An Imperceptible and Blind Watermarking Scheme Based on Wyner-Ziv Video Coding for Wireless Video Sensor Networks

Noreen Imran^{a,*}, Boon-Chong Seet^a, A. C. M. Fong^b

^aSchool of Engineering, Auckland University of Technology, Auckland 1010, New Zealand.

^bSchool of Computing and Mathematical Sciences, Auckland University of Technology, Auckland 1010, New Zealand.

*Corresponding Author: nimran@aut.ac.nz

Abstract

Distributed video coding (DVC) is an emerging video coding paradigm designed to allow low-complexity compression by resource-constrained video sensor nodes in wireless video sensor networks (WVSN). Several DVC architectures have been proposed in literature but there are hardly any security mechanisms designed to validate the integrity and authenticity of the decoded video. In this paper, we propose a novel, low-complexity, blind, and imperceptible video watermarking scheme based on DVC for WVSNs. The watermark information is embedded into key frames under selected GOP (Group of Pictures) configurations. Considering the information-intensive yet resource-constrained WVSN environment, the proposed scheme aims to limit the impact of watermark embedding complexity by hiding the secret information using non-zero quantized AC coefficients. We implemented the proposed watermarking scheme for a DVC based codec and evaluated its processing complexity (encoding/decoding). We also evaluated the average normalized correlation between the embedded and extracted watermarks for different video sequences by varying the redundancy in embedding.

Keywords: Watermarking, Distributed Video Coding, Wireless Video Sensor Networks.

1. Introduction

Recent technological advancements in low-cost miniaturized video sensors and wireless communications are enabling the development of video-enabled wireless

sensor networks for real-time applications such as video-assisted navigation, critical infrastructure surveillance, and health monitoring. A wireless video sensor network (WVSN) generally comprises of several sensing nodes, each equipped with a video camera and wireless transceiver. Due to limited processing and energy capacities of the nodes, the traditional predictive video coding (PVC) paradigm based on complex encoding and simple decoding architecture is not well suited for WVSN environment. On the other hand, distributed video coding (DVC) is an emerging video coding paradigm designed to allow low-complexity compression by resource-constrained video sensor nodes in WVSN. Digital video watermarking is a technique for validating the integrity and authenticity of the received video. Existing video watermarking schemes are also mostly if not entirely based on PVC, which are not suitable for DVC based codecs due to the significant disparity between their coding frameworks⁽¹⁾. Hence, new watermarking techniques based on DVC need to be designed for the WVSN environment.

Standard video coding solutions such as ITU-H.26x⁽²⁾ and ISO-MPEGx⁽³⁾ work well when it comes to the compression efficiency and reconstruction quality of the coded video. These standards are based on PVC paradigm which performs complex operations such as motion estimation and compensation at the source coding site, which aggravates the computational complexity of the encoder 4-5 times more than that of the decoder. Such encoder-decoder formation is appropriate for the *downlink application architectures* where encoder is located at the energy unconstrained base-station, while the decoder is a comparatively low-complexity module that may operate on energy constrained devices. Video broadcasting and telephony are the examples of downlink architectures.

On the other hand, the *uplink application architectures* represent the reverse paradigm, where low-complexity encoder is required and the complexity of the decoder is not a major concern. In a WWSN, several battery-operated video sensing nodes with limited on-board processing and wireless communication functionality are used to capture, compress and transmit compressed video data from their surroundings to a centralized location (base-station) over single or multi-hop environment. Such application architectures require a low-complexity encoding module that not only can enhance the lifetime of the sensing devices but also the lifetime of the entire network.

DVC is a video coding technology that matches well with uplink application architectures, as it relocates the processing complexity from the encoder to the decoder (or intermediate transcoder), resulting in simpler and less energy consuming encoders to be used while maintaining a compression ratio comparable to that of conventional codecs⁽⁴⁾. This is made possible by exploiting source statistics on the decoder which eventually transforms encoder into a simple and low-complexity module⁽⁵⁻⁶⁾. Such encoder-decoder formation makes DVC a well suited coding alternative in uplink application architectures.

In this paper, we proposed a novel, low complexity, blind, and imperceptible watermarking scheme for WWSNs based on Wyner-Ziv (WZ) video coding, which is one of the most popular DVC techniques. The remaining of the paper is organized as follows: Section 2 covers the only available related work, Section 3 introduces the proposed watermarking scheme, Section 4 shows the experimental setup for the evaluated results and discusses the presented results, and finally Section 5 concludes the paper with some suggestions for the future research.

2. Related Work

To the best of our knowledge, there exists only a single work by Ning et al.⁽⁷⁾ on a DVC based video watermarking scheme where the watermark is embedded into least significant bits (LSB) of selected discrete cosine transform (DCT) coefficients of each key frame. The scheme used Arnold transformation⁽⁸⁾ and Harris corner detector⁽⁹⁾ to scramble the watermark image, and identify the interest points in each key frame for watermark embedding, respectively. Harris corner detector processes each key frame to evaluate the number of interest points which by

itself is a complex operation that may cause additional delay in real-time applications. Moreover, the capacity of embedding is dependent on the type of frame itself, while embedding in DCT coefficient increases the bitrate requirement and may cause a drift error in WZ frame reconstruction. We also implemented the scheme under discussion to present a comparative analysis of our proposed scheme.

3. Proposed Scheme

In this paper, we proposed a novel, low complexity, blind, and imperceptible video watermarking scheme based on WZ coding⁽¹⁰⁾, which builds on the principles of DVC. The WZ coding architecture encodes video sequence using GOP¹ configuration. For a given GOP size n , every first frame in a GOP is classified as a key frame and compressed via conventional intra-frame encoding, while the remaining $n - 1$ non-key frames are compressed via WZ encoding. At the decoder site, key frames are reconstructed using conventional intra-frame decoder, whereas decoding of the WZ frames requires side-information generated from both previously decoded key and WZ frames. The WZ video coding architecture along with our proposed watermarking module is presented in Fig.1.

Due to the resource constraints and codec related dependencies, we proposed to watermark only the key frames in every GOP configuration. Key frame is the most significant frame, since it provides the basis (side-information) for the rest of the non-key (WZ) frames in a given GOP. The Low Density Parity Check (LDPC) encoder-decoder is being utilized as a Slepian-Wolf codec. Each bit-plane vector is decoded when side-information and residual statistics are available. However, if decoder cannot decode a bit-plane, it requests for additional parity bits from encoder via feedback channel, and the process continues until a certain acceptable level of bit error rate (BER) performance is achieved. Such an arrangement ensures that the WZ encoder will transmit only a small amount of parity bits to the decoder for reconstruction of quantized bitstream. However, the decoder continues to generate feedback requests until the quantized bitstream is reconstructed with desired quality parameter.

¹A configurable setting GOP (Group of Pictures) determines the distance between key frames in the video sequence, and thus the coding efficiency.

Feedback channel is also used to transmit rate control information to the encoder, which is justified since the encoder does not need to perform any computation associated with the selection of encoding bit rate. Similar to inter-frames, WZ-frames are highly compressed and have lower payload due to fewer residual statistics. Therefore, embedding watermark in these frames may adversely affect the coding efficiency as well as the complexity of the watermark embedding module. Hence, for WZ coding, key frame watermarking is a viable and energy efficient solution for uplink application architectures.

The WZ encoder implementation in ⁽¹¹⁾ employed H.264/AVC Intra mode for key frame encoding. H.264/AVC has three alternate macroblock (MB) settings for Intra prediction mode, namely *Intra 4×4*, *Intra 16×16*, and *I_PCM*. We utilized *Intra 16×16* mode for our watermarking scheme which uses Hadamard transform to encode discrete cosine (DC) coefficients. Each *16×16*

macroblock is decomposed into *4×4* sub-blocks and processed one by one. In this mode, macroblock is predicted from top and left neighboring pixels and subsequently executes in four modes namely, horizontal, vertical, DC and plane modes. We proposed to embed the watermark information only into selective *4×4* sub-blocks of luminance macroblocks by suitably modifying the number of non-zero quantized AC coefficients (NZQAC) with the aim of restraining the bitrate escalation due to watermark embedding. The key frame watermarking framework based on H.264/AVC Intra is shown in Fig.2. Most of the existing video watermarking schemes works on modifying the DCT coefficients for watermark embedding, which not only adversely elevates the bit rate requirement but also degrades the video reconstruction quality. In contrast, working with AC coefficients for watermark embedding may cause only a slight distortion to the signal and minimal impact to the required bitrate. Details of the watermarking embedding algorithm are as follows:

Fig. 1. Wyner-Ziv Video Coding based Watermarking Framework

Fig. 2. H.264/AVC Intra based Key Frame Watermarking Framework

3.1 Embedding Algorithm

Step 1- Calculating Macroblock Mean Matrix from Camera Fingerprint

Every digital camera has its own distinctive characteristics; even identical units produced by the same manufacturer under the same conditions have their own unique features. This is due to the infinitesimal physical irregularities such as variation among pixels in relation to their sensitivity to light, which is also referred to as Photo Response Non-Uniformity (PRNU)⁽¹²⁾. Every image/frame captured from a particular camera exhibits a distinctive PRNU pattern which represents its unique fingerprint. We assumed that such fingerprint can be captured and placed within the video sensor's internal memory at the time of network installation. The results presented in this paper are based on the fingerprint we computed from our CITRIC smart camera⁽¹³⁾.

Input: *Camera_Fingerprint_Matrix* [] of dimension (176×144)

Output: *Macroblock_Mean_Matrix* [] of dimension (1×99)

- 1: Decompose *Camera_Fingerprint_Matrix* [] into
- 2: 99 macroblocks of dimension (16×16) . Calculate
- 3: the mean of each macroblock and store the result
- 4: in *Macroblock_Mean_Matrix* [].

Step 2- Evaluating Reference Point from *Macroblock_Mean_Matrix* []

Determine the *Reference_Point* in *Macroblock_Mean_Matrix* [] which decomposes the matrix into approximately two equal halves based on the

limit_{upper} and *limit_{lower}* variables.

Input: *Macroblock_Mean_Matrix* []

Output: *Reference_Point*

- 1: *limit_{upper}* = $\max(\text{Macroblock_Mean_Matrix}[])$
- 2: *limit_{lower}* = $\min(\text{Macroblock_Mean_Matrix}[])$
- 3: $\{ \text{limit}_{lower}, \dots, i-1 \} < \text{Reference_Point}$
- 4: $\leq \{ i+1, \dots, \text{limit}_{upper} \}$

Step 3- Watermark Scrambling

Each (sensor node, base-station) pair is assumed to share a 99-bit secret key. The watermark that needs to be embedded into the key frames is also a 99-bit sequence which could be derived from a secret binary message or a binary logo.

Input: *Macroblock_Mean_Matrix* [] , *Key* [] , *Watermark* [] all of dimension (1×99) , and a *Reference_Point* variable.

Output: *Scrambled_Watermark* [] of dimension (1×99)

- 1: for each *i*, where $i \in \{0, 1, 2, \dots, 98\}$ {
- 2: if (*Macroblock_Mean_Matrix* [*i*] >
- 3: *Reference_Point*)
- 4: *Scrambled_Watermark* [*i*] =
- 5: *Key*[*i*] $\otimes$ *Watermark* [*i*]
- 6: else
- 7: *Scrambled_Watermark* [*i*] =
- 8: *Watermark* [*i*]
- 9: }

Step 4 - Watermark Embedding

The proposed watermarking scheme embeds the

watermark in the luminance component of the Intra frame after DCT transformation and quantization within the reconstruction loop which employs (16×16) macroblock as a basic processing unit. Each (16×16) macroblock is then further decomposed into (4×4) sub-blocks as shown in Fig. 3. Following the integer DC transformation and quantization, the number of nonzero AC coefficients $Count_NZAC[i][j]$ in the selected sub-block j of macroblock i is counted. Thereafter, the last non-zero AC coefficient is marked as AC_l , where l is the order of this AC coefficient by Zig-Zag scan. Given the 99-bit $Scrambled_Watermark[]$ and 99 macroblocks per frame, each macroblock i is embedded with the corresponding $Scrambled_Watermark[i^{th}]$ bit. However, each macroblock is able to embed redundant copies of the given watermarking bit ranging from $\{1, \dots, 16\}$.

Fig. 3. Luminance component of 16×16 Macroblock and expanded 4×4 sub-block

The Boolean data structure $Embed_Watermark[]$ is utilized to choose a sub-block for embedding in each macroblock and to cause the disparity in the embedding pattern in each key frame. Watermark bit will be embedded in a particular sub-block if and only if the corresponding sub-block entry in $Embed_Watermark[]$ is set to 1. If the entire $Embed_Watermark[]$ is set to 1, it implies that the same watermark bit will be embedded in all 16 sub-blocks. At least one bit in $Embed_Watermark[]$ should be set to 1 in order to embed one complete watermark in a given key/Intra frame. We employed a simple hashing algorithm at the encoding and decoding nodes to generate similar $Embed_Watermark[]$ array for each frame using seed parameters such as frame number

Input: $Key_Frame[]$, $Scrambled_Watermark[]$, and $Embed_Watermark[]$ of dimension (176×144) , (1×99) , and (1×16) respectively.

Output: $Watermarked_Key_Frame[]$ of dimension (176×144)

```

1:  for each macroblock i {
2:      for each subblock j in macroblock i {
3:          Count total number of nonzero AC
4:          coefficients and store the result in
5:          Count_NZAC [i][j]
6:          if (Count_NZAC[i][j] is odd) AND
7:          (Scrambled_Watermark [i] is 0)
8:              Change last nonzero AC
9:              coefficient to 1
10:         else
11:         if (Count_NZAC[i][j] is odd) AND
12:         (Scrambled_Watermark [i] is 1)
13:             Do not change last nonzero
14:             AC coefficient
15:         else
16:         if (Count_NZAC[i][j] is even) AND
17:         (Scrambled_Watermark [i] is 0)
18:             Do not change last nonzero
19:             AC coefficient
22:         else
23:         if (Count_NZAC[i][j] is even) AND
24:         (Scrambled_Watermark [i] is 1)
25:             Change last nonzero AC
26:             coefficient to 1
27:         else
28:         if (Count_NZAC[i][j] is zero) AND
29:         (Scrambled_Watermark [i] is 1)
30:             Change last AC coefficient of
31:             the subblock to 1
32:         else
33:         if (Count_NZAC[i][j] is zero) AND
34:         (Scrambled_Watermark [i] is 0)
35:             Do not change last nonzero
36:             AC coefficient
37:     }
38: }

```

3.2 Hashing Algorithm

Input: Number of redundant watermarks to be embedded $redunt_wat$, the current frame number $frame_no$, and the maximum number of embedding corresponding to given macroblock configuration max_embed .

Output: A binary array $Embed_Watermark[]$ of dimension (1×16)

```

1: Initialize an array Embed_Watermark[ ] of
2: dimension (1 × 16) with zero.
3: if ( redunt_wat equals max_embed )
4:     Set entire Embed_Watermark[ ] to 1
5: else
6:     for each i where  $i \in \{1..redunt\_wat\}$  {
7:         index =  $i * frame\_no + \sigma$ 
8:         % max_embed,
9:     where  $\sigma$  is a constant.
10:    if (Embed_Watermark [index]
11:        is 1)
12:        Look out for first index in
13:        Embed_Watermark with Zero
14:        value and set it to 1.
15:    else
16:        Set Embed_Watermark[index] to 1
17:    }
18:

```

3.3 Detection Algorithm

The watermark detection algorithm is similar to the embedding algorithm in that it also performs the steps such as calculating the fingerprint macroblock mean, reference point value for macroblock mean matrix, watermark scrambling, and hashing for *Extract_Watermark* []. The main difference from the embedding algorithm is that rather than embedding the watermark and modifying non-zero AC coefficients, the detection algorithm will count the total number of non-zero AC coefficients in a particular sub-block. If it is odd, the extracted watermark bit is one, or zero otherwise. The watermark extraction function from a sub-block *i* can be expressed as:

$$W'_i = \begin{cases} 1, & \text{if } Count_NZAC'_i \% 2 \approx 1 \\ 0, & \text{if } Count_NZAC'_i \% 2 \approx 0 \end{cases} \quad (1)$$

The watermark bit is also extracted only from those sub-blocks with corresponding value set to 1 in *Extract_Watermark*[] array. The procedure to populate *Extract_Watermark*[] is the same as *Embed_Watermark*[].

4. Experimental Setup and Results

The DVC implementation from CMLAB ⁽¹¹⁾ is used and experiments were performed on 150 frames of Hillview and CoffeeCup video sequences, captured from our CITRIC smart camera ⁽¹³⁾. These sequences have been

selected to analyze the impact of our watermarking scheme on videos with varying characteristics. For example, the Hillview sequence captured in cloudy outdoor environment has regions with uniform texture and low-complexity content, while the CoffeeCup represents indoor office scenario with objects having sharp corner and edges and some foreground motion ⁽¹⁴⁾. A 99-bit binary image watermark is embedded into video all four video sequences. A frame rate of 15 fps is employed to encode raw video (YUV) of QCIF resolution using GOP size of 2 and 4 frames respectively. The DVC codec implemented in ⁽¹¹⁾ employed JM9.5 reference software ⁽¹⁵⁾ to encode key frames in GOP configuration. The encoder settings used for the key and WZ frames are shown in Table.1

Table 1. WZ encoder configuration parameters

H.264 Intra Encoder Configuration for Key Frames		DVC Encoder Configuration for WZ Frames	
Parameter	Value	Parameter	Value
Profile/Level IDC	77 main	Intra Mode	1 (Enable key frame sequence)
Frame Rate	15fps	Frame Rate	15fps
QP	28	QIndex	4
Period of I Frame	GOP Dependent	IntraQP	28
Rate Control	0 (Default)	Quantization parameter of Intra Deblocking filter	1 (Enable)
RD Optimization	0 (Low Complexity Mode)	Skip Block	1 (Enable)
		GOP Size	2 or 4

4.1 Capacity and Imperceptibility

Each 16 × 16 macroblock in a key frame is able to carry watermark bits ranging from 1-16, which means the maximum embedding capacity using QCIF resolution will be 1584 (99 macroblocks × 16 sub-blocks/macroblock) bits per key frame. Instead of embedding a single watermark of 1584 bits, we have chosen to embed multiple smaller watermarks (collectively comprising of 1584 bits) since radio transmission is more prone to channel errors which may cause a valid frame to be rejected due to failure in watermark extraction process.

Fig. 4 shows the average normalized correlation (NC) between the original watermark and extracted watermarks from two video sequences. A video frame from each of the video sequence is also shown in Fig. 4 watermarked at its maximum embedding capacity while maintaining visual imperceptibility. We gradually increased the watermark redundancy from 2 to 16 and extracted the watermark(s) from the decoded video. The average NC between the original and the extracted watermark for the proposed and Ning's scheme is at least 0.97 and 0.84 respectively. In contrast to the video watermarking scheme in ⁽⁷⁾ that embeds watermark into selective interest points around

corners and edges within the key frame, our proposed scheme has uniform and consistent embedding capacity irrespective of the content of the frame itself. Furthermore in ⁽⁷⁾, the watermark cannot be embedded if the entire frame content comprises of smooth and uniformly textured regions that has no corner or edges inside it. However, the video sequences have been selected carefully, such that both schemes exhibit similar embedding capacity in key frames.

Video Sequences	No. of Embeddings	2	4	8	16
		Average Normalized Correlation(NC)
Proposed Scheme				
		1.0	0.97	0.98	0.97
Ning's Scheme				
		0.97	0.95	0.91	0.84
Proposed Scheme				
		1.0	0.97	0.98	0.98
Ning's Scheme				
		0.98	0.96	0.93	0.86

Fig. 4. Normalized Correlation (NC) and watermarked video frame under maximum embedding capacity

4.2 Encoding and Decoding Complexity

The computational load on resource-limited sensor nodes is considered as a primary design issue not only for the video coding schemes but also for designing security mechanisms in WVSN environment. We evaluated the relative change in the encoding complexity of WZ codec for the video test sequences with and without watermark embedding under GOP 2 and GOP 4 configurations for both schemes (Fig. 5). It can be seen that the encoding complexity level increases with increasing redundancy in watermark embedding, and reaches its peak when 14 or 15 watermarks are embedded. Once again, a slight drop in the encoding complexity has been witnessed for all the video sequences for both GOP configurations with maximum embedding. This could be due to that no iterations for the hashing algorithm need to be executed in order to determine the subblocks for watermarking bit embedding.

The proposed watermarking scheme has a maximum

embedding capacity of 1584 bits per key frame. Based on the encoder settings, a key frame has a total of 99 macroblocks, each being able to embed up to 16 bits (1 bit in each subblock). The total capacity can be utilized by embedding a single watermark of 1584 bits or multiple redundant copies of a small watermark with less than 1584 bits. We chose the later option and embed redundant copies of a 99-bit watermark.

Fig. 5. Encoding Complexity vs. Embedding Capacity under GOP 2 and 4 configurations

Fig. 6. Decoding Complexity vs. Embedding Capacity under GOP 2 and 4 configurations

The encoding complexity associated with zero redundant watermark represents encoding-only computations, while that for sixteen redundant watermarks represents encoding computations along with embedding under maximum capacity utilization. It can be observed that video sequences with GOP 2 configuration have higher complexity than GOP 4 due to the alternate key frame encoding but the relative change in computational load for both GOP settings, even at maximum capacity utilization, is not more than 17%.

In contrast to encoding complexity, the impact of watermark detection on decoding algorithm with varying watermarking payload for both GOP configurations is quite negligible as shown in Fig. 6. The computations for embedding and detection algorithm are more or less the same but this behaviour is due to the heavy decoder which makes the additional computations for the detection algorithm relatively insignificant in reference to the video decoding.

5. Conclusions

In this paper, we proposed a novel, low complexity, blind, and imperceptible watermarking scheme based on WZ video coding. The encoding and decoding complexity of the WZ codec is evaluated with and without using watermarking scheme in order to analyze the additional computational load. It can be seen that the relative change in encoding and decoding complexities under different video sequences, GOP settings, and maximum capacity utilization, is not more than 17%, and 2%, respectively. Beside this, the impact of watermark embedding on video sequences in terms of imperceptibility is also very trivial as examined with maximum capacity utilization under various quantization levels for both GOP settings. For our future work, we would like to analyze the impact of channel errors on detection mechanism for indoor and outdoor environments over single/multi-hop communication in WVSNS.

References

- (1) L. A. Grieco, *et al.*, "Secure Wireless Multimedia Sensor Networks: A Survey," in *Mobile Ubiquitous Computing, Systems, Services and Technologies. UBICOMM '09. Third International Conference on*, 2009, pp. 194-201.
- (2) "Draft ITU-T Recommendation and Final Draft International Standard of Joint Video Specification," *ITU-T Rec. H.264 and ISO/IEC 14496-10 AVC*, 2003.
- (3) L. Hanzo, *et al.*, *Video compression and communications: from basics to H.261, H.263, H.264, MPEG4 for DVB and HSDPA-style adaptive turbo-transceivers*: IEEE Press, 2007.
- (4) T. Melodia and I. Akyildiz, "Research Challenges for Wireless Multimedia Sensor Networks," in *Distributed Video Sensor Networks*, B. Bhanu, *et al.*, Eds., ed: Springer London, 2011, pp. 233-246.
- (5) D. Slepian and J. Wolf, "Noiseless coding of correlated information sources," *Information Theory, IEEE Transactions on*, vol. 19, pp. 471-480, 1973.
- (6) A. Wyner and J. Ziv, "The rate-distortion function for source coding with side information at the decoder," *Information Theory, IEEE Transactions on*, vol. 22, pp. 1-10, 1976.
- (7) Z. Ning, *et al.*, "A Novel Watermarking Method For Wyner-Ziv Video Coding," in *Intelligent Information Hiding and Multimedia Signal Processing. IHHMSP '08 International Conference on*, 2008, pp. 857-860.
- (8) H. Miao, "Adaptive Image Digital Watermarking Algorithm based on Best Scramble," Master's dissertation. Wuhan, China: South-Central University for Nationalities, 2007.
- (9) C. Harris and M. Stephens, "A combined corner and edge detector," in *Proceedings of Fourth Alvey Vision Conference*, 1988, pp. 147-151.
- (10) A. Aaron, *et al.*, "Transform-domain Wyner-Ziv codec for video," *Visual Communications and Image Processing, January 2004, San Jose, CA, USA, Proceedings of SPIE*, vol. 5308, pp. 520 - 528.
- (11) CMLAB: DSP, Graphic and Network Systems. URL: http://www.cmlab.csie.ntu.edu.tw/new_cml_website/index.php
- (12) M. Goljan, *et al.*, "Sensor noise camera identification: Countering counter-forensics," in *IS&T/SPIE Electronic Imaging*, 2010, pp. 75410S-75410S-12.
- (13) CITRIC Smart Camera Platform and Applications. URL: <http://www.eecs.berkeley.edu/~yang/software/CITRIC/>
- (14) YUV Sequences. URL: <http://trace.eas.asu.edu/yuv/>

- (15) *H.264/AVC Software Coordination.* URL:
<http://iphome.hhi.de/suehring/tml/>