INFORMATION HIDING IN HEVC STANDARD USING ADAPTIVE CODING BLOCK SIZE DECISION

Yiqi Tew, KokSheik Wong

Faculty of Computer Science and Information Technology,
University of Malaya, Kuala Lumpur, Malaysia.
Email: yiqi@siswa.um.edu.my, koksheik@um.edu.my

ABSTRACT

In this work, an information hiding techniques is proposed using the coding block size decision in HEVC. This approach manipulates the CB (coding block) size decision on every coding tree unit to embed information based on the predefined mapping rules. Each CB is forced to assume certain size to encode the external information without significantly compromising perceptual quality. To improve payload, the odd-even based information hiding technique is further deployed by manipulating the nonzero DCT coefficients in certain ranges, in which case each range depends on the CB size. Results suggest that by combining both approaches, improvement is achieved in the terms of payload for the higher bitrate scenario and insignificant degradation in perceptual video quality for the low bitrate scenario.

***Index Terms*—** Information hiding, coding block size, HEVC, H.265, compressed video

1. INTRODUCTION

Encoding video content into a bitstream (i.e., video compression) without compromising perceptual quality for efficient storage and transmission has received much attention in recent years. HEVC (high efficiency video coding), which is the latest video compression standard, is designed to fulfill the broad spectrum of application requirements. However, efficient video compression standard itself only solved part of the problem because nowadays videos appear in large number thanks to the advancement of end-user capturing devices (notably the mobile smart devices) at affordable prices. Thus, there is an urgent need to manage these videos efficiently, including authenticating genuine video, labeling video content with appropriate tag, hyper-linking related videos, etc. [1]. A possible solution to these applications is information hiding, which inserts an external information into a video.

Information hiding techniques are well researched for the previous state-of-the-art compression standards, including, MPEG1/2, and H.264/AVC/SVC [2]. These techniques manipulate selected part of the video coding structure, including intra prediction [3], motion vector [4], DCT (discrete cosine transform) coefficient [5], syntax element [6], etc., to insert information. The application of information hiding includes watermarking which inserts copyright information, stagenography that camouflages secret information, error concealment which aims at improving quality when transmission error occurs, etc.

Since HEVC is recently finalized, to the best of our knowledge, there is still no information hiding technique designed to specifically exploit its coding structure. Hence, in this paper, the coding block size feature in HEVC is utilized for information hiding purposes. Here, the nonzero DCT coefficients are manipulated based on the transform block size in all slices and an information hiding technique is proposed to adaptively manipulate the prediction block size. These techniques have the potential to be further fine-tuned to handle specific applications, including content authentication.

2. OVERVIEW OF HEVC STANDARD

HEVC is the latest video coding standard published by the ITU-T VCEG and ISO/IEC MPEG [7]. The main achievement of the HEVC standard is its significant improvement in compression performance when compared with the previous state-of-the-art standard (i.e., H.264/AVC), with at least 50% reduction in bitrate for producing video of similar perceptual quality [8]. HEVC standard is designed to address essentially all existing applications of H.264/AVC. It achieves two additional major achievements, namely: (a) handle higher video resolution by introducing larger coding block sizes, and (b) capitalize on parallel processing architecture in the video encoder design for boosting the encoding time.

Similar to the existing standards, HEVC divides a video sequence into several GOPs (groups of picture). In each GOP, these pictures are labeled as I (Intra), P (Predicted) and B (Bi-directionally predicted)-slices, depending on the order in which they appear. Fig. 1 illustrates a possible arrangement of I, P and B-slices in a GOP.

HEVC also introduced several new features to achieve higher video compression, such as various coding block sizes,

This paper was written under the project title Unified Scalable Information Hiding and entirely supported by the University of Malaya HIR Grant UM.C/625/1/HIR/MOHE/FCSIT/01, B000001-22001.

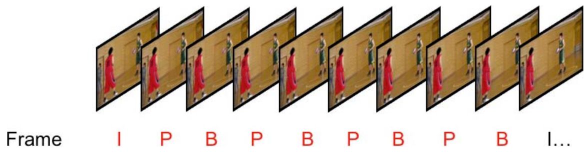


Fig. 1. I, P and B-slices in a GOP of HEVC standard.

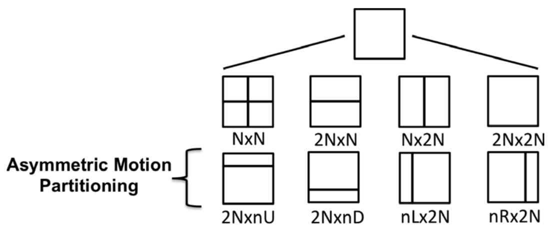


Fig. 2. List of PB sizes.

more intra prediction modes, residual quad tree, sample adaptive offset, tiles and wavefront processing, etc. [7]. Among these features, implementation of the variable prediction and transform block size are exploited in this work for information hiding purposes. Here, the GOP arrangement of IPBPBPBPB is considered to increase the achievable payload while preserving perceptual video quality.

2.1. Partition Size Decision

A slice consists of certain number of CTU (coding tree units) in HEVC. Each CTU consists of some number of CB in the size of 64×64, 32×32, 16×16 and 8×8 pixels. Each 8×8 CB can be further split into 4×4 in the prediction process. The availability of CB in various sizes allows the video encoder to encode each part of the video slice based on its local characteristic (i.e., spatial activity). In particular, a region with high spatial activity (e.g., water waves) requires smaller CB size to precisely capture the variation in pixel intensity values. On the other hand, a smooth region (e.g., plain wall, background, or cloudless sky) can be encoded by using larger CB size. In the HEVC encoder, the prediction and transformation processes utilize the CB structure to perform intra/inter prediction, DCT and quantization. The CB utilized in the prediction and transformation processes are called PB (prediction block) and TB (transform block), respectively.

Previously, H.264 implements CB with square block of size ranging from 16×16 to 4×4 pixels in the I-slices, and it includes rectangular (i.e., non-square) blocks with size of $2N \times N$ and $N \times 2N$ for $N \in \{4, 8\}$ in the P and B-slices. In HEVC, the size of CB can be as large as 64×64 pixels. Specifically, the I-slices can only be coded by using square blocks of various sizes ranging from 64×64 to 4×4 pixels. In addition, the P and B-slices are encoded based on all the possible block arrangements, including $2N \times 2N$, $2N \times N$, $N \times 2N$, $N \times N$ and AMP's (Asymmetry Motion Partition).

Fig. 2 illustrates the basic ideas of AMP, which is a unique partitioning technique introduced in HEVC to divide a PB into two unequal rectangles. The ratio is considered in the

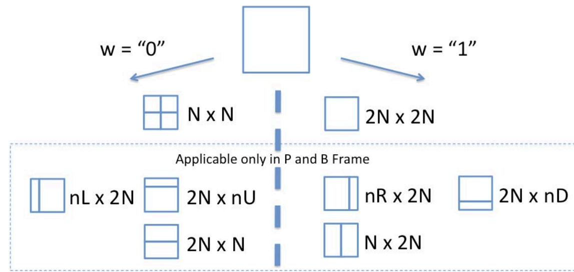


Fig. 3. Two categories of PB size utilized in the proposed information hiding technique.

form of *top : bottom* or *left : right* as depicted in Fig. 2. These four types of AMP are denoted by $2N \times nU$, $2N \times nD$, $2N \times nL$, and $2N \times nR$, respectively. The implementation of AMP in HEVC provides better prediction reference and less bitstream size overhead for PB that contains slight movement at either the up, down, left or right part in the P and B-slices. Note that AMP is only applicable on CTU for block prediction purposes in the HEVC encoder.

3. PROPOSED INFORMATION HIDING TECHNIQUE

The core ideas are: (a) exploit size of PB in HEVC, and; (b) manipulate the nonzero DCT coefficients (referred to as Coeff) based on the TB size in which the coefficient resides. The following subsection describe each technique in detail.

3.1. Manipulation on PB Size Selection

During encoding, the rate distortion optimizer (RDO) calculates the cost function (i.e., a tradeoff between the distortion produced and the number of bits spent) of each possible block size for coding a given PB [9]. RDO selects the PB size with the lowest cost as the final decision to achieve the best compression ratio based on the desired bitrate. Instead of using the size suggested by RDO, the size of PB is forced to be the one representing the information to be embedded based on a predefined mapping rule. A possible implementation is shown in Fig. 3. In particular, different PB size selection technique (i.e., PBSize) is applied when handling I, P and B-slices.

In each I-slice, all the PB are forced to be encoded using 8×8 or 4×4 mode to attain higher payload. In particular, the PBs are forced to be encoded in the respective sizes according to the mapping rules shown in Fig. 3. In this case, the PB size of 8×8 and 4×4 pixels denote '1' and '0', respectively. In P and B-slices, the PB size is decided based on two categories, where one encodes '0' and the other encodes '1'. In particular, category '0' includes $2N \times N$, $2N \times nU$, $nL \times 2N$, and $N \times N$, while category '1' includes $N \times 2N$, $2N \times nD$, $nR \times 2N$, and $2N \times 2N$. These categorizations are summarized in Fig. 3. Note that in I-Frame, the PB size is forced to be 8×8 for $w = 1$, i.e., the same mapping rule for P and B slides is applied without the consideration for $2N \times nU$, $nL \times 2N$, $2N \times nD$ and $nR \times 2N$ depicted in Fig. 3. For instance, if the PB size decided by RDO is 16×8

and $w = 1$, then the proposed technique will force the RDO to recalculate the cost of 8×16, 16×16, and two AMP's (i.e., $2N \times nD$, $nR \times 2N$), then choose the size with the smallest cost as the PB size. For PB with larger block size (e.g., 32×32), it is reasonable to encode it by using some combination of smaller block sizes (e.g., two 32×16, four 16×16, etc.). This approach maintains the video quality at the expense of slight bitstream size expansion. The embedded information can be extracted in the decoding process by examining the PB size based on Fig. 3.

3.2. Manipulation on Nonzero DCT Coefficient

LSB manipulation (i.e., a special case of odd-even embedding) is the most frequently utilized technique for information hiding [10] [11]. This technique allows external information to be conveniently inserted into a digital content (i.e., image, video) without causing significant impact on perceptual quality. Here, the odd-even based technique is implemented on the nonzero DCT coefficients in each CTU. However, the implementation is restricted to AC coefficients in the interval of $[-8, 8] \setminus \{0\}$ and these coefficients are modified based on the size of the TB in which they reside. We empirically observed that modification on this range is adequate to maintain the perceptual quality while providing sufficient payload simultaneously. In particular, AC coefficients in the luminance channel is divided into four categories (i.e., ranges, R) based on the TB size in CTU. Only AC coefficients in $R \setminus \{0\}$ are considered for information hiding. Let Y_c and Y'_c denote the original and modified DCT coefficient values, respectively, and let w be the information bit to be embedded. The embedding process is summarized as follows:

$$Y'_c = \begin{cases} Y_c + (-1)^w & \text{if} \quad \bmod(|Y_c|, 2) \neq w, \\ Y_c & \text{otherwise}, \end{cases} \tag{1}$$

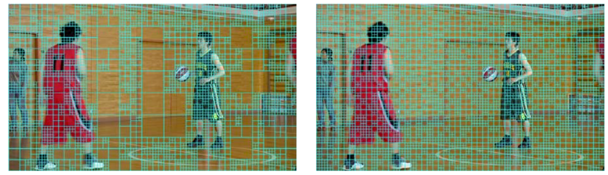
where $Y_c \in R \setminus \{0\}$ for

$$R = \begin{cases} [-8, 8] & \text{if } TB_{size} = 4 \times 4, 8 \times 8 \\ [-6, 6] & \text{if } TB_{size} = 16 \times 16 \\ [-4, 4] & \text{if } TB_{size} = 32 \times 32 \\ [-2, 2] & \text{if } TB_{size} = 64 \times 64 \end{cases} \tag{2}$$

To extract the inserted information, the LSB of each coefficient in the range specified by Eq. (2) is considered.

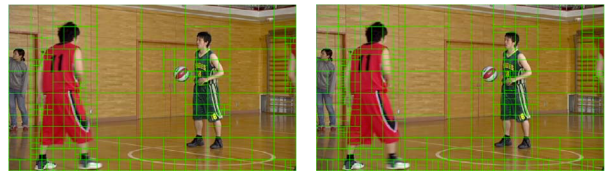
4. PERFORMANCE EVALUATION

The standard test video sequences for HEVC (i.e., *BasketballPass*, *BasketballDrill*, *FourPeople*, *Tennis*) from [12] are considered to evaluate the basic performance of the proposed technique under various bitrates. The HEVC reference model video encoder version HM10 [13] is modified to encode the video sequences while hiding information into it. These video sequences are encoded by using a targeted bitrate ranging from 100kbps to 50Mbps (bits per second). Here, PBSize (proposed in Section 3.1) can be considered as the improved version of [14] in HEVC for comparison purposes. To combine both Coeff and PBSize techniques, the PBSize technique is first invoked, followed by the Coeff technique.



(a) Original I-slice. (b) I-slice with embedded info.

Fig. 4. I-slice PB structure of compressed video at 1Mbps.



(a) Original P-slice. (b) P-slice with embedded info.

Fig. 5. P-slice PB structure of compressed video at 1Mbps.

Fig. 4(a) shows the first I-slice of the original compressed video of *BasketballPass*. The external information is embedded into the same I-slice and the output video is illustrated in Fig. 4(b). Note that almost all large blocks in Fig. 4(a) are decomposed into combinations of smaller blocks to embed information as suggested by Fig. 4(b). It is observed that the changes in block size between 8×8 and 4×4 are affecting the perceptual quality insignificantly. Results suggest that smaller PB size can be implemented for all I-slices to achieve higher payload while maintaining video quality because smaller PB size generally results in better video quality. Similar conclusions can be drawn for the P-slices. Fig. 5(a) and 5(b) show the original and modified P-slices, respectively. It is observed that some of the CU's are replaced by combinations of two (non-square) rectangles, including the AMP's which are not available in H.264.

To quantify the effect of information hiding on perceptual image quality, the PSNR (peak signal-to-noise ratio) and SSIM (structural similarity index) [15] are computed using the average value over the video sequence, and the results are recorded in Table 1. Quality of the original compressed video sequences are also recorded for reference purposes. Here, the results of the implemented techniques (i.e., Coeff, PBSize and the combination of both) are collected for the bitrate ranging from 100kbps to 50Mbps.

First, it is observed that at low bitrate (i.e., 100kbps), regardless of the video sequence (and hence the resolution), PBSize consistently offers higher payload when compared to Coeff. It is because at low bitrate, most of the coefficients are quantized to zero, while the numbers of CU are relatively consistent regardless of the bitrate. This trend is particularly obvious for video of high resolution (i.e., *Tennis*). On the other hand, as bitrate increases, the opposite trend is observed, i.e., Coeff offers significantly higher payload when compared to PBSize. This justifies the combination of both PBSize and Coeff techniques to ensure the availability of payload for information hiding purposes.

Secondly, the perceptual video quality of the video manipulated by PBSize is, in general, lower than that of Coeff,

Table 1. Video quality and payload of the proposed techniques for various bitrates.

Video Sequences	Bitrate	Original		Coeff			PBSize			Coeff + PBSize		
		PSNR	SSIM	PSNR	SSIM	bits/sec	PSNR	SSIM	bits/sec	PSNR	SSIM	bits/sec
BasketballPass (416 × 240)	100k	28.50	0.789	28.17	0.780	5571	26.25	0.726	7752	26.10	0.722	11779
	500k	36.18	0.931	35.95	0.929	49992	35.23	0.959	13878	34.99	0.912	58150
	1M	39.87	0.965	39.58	0.963	116922	39.11	0.994	17126	38.82	0.956	127241
BasketballDrill (832 × 480)	100k	25.62	0.715	25.51	0.710	5452	23.08	0.624	25993	23.08	0.625	24637
	500k	32.43	0.912	32.24	0.908	35517	30.78	0.885	30869	30.78	0.881	54378
	1M	35.24	0.954	35.04	0.951	86693	34.15	0.940	38548	34.32	0.937	112633
	5M	42.48	0.994	42.21	0.993	594515	41.77	0.993	59230	42.06	0.992	656513
	10M	45.33	0.996	45.08	0.996	1327068	44.78	0.996	69906	45.05	0.996	1420820
FourPeople (1280 × 720)	100k	27.13	0.835	26.94	0.828	8559	22.47	0.730	61827	22.33	0.725	51712
	500k	36.41	0.983	36.18	0.981	48901	31.63	0.932	61883	31.34	0.927	79372
	1M	39.83	0.992	39.64	0.992	110236	38.57	0.989	66571	38.38	0.988	140880
	5M	44.18	0.997	44.07	0.997	608916	43.96	0.997	94786	43.85	0.997	683097
	10M	45.31	0.998	45.15	0.998	1297460	45.13	0.998	116141	44.98	0.998	1411533
Tennis (1920 × 1080)	100k	27.18	0.782	27.21	0.781	2347	21.84	0.566	144466	22.06	0.578	119117
	500k	32.23	0.907	31.89	0.902	34627	24.02	0.700	146326	24.06	0.698	142426
	1M	36.06	0.953	35.92	0.951	82913	30.62	0.879	147232	30.48	0.876	164993
	5M	41.25	0.992	41.14	0.991	543583	40.30	0.987	188189	40.19	0.987	600679
	10M	42.79	0.996	42.69	0.996	1183906	42.28	0.995	223226	42.18	0.994	1223809
	50M	46.15	0.999	45.81	0.998	6945862	45.84	0.998	425027	45.55	0.998	7190970

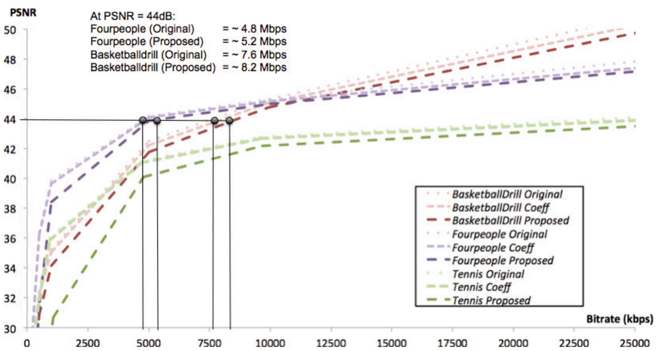


Fig. 6. Rate distortion curve for the original compressed video, coeff technique and proposed combined technique.

especially at lower bitrates. As bitrate increases, the quality attained by both the PBSize and Coeff techniques are similar. These observations are also applicable to the combined technique, where the distortion is mainly caused by PBSize. These results also suggest the bitrate from which the performance of the HEVC encoder starts to saturate for purposes of information hiding when a given a video / resolution. For example, in the case of *FourPeople*, when the bitrate is greater than 1Mbps, both PBSize and Coeff (as well as the combined technique) are equally viable for information hiding. Similar, these results may also suggest the bitrate from which the performance of the HEVC encoder starts to saturate for encoding purposes, i.e., determining the maximum bitrate when given a video / resolution, and we shall research in this direction as our future work.

Next, to visualize the performance of the proposed combined technique, part of the results in Table 1 are translated into Fig. 6. The graphs suggest that by implementing the combined technique at higher bitrate, PSNR decreases with a magnitude of < 3dB, while the perceptual quality of all video sequences are maintained as suggested by the SSIM results. From the perspective of bitrate, to achieve the PSNR of 44dB, the proposed combined technique requires an additional 7.9% and 8.3% of bitrates in *BasketballDrill* and *FourPeople*, respectively, when compared to their original counterparts (i.e., compressed videos). These results suggest that the proposed method has negligible impact on the bitrate when considering the amount of payload that can be hidden.

All in all, the video produced by the combined technique is of slightly lower quality than that by Coeff embedding itself. However, the quality improves when the bitrate increases. Naturally, the payload in the combined technique is higher than each individual technique. Therefore, the combined technique can be considered to achieve higher payload, with acceptable perceptual quality. For coding complexity, based on the information to be hidden, the encoder evaluates only the selected PB sizes instead of every possible PB sizes, which reduces the encoding time up to 20% in cases of higher bitrates (e.g., *BasketballDrill* at 1.25Mbps). Nonetheless, due to the variation of video content and processor capability, more accurate evaluations will be carried out as our future work to study the reduction in coding complexity.

5. CONCLUSION

An information hiding technique was proposed to insert external information in HEVC compressed video. This technique encodes information by manipulating the PB size. In addition, the proposed technique was combined with odd-even embedding using nonzero coefficients belonging to selected ranges of value. The ranges, in turn, depend on the TB size to achieve similar perceptual quality as the original video. Simulation results suggest that the proposed PBSize technique maintains the perceptual quality of the video for higher bitrate scenarios and improves the conventional odd-even embedding in terms of payload. As bitrate increases, PBsize manipulation becomes negligible in terms of quality degradation and capacity. However, in the case of lower bitrates, PBsize offers significantly in terms of payload at the expense of quality degradation.

For future work, we aim to utilize other features in HEVC as the new opportunities for information hiding, where these techniques can be further improved by applying matrix encoding [16].

6. REFERENCES

[1] Y. Tew and K. Wong, "An overview of information hiding in H.264/AVC compressed video," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 24, no. 2, pp. 305–319, Jan. 2014.

[2] T. Wiegand, G. J. Sullivan, G. Bjontegaard, and A. Luthra, "Overview of the H.264/AVC video coding standard," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 13, no. 7, pp. 560–576, Jul. 2003.

[3] D.-W. Kim, Y.-G. Choi, H.-S. Kim, J.-S. Yoo, H.-J. Choi, and Y.-H. Seo, "The problems in digital watermarking into intra-frames of H.264/AVC," *Image Vision Comput.*, vol. 28, no. 8, pp. 1220–1228, Aug. 2010.

[4] Y. Guo and F. Pan, "Information hiding for H.264 in video stream switching application," *Proc. IEEE Int. Conf. Inform. Theory Inform. Security*, pp. 419–421, Dec. 2010.

[5] T.-J. Lin, K.-L. Chung, P.-C. Chang, Y.-H. Huang, H.-Y. M. Liao, and C.-Y. Fang, "An improved DCT-based perturbation scheme for high capacity data hiding in H.264/AVC intra frames," *J. Syst. Software*, vol. 86, pp. 604–614, Mar. 2013.

[6] J. Li, H. Liu, J. Huang, and Y. Zhang, "A robust watermarking scheme for H.264," *Int. Workshop on Digital Forensic and Watermarking*, vol. LNCS 5450, pp. 1–15, 2009.

[7] G. J. Sullivan, J.-R. Ohm, W.-J Han, and T. Wiegand, "Overview of the High Efficiency Video Coding (HEVC) standard," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 22, no. 12, pp. 1649–1668, Dec. 2012.

[8] J.-R. Ohm, G. J. Sullivan, H. Schwarz, T. K. Tan, and T. Wiegand, "Comparison of the coding efficiency of video coding standards - including High Efficiency Video Coding (HEVC)," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 22, no. 12, pp. 1669–1684, Dec. 2012.

[9] G. J. Sullivan, and T. Wiegand, "Rate-distortion optimization for video compression," *IEEE Signal Process. Mag.*, vol. 15, pp. 74–90, Nov. 1998.

[10] R.-Z. Wang, C.-F. Lin, and J.-C. Lin, "Image hiding by optimal LSB substitution and genetic algorithm," *Pattern Recognition*, vol. 34, no. 3, pp. 671–683, Mar. 2001.

[11] C.-K. Chan and L. Cheng, "Hiding data in images by simple LSB substitution," *Pattern Recognition*, vol. 37, no. 3, pp. 469–474, Mar. 2004.

[12] Institut für Informationsverarbeitung. (2013). *YUV sequences repository* [Online]. Available FTP: ftp://hvc:US88Hula@ftp.tnt.uni-hannover.de/testsequences

[13] Fraunhofer Heinrich Hertz Institute. (2013) *High Efficiency Video Coding: HEVC software repository* [Online]. Available: https://hevc.hhi.fraunhofer.de

[14] S. Kapotas and A. Skodras, "A new data hiding scheme for scene change detection in H.264 encoded video sequences," *IEEE Int. Conference on Multimedia and Expo*, pp. 277–280, Apr. 2008.

[15] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: from error visibility to structural similarity," *IEEE Trans. Image Process.*, vol. 13, no. 4, pp. 600–612, Apr. 2004.

[16] J. Fridrich and D. Soukal, "Matrix embedding for large payloads," *IEEE Trans. Inf. Forensics Securityn*, vol. 1, no. 3, pp. 390–395, Sep. 2006.