



Extending JPEG-LS for low-complexity scalable video coding

Ukhanova, Anna; Sergeev, Anton; Forchhammer, Søren

Published in:

Proceedings of SPIE - The International Society for Optical Engineering

Link to article, DOI:

[10.1117/12.887416](https://doi.org/10.1117/12.887416)

Publication date:

2011

Document Version

Publisher's PDF, also known as Version of record

[Link back to DTU Orbit](#)

Citation (APA):

Ukhanova, A., Sergeev, A., & Forchhammer, S. (2011). Extending JPEG-LS for low-complexity scalable video coding. *Proceedings of SPIE - The International Society for Optical Engineering*, 7870, 78701D. <https://doi.org/10.1117/12.887416>

General rights

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal

If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.

Extending JPEG-LS for low-complexity scalable video coding

Anna Ukhanova^a, Anton Sergeev^b and Søren Forchhammer^a

^aDTU Fotonik, Kgs. Lyngby, Denmark;

^bSUAI, St. Petersburg, Russia

ABSTRACT

JPEG-LS, the well-known international standard for lossless and near-lossless image compression, was originally designed for non-scalable applications. In this paper we propose a scalable modification of JPEG-LS and compare it with the leading image and video coding standards JPEG2000 and H.264/SVC intra for low-complexity constraints of some wireless video applications including graphics.

Keywords: scalable JPEG-LS, H.264/SVC, JPEG2000, wireless video transmission

1. INTRODUCTION

JPEG-LS¹ was designed as an efficient algorithm for completely lossless compression of still images and controlled lossy mode where a precise upper bound on the maximal error in pixel value could be predefined by the user. The main advantages of these algorithms are very low computational complexity, perfect coding efficiency at high rates and possibility of near-lossless compression.^{2,3} All this allows JPEG-LS to successfully compete for some applications with the state-of-the-art compression algorithms for still images (JPEG2000⁴) and video (H.264⁵). JPEG-LS comes to foreground for application with strong constraints for implementation complexity and memory consumptions: processing and storage of medical, airspace images, maps; mobile video; desktop graphics; transmission of HD video and etc. Unfortunately JPEG-LS has also some evident shortcomings. In contrast to its competitors, the original JPEG-LS has no means for building scalable video streams and organizing multithreaded transmission. In scalable video coding partial loss of the scalable stream does not irreparably affect the decoding process: the decoder may reconstruct image with reduced quality level (without re-compression and retransmission). The successful reception of all the compressed streams results in decoding and reconstruction of the whole image with the same quality level as at the coder side. Scalability is extremely important in modern video coding and useful in video transmission, downloading, providing variable quality access etc..^{6,7} There are no such a features in the standard JPEG-LS. In this work we propose a new scalable extension of JPEG-LS and compare it with two generally recognized leaders in the area of image and video coding: JPEG2000 and H.264/SVC in intra mode. Modern full-featured image/video compression algorithms support a variety of compression modes, each with different trade-offs among efficiency of compression, loss of fidelity in the compression and the amount of computation required to encode/decode. Before designing and comparing the compression algorithms, we should determine which modes are suitable based on target application scenarios and limits. Today video transmission in wireless networks (WPAN/WMAN/WLAN) has become a hot topic in the industry.⁷⁻⁹ Therefore this paper aims at providing a performance evaluation of the proposed and other algorithms for the task of real-time wireless high-quality low-complexity video transmission. Comparison parameters are selected accordingly to the requirements of wireless video applications:

- Reduced computational complexity because in mobile terminals (laptops, phones, etc) battery life and processing power are strongly limited.
- High coding efficiency at high rates. Throughput of wireless technologies increased during the last decade (11 Mbps 802.11b to 160 Mbps 802.11n in WLAN, 3Mbps of Bluetooth to 3Gbps of 802.15.3TG3 in WPAN). So looking forward we focus on high transmission rates and consequently consider low compression ratios.
- Scalability and error resilience, controlled quality reduction achieved by partial loss of bit stream.

Further author information: (Send correspondence to Anna Ukhanova)

E-mail: annuk@fotonik.dtu.dk, Telephone: +45 4525 6567

- Universality i.e. effective coding of different types of images including photo, computer graphics and synthetic images.

The paper is organized as follows. Section 2 includes a short overview and basis for profile selection of JPEG-LS and competitors. In Section 3 the scalable JPEG-LS modification is proposed. Section 4 describes comparison methodology and presents comparison of the compression algorithms according to the selected requirements. Section 5 presents results of the performance comparison of the suggested scalable JPEG-LS, standard H.264 intra and JPEG 2000. Section 6 concludes the paper.

2. CODECS OVERVIEW

An overview of the selected codecs is given here.

2.1. JPEG2000 compression

JPEG2000 is the current ISO/ITU-T standard for still and motion image coding. The RCT (Reversible Color Transform) color format is used, described in the standard. JPEG2000 supports many interesting features such as lossless and lossy compression, multi-resolution representation, scalable and Region Of Interest (ROI) coding, tiling, blocking, error resilience and a flexible file format. But on the other hand JPEG2000 is more complex and slower than the prior and still widely-used JPEG standard.¹⁰

2.2. H.264/SVC compression

The Moving Picture Experts Group (MPEG) has introduced the Scalable Video Coding standard, which is an extension of the H.264/MPEG-4 Advanced Video Coding (AVC) standard.¹¹ It has additional properties like scalability. So, on one hand, H.264/SVC could be used in the situation when we have many receivers and it is necessary to receive the data at different bitrates. Another case is when we have to control the transmission rate depending on the situation in the channel. Other important features of H.264 are integrated rate control, deblocking filter and error resilience.

2.3. JPEG-LS compression

JPEG-LS^{1, 12} is the international standard for lossless and near lossless still images compression. The main advantage of JPEG-LS is a possibility to set up the maximum error value per pixel by choosing a bound on the differences for near-lossless coding (so-called near parameter or lossy-factor). Another plus of JPEG-LS is an extremely low level of implementation complexity⁸ and memory consumption at the encoder side. The encoding process requires less than two rows of samples only (less than 10KByte). Unfortunately the original JPEG-LS standard has no means for building scalable video streams and organize multithreaded transmission in contrast with its competitors. Scalability (see Section 4.2) here means a possibility of lossy reconstruction with smaller quality level in case of partial data loss. This feature is very important for wireless video applications because wireless channel may be quite non-stable and some data could be lost during the transmission. Therefore in the next section we propose a scalable version of the JPEG-LS to address this shortcoming.

2.4. Codec Profiles

During the experiments we compared different profiles and feature sets for the JPEG2000, H.264/SVC and JPEG-LS image or video compression algorithms. The following configuration sets are selected for analyzing the codecs from the point of the tradeoff between rate-distortion characteristics and complexity:

- JPEG2000: RCT colorpace transforms, tile size 1280x8, codeblock size 512x8, number of DWT levels is equal 2, reversible wavelet 5-3;
- JPEG-LS: scalable modification, per pixel processing, lossless RCT colorspace transform, no subsampling, Golomb coding. It was noted⁸ that for HD picture size and 60 FPS it is possible to design and implement in RTL (150MHz clock rate, TSMC 60nm technology) the encoder with total power consumption of around 10mW;

- H.264/SVC intra: YUV colorspace, all intra prediction, 4x4 DCT, Intra/IPCM detection, CABAC. The power consumption for H.264 for similar parameters was given as well as for JPEG-LS as stated above.⁸ H.264/SVC has a power consumption, which is roughly at least twice that of JPEG-LS. However, if the channel fluctuates, it is possible to use a scalable stream for power saving.¹³

3. SCALABLE JPEG-LS COMPRESSION

A simple scalable version of JPEG-LS has been proposed¹⁴ by sending one or more least-significant bit-planes uncoded and only coding the reduced precision image using JPEG-LS. We consider another approach for better performance and/or finer granularity in the scalable format.

Our proposed scalable JPEG-LS-based algorithm utilizes the well-known differential coding approach.^{15–17} Firstly data is compressed in lossy mode, then the main (primary) compressed stream is constructed providing the base quality level. Then the difference between an original and compressed image is calculated pixel by pixel. This difference is also encoded and forms the second (secondary) stream. The main question here is: what compression method should be used for the second stream encoding?

The main idea of the proposed scalable JPEG-LS is to compress the input image (or video intra-frame) in two (or more) steps and create two (or more) sub-streams joint in one global scalable bit-stream. At the first step JPEG-LS is used in lossy mode for encoding the original image. The compressed data forms the primary (or the first) sub-stream. The lossy mode increases the compression rate (and thereby decreases the amount of data to be transmitted) but distorts the original image. At the second step at the transmitter side a residual image is calculated as the difference between the original and the reconstructed images (after the first step). Then the residual image is compressed by lossless JPEG-LS and this constitutes the second bit stream for the wireless transmission.

Therefore the encoder generates a single global bitstream, which may include the primary bit stream and the second bit stream. The global bit stream could be transmitted over a stable or an un-stable communication wired or wireless communication channel. The codec may use a progressive approach to provide SNR scalability that allows supporting features such as multi-streaming and prioritized transmission.

A successful reception and decoding of both sub-streams provides a lossless image reconstruction at the receiver side (i.e. reconstructed image is equal the original one). At the same time if there are any problems with the second stream there is a possibility to decode only the primary stream and show the image with visually acceptable quality level (around 42db PSNR for for a maximum pixel value error of 2). Similarly if the quality level of the primary stream is satisfactory for the end-user then the decoding of the second stream (even if it is successfully received) could be omitted.

The maximum error value per pixel (near parameter, n) for constructing the primary bit stream should be selected according to the requirements of the real applications. Figure 1 gives an overview of the algorithm. It is summarized by the following formulas: Consider x_{ij} to be the pixel of the input image with coordinates i and j accordingly. This pixel is firstly compressed with the lossy factor n_1 . After successful reconstruction of the pixel it is equal \hat{x}_{ij} . The corresponding difference between the original pixel and the reconstructed one, $x_{ij} - \hat{x}_{ij}$, is denoted d_{ij} . Prior to making the additional lossless compression of the differential image, all values are shifted by the value of lossy factor n_1 , $d'_{ij} = d_{ij} + n_1$, to get non-negative values. Then the same block of lossless compression is applied to the image (d'_{ij}) of restricted range. This forms the second bit stream of compressed pixels.

The bigger n is selected, the smaller the size of the primary stream will be and the more stable it is to the sudden throughput fluctuations. In our investigations we selected $n = 1$ because in this case the total size of the primary and secondary bit streams is minimum while the size of both bitstreams are approximately equal. One could notice that JPEG2000 and SVC are truly scalable while the presented modification of JPEG-LS relates mainly to progressive two-step scalability.

It should be noted though that the simple two-step coding is well-correlated with the existing usage models for the end-user side:

- the picture of the ideal quality is shown while the wireless channel is in the "good" state,

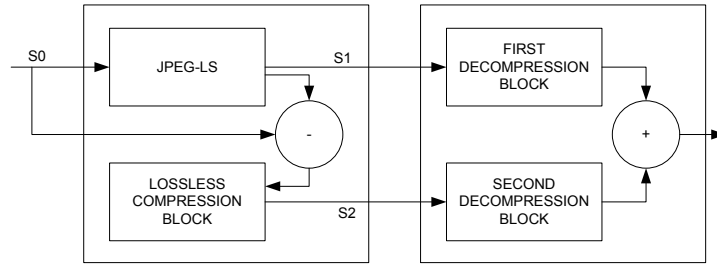


Figure 1. Full lossless decompression

- there is a way of the instantaneous and visually graceful, ideally imperceptible quality degradation in case of sudden noise level fluctuations.

Depending on the image type in lossless mode the compression performance of Scalable JPEG-LS is 2-23% worse than Standard JPEG-LS. With the growth of n this gap slowly increases.

Our proposal also may be extended to a nested structure (as e.g. for near-lossless coding¹⁸): For the JPEG-LS parameter, $NEAR = n_i, i = 1, 2, \dots$ we get the interval: $\prod_i (2 \times n_i + 1)$. Using $n_1 = n_2 = 1$ we have intervals of 9, 3, 1 (=lossless) in a scalable manner. For finer granularity we may use, the visual quantization in JPEG-LS part 2, which context adaptively chooses between near parameters n and $n + 1$. This may readily be treated as above based on the larger value, $n + 1$. For more efficient coding, the refinement JPEG-LS coder should be modified so it reconstructs the near parameter for each pixel and uses this information in coding the difference pixel. Compared with other results,¹⁴ our versions have the advantage of a better control of the (maximum) error of the lossy coding and if using visual quantization, a finer granularity. It is possible to use near-lossless factor up to 5 without any significant visual losses.

For example the simple scalable coding mentioned based on selecting k LSB planes¹⁴ has the following drawbacks:

1. The intervals steps increase exponentially as 2^k , i.e. 2, 4, 8, 16,
2. As can be seen from the result,¹⁴ the convex hull of the rate-distortion (measured by PSNR) of the baselayer has lower quality than the convex hull of JPEG-LS measured at different near values.
3. Also measured by the inf-norm truncation it is inferior to using JPEG-LS in near-lossless mode. (Points 2 and 3 are in part due to reconstructing to integer points.) Due to these relative advantages of JPEG-LS in near-lossless mode, it is interesting to explore the possibilities of using JPEG-LS near-lossless directly as the base-layer, but still have a simple coding and processing of enhancement layers for a scalable codec.

4. COMPARISON METHODOLOGY

To decrease the level of power consumption and complexity cost it is suggested here to use video encoders with relatively small level of operational complexity and very small memory utilization.

4.1. Intra-Coding and Slicing

To decrease complexity costs at the encoder side intra-coding mode only is used, i.e. the frames are compressed independently and are not stored after compression. Therefore we can not apply algorithms that exploit temporal redundancy such as motion estimation/compensation (ME/MC) or differential coding.

An additional way to decrease memory consumption for some algorithms like JPEG2000 is to split an input frame into smaller pieces and to process each piece independently and sequentially. More specifically, a frame is partitioned into (one or) several disjoint rectangular regions called slices. Small slices can

significantly decrease the level of memory consumption. The slice size in our tests are 1280x16 (less than 100 KByte if a pixel depth is 24 bits) and for hardware implementation a low-cost internal chip-based memory can be used. Slicing can be viewed from an implementation point of view: we have one slice in memory at a time. This will restrict wavelet based coders, whereas JPEG-LS only requires 2-lines in memory (nevertheless, coded as one frame). Meanwhile, slicing can also be viewed from the point of the coding format, which may influence performance. For JPEG-LS we can code the whole image as one entity (still not requiring more than two lines in memory) or we can slice it. The later approach for slicing will come at a price in coding performance for adaptive coders as JPEG-LS. However, slicing also allows us to enhance the use of unequal error protection (UEP)¹⁹ to the transmission of the video data.

4.2. Scalability

Data transmission with two sub-streams is one of the most commonly used scalability schemes satisfying most of the use cases. It allows for end-users to receive pictures even in the case of unexpected data loss and at the same time the overhead costs for (unequal) error protection for two streams instead of one are quite acceptable. So, the main idea of scalable coding is that coder forms the bitstream from several layers. All three competitors support scalable and progressive coding i.e. in case of partial data loss during transmission only a part of the encoded data could be received, extracted and decoded from a global scalable bit-stream and an image with smaller quality level can be successfully reconstructed at the receiver. Scalability is one of the most prominent features of JPEG2000 that leads to its ability of extracting different resolutions, fidelities, components or spatial locations from a single compressed bitstream. The scalable extension (SVC) of the H.264/AVC Standard, is a highly attractive solution to the problems posed by the characteristics of modern video transmission systems. Current version of JPEG-LS standard (excluding the one proposed in this paper) does not support scalability.

4.3. Complexity Costs and Features

It is clear that implementation complexity and memory consumption should be estimated not for the whole coding standard but for a selected core number of functionalities required by wireless applications. The standardized codecs provide a rich set of instruments for putting certain restrictions on the encoding parameters such that some kind of complexity scalability can be achieved. Anyway different investigations² demonstrate that JPEG2000 is more complex than H.264/SVC due to integral arithmetic coding and multiple bitwise operations. It was mentioned^{20,21} that the Tier-1 block of JPEG2000 that includes those operations consumes more than 50% of total computation power. On the other hand JPEG-LS was originally designed as a low-complexity coding system that has extremely low complexity level in comparison with H.264/SVC and JPEG2000. Error concealment in JPEG2000 can be ensured by different means including markers, regular termination of the arithmetic coder, error resilient termination and segment symbols, possibility to move the sensitive packet head information to the bit stream header and etc.⁴ H.264/SVC also includes a rich set of tools particularly designed for that purpose.⁵ JPEG-LS unfortunately does not offer a proper error resilience support, but as described may easily be extended..

5. RESULTS

In our experiments HD video sequences of resolutions 1280x720, 30fps are tested. They have a 24 bits/pixel depth. The raw bitrate is 0.663 Gbps. We have used RGB 4:4:4 color space format for JPEG2000 and JPEG-LS and YUV 4:4:4 for H.264/SVC. For more detailed description of the codecs settings please refer to Subsection 2.4. "Kungfu" (427 frames) and "Breeze" (461 frames) are natural image sequences, with fast and slow motion, respectively. "Desktop" (1880 frames) is a computer desktop snapshot, mostly consisting of computer graphics. (For the sake of simplicity and ease of reporting on the figures lossless is depicted at 70dB is for JPEG-LS and JPEG2000). The following graphs (Figs.2-4) present the comparison of results. It is necessary to note that one point for JPEG-LS (between n=0 and n=1) is obtained as a compression of one half of the frames with n=0 and the other - with n=1.

Obviously, JPEG-LS was not constructed for the low rates, in contrast to H.264. JPEG-LS has the best relative performance for computer graphics, that is why it shows outstanding results on the "Desktop"

sequence (Fig.2). Table 1 shows compression efficiency comparison for Standard JPEG-LS and the proposed scalable solution in Gbps.

Table 1. Compression efficiency comparison (bit rate, Gbps)

Codec type	desktop	kungfu	breeze
JPEG-LS	0.128	0.229	0.262
Scalable JPEG-LS	0.132	0.247	0.276

6. CONCLUSION

This paper proposes simple and low-complexity solution for two-step scalable JPEG-LS. Comparison with other widely used scalable solutions for video compression like H.264/SVC and JPEG2000 show that the proposed idea can compete with standard solutions for specific video content at high rates.

7. ACKNOWLEDGMENT

This research work was supported in part by a special grant of Intel CTG Research Council. We are especially grateful to Dr. Andrew Belogolovy and Dr. Andrey Turlikov for plentiful discussions and helpful comments.

REFERENCES

1. ISO/IEC JTC 1/SC 29/WG1 FCD 14495-public draft, "JPEG-LS: lossless and near-lossless coding of continuous tone still images," July 16th, 1997
2. D. Santa Cruz et al., "An analytical study of JPEG 2000 functionalities," *Proceedings of SPIE*, Vol. 4115, 2000.
3. S. D. Rane, G. Sapiro, "Evaluation of JPEG-LS, the New Lossless and Controlled-Lossy Still Image Compression Standard, for Compression of High-Resolution Elevation Data," *IEEE Transactions on Geoscience and Remote Sensing*, Vol. 39, No. 10, October 2001.
4. ITU-T and ISO/IEC JTC 1, "JPEG 2000 Image Coding System: Core Coding System, ITU-T Recommendation T.800 and ISO/IEC 15444-1," *JPEG 2000 Part 1*, 2000.
5. "Introduction to SVC Extension of Advanced Video Coding, ISO/IEC JTC1/SC29/WG11" *International Organization for Standardization, Coding of Moving Pictures and Audio*, Pozna , Poland, July 2005. <http://www.chiariglione.org/mpeg/technologies/mp04-svc/svc/>.
6. D. S. Taubman, "Directionality and Scalability in Image and Video Compression," *PhD dissertation*, 1994.
7. K. N. Ngan, C. W. Yap, K. T. Tan "Video coding for wireless communication systems," Marcel Dekker, 2001.
8. A. Belogolovy, E. Belyaev, A. Sergeev, A. Turlikov, "Video Compression for Wireless Transmission: Reducing the Power Consumption of the WPAN Hi-speed Systems," *New2AN*, 2009.
9. A. H. Sadka "Compressed video communications," John Wiley and Sons, 2002
10. ISO/IEC 10918-1 ITU-T Recommendation T.81) <http://www.w3.org/Graphics/JPEG/itu-t81.pdf>
11. T. Wiegand, G. J. Sullivan, G. Bjontegaard, and A. Luthra, "Overview of the H.264/AVC Video Coding Standard," *IEEE Trans. on Circuits and Systems for Video Technology*, vol. 13, no. 7, July 2003.
12. M.J. Weinberger, G. Seroussi, G.Sapiro, "The LOCO-I lossless image compression algorithm: principles and standardization into JPEG-LS," *Image Processing, IEEE Transactions*, Volume 9, Issue 8, Aug 2000 Page(s):1309 - 1324.
13. E.Belyaev, V. Grinko, A. Ukhanova, "Power Saving Control for the Mobile Receivers in the DVB-H based on the Scalable Extension of H.264/AVC Standard," *Wireless Telecommunications Symposium*, 2009
14. R.J.Vleuten, S. Egner, "Lossless and Fine-Granularity Scalable Near-Lossless Color Image Compression," *Data Compression Conference*, 2002, DCC Proceedings.
15. D.G. Korn, K.P. Vo, B. Krishnamurthy, "Vdelta: Differencing and Compression, Practical Reusable Unix Software" John Wiley Sons, 1995.

16. D. S. Taubman, "Directionality and Scalability in Image and Video Compression," *PhD dissertation*, 1994.
17. "DPCM - Differential Pulse Code Modulation," <http://einstein.informatik.uni-oldenburg.de/rechnernetze/dpcm.htm>.
18. A. Krivoulets, "Design of efficient algorithms for image compression with application to medical images," *Ph.D. thesis at IT University of Copenhagen*, ISBN: 87-7949-060-3, 2004.
19. R.M. Joohee Kim Mersereau, Y. Altunbasak, "Error-Resilient Image and Video Transmission Over the Internet Using Unequal Error Protection," *Transactions on Image Processing*, Vol. 12, No. 2, pp. 121 - 131, February 2003.
20. Chen, Liang-Gee, *United States Application 20060008162*, <http://www.freepatentsonline.com/20060008162.html>.
21. G. Pastuszak, "A high-performance architecture for EBCOT in the JPEG 2000 encoder," *Signal Processing Systems Design and Implementation*, IEEE Workshop on Volume , Issue , 2-4 Nov. 2005 pp. 693 - 698.
22. B. Martins and S. Forchhammer, "Lossless Compression of Video using Motion Compensation,"
23. S. Andriani, G. Calvagno, M. Durigon, R. Rinaldo, M. Knee, P. Walland, M. Koppetz, T. Erseghe, G.A. Mian, "Comparison of lossy to lossless compression techniques for digital cinema," *International Conference on Image Processing*, Singapore, 2004.
24. D. Brunello, G. Calvagno, G.A. Mian, and R. Rinaldo, "Lossless Compression of Video Using Temporal Information," *IEEE Transactions on image Processing*, Vol. 12, No. 2, February 2003.

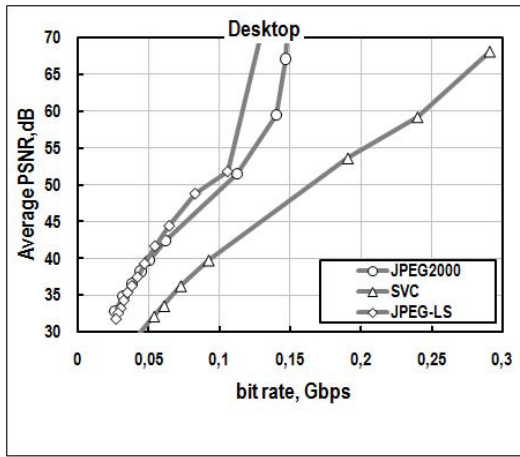


Figure 2. RD comparison for "Desktop" sequence

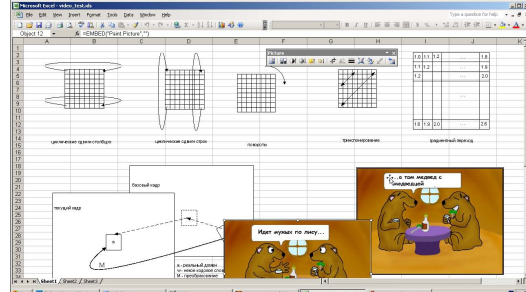


Figure 5. Example of "Desktop" sequence

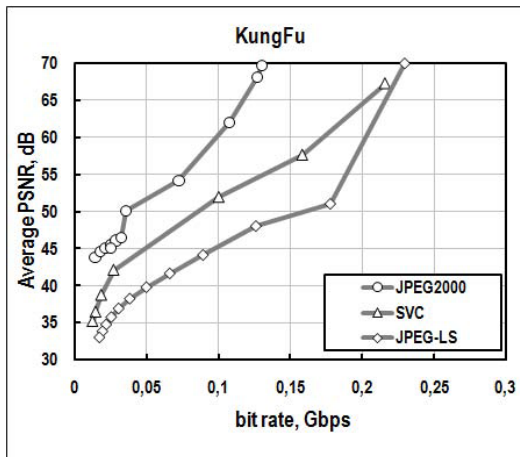


Figure 3. RD comparison for "Kungfu" sequence

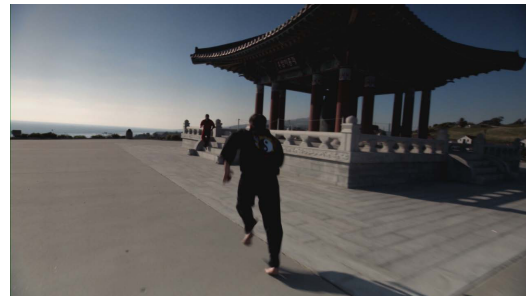


Figure 6. Example of "Kungfu" sequence

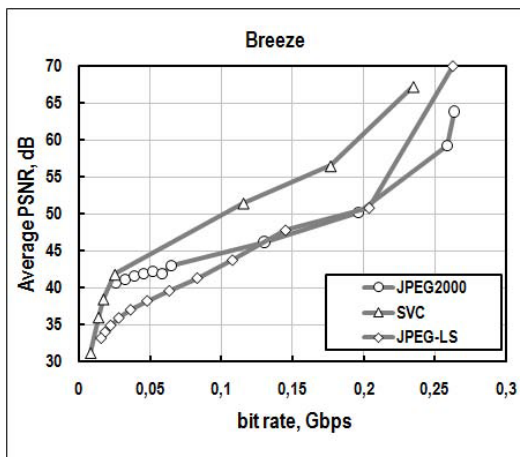


Figure 4. RD comparison for "Breeze" sequence

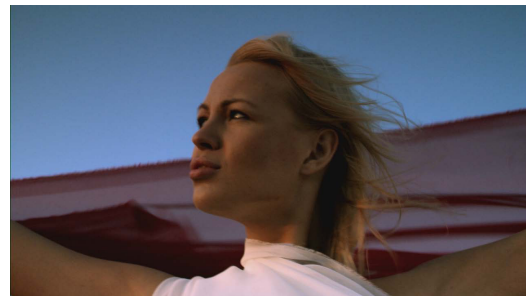


Figure 7. Example of "Breeze" sequence