candidate frame. If it is not affine transform, KLT gives wrong parameters, and the performance after spatial registration can be worse than that without it. Hence, spatial registration is set optional in our detector. Typically, detector can control to switch on/off the spatial registration if it has manipulation information. Otherwise, we can always try both and choose the one with better detection result.

2) *Temporal Registration*: In temporal registration, we use the simple rule of minimizing the mean-square-error (MSE) to register candidate frame to the original frame. We scan original sequence to find the best match for current candidate frame that minimize MSE [19]. One causal constraint is put so that no frame displayed in the past can be captured in the future. That is, if two frames  $i, j$  in candidate video with  $i < j$ , and then the registered frame  $\alpha(i), \alpha(j)$  in original video must satisfy  $\alpha(i) \leq \alpha(j)$ . For frame  $k$  in the candidate video, it computes the MSE with  $n$  consecutive frames in the original video  $\alpha(k-1) + 1, \dots, \alpha(k-1) + n$ , where  $\alpha(k-1)$  is the previous registered frame, and register the current frame to the one with the minimal MSE.

Likewise, temporal transform may or may not appear in the candidate video. The performance after temporal registration could be worse than that without it if no temporal manipulation occurs. Therefore, temporal registration is also set optional in our detector. If temporal registration is enabled, it is usually performed ahead of spatial registration.

## B. Watermark Extraction and Payload Recovery





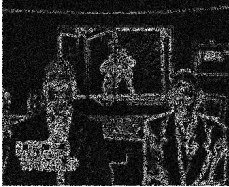





After registration, it is assumed each frame in candidate video has found its match frame in the original video. Note the watermark is a additive system that adds watermark pattern into original frame. Hence, we can detect the existence of watermark signal by computing the cross-correlation between the watermark pattern and the true frame difference. We use a key exactly corresponding to Key3 at embedder to re-generate the watermark pattern, a frame-size PN sequence at the detector. The normalized cross-correlation is defined as:

$$NC(P, \hat{P}) = \left\langle \frac{P}{\|P\|_F}, \frac{\hat{P}}{\|\hat{P}\|_F} \right\rangle \quad (15)$$

where  $P$  is the watermark pattern,  $\hat{P}$  is the true difference between candidate frame and its registered frame;  $\langle \cdot, \cdot \rangle$  denotes inner product, and  $\|\cdot\|_F$  is Frobenius norm.

The range of the normalized cross-correlation is  $[-1, 1]$ . The larger the absolute value of the coefficient, the better chance the candidate frame contains the regenerated pattern, i.e. it has the watermark information embedded. Each candidate frame

TABLE I  
STEP BY STEP RESULT OF WATERMARKING EMBEDDER

Sequences	PWM	Watermarked	Geo-transformed
			
			
			
			

corresponds to one coefficient value. A hard decision threshold of 0 is used to make the coefficient sequence to a binary  $-1/+1$  sequence. If the coefficient is larger than 0, we denote it as "1", otherwise it is "-1". The extracted  $-1/+1$  sequences  $\{X'\}$  is then demodulated to 0/1 sequence  $\{X\}$  as:

$$X = (X' + 1)/2 \quad (16)$$

The watermark payload recovery is the reverse process of payload generation. The binary payload sequence  $X'$  needs to be decoded and decrypted to derive the string. For ECC decoding, we use Viterbi algorithm to decode the convolutional code [20]. And AES decryption method is described in standard [9]. The 128-bit AES key used in decryption is denoted as  $Key1'$ , usually set to be identical to  $Key1$ .

## V. EXPERIMENTAL RESULTS

The step by step results of watermark embedding are listed in Table I. The test video sequences are downloaded from [21] for watermarking embedding. They are YUV sequences of CIF format including *Foreman*, *Mobile*, *News* and *Stefan*. Column 1 shows the Y components of the 90th frames in the original sequences. Column 2 represents their corresponding watermark patterns after perceptual weighting model. Column 3 shows the watermarked sequences from which the watermark is imperceptible. And Column 4 is the watermarked frames after geometric transform with anti-collusion ability. They all undergo up-right shifting 3 pixels, clockwise rotate  $2^\circ$ , and resizing 101%. We can hardly distinguish the watermarked frames in Column 3 with the original frames in Column 1, which meets our perceptual requirement for watermarking system. Furthermore, geometric anti-collusion code does not cause much distortion neither, as frames in Column 4 and Column 3 look exactly alike. The PSNR of the watermarked sequences are shown in Fig. 4, which falls in the range of 34 and 47 dB.

At detector side, we test the capability of detector to correctly extract watermark information under various attacks. Among them, the most important task is to verify the capability to resist affine transforms, not only because they are used for anti-collusion coding at embedder for security purpose, but the distributed videos can encounter malicious geometric manipulations as well. Fig.5 lists the watermarked frame under various affine transforms. The test sequence is 300 frames QCIF *Grandma*, and the 5th frame is selected to illustrate the effect of multiple geometric transforms, including 25 pixel rotation (around  $8^\circ$  rotation) (5(c)), 10 pixel expanding (around 105.7%) (5(d)), 10 pixel shrinking (around 94.3%) (5(e)) and 40 pixel shifting 5(f).



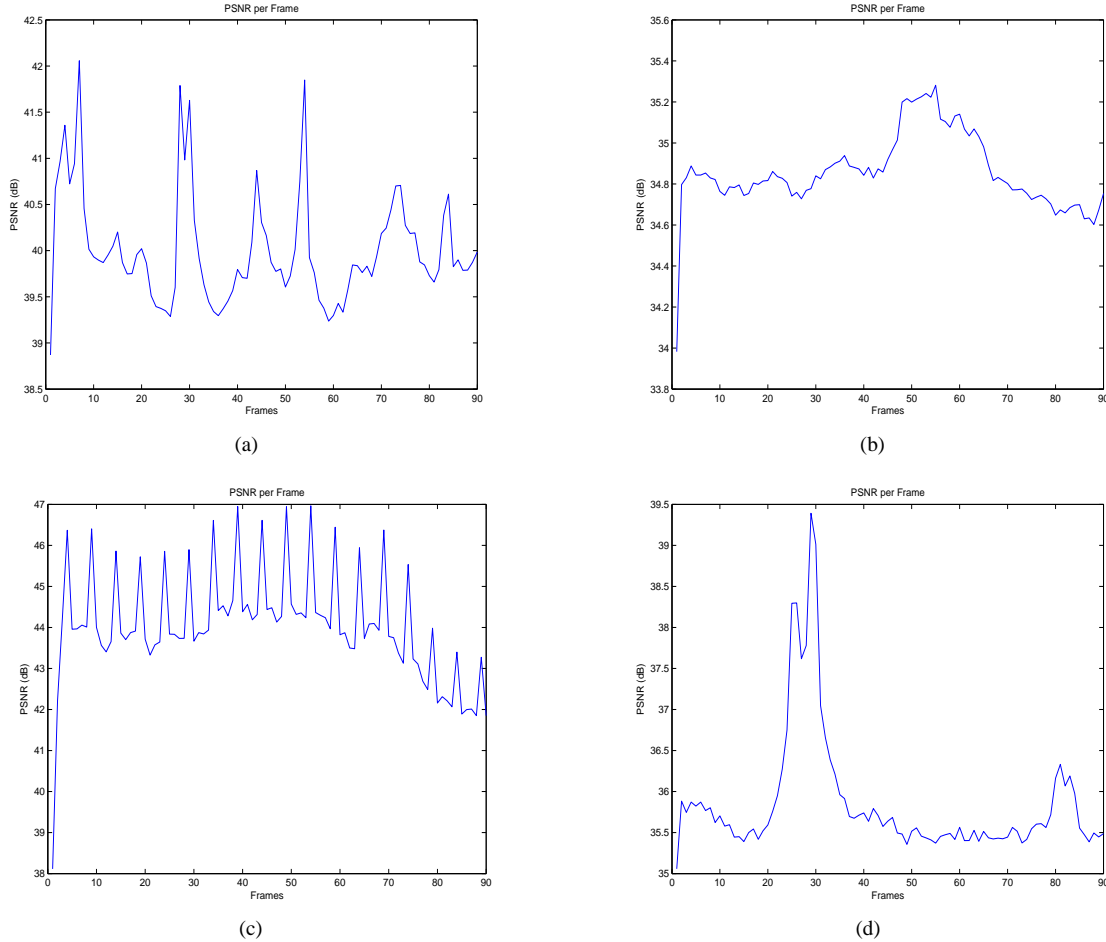


Fig. 4. PSNR of watermarked video sequences in Table I. (a) Foreman. (b) Mobile. (c) News. (d) Stefan.

Note how significantly the last three transforms change the frame structure. The geometric transforms to such extent have been easily detected by human eyes, hence they may be out of the range the anti-collision coding can carry out on watermarked video, and very likely the result of third party attacks. Therefore, the performance the detector achieves on these videos can fully justify it under affine transform attacks. Table II shows the error rate of cross-correlation coefficients the detector obtains under the above mentioned transform scenarios. It is defined as:

$$R_e = \frac{M_e}{M} \quad (17)$$

where  $M_e$  is the number of erroneous demodulated binary bits, and  $M$  is the number of frames in the sequence. Note this error rate is obtained before ECC decoding, which can further correct the bit error. The first row shows the error rate without frame registration. The error rate is too high for ECC to correct. And it turns out we cannot get the correct watermark information at detector. One time KLT registration has significantly reduced the error rate. The iterative KLT registration can further improve the performance (the third row) but not so significant as what one time registration to no registration at all. We notice for 25 pixel rotation, iterative KLT is actually identical to one time KLT as it only operates the registration once. This is because these are all single kind transforms, either rotation, resizing or shifting. And one time KLT is good enough to track the correct transform parameters. While combinational transforms pose greater challenge for KLT based spatial registration. As shown in Table III, complex affine transforms and single transform of higher magnitude require iterative KLT to enable the detector to extract the correct watermark. Note that for resizing, positive value means shrinking (5(e)), and negative value indicates expanding (5(d)). There are some combinatorial transforms in which KLT registration fails (indicated by "N/A"), i.e. iterative KLT registration will not converge after maximum iteration times and fails to estimate the correct parameters.

## VI. CONCLUSION

In this paper, we propose a robust track-and-trace anti-collision watermarking system. At the embedder, the user and copyright information is securely mapped to binary sequences using AES, ECC, which results in watermark payload. Orthogonal frame



Fig. 5. Geometric transform/attacks to 5th frame of *Grandma*. (a) Original frame. (b) Watermark frame. (c) Watermark frame with 25 pixel rotation. (d) Watermark frame with 10 pixel expanding. (e) Watermark frame with 10 pixel shrinking and truncated to original size. (f) Watermark frame with 40 pixel shifting.

TABLE II  
CROSS CORRELATION COEFFICIENT ERROR RATIO (%) WITH FRAME REGISTRATION IN *Grandma* SEQUENCE

	25 Pixel Rotation	10 Pixel Expanding	10 Pixel Resizing	40 Pixel Shifting
No registration	41.67	3	48.67	47.67
1-KLT Registration	0	2.33	0.67	2.33
Iterative KLT Registration	0	0	0	0.33

TABLE III  
CAPABILITY OF KLT BASED VIDEO REGISTRATION FOR VARIOUS GEOMETRIC TRANSFORMS (N/A: NOT APPLICABLE)

shift	resize	rotate	KLT iteration time	Capability
0	-15	0	2	Y
0	-20	0	N/A	N
0	15	0	2	Y
0	20	0	N/A	N
0	0	40	2	Y
10	10	10	4	Y
20	10	20	N/A	N
20	5	20	4	Y
20	-5	20	3	Y
30	5	10	N/A	N

size PN-sequence is generated with secret key as watermark pattern. The pattern is then perceptually weighted and integrated with the original video sequence frame by frame according to watermark payload. For anti-collusion purpose, the watermarked video will be geometrically transformed before distribution. At the detector, candidate video will be spatially and temporally registered to the original video if needed. We compute the cross correlation between the re-generated watermark pattern and frame difference to extract the payload. Then the payload is ECC decoded and AES decrypted to get the final watermark information. Experimental results show that the proposed system is robust against geometric attacks and collusion attacks, and meets the requirement of invisibility to HVS.

Meantime, it also shows that iterative KLT registration has limitations. Our future work includes further investigation into the transform attacks and enhancing the detector capability to cope with complicated combinatorial affine transforms and non-affine transforms. Moreover, we will test the detector under other types of video-related attacks such as compression, erroneous transmission, and reverse order display.

#### REFERENCES

- [1] V. Potdar, S. Han, and E. Chang, "A survey of digital image watermarking techniques," in *Proceedings of IEEE International Conference on Industrial Informatics*, 2005, pp. 709–716.
- [2] P. Bas and J. Chassery, "A survey on attacks in image and video watermarking," *Applications of digital image processing XXV: 8-10, Seattle, Washington, USA*, p. 169, 2002.
- [3] W. Trappe, M. Wu, Z. Wang, K. Liu *et al.*, "Anti-collusion fingerprinting for multimedia," *IEEE Transactions on Signal Processing*, vol. 51, no. 4, pp. 1069–1087, 2003.
- [4] M. Wu, Z. Wang, and K. Liu, "Anti-collusion Fingerprinting for Multimedia," *IEEE Transactions on Signal Processing*, vol. 51, pp. 1069–1087, 2003.
- [5] Z. Wang, M. Wu, W. Trappe, and K. Liu, "Anti-collusion of group-oriented fingerprinting," in *Proceedings of IEEE International Conference on ICME'03.*, vol. 2, 2003.
- [6] B. Cha and C. Kuo, "Design and analysis of high-capacity anti-collusion hiding codes," *Circuits, Systems, and Signal Processing*, vol. 27, no. 2, pp. 195–211, 2008.
- [7] G. Tardos, "Optimal probabilistic fingerprint codes," *Journal of the ACM (JACM)*, vol. 55, no. 2, pp. 1–24, 2008.
- [8] D. Boneh and J. Shaw, "Collusion-secure fingerprinting for digital data," *Advances in Cryptology CRYPT095*, pp. 452–465, 1995.
- [9] J. Daemen and V. Rijmen, *The design of Rijndael: AES—the advanced encryption standard*. Springer Verlag, 2002.
- [10] S. Lin and D. Costello, *Error control coding: fundamentals and applications*. Prentice-hall Englewood Cliffs, NJ, 1983.
- [11] I. Cox, J. Kilian, F. Leighton, and T. Shamoan, "Secure spread spectrum watermarking for multimedia," *IEEE transactions on image processing*, vol. 6, no. 12, pp. 1673–1687, 1997.
- [12] H. Rheingold, *Smart mobs: The next social revolution*. Basic Books, 2003.
- [13] R. Achanta, S. Hemami, F. Estrada, and S. Susstrunk, "Frequency-tuned salient region detection," in *Proceedings of IEEE Conference on CVPR 09*. IEEE, 2009, pp. 1597–1604.
- [14] D. Booth and R. Freeman, "Discriminative measurement of feature integration in object recognition," *Acta Psychologica*, vol. 84, pp. 1–16, 1993.
- [15] I. Cox, *Digital watermarking and steganography*. Morgan Kaufmann, 2008.
- [16] R. Keys, "Cubic convolution interpolation for digital image processing," *IEEE Transactions on Acoustics, Speech and Signal Processing*, vol. 29, no. 6, pp. 1153–1160, 1981.
- [17] B. Lucas and T. Kanade, "An iterative image registration technique with an application to stereo vision," in *International joint conference on artificial intelligence*, vol. 3. Citeseer, 1981, pp. 674–679.
- [18] J. Shi and C. Tomasi, "Good features to track," in *Proceedings of IEEE Computer Society Conference on CVPR'94.*, 1994, pp. 593–600.
- [19] H. Cheng, "A review of video registration methods for watermark detection in digital cinema applications," in *Proceedings of IEEE International Symposium on Circuits and Systems'04.*, vol. 5, 2004, pp. 704–707.
- [20] T. Moon, *Error correction coding: mathematical methods and algorithms*. Wiley-Blackwell, 2005.
- [21] "Yuv video sequences." [Online]. Available: <http://trace.eas.asu.edu/yuv/index.html>