COMPOUND IMAGE COMPRESSION USING LOSSLESS AND LOSSY LZMA IN HEVC

Cuiling Lan[1], Jizheng Xu[1], Wenjun Zeng[1], Feng Wu[2]

[1]Microsoft Research Asia, [2]University of Science and Technology of China
{culan, jzxu, wezeng}@microsoft.com, fengwu@ustc.edu.cn

ABSTRACT

We present a compound image compression scheme based on the dictionary-based Lempel-Ziv-Markov chain algorithm (LZMA), under the framework of High Efficiency Video Coding (HEVC). Through matching strings from the sliding window dictionary, LZMA exploits the characteristics of the repeated patterns over the text and graphics regions of compound images, and represents them compactly. To obtain high compression efficiency even for noisy text and graphics contents, we have modified LZMA to support both lossless and lossy compression. We develop and treat it as a new intramode of HEVC. Experimental results show that the proposed scheme achieves significant coding gains for compound image compression. Thanks to the introduction of the lossy LZMA, the compression performance for noisy compound images is improved for more than 5dB in terms of PSNR in comparison with the lossless LZMA scheme.

Index Terms **—** Compound image compression, dictionary-based algorithm, lossless and lossy LZMA

1. INTRODUCTION

Compound image is a combination of the natural image, text and graphics contents. Representation and transmission of data in the form of compound images is becoming more and more popular especially with the rapid development of electronic devices, computers and networks. Compound images can either be directly generated by computers, such as the documents, slides, webpages that displayed on screen, or be captured by the electronic devices, such as cameras and scanners. Image noises are usually generated as a by-product of image capturing [1]. Transmission or storage of compound images favors many applications such as screen sharing, remote desktop control [2], cloud computing [3][4], electronic record storage. How to efficiently compress compound images is still a hot and valuable research topic.

Different from natural images, contents in the text and graphics regions of compound image are usually not continuous. They present abrupt changes, sharp edges, and high contrast. Transform based compression methods such as those in H.264/MPEG-4 AVC [5], HEVC [6] are not very efficient for compressing such contents since the transform cannot compact the energy to low-frequency components but may spread it to high frequency bands. To improve the coding efficiency over the text and graphics regions, some approaches have been proposed in the past decade. Based on the tolerance to distortion, they can be classified into lossless approaches [7-11] and lossy approaches [11-16]. In considering the basic ideas behind, they can be categorized into four classes: image-based approaches, layer-based approaches [15], block-based approaches [7, 11, 16] and dictionary-based approaches [8-13]. A detailed review of the first three categories of approaches can be found in [16].

In this paper, we focus on dictionary-based approaches. The idea of using dictionary-based compression has a long history in the compression fields. The basic idea is to exploit the repeated patterns in the data and represent them in compact ways which require very few bits. There are two categories of dictionary-based approaches that have been developed independently.

Lossy dictionary-based approaches: Vector quantization, also referred to as "pattern matching quantization", divides the data into blocks (or vectors) and represents them by similar ones in a dictionary [17]. In general, the dictionary can be generated and maintained on the fly to have the adaptability to image contents. The multidimensional multiscale parser (MMP) coding algorithm [12, 13] belongs to this category. It uses approximate 2D pattern matching with adaptive multiscale dictionaries for compression. Significant gain is achieved for compound image compression. Due to the nature of lossy compression, it has high probability of finding approximate matches and is tolerant to noises. However, the matching is constrained to be block-wise and it lacks the flexibility in the selection of different match combinations, e.g., matching lengths, matching start positions. Moreover, the lossless encoding alternative is inefficient under such framework due to the constraint on matching of the entire block or subblock. Besides, the maintenance of the dictionaries for different block sizes is complicated.

Lossless dictionary-based approaches: Lempel-Ziv LZ77 [18] and LZ78 [19] are two lossless data compression algorithms which are the bases of many variations including LZW, LZSS, LZMA [20] and others. They are dictionary based coders for sequential data compression, where LZ77 maintains the dictionary by a sliding window while LZ78 uses the explicitly constructed dictionary. The common idea is to represent current data with previously occurred data by the compact description of "*length-distance*", telling that the next *length* symbols is exactly equal to the symbols that are in *distance* behind it in the input data stream. LZ78 based method LZW is used in GIF images. One of the most popular LZ77 based compression methods DEFLATE is used in PKZIP, Gzip and PNG for file or image compression. LZMA, as an advanced LZ77-based compression algorithm, is the default and general compression method in 7-Zip. Because of the one dimensional matching nature, LZ-based algorithms have more flexibility in the selections of matching lengths and start positions of the match. A combination of some lossless LZ-based approaches, such as Gzip and PNG, with H.264/AVC or HEVC was proposed in [8-10] for compound image compression. Making use of LZMA in HEVC for compound image compression was proposed in [11], which achieves significant coding gain. However, the lossless dictionary-based algorithms are not very efficient in compressing noisy compound images.

Our contribution in this paper is that we have designed a unified framework which can support both the lossless and lossy dictionary-based compression. We achieve this by extending the LZMA algorithm to support lossy compression. Different from the original LZMA which uses the coding rate cost as the criteria to determine the optimal encoding matches, we take the rate-distortion cost as the optimization target. Both exact matching and approximate matching are supported. We design the extended LZMA as a new intramode in HEVC and use rate-distortion optimization to select the best coding manner among lossy LZMA, lossless LZMA and the original HEVC methods. Experimental results show that the coding performance for noisy compound images can be greatly improved thanks to the introducing of lossy LZMA. It outperforms the compound image compression scheme as in [11] which uses lossless LZMA often more than 5dB in terms of PSNR.

The rest of this paper is organized as follows. Section 2 gives an overview of the proposed scheme and the HEVC framework. Section 3 describes the proposed unified LZMA mode in detail. Experimental results are presented in Section 4. We draw the conclusions in Section 5.

2. OVERVIEW OF PROPOSED SCHEME AND HEVC

We construct a compound image compression scheme by modifying the LZMA algorithm to be a new intramode that supports both lossy and lossless dictionary-based compression, referred to as LZMA mode, and implementing it into HEVC. This mode is applied on various coding unit sizes to adapt to the characteristics of the coding contents. In general, text and graphics contents prefer to be encoded by LZMA modes while natural image contents prefer to be encoded by the original modes of HEVC. Rate-distortion optimization as used in HEVC is adopted to select the best mode and coding unit size, i.e., selecting the one with the minimum rate-distortion cost.

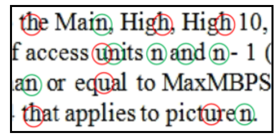


Fig. 1 Repeated patterns in text regions.

Design of a unified lossless and lossy LZMA mode for text and graphics content compression is inspired by the following considerations. Firstly, text and graphics contents usually present many repeated patterns as shown in the examples in Fig. 1 that are marked by circles. Dictionary-based strategy can exploit these local and non-local correlations by representing current strings with previously appeared strings in the dictionary. Note that a pixel value (on luminance) is the basic unit, i.e., a symbol in a string. Secondly, as a dictionary-based method, the lossless compression algorithm LZMA is mature and has high compression efficiency. Thirdly, the existence of image noise may break the long matching string to many short strings, which increases the encoding bits and thus attenuates the compression performance. The use of approximate matching, i.e., lossy compression, can be tolerant to noises and thus alleviates this problem. Therefore, we propose the use of both lossy LZMA and lossless LZMA to compress compound images. They share the same LZ-based framework and thus we call this a unified framework.

We take HEVC as the benchmark to develop our scheme considering its high coding performance. HEVC was designed to reduce the bitrate by half with comparable image quality with respect to H.264/AVC [21]. It still adopts the hybrid block-based video coding framework which uses inter/intra prediction and 2D transform. A picture is uniformly split to coding-tree unit (CTU) with the maximum size of 64x64 pixels. A CTU can be further split into smaller coding units (CUs) with the quadtree structure. Each CU contains one or more prediction units (PUs). A CU is associated with a transform quadtree which transforms the prediction residues. Fig. 2 shows an illustrative example.

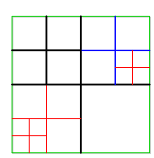


Fig. 2. Example of the HEVC partition on a CTU. Black lines illustrate the dividing of CTU to hierarchal smaller CUs. Blue lines illustrate the dividing of a CU to PUs. Red lines illustrate the partitioning of TUs.

3. UNIFIED LOSSLESS AND LOSSY LZMA

3.1. Review of LZMA

LZMA has superior compression performance among the LZ-based algorithms. It has a well-designed set of context models for encoding the representations using a range encoder. It uses the dynamic programming algorithm for the optimized encoding selections [20, 22].

LZMA stores certain amount of the most recent data within a sliding window in a buffer, which is the dictionary for searching matches. To enable the fast search, it uses three hash tables for hashing two, three and four bytes (*hash4*) respectively. For *hash4* table, a binary tree is maintained for each node, which facilitates the tracing down in search for longer matches [23]. The default maximum tree depth is 128, corresponding to a match of 128 symbols. The dictionary and hash tables are online updated along the processing of every symbol.

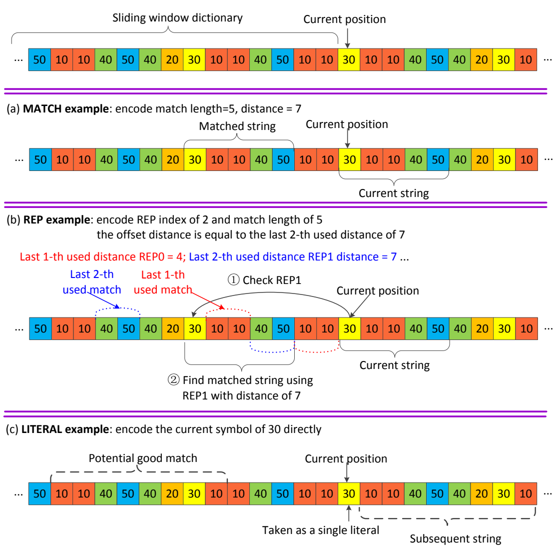


Fig. 3. Examples for illustration of possible encoding options: *MATCH*, *REP*, *LITERAL*.

For a string, there is much freedom in selecting the matches, such as the options on which matches to select and how long the match length is. LZMA has several possible encoding options for the current position string, i.e., *MATCH*, *LITERAL* and *REP0* - *REP3*. (1) *MATCH* indicates a typical match described by the matching length and distance with an example shown in Fig. 3 (a). Note that there can be several match manners with different offset distances. In Fig. 3, the nearest symbols on the left of the current position is in the sliding window dictionary. (2) For *REPi* case (i = 0, 1, 2, 3), the offset distance is equal to the latest (i+1)-th used distance. Thus it requires very few bits to indicate the offset distance but still needs to describe the matching length. An example is shown in Fig. 3 (b). (3) LITERAL means it is encoded as a single byte. Fig. 3(c) shows an example. In Fig. 3(c), encoding the current symbol as a LITERAL may not be the locally optimal. But it may result in longer potential matches for the following strings. Smart selection of the combinations for the encoding options is important for its superior compression performance.

For lossless compression, it is expected to select the encodings that minimize the global encoding rate. For a portion of a string, there are abundant sequential combinations of the encoding options. LZMA uses dynamic programming to approximately find the optimal combinations. The size of the portion to be processed in the dynamic programming algorithm is determined by the found maximum match length for the current position string from the *MATCH*s and *REP0* – *REP3*. This size can be extended during the solving of the sub-problems while it is limited by an upper bound, e.g., 4096. The dynamic programming algorithm stops when the sub-problems within the range are all solved or when a match with length longer than a predefined length (e.g., 128) is found at any point in the range. The solution in terms of the concatenation of a series of encoding options is the output for this portion of the string. These representations are then encoded using a context-based range encoder. After that, a new dynamic programming instance is started to process the hereafter strings. This procedure continues until the entire input stream is encoded.

3.2. Proposed unified lossless and lossy LZMA

We design LZMA modes to work on intra coding units (CUs) with block sizes ranging from 64x64 to 8x8 on the luminance components in HEVC. A flag at the CU level is coded to identify whether the CU is coded with LZMA mode or not. Note that it does not require overhead bit for distinguishing the lossless and lossy LZMA. We set the prediction direction of the LZMA mode as DC and the transform depth as 0 in the syntax. For the 64x64 CU, we set the transform depth as 1 in the syntax to be consistent with the available smallest transform depth of 1.

In the LZMA mode, for a coding block, we scan the pixel luminance values in the raster order to get a one dimensional string and concatenate it to the end of the LZMA input stream for compression. We online update the dictionary by the contents coded by the LZMA mode. A dictionary with larger size will result in higher coding performance. We set it to 32K bytes to save the memory and computation time.

For the input string of a block, LZMA uses the dynamic programming algorithm to determine the encoding options by minimizing the estimated price for every portion of the string. The output is encoded using the context-based range encoder. A new dynamic programming instance starts at the next symbol following the previous portion of the string. This procedure continues until the entire input string of this block is processed. We extend the lossless LZMA to unified lossless and lossy LZMA by redefining the price for the i-th portion of the string as

$$P_i = D_i + \lambda \times R_i \quad (1)$$

where λ is the Lagrange multiplier as that in HEVC. R_i indicates the estimated bits while D_i represents the distortion. For lossless LZMA, D_i is always zero since only the exact matches are allowed. For the proposed unified LZMA, approximate matches during the dynamic programming decision procedure is allowed. A symbol a is said to have approximate match with symbol b if their absolute difference value is not larger than a threshold $thre$ as

$$|a-b| \le thre$$
$$thre = \begin{cases} 0.5Q_{step}, & \text{for lossy LZMA} \\ 0, & \text{for lossless LZMA} \end{cases} \quad (2)$$

Q_{step} represents the quantization step in HEVC given the quantization parameter (QP). The lossless LZMA can also be represented as above with the threshold being zero. Note that it is the reconstructed symbols that are used for the update of the dictionary in the lossy case.

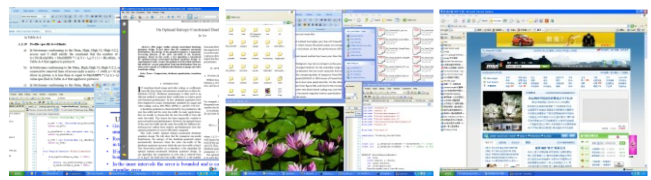
Fast search using hashing is still utilized for looking for the exact matches in lossless LZMA. But using hash may miss some approximately matched candidates for lossy LZMA. This can be remedied by using distance-based hashing [24] fast search algorithm or performing full search to find approximate matches. For simplicity, we use full search in this paper and leave the implementation of the fast algorithm as future work. In addition, for the lossy LZMA, during the dynamic programming procedure, when the optimal encoding options are not finally determined for a portion of string, the reconstruction symbols within this portion of string are unknown, thus we do not use the symbols within this portion of string as reference symbols for matching since the calculated distortion is not accurate.

Recovery to the former states is required when the better mode is not LZMA during the rate-distortion optimization process. Thus, the context models, entropy coder states and the pointers for the dictionary are stored to enable the recovery. For the hash table recovery, to avoid the time consuming memory copying operation, we do not update the hashes during the rate-distortion optimization procedure and only update them in the real coding where the best coding modes and partitions have been determined for the entire CTU. Full search over the local symbols which have not been hashed is performed to remedy the loss due to delayed hash updating.

In the block based coding structure, when we encode a block, we do not know what the reconstructed contents would be for the next block. Thus, we cannot involve symbols belonging to the next block to avoid mismatch. In addition, the hashes, such as the *hash4* that are linked with the binary trees, should be updated only when there are enough known look-ahead symbols. For instance, for the last 127 symbols (maximum binary tree depth being 128) of the coding block, when there is no decided LZMA block ahead, the *hash4* will not be updated for those symbols. Otherwise, the binary trees will be destroyed due to the uncertainty of some look-ahead symbols. As a remedy, we can delay the updating until there are enough available symbols.

4. EXPERIMENTAL RESULTS

We implemented the proposed unified lossless and lossy LZMA in the HEVC test model (HM) software HM2.0 [25]. In the experiments, we set the quantization parameters (QPs) to be 17 to 47 with a step of 10 to have wider bitrate ranges. We use the default setting of the all-intra configuration in HM except that the bitdepth is set to be 8 and the deblock filtering is disabled.



(a) Files (b) Various (c) Webpage

Fig. 4. Set 1 images (1280x1024). The other two sets of images are the same pictures but with noises.

We use three categories of screen images to evaluate the performances. (1) *Set 1 images*, including *Files*, *Various* and *Webpage*, are compound images from [11] including computer generated text and graphics contents as shown in Fig. 4. They are noise-free images. (2) We know that image captured by electronic devices are usually polluted with noises [1]. We mimic this by adding random uniform distributed noises to those screen images with the lower bound and upper bound of noises being -2.5 and 2.5. We take them as *Set 2 images* and name them as *Files_noises*, *Various_noises*, *Webpage_noises*. (3) It is common to display the compressed compound images on screen, where compression noises are embedded. Applications such as remote desktop will inevitably display the already compressed images and need to recompress such images. We take the compound images compressed by HEVC with quantization parameter (QP) 27 as *Set 3 images* with names as *Files_rec*, *Various_rec*, *Webpage_rec*.

Our proposed scheme encodes a block by both lossless LZMA and lossy LZMA. We refer to it as *HM+UnifiedLZMA*. We take two other schemes, *HM* and *HM+LosslessLZMA* [11], for performance comparision. *HM* denotes the original HEVC scheme. *HM+LosslessLZMA* indicates the scheme where only the lossless LZMA mode is integrated into HEVC. In addition, we also show the performance of our scheme which only integrates the proposed lossy LZMA into HEVC, referred to as *HM+LossyLZMA*.

Fig. 5, Fig. 6 and Fig. 7 show the rate-distortion curves of those schemes for the three different sets of compound images respectively. Fig. 5 shows the results for the *Set 1 images*, which are noise-free compound images. The performance after integrating the LZMA mode is greatly improved in comparison with HEVC. *HM+UnifiedLZMA* and *HM+LosslessLZMA* provides similar performance for this set of images since the approximate match is seldom needed for noise-free contents. It indicates the lossless LZMA is good enough to handle noise-free contents.

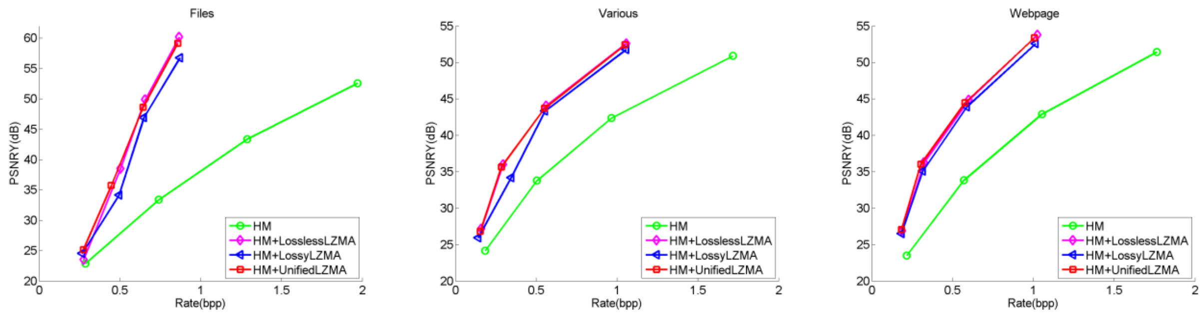


Fig. 5. Rate-distortion performance for *Set 1 images*: captured desktop compound images.

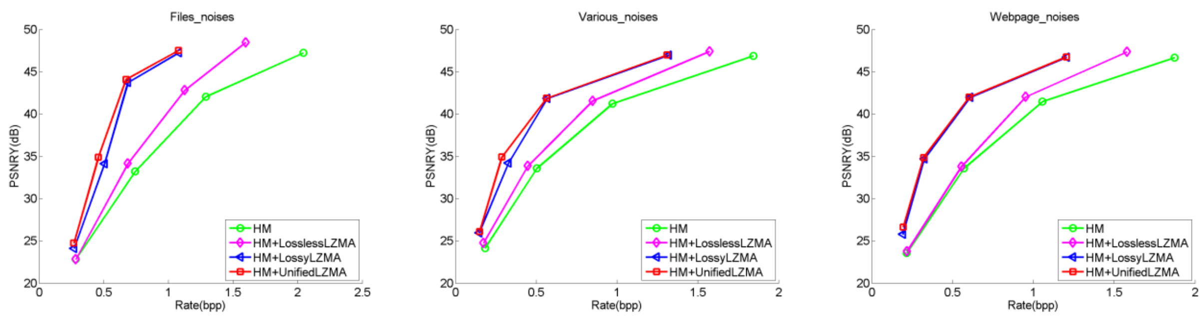


Fig. 6. Rate-distortion performance for *Set 2 images*: compound images with noises.

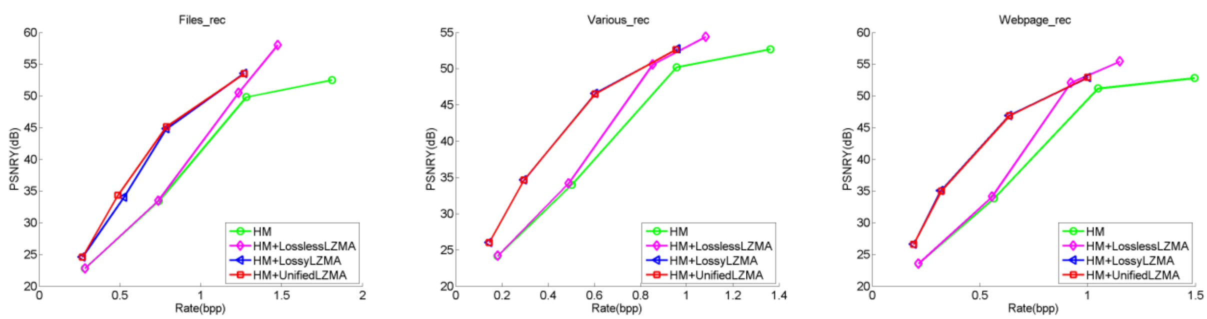


Fig. 7. Rate-distortion performance for *Set 3 images*: compressed compound images containing compression noises.

Fig. 6 and Fig. 7 show the results on noisy compound image sets, *Set 2 images* and *Set 3 images* respectively. The proposed scheme *HM+UnifiedLZMA* outperforms the lossless LZMA scheme *HM+LosslessLZMA* [11] greatly, often more than 5dB. Due to the tight constraint on the exact match in the lossless LZMA mode, *HM+LosslessLZMA* would lose many opportunities for finding matches with long length and thus has interior performance. In contrast, lossy LZMA relaxes the matching constraint and uses the rate-distortion optimization criteria to search the optimal matches. Checking both lossy LZMA and lossless LZMA as in *HM+UnifiedLZMA* gives a little better performance than *HM+LossyLZMA* since the newly introduced lossy LZMA can handle well most of the text/graphics contents. Note that we integrate the proposed methods as intramode. To validate their efficiency, we only show the performance on intra frame/image coding. Actually, they can improve the performance on compound video coding too.

Intuitively, lossy LZMA should give similar performance as lossless LZMA when dynamic programming is utilized to determine the optimal encodings. In fact, lossy LZMA is a little weaker than lossless LZMA as observed in Fig. 5. This is because the symbols within the same portion of string in the dynamic programming instance are not taken as the reference symbols in the lossy LZMA implementation, where the reconstruction values are available only until the end of the dynamic programming procedure.

The decoding complexity of the proposed scheme is lower than *HM* since LZMA modes only require to look up the dictionary without prediction and inverse transform. The current encoding complexity is much higher than HM. Fortunately, there is large room for complexity reduction which is left as the future work. Firstly, we can skip the checking of LZMA modes for some regions based on some simple measures of the content characteristics, e.g., the gradient of a block. Secondly, distance based hashing for fast approximate matching can be utilized to reduce the search complexity for lossy LZMA.

5. CONCLUSION

This paper presents a unified lossless and lossy LZMA dictionary-based scheme for compound image compression. Approximate match in the lossy LZMA makes the proposed scheme tolerant to noises. We use the rate-distortion cost as the metric to optimally determine the matches by dynamic programming in LZMA. Experimental results show that the proposed scheme can often provide more than 5dB coding gain than the scheme using lossless LZMA for coding compound images with noises.

6. REFERENCES

[1] http://en.wikipedia.org/wiki/Image_noise.

[2] H. Shen, Y. Lu, F. Wu and S. Li, "Low-cost real-time screen sharing to multiple clients," in Proc. ICME 2010, Singapore, pp. 980-985, July 2010.

[3] M. Armbrust et.al, "Above the clouds: a Berkeley view of cloud computing," Technical Report No. UCB/EECS-2009-28.

[4] T. Lin and S. Wang, "Cloudlet-screen computing: a multi-core-based,cloud-computing-oriented, traditional-computing-compatible paralled computing paradigm for the masses," in Proc. ICME, pp. 1805-1808, Jul. 2009.

[5] Advanced Video Coding for Generic Audiovisual services, ITU-T Recommendation H.264 - ISO/IEC 14496-10 (AVC), ITU-T and ISO/IEC JTC 1, 2003.

[6] 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. 1648-1667, Dec. 2012.

[7] T. Lin and P. Hao, "Compound image compression for real-time computer screen image transmission," IEEE Trans. Image Process., vol. 14, no. 7, pp. 993–1005, Jul. 2005.

[8] S. Wang and T. Lin, "A unified LZ and hybrid coding for compound image partial-lossless compression," in Proc. ICISP, pp. 99-103, Oct. 2009.

[9] S. Wang and T. Lin, "United coding for compound image compression," in Proc. ICISP, pp. 566-570, Oct. 2010.

[10] T. Lin, P. Zhang, S. Wang, K. Zhou and X. Chen, "Mixed Chroma Sampling-rate high efficiency video coding for full-chroma screen content," IEEE Trans. On Circuits Syst. Video Technol., vol. 23, no. 1, Jan. 2013.

[11] C. Lan, J. Xu and F. Wu, "Compression of compound images by combining several strategies," in Proc. MMSP, Hangzhou, China, pp. 1-6, Oct. 2011.

[12] M. B. de Carvalho, E. A. B. da Silva and W. Finamore, "Multidimensional signal compression using multiscale recurrent patterns," Elsevier Signal Processing, vol. 82, pp. 1559-1580, Nov. 2002.

[13] N. C. Francisco, N. M. M. Rodrigues, E. A. B. da Silva, M. B. de Carvalho, S. M. M. de Faria and V. M. M. da Silva, "Scanned compound document encoding using multiscale recurrent patterns," IEEE Trans. on IP, vol. 19, no. 10, pp. 2712-2724, Oct. 2010.

[14] A. Zaghetto and R. L. de Queiroz, "Segmentation-driven compound document coding based on H.264/AVC-INTRA,"IEEE Trans. Circuits Syst. Video Technol., vol. 16, pp. 1755–1760, Jul. 2007.

[15] R. L. de Queiroz, "Compressing compound documents," in The Document and Image Compression Handbook, M. Barni, Ed. New York: Marcel-Dekker, 2005.

[16] C. Lan, G. Shi and F. Wu, "Compress compound images in H.264/MPEG-4 AVC by exploiting spatial correlation," IEEE Trans. on IP, vol. 19, no. 4, pp. 946-957, Apr. 2010.

[17] A. Gersho and R. Gray, Vector Quantization and Signal Compression. New York: Springer, 1991.

[18] J. Ziv and A. Lempel, "A universal algorithm for sequential data compression," IEEE Trans. On Inform. Theory, vol. 23, no. 3, pp. 337-343, May 1977.

[19] J. Ziv and A. Lempel, "Compression of individual sequence via variable-rate coding," IEEE Trans. On Inform. Theory, vol 24, no. 5, pp. 530-536, Sep. 1978.

[20] http://en.wikipedia.org/wiki/Lempel%E2%80%93Ziv%E2%80%93Markov_chain_algorithm

[21] B. Li, G. J. Sullivan and J. Xu, "Compression performance of high efficiency video coding (HEVC) working draft 4," in Proc. ISCAS, Seoul, Korea, pp. 886-889, May 2012.

[22] LZMA SDK: http://www.7-zip.org/sdk.html.

[23] D. Salomon and G. Motta, *Handbook of data compression*. Springer, 2010.

[24] V. Athitsos, M. Potamias, P. Papapetrou and G. Kollios, "Nearest neighbor retrieval using distance-based hashing," in Proc. ICDE, Cancun, Mexico, pp. 327-336, April 2008.

[25] http://hevc.kw.bbc.co.uk/trac/browser/branches/HM-2.0-dev.