A SYSTEM FOR EVALUATING PERFORMANCE OF VIDEO CODECS IN IMAGE COMPRESSION

Marek Parfieniuk, Andrzej Szpakowicz

Faculty of Computer Science, Bialystok University of Technology, Białystok, Poland

Abstract: This article presents a system for evaluating how well do video codecs perform in still image compression. As it is time consuming and tedious to directly configure and run the reference H.265 (HEVC) and H.264 (MPEG4 AVC) codecs, our tool makes experimenting easier and more efficient. The system provides a graphical user interface for conveniently and quickly preparing input files, for repeatedly running video codecs, and for analysing output data. In the background, there are algorithms for forming video sequences and configuration files, for batch executing an encoder, and for extracting information from output files. Our research was mainly aimed at speeding-up experiments related to compressing an image represented as a video sequence composed of its polyphase components. The system has allowed us to experimentally verify that, like its predecessor, the H.264, the HEVC standard needs modified entropy codes for effectively processing decimated images.

Keywords: video, image, coding, compression, codec, HEVC, H.264, MPEG4 AVC, polyphase, decomposition

1. Introduction

No so many years ago, it has been verified that video codecs can compress images better than advanced algorithms for coding still pictures, like JPEG2000 [1,2,3,6,7,10,11,19,22]. This has been shown for both the H.264 (MPEG4 AVC: Advanced Video Coding) standard [15], which currently dominates the multimedia market as the basis of the DVB-T terrestrial television and Blu-Ray discs, and the H.265 (HEVC: High Efficiency Video Coding) standard [17,18,21], which should be implemented soon for applications that involve UltraHD video. As a result, the HEVC standard has been extended so that it includes the Still Picture Profile [11], and a special file format called HEIF (High Efficiency Image File) have been standardized [4]. A light-weight variant of the reference HEVC encoder has also been developed [5], which is optimized for compressing a single image.

Advances in Computer Science Research, vol. 13, pp. 59-73, 2016.

In the majority of works, an image that has to be compressed using a video codec is represented as a single-frame video sequence. It is encoded only by means of intra prediction, while motion-compensated prediction (MCP) remains unused, even though it is advanced and effective. This apparently seems appropriate as motion estimation-compensation mechanisms are aimed at removing temporal correlation among frames, which has nothing to a single image. However, it has been shown in [12] that dependencies among pixels of an image can be converted into inter-frame correlations, which can be removed by means of motion estimation-compensation. It is only necessary to represent this image as a video sequence that comprises its polyphase components, which result from pixel decimation.

Our experiments, conducted using the H.264 reference software and presented in [12], have shown that the polyphase approach in general performs slightly worse than the single-frame counterpart, but this is not caused by a conceptual fault. The problem is more because the H.264 standard has not been optimized toward processing polyphase components, which have properties different than those of still images.

With the advent of the HEVC, it seems reasonable to check how well the new standard suits the polyphase approach. As the H.265 is more than the H.264 oriented toward still image compression, one can expect that it should effectively compress video sequences constructed from the results of the polyphase decomposition of a picture.

Appropriate tests can be performed using the reference HEVC codec, but experimenting is tedious and time consuming if not supported with a higher-level software. The reference software is difficult to configure, being controlled by more than a hundredth of parameters. An image must be converted into a video sequence before

Table 1. Analyzers of compressed video bitstreams

Name	Company	Web page
CodecVisa	Codecian	www.codecian.com
Zond 265	Solveigmm	http://www.solveigmm.com/en/products/zond
Elecard HEVC bitstream analyzer	Elecard	www.elecard.com
Gitl HEVC bitstream analyzer	Sun Yat-sen University	www.github.com/lheric/GitlHEVCAnalyzer
HEVC Analyzer for Rapid Prototyping (HARP)	Friedrich-Alexander-University	www.lms.lnt.de/HARP
Parabola Verifier	Parabola Research	http://www.parabolaresearch.com/verifier-hevc-validation.html
Intel Video Pro Analyzer (VPA)	Intel	https://software.intel.com/en-us/intel-video-pro-analyzer

running an encoder, and then it is necessary to restore the decoded picture from video frames, in order to evaluate the rate-quality tradeoff. Finally, an encoder must be run many times, for various images and combinations of parameter values.

Table 1 shows a list of the existing analyzers of video bitstreams. Such analyzers are aimed at helping codec developers, television engineers, and creators of multimedia contents. They are advanced but provide means mainly for analyzing existing bitstreams with respect to standard compliance, bit rate, and quality of decoded video. There is no specific support for processing still images, creating bitstreams, and comparing compression results for different settings of an encoder. So we had a reasonable motivation for developing a system tailored to our research needs.

2. The H.265 and H.264 standards for video coding

Video compression requires transforming picture (frame) sequences in such a way that two kinds of redundancy are removed. The first is temporal redundancy, which is related to correlations among pixels of adjacent frames. The second is spatial redundancy, which is a result of similarity of adjacent pixels in a single frame.

Both H.265 and H.264 standards specify hybrid decoders [14], which use both prediction and transforms to convert highly correlated pixels into uncorrelated values, which can be effectively entropy coded. In particular, temporal redundancy is removed by means of motion estimation/compensation among frames, while the spatial redundancy is removed by predicting pixels of a block to be coded from reconstructed pixels of blocks that can be decoded in advance [10]. In addition, DCT-like and Hadamard transforms are applied to prediction residuals [9].

Both standards only specify a bitstream format and explain how to interpret data so as to successfully decode a video sequence. They provide no ready-to-use recipes for implementing a codec which would be able to produce a format-compliant bitstream from video data and to efficiently reconstruct frames. However, both HEVC and H.264 standards are supported by reference software, which makes it easier to understand a standard as well as to develop and evaluate conformant programs, devices, and bitstreams.

The reference H.265 coded is called the HM (HEVC Test Model), while the reference H.264 software is called the JM (Joint Test Model). Their source codes are publicly available, in the C++ and C languages, respectively. Both projects have been maintained by the Fraunhofer Heinrich Hertz Institute (`www.hhi.fraunhofer.de`), as joint standardization projects of the ITU-T Video Coding Experts Group and ISO/IEC Moving Picture Experts Group.

The reference software are command-line applications. Before encoding a video sequence, about one hundredth of parameters must be set, which determine both effectiveness and speed of compression. This can be done entirely via the command line, but it is also possible to provide only one command-line parameter, with is the path to a configuration file, in which particular settings have been specified.

3. Converting images into video sequence by means of polyphase decomposition

Polyphase components of an image result from decimating it, by taking every second pixel vertically and horizontally, as explained in Fig. 1. As either even or odd pixels can be taken, polyphase decomposition results in 4 subimages. Polyphase decomposition and decimation are operations widely used in the research field of filter banks and transforms [20,8], as they allow for simplifying analysis and design of circuits and for efficiently implementing multirate systems.

Figure 2 shows the polyphase decomposition of a fragment of the well-known "Barbara" image. It is difficult to notice differences among polyphase components. They look like the same image, similarly as video frames. So it is intuitively justified to arrange them into a video sequence and to apply a video encoder to such data.

However, polyphase components have characteristics different from those of video frames. Variations among frames result from movements of the video camera and of objects it sees, while image contents (directions of edges and of intensity changes) determines which polyphase components are most similar in a given area. Moreover, video frames are smoother than polyphase components, which suffer from aliasing distortions, as no anti-aliasing pre-filtering precedes decimation.

Regardless of these issues, inter-prediction should work well on decimated images. For smooth regions of an image, a pixel of a polyphase component is usually the average of pixels of another component [13]. This suggests that bidirectional prediction should work better than unidirectional prediction, or that it is better to compress polyphase components as B-frames than as P-frames.

However, in order to employ bidirectional prediction, it is insufficient to straightforwardly arrange polyphase components into a 4-frame sequence. This is because the state-of-art video codecs cannot use a single frame as two reference frames for bidirectional prediction of another frame. A workaround is to put a reference frame twice into a video sequence, before and after the latter frame. Additionally, an encoder must be instructed how to handle so specific, redundant video sequences: in which order frames should be processed, and which kind of prediction should be ap-

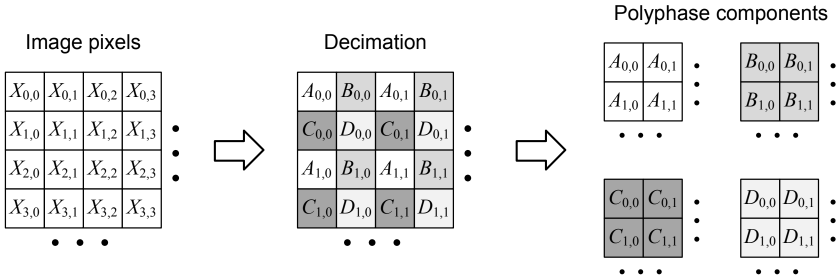

Fig. 1. The principle of polyphase decomposition.

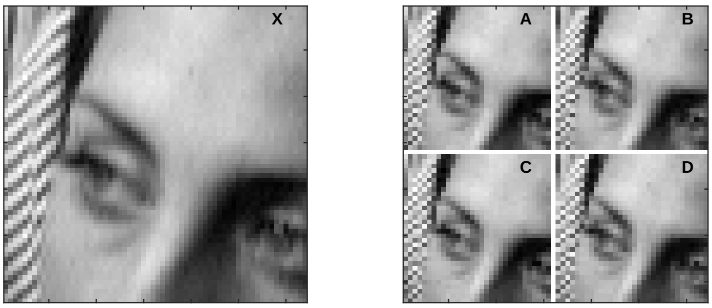

Fig. 2. The polyphase decomposition of a fragment of the "Barbara" image.

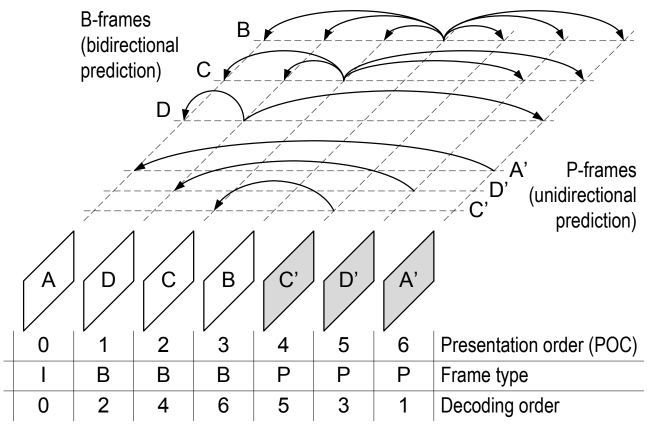

Fig. 3. The GOP structure used to compress polyphase components.

plied to a particular frame. For this purpose, it is necessary to carefully structure the group of pictures (GOP), which is one of the essential settings of a video encoder.

In our experiments, we used the GOP that is shown in Fig. 3, and video sequences in which three polyphase components occurred twice. Only the A component is encoded using intra predictions. The essential copies of the C, B, and D components are encoded using bidirectional prediction, as B-frames, and serve as references for encoding the extra frames as P-frames.

An encoder encodes such video sequences very efficiently, even thought it must process much more data than in the original image. In particular, it represents the extra frames using a negligible number of bits in the output stream, as demonstrated in Table 2.

Table 2. HEVC encoding statistics of video sequences composed of the polyphase components of the "Barbara" and "Finger" images

(a) "Barbara" image, 512 × 512 pixels

Frame	Quantization parameter									
	QP = 9		QP = 17		QP = 25		QP = 33		QP = 41	
	Bit count	**PSNR [dB]**	**Bit count**	**PSNR [dB]**	**Bit count**	**PSNR [dB]**	**Bit count**	**PSNR [dB]**	**Bit count**	**PSNR [dB]**
A	298728	54.09	199416	46.15	111472	38.90	55416	32.62	19056	26.69
A’	104	54.09	96	46.15	96	38.90	88	32.62	88	26.69
D	248520	53.43	150752	45.67	62192	38.19	14208	31.21	1712	25.98
D’	112	53.43	112	45.67	96	38.19	96	31.21	96	25.98
C	197752	53.20	104408	45.65	25616	38.07	1944	31.94	224	26.63
C’	112	53.20	112	45.65	104	38.07	104	31.94	104	26.63
B	196728	53.20	102920	45.61	26736	38.00	1712	31.07	248	26.00

(b) "Finger" image, 256 × 256 pixels

Frame	Quantization parameter									
	QP = 9		QP = 17		QP = 25		QP = 33		QP = 41	
	Bit count	**PSNR [dB]**	**Bit count**	**PSNR [dB]**	**Bit count**	**PSNR [dB]**	**Bit count**	**PSNR [dB]**	**Bit count**	**PSNR [dB]**
A	110376	54.35	84408	46.92	61840	38.97	36440	30.16	11728	22.19
A’	72	54.35	72	46.92	80	38.97	80	30.16	80	22.19
D	104304	53.67	77376	46.21	53424	37.87	27656	28.93	720	20.26
D’	88	53.67	88	46.21	80	37.87	80	28.93	80	20.26
C	100368	53.29	74256	46.26	50488	37.86	23936	28.74	144	20.62
C’	88	53.29	88	46.26	80	37.86	88	28.74	88	20.62
B	100104	53.47	74208	46.20	50384	37.80	24376	28.81	120	20.33

4. Functionalities and architecture of the system

The main functionality of our system is to provide a graphical user interface (GUI) for preparing configuration files for a video encoder, for running a reference software and evaluating its output data, and for managing and analyzing archived results of many experiments. Figure 4 shows three main windows that have been created for these purposes.

The window in Fig. 4a allows for selecting an image to be compressed, for choosing between HEVC and H.264 reference encoders, and for designating a combination of parameter values whose effect on compression results has to be tested. Before running an encoder, a configuration file is created based on a template, whose fragments are substituted in accordance with user's preferences. Also a data file with video frames is created in the planar YUV format [16], which is the only format accepted by the reference software.

The YUV format represents video frames without any compression. Row-by-row and column-by-column, for each pixel, values are listed that describe luminance, which is denoted as Y, as well as two, blue and red, chrominances, denoted by U and V, respectively. There are neither headers nor marks that would determine frame boundaries, and there is no indication of chroma subsampling. A software that has to process a YUV file must be informed by a user about the video resolution and color format of this file.

Both H.265 and H.264 reference encoders produce three files: a report, a bit stream of the encoded image, and the decoded video sequence in the YUV format. Our system converts the sequence back into an image file and archives all files in a newly-created directory for later usage. The window in Fig. 4b allows for investigating the results of a single run of the encoder. The original and decoded images can be compared visually, at the background of the compressed bit rate and objective measures of quality, the PSNR and MSE.

The window in Fig. 4c allows for extracting information from many directories with archived data. Data can be reviewed in a tabular form or plotted if this is preferred by the user. It is also possible to copy data to the system clipboard so as to export them to other applications.

The system has been implemented in Java, using the JDK 8 and Netbeans 8.1 development environments. It uses mainly the standard Java libraries, as the only exception is the "JFreeChart" library for drawing plots. The GUI is based on the Java Swing API. As to the reference software, the HM version 16.2 and the JM version 19.0 were used in our research, compiled for 32-bit Windows operating systems.

(a)
(b)
(c)

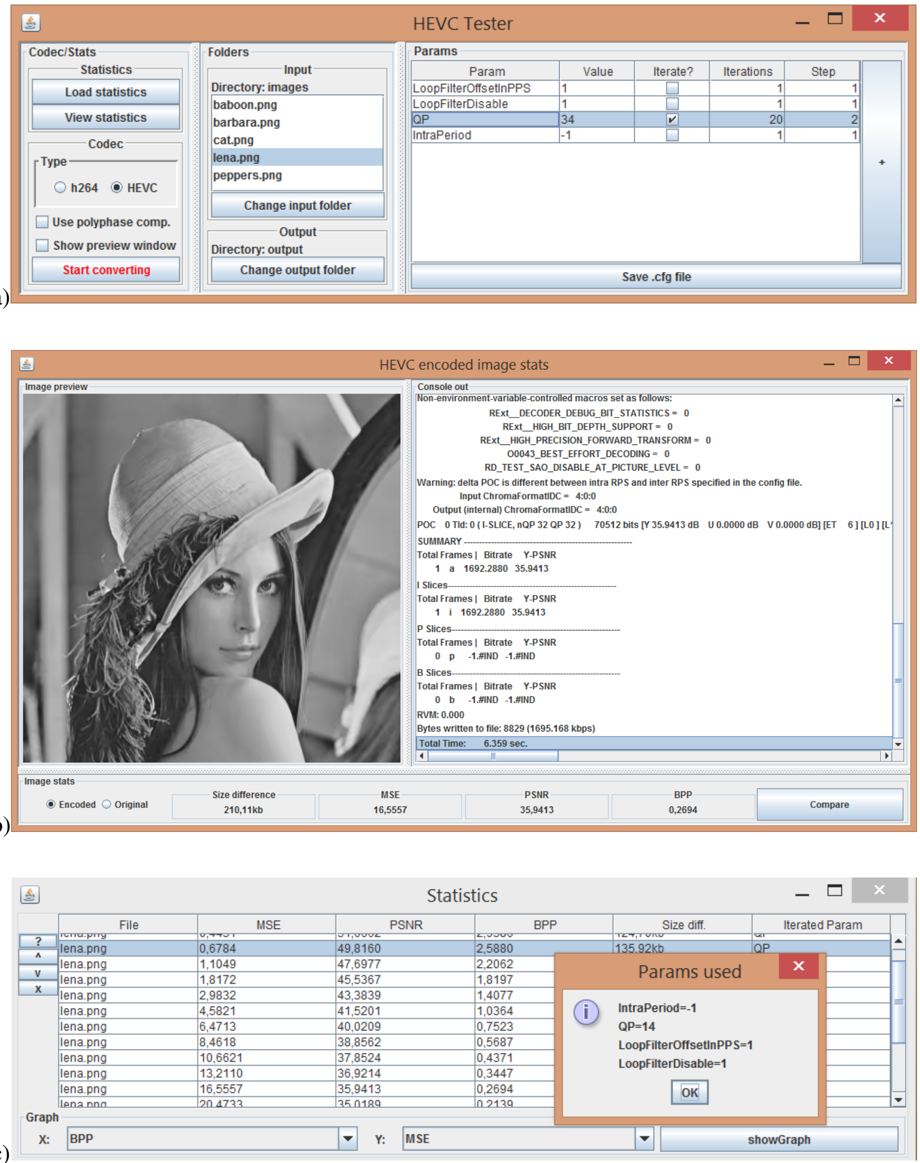

Fig. 4. The graphical user interface for evaluating performance of video codecs: (a) the window for configuring experiments, (b) the window for evaluating results of a single run of an encoder, and (c) the window for reviewing archived results and determining trends in data.

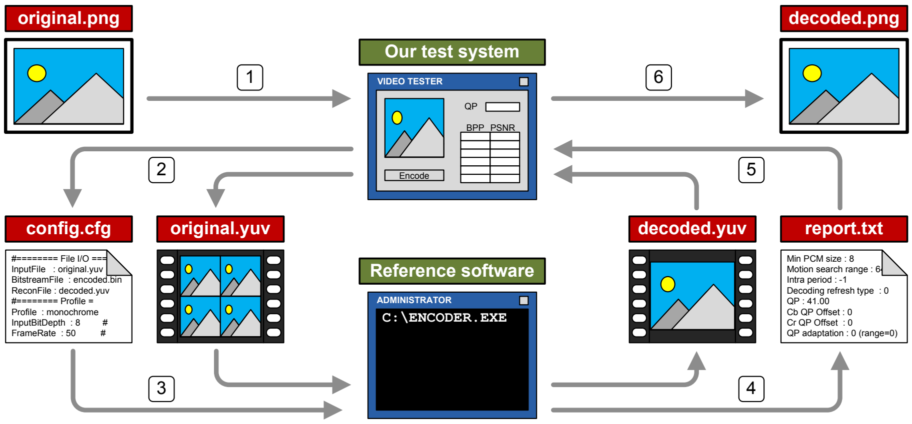

Fig. 5. The dataflow in our system for testing video codecs.

The algorithms and data structures behind the windows are not trivial, even though our application is essentially a wrapper to the reference implementations of the HEVC and H.264 standards. They cooperate in accordance with the dataflow shown in Fig. 5, exchanging data via files.

The source code of our system consists of about 3000 lines and about 20 classes, about half of which are related to video encoding. In particular, the main class, "CodecManager", implements interfaces to the reference encoders and means for managing their input and output files. The main data structures have been encapsulated in the "GrayscaleMedia" class, which is responsible for storing both original image and corresponding video sequence, and allows for converting files into 2D arrays of numbers, which represent image pixels. Another important class is "ImageUtils", which provides functions for converting images between disk files and numerical arrays. In particular, it provides methods for converting a pixel array into its polyphase components.

Video encoding is realized by external executables, which are launched by the system in accordance with user preferences, determined using the GUI. Before launching an encoder chosen by the user, the system performs two tasks. Firstly, it converts a given image into a YUV file, which may be a seven- or single-frame video sequence, depending on whether the polyphase approach has to be tested. The second step is to analyze GUI controls and to produce a configuration file for the encoder.

After launching a reference software, our system monitors the progress in encoding, collects output data, and saves results for later analysis. In case of successful execution, the encoders produce two main files: a bitstream with encoded information (a file with the ".hevc" or ".h264" extension) and the decoded video sequence in the YUV format. Our system converts the latter into a PNG file, so as to allow the decoded image to be evaluated visually and compared to the original.

5. Representative experimental results

The system we have developed allows for easily gathering various and thorough experimental results. Firstly, data can be obtained that are necessary to plot rate-quality curves like those in Fig. 6. Secondly, details about how a particular polyphase component has been encoded can be extracted from encoder reports, in order to create tables like Table 2. Finally, original and reconstructed images can be compared visually so as to evaluate subjective quality and compression artifacts, whose examples are shown in Fig. 7.

In general, our results obtained for the HEVC conform with those presented in [12] for the H.264 standard. Coding a single frame is usually more effective than coding a sequence of polyphase components, but the latter method performs only slightly worse and sometimes works equally well. This suggests that the polyphase approach should be successful after adapting the reference software to processing alias-contaminated images, especially by adjusting entropy codewords to a limited set of motion vectors and to a reduced search range for motion estimation.

Figure 7 shows artifacts caused by compressing an image as a single frame and as a sequence of polyphase components. It can be seen that one cannot depend on objective evaluation of decoded images if he would like to compare the approaches. The difference between images seems considerable if measured in terms of the PSNR, but they are more or less similar visually. For compressing polyphase components, the reconstructed image looks even better, as it is smoother, and quantization noise is less noticeable, even though the objective measure suggests the opposite.

Our experimental data also confirm the known fact that the HEVC performs better than the H.264, regardless of whether an image is represented as a single frame or as a sequence of polyphase components.

6. Summary

The primary aim of our research has been achieved. We have developed a functional system that is really helpful in configuring and running the reference video codecs, and then in analyzing their output data. In particular, the system has allowed us to thoroughly test whether the HEVC reference software is able to efficiently compress images represented as multi-frame video sequences using polyphase decomposition.

Unfortunately, even equipped with such a handy tool, we were unable to tune settings of the HEVC encoder so as to prove that it is better to compress a sequence of polyphase components than to encode an image as a single frame. In general, the latter approach performs slightly better, but sometimes both techniques work equally well. This suggests that only a video codec with carefully optimized internals, especially entropy codes, would be able to handle decimated images effectively. We have recently studied the heart of such codec, a wavelet-like transform inspired by motion estimation/compensation [13].

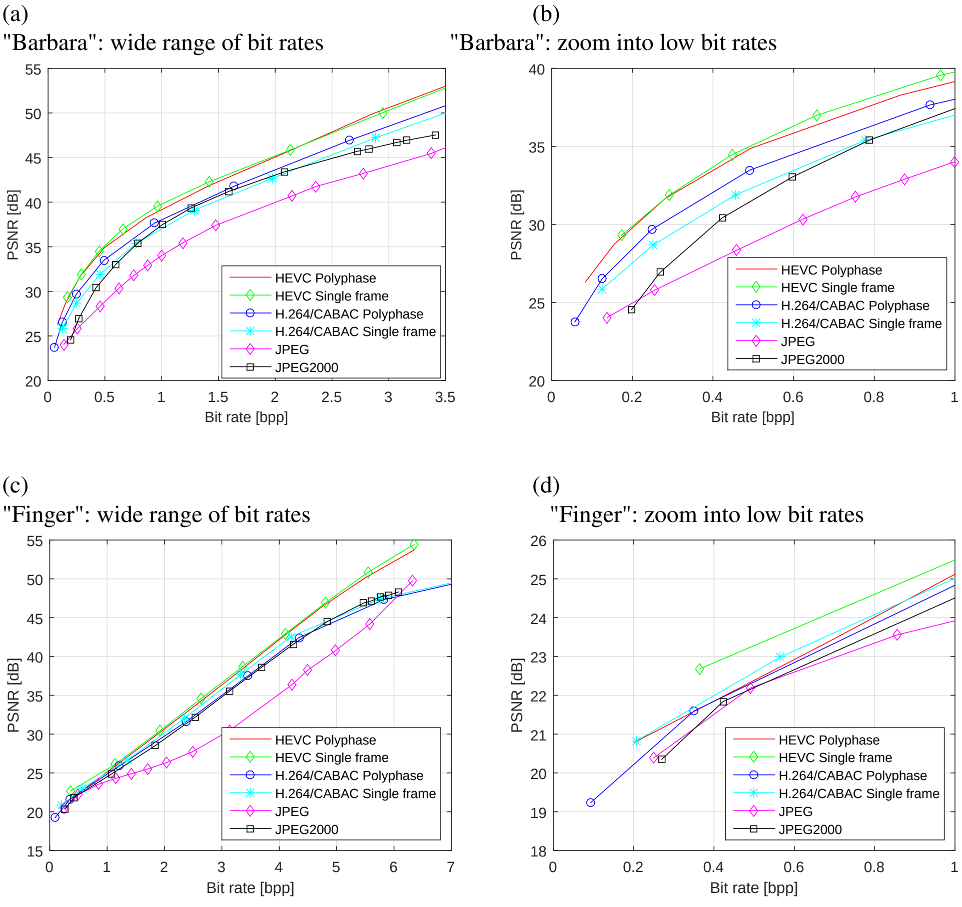

Fig. 6. Effectiveness of various standards in coding the ”Barbara” (512×512 pixels) and ”Finger” (256×256 pixels) images.

(a) Original | (b) Polyphase 0.2431 bpp, 34.47 dB | (c) Single-frame 0.2397 bps, 35.48 dB

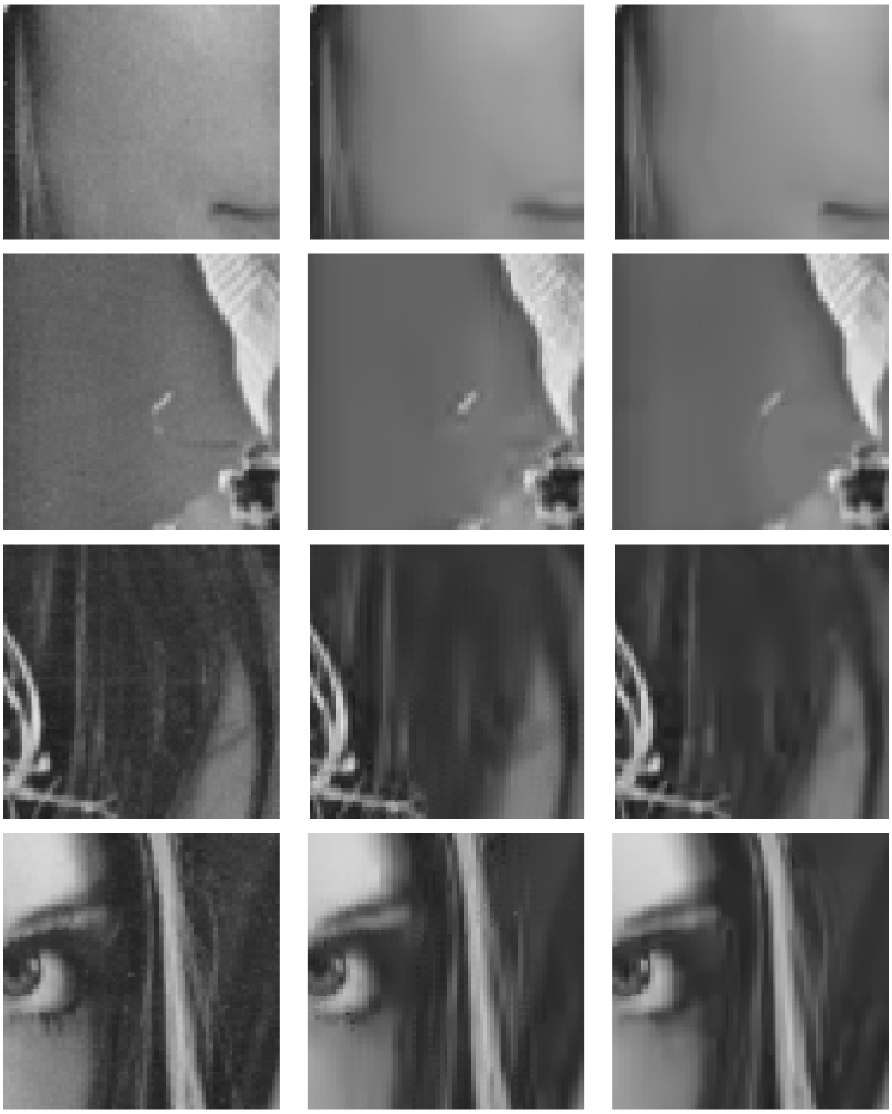

Fig. 7. Artifacts in images compressed using the reference HEVC codec: (a) fragments of the original "Lena" image, (b) fragments reconstructed from a bitstream obtained using the polyphase approach, and (c) fragments reconstructed from a bitstream obtained by encoding a single frame.

Acknowledgment

This work was supported by the Polish National Science Centre under Decision No. DEC-2012/07/D/ST6/02454.

References

[1] A. Al, B. P. Rao, S. S. Kudva, S. Babu, S. David, and . Ajit V. Rao. Quality and complexity comparison of H.264 intra mode with JPEG2000 and JPEG. In *Proc. IEEE Int. Conf. Image Process. (ICIP)*, pages 525–528, Singapore, 24-27 Oct. 2004.

[2] Q. Cai, L. Song, G. Li, and N. Ling. Lossy and lossless intra coding performance evaluation: HEVC, H.264/AVC, JPEG 2000 and JPEG LS. In *Proc. Asia-Pacific Signal & Information Process. Association Annual Summit and Conf. (APSIPA ASC)*, pages 1–9, Hollywood, CA, 3-6 Dec. 2012.

[3] P. Hanhart, M. Rerabek, P. Korshunov, and T. Ebrahimi. Subjective evaluation of HEVC intra coding for still image compression. In *Proc. 7th Int. Workshop Video Process. Quality Metrics Consumer Electronics (VPQM)*, pages 58–63, Scottsdale, Arizona, USA, Jan. 30-Feb. 1 2013.

[4] M. M. Hannuksela, J. Lainema, and V. K. M. Vadakital. The High Efficiency Image File format standard. *IEEE Signal Process. Mag.*, 32(4):150–156, July 2015.

[5] F. Liang, X. Peng, and J. Xu. A light-weight HEVC encoder for image coding. In *Proc. Visual Communications Image Process. (VCIP)*, pages 1–5, Kuching, Malaysia, 17–20 Nov. 2013.

[6] D. Marpe, V. George, H. Cycon, and K. Barthel. Performance evaluation of Motion-JPEG2000 in comparison with H.264/AVC operated in pure intra coding mode. *Proc. SPIE*, 5266 (Wavelet Applications in Industrial Process.):129–137, 2004.

[7] D. Marpe, V. George, and T. Wiegand. Performance comparison of intra-only H.264/AVC HP and JPEG2000 for a set of monochrome ISO/IEC test images: Contribution JVT ISO/IEC MPEG and ITU-T VCEG. Technical Report JVT M-014, Oct. 2004.

[8] L. Milić. *Multirate filtering for digital signal processing: MATLAB applications*. Information Science Reference / IGI Global, Hershey, PA, 2009.

[9] T. Nguyen, P. Helle, M. Winken, B. Bross, D. Marpe, H. Schwarz, and T. Wiegand. Transform coding techniques in HEVC. *IEEE J. Sel. Topics Signal Process.*, 7(6):978–989, Dec. 2013.

[10] T. Nguyen and D. Marpe. Performance analysis of HEVC-based intra coding for still image compression. In *Proc. Picture Coding Symp. (PCS)*, pages 233–236, Cracow, Poland, 7-9 May 2012.

[11] T. Nguyen and D. Marpe. Objective performance evaluation of the HEVC Main Still Picture Profile. *IEEE Trans. Circuits Syst. Video Technol.*, 25(5):790–797, May 2015.

[12] M. Parfieniuk. Polyphase components of an image as video frames: a way to code still images using H.264. In *Proc. 2012 Picture Coding Symposium (PCS)*, pages 189–192, Cracow, Poland, 7-9 May 2012.

[13] M. Parfieniuk. *Wavelet-Like Lifting-Based Transform for Decomposing Images in Accordance with the Inter-prediction Principles of Video Coding*, volume 9280 of *Lecture Notes in Computer Science (LNCS)*, pages 162–171. Springer, 2015.

[14] K. R. Rao, D. N. Kim, and J. J. Hwang. *Video coding standards: AVS China, H.264/MPEG-4 PART 10, HEVC, VP6, DIRAC and VC-1*. Springer, 2014.

[15] I. E. G. Richardson. *The H.264 advanced video compression standard*. Wiley, 2 edition, 2010.

[16] G. Sullivan and S. Estrop. Recommended 8xbit YUV formats for video rendering. Technical report, Microsoft, 2008.

[17] 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.*, 22(12):1649–1668, Dec. 2012.

[18] V. Sze, M. Budagavi, and G. J. Sullivan, editors. *High Efficiency Video Coding (HEVC): algorithms and architectures*. Springer, 2014.

[19] T. Tran, L. Liu, and P. Topiwala. Performance comparison of leading image codecs: H.264/AVC Intra, JPEG2000, and Microsoft HD Photo. *Proc. SPIE*, 6696 (Applications of Digital Image Processing XXX):66960B, 2007. doi:10.1117/12.775472.

[20] P. P. Vaidyanathan. Multirate digital filters, filter banks, polyphase networks, and applications: A tutorial. *Proc. IEEE*, 78(1):56–93, Jan. 1990.

[21] M. Wien. *High Efficiency Video Coding: coding tools and specification*. Springer, 2015.

[22] A. Zaghetto and R. L. de Queiroz. Segmentation-driven compound document coding based on H.264/AVC-Intra. *IEEE Trans. Image Process.*, 16(7):1755–1760, July 2007.

SYSTEM DO BADANIA SKUTECZNOŚCI KODEKÓW WIDEO W KODOWANIU OBRAZU

Streszczenie Artykuł prezentuje system do badania sprawności kodeków wideo w kompresowaniu obrazu. Ponieważ bezpośrednie konfigurowanie i uruchamianie referencyjnych kodeków H.265 (HEVC) i H.264 (MPEG4 AVC) jest pracą czasochłonną i żmudną, omawiane narzędzie pozwala przeprowadzać eksperymenty łatwiej i szybciej. System jest wyposażony w graficzny interfejs użytkownika do wygodnego i sprawnego przygotowywania plików wejściowych, do wielokrotnego uruchamiania kodeków wideo i do analizowania danych wyjściowych. Interfejs opiera się na podprogramach do formowania sekwencji wideo i plików konfiguracyjnych, do wsadowego uruchamiania koderów i do wydobywania informacji z plików wynikowych. Głównym celem pracy nad systemem było przyśpieszenie eksperymentów nad kodowaniem obrazu reprezentowanego jako sekwencja wideo złożona z jego składowych polifazowych. System umożliwił eksperymentalne sprawdzenie, że podobnie jak H.264, standard HEVC może skutecznie przetwarzać obrazy zdecymowane, ale wymaga to zoptymalizowania kodów entropijnych.

Słowa kluczowe: wideo, obraz, kodowanie, kompresja, kodek, HEVC, H.264, dekompozycja/składowa polifazowa