Compression of digital hologram sequences using MPEG-4

Emmanouil Darakis[a] and Thomas J. Naughton[a,b]

[a] Department of Computer Science, National University of Ireland - Maynooth,
County Kildare, Ireland;

[b] RFMedia Laboratory, University of Oulu, Oulu Southern Institute, 84100 Ylivieska, Finland

ABSTRACT

Recording and real time reconstruction of digital hologram sequences have recently become feasible. The amount of information that such hologram sequences contain results in voluminous data files, rendering their storage and transmission impractical. As a result, compression of digital hologram sequences is of utmost importance for practical applications of hologram sequences. In the absence of a specific hologram sequence compression technique, a first concern is how a high-performance conventional video compression technique would perform. Such a technique would not be optimized for hologram sequences but would provide a threshold that all hologram sequence compression techniques should reach.

In this paper, the use of MPEG-4 part 2 video coding algorithm for the compression of hologram sequences is investigated. Although the algorithm was originally developed for the compression of ordinary video, we apply it on digital hologram sequences and investigate its performance. For this, appropriate digital hologram sequences are used to asses how the coding algorithm affects their information content. In addition, we investigate whether MPEG-4 interframe coding, which aims to achieve compression by exploiting similarities across adjacent frames of the sequence, offers any advantage compared to intraframe coding, where each frame is coded independently.

Results show that the MPEG-4 coding algorithm can successfully compress hologram sequences to compression rates of $\sim$ 20 : 1 while retaining the reconstruction quality of the hologram sequences.

Keywords: Digital holography, Digital hologram sequences, Holographic data coding, Digital hologram sequence coding

1. INTRODUCTION

Digital holography is a technique where object wavefields are recorded digitally.[1] Numerical reconstruction of these wavefields can reproduce views of the object scene. Digital holography has, so far, been mainly utilized in metrology applications. Nevertheless, digital holograms have several characteristics that make them appealing for imaging applications too.[2] Recording hologram sequences of moving objects has recently been reported.[3,4] In addition, recent advances in display technologies have made possible the optical reconstruction of hologram sequences.[5] In general, the interest for imaging applications of digital holography is increasing constantly.

As the applications of digital holographic imaging increase, so does the need for efficient storage of holographic data. Digital holograms are by nature speckled with very low spacial redundancies making their lossless coding rather inefficient.[6] We have developed a series of lossy methods to compress still holographic data.[6,7] We have also shown that such compression methods can retain the information content of the holograms both in terms of different perspective reconstructions[8] and in terms of phase.[9]

In this paper, the use of MPEG-4 is investigated for the compression of hologram sequences that can be used for imaging applications. MPEG-4 is a collection of methods for the compression of audio and visual data.[10] MPEG-4 includes a broad range of coding tools (called parts) several of which are still under development. It has to be emphasized that whenever we refer to MPEG-4 within this paper, we actually mean MPEG-4, Part 2, Advanced Simple Profile.

Further author information:
Emmanouil Darakis E-mail: emmdarakis@ieee.org,
Thomas J. Naughton E-mail: tomn@cs.nuim.ie.

Holography: Advances and Modern Trends, edited by Miroslav Miler, Miroslav Hrabovský,
Proc. of SPIE Vol. 7358, 735811 · © 2009 SPIE · CCC code: 0277-786X/09/$18 · doi: 10.1117/12.820632

Although MPEG-4 is a coding standard developed for the compression of visual digital data, we investigate its performance for the compression of hologram sequences. The sequences we use to asses the performance of the algorithm have been recorded from four rotating objects. Each hologram sequence is coded and its frames are reconstructed. For quality evaluation, the obtained reconstructions are compared to the reconstructions from the uncompressed data.

We investigate what kind of artifacts MPEG-4 compression causes to the reconstructions of the digital holograms. In addition to this, we investigate whether the interframe coding (coding that exploits similarities among adjacent frames of the sequence) that MPEG-4 uses, offers any advantage over the intraframe coding (coding that compresses each frame independently of adjacent frames). Adjacent frames of hologram sequences appear to have limited similarities to each other, hence it is not clear whether coding of hologram sequences would benefit from interframe coding. If interframe mode does not offer any advantage, then coding each frame of the hologram sequence separately by using previously developed still hologram compression techniques,[6, 7] would be enough. Our analysis shows that the interframe coding that MPEG-4 uses offers higher compression performance compared to intraframe coding, hence holographic data compression can benefit from its use.

All in all, MPEG-4 is shown to be a good coding tool for the compression of hologram sequences, although it was originally developed and optimized for coding of ordinary video sequences.

2. FUNDAMENTALS OF VIDEO CODING

2.1 Block-based motion compensated hubrid coder

Video signals usually contain two main types of redundancies: spatial (within a frame) and temporal (among adjacent frames), the reduction of which is targeted by the video compression algorithms. In order to increase interoperability, MPEG-4 does not standardize the encoder.[10] It only defines the form of the produced bitstream and the procedure to decode it. Nevertheless, all the implementations of MPEG-4 use the basic block-based motion compensated hybrid coder that is briefly described below. Fig. 1 shows a simplified diagram of the coder and the decoder.

There are 2 main coding modes: intraframe where the frame (in this case called I-frame) is coded independently of other frames and interframe where the frame (in this case called P-frame) is coded with respect to previous or future coded frames. Intraframe coding aims to reduce spatial redundancy while interframe coding aims to reduce temporal. In general the frames of a sequence are coded following a predefined sequence.

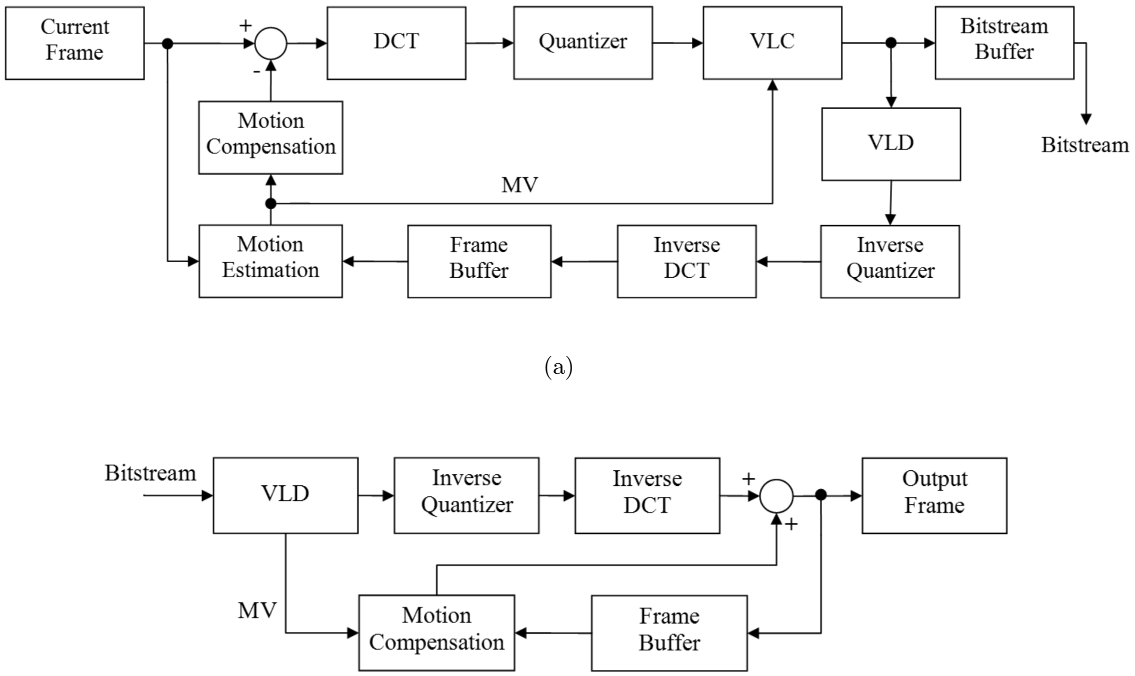
One I-frames is coded first, followed by few P-frames forming a Group Of Pictures (GOP) and the sequence is repeated until the end of the video is reached. For each frame the coding procedure is as follows. Initially the frame to be coded is divided into macroblocks (MB), a 16×16 (or optionally 8×8) group of pixels. Each MB is processed separately depending on the coding mode (intraframe or interframe coding). These modes are described below.

2.1.1 Intraframe coding - spatial redundancy reduction

I-frames are the least compressible frames. They are used as a reference to compress subsequent frames as well as to facilitate change of scenes that might happen in the video sequence and fast seeking during playback. I-frames also prevent long propagation of coding artifacts.

They are coded using 2D Discrete Cosine Transform (DCT),[10] as it is shown in Fig. 1(a). The DCT is applied on the values of the macroblock and the resulting coefficients are quantized using quantization tables. DCT tends to concentrate the power of the signal to low frequency coefficients. As a result, it is expected that after quantization a lot of high frequency coefficients will be zero.

The quantized coefficients are then converted to an 1D vector, coded using a Variable-Length Coding (VLC) and stored in the bitstream. Following this and before the next frame begins to be processed, the current coded frame is decoded by variable-length decoding (VLD), inverse quantizing and inverse DCT. The decoded current frame is stored in a frame buffer and will be used as a reference for the processing of the next frame.



(a)

(b)

Figure 1. Simplified block diagram of a block-based motion compensated hubrid coder.

2.1.2 Interframe coding - temporal redundancy reduction

P-frames are predicted from a previously encoded I- or P-frame called reference frame. As explained in the previous section, the reference frame is available in a frame buffer. The prediction is made by extracting motion information for each MB of the current frame in relation to the reference frame with a procedure called motion estimation. A search is performed for each MB of the current frame within the reference frame to locate the best matching position. The coordinates of the best match position relative to the reference frame define a Motion Vector (MV). Given the reference frame and all the MVs, a prediction of the current frame can be obtained.

The predicted current frame is then subtracted from the actual current frame resulting in a difference (error) frame. The error frame is expected to have values close to zero. To further encode it, the intraframe coder described in Section 2.1.1 is used. In this case, as most of the values of the error frame are expected to be close to zero, a different quantization table than the one used for I-frame coding is utilized. The resulting quantized DCT coefficients together with the MVs are coded by the VLC and stored in the bitstream.

2.2 Decoding

A simplified block diagram of the decoder is shown in Fig. 1(b). In the case of I-frames the quantized DCT coefficients are extracted from the bitstream by VLD. Inverse quantization and inverse DCT follows to restore the current frame. The restored frame is stored in a buffer so that it will be used as a reference frame for the decoding of the following P-frame.

In the case of P-frames, the MVs and the quantized DCT coefficients of the error frame are extracted from the bitstream. By using the MVs and the reference frame, the prediction of the current frame is done. Also, the error frame is restored by inverse quantization and inverse transforming the coded DCT coefficients. By adding the error frame to the predicted frame the current frame is restored.

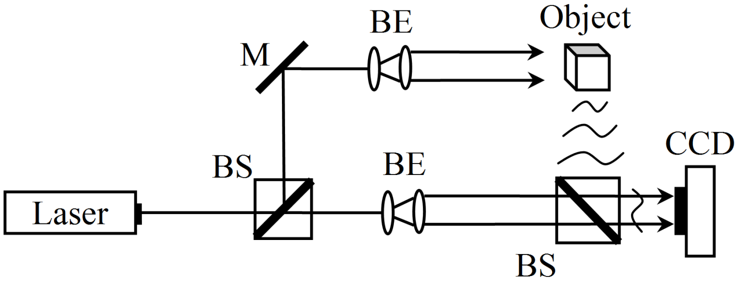


Figure 2. Simplified diagram of the recording setup. BE are beam expanders, BS beam splitters, and M is a mirror.

The basic coding and decoding procedures described above are used in all the MPEG video coders (MPEG-1, MPEG-2, MPEG-4 part 2, and MPEG-4 part 10 AVC/H.264). Compared to the older MPEG-1 and MPEG-2 standards, MPEG-4 introduced some advanced features such as subpixel ME accuracy, 4 motion vectors per MB, direct bidirectional prediction (b-frames), and global motion compensation. Description of these in details falls outside the scope of this paper and hence they are not covered here. Interested reader can find more information in the literature.[10]

For this paper, MATLAB was used to process the holographic data. The videoIO[11] toolbox has been used to control FFmpeg,[12] which is the program that was used to compress the holographic data. The Xvid[13] coder was used for the implementation of the MPEG-4 compression.

3. RECORDING AND PROCESSING OF HOLOGRAM SEQUENCES

3.1 Recording

Fig. 2 shows the setup used to record the hologram sequences. A laser beam of $\lambda = 785$ nm is separated by a beam splitter to two parts. Both are expanded and collimated by two beam expanders. One of the collimated beams shines directly onto the camera providing the reference beam. The other collimated beam illuminates the object to be recorded. The object beam propagates towards the camera through a second beam splitter and interferes with the reference beam. The camera that captures the holograms has $N_x \times N_y = 1392 \times 1040$ square pixels measuring 6.45 μm each, with a resolution of 16bits per pixel.

The object is positioned on a rotational stage which rotates with a step of 4°. The camera captures one hologram for each step and stores it in an uncompressed image file. Following recording, the resolution of the image files is reduced to 8 bits per pixel, as 8 bit per pixel is the only resolution supported by the MPEG-4 implementation we used. These 8 bit per pixel holograms are considered as the original uncompressed data. The size of the uncompressed hologram sequence can be calculated in bytes as:

$$S = N_x \times N_y \times L, \tag{1}$$

where L is the length of the hologram sequence in frames.

3.2 Reconstruction

To reconstruct the uncompressed hologram sequence each frame is filtered with a high-pass filter to reduce the zero order term of the reconstruction,[1] and then is propagated to the desired distance using the Fresnel transform.[1] Figs. 3 (a), (c), (e), and (g) show reconstructed frames from the hologram sequences we used for the numerical analysis. Four object scenes are used, namely: a stormtrooper, a die, a cat, and two screws. Following reconstruction, no speckle or other form of noise reduction is used in this paper.

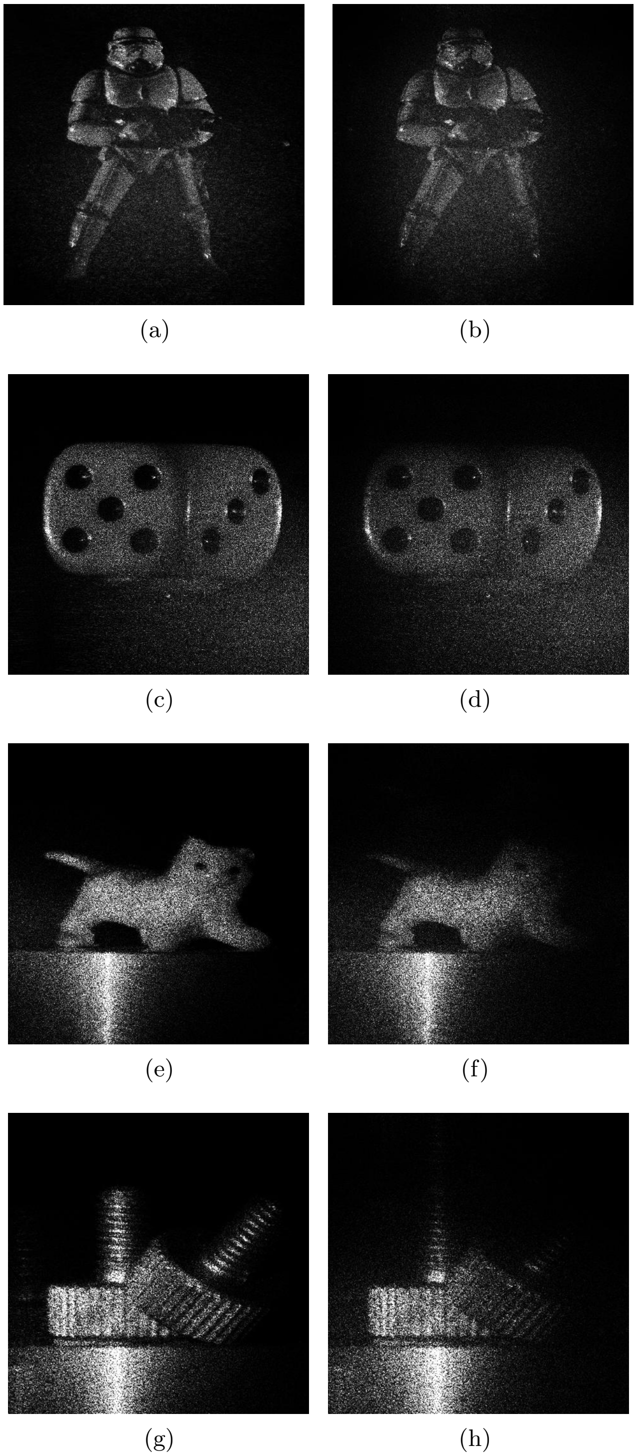


Figure 3. Reconstructions of the hologram sequences used for the numerical analysis. (a), (c), (e), and (g) are reconstructions from the uncompressed data and (b), (d), (f), and (h) are reconstructions from the compressed sequences.

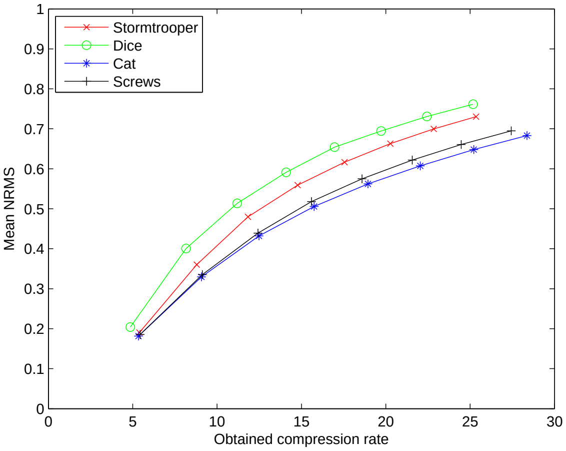


Figure 4. NRMS vs compression rate numerical results for the holograms

3.3 Coding

MPEG-4 is used to compress the uncompressed hologram sequences. Before compression a target compression rate needs to be defined. In this paper we have used target compression rates of $C = 5, 10, 15, ...40$. The coder creates Audio Video Interleave (AVI) files of the compressed hologram sequences.

Following coding, the frames of the compressed sequence are extracted from the AVI files. Then, they are filtered with a high-pass filter to reduce the zero order term from the reconstruction and are reconstructed as described in section 3.2. Figs. 3 (b), (d), (f), and (h) show the reconstructions of the scenes shown in Figs. 3 (a), (c), (e), and (g), respectively after the corresponding sequences have been compressed.

4. NUMERICAL RESULTS

4.1 Coding performance

In order to evaluate the coding performance, the hologram sequences are compressed to the target compression rates. Then, in order to quantify the quality of the reconstructions obtained from the compressed sequences the Normalized Root Mean Square (NRMS) error metric is used. NRMS is defined as:[6]

$$NRMS = \left[\sum_{N_x} \sum_{N_y} \left(|U_d|^2 - |\tilde{U_d}|^2 \right)^2 / \sum_{N_x} \sum_{N_y} \left(|U_d|^2 \right)^2 \right]^{1/2} \tag{2}$$

where U_d and $\tilde{U_d}$ are the object plane complex wavefronts from the original and the compressed data, respectively. NRMS is calculated for each frame of the sequence using Eq. 2 and its mean value is used. Also, the obtained compression rate is calculated as $c = S/s$, where s is the size of the compressed AVI file in bytes. Fig. 4 shows the numerical results obtained for the four hologram sequences that were studied. It can be seen that the compression performance has similar trend for all the sequences.

The reconstructions shown in Figs. 3 (b), (d), (f), and (h) are from sequences coded to an obtained compression rate of $\sim 20:1$. The corresponding NRMS error values are: 0.66, 0.69, 0.61, and 0.62 for the stormtrooper, dice, cat, and screws holograms, respectively. As it can be seen, compression affects the reconstructions in two ways. First, the speckle noise is increased. Secondly, it can be noticed by closer inspection that some parts of the objects have become darker. In particular, the head and the legs of the stormtrooper, the top side of the die, the head and the tail of the cat, and the spiral parts of the screws appear darker after compression. These parts correspond to the edges of the objects.

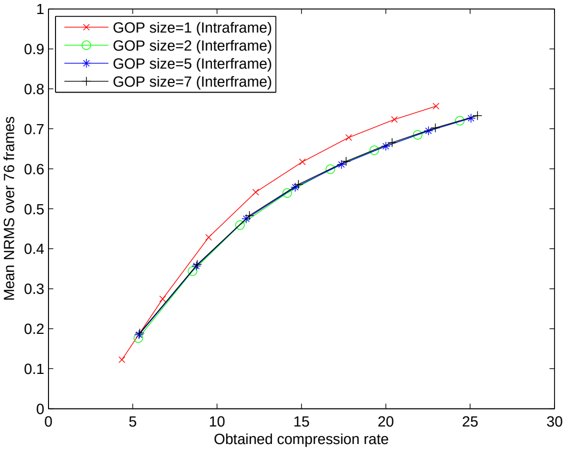


Figure 5. Comparison of numerical results for intraframe (GOP size=1) and interframe (GOP size=2, 5, and 7) compression for the stormtrooper hologram sequence.

Light diffracted from the object's edges has the highest incident angle with the reference wave producing the highest frequencies (smallest fringe spacing) in the holograms.[1] It is apparent that compression operates as a low-pass filter, reducing the high frequency content of the hologram and causing the darkening of the objects' edges. MPEG-4 is a lossy coding technique and as such, it is expected that the practical obtained compression rates will be limited. Fig. 3 shows that compression rates of $\sim 20:1$ with small degradation of the reconstructed quality are obtainable.

4.2 Intra-/Inter-frame coding

MPEG-4 assumes that similarities across subsequent frames of a sequence exist and exploits them to achieve compression. However, this assumption is not clear in the case of digital hologram sequences. In this section we investigate whether the MPEG-4 algorithm can find such similarities within hologram sequences. If not, then the use of interframe coding would be unjustified, as intraframe coding would suffice.

For this, we force the coder to compress a hologram sequence using different GOP sizes, namely GOP size of 1 for intraframe coding and GOP sizes of 2, 5, and 7 for interframe coding. Fig. 5 shows the obtained numerical results for the stormtrooper hologram. The numerical results for the rest of the hologram sequences have similar trend. As it can be seen, coding performance is better in the case of interframe coding especially for higher compression rates. This means that the MPEG-4 algorithm can successfully identify and exploit similarities across different frames of the hologram sequence. Hence, even in the case of hologram sequences, the interframe coding that MPEG-4 offers, yields better performance compared to the intraframe coding. This indicates that hologram sequences have temporal redundancies and as a result, MPEG-4 would probably yield higher coding performance compared to methods based on the compression of the frames of a sequence independently using still interferograms coding techniques.[6]

Also, from Fig. 5 it can be seen that GOP size does not affect the performance of interframe coding, as the coding performance remains practically the same for all the GOP size values other than one.

4.3 Conclusions and Discussion

In this paper we investigate the use of MPEG-4 for the compression of hologram sequences. Digital hologram sequences of four rotating objects have been recorded and used to assess the coding performance of the algorithm.

By visually analyzing the reconstructions obtained from the coded sequences it becomes apparent that coding tends to increase the speckle noise and to darken the edges of the objects. This is probably because coding operates as a low-pass filtering reducing the high frequency content of the holograms which is caused by the objects' edges.

In order to quantify the quality performance, the NRMS quality metric has been used in conjunction with the obtained compression rate. It is shown that compression rates of up to $\sim$ 25 : 1 can be obtained while retaining the average NRMS error below $\sim$ 0.7.

It is also shown that MPEG-4 interframe coding offers higher performance compared to independent coding the sequence's frames (intraframe coding). This indicates that some temporal redundancy among the frames of a hologram sequence exists. Still interferogram coding methods[6] cannot exploit such redundancy and as a result, it is expected that MPEG-4 is better suited for the compression of hologram sequences than these methods.

MPEG-4 has been shown to be a good tool for the compression of hologram sequences. Apart from its coding performance, MPEG-4 video coding inherently supports color, as a result, the proposed method, has great potential for the compression of color digital hologram sequences.

ACKNOWLEDGMENTS

This work was funded by the Irish Research Council for Science, Engineering and Technology, and Science Foundation Ireland, under the National Development Plan through grant no. PD/2007/11 and a European Commission Marie Curie Intra-European Fellowship.. The authors would also like to thank Nitesh Pandey for assisting with the recording setup.

REFERENCES

[1] Schnars, U. and Jueptner, W., [*Digital Holography: Digital Hologram Recording, Numerical Reconstruction, and Related Techniques*], Springer, Berlin (2005).

[2] Frauel, Y., Naughton, T. J., Matoba, O., Tajahuerce, E., and Javidi, B., "Three-dimensional imaging and processing using computational holographic imaging," *Proc. of the IEEE* **94**(3), 636–653 (2006).

[3] Dyomin, V. V. and Olshukov, A. S., "Digital holographic video of plankton," **7073**, 70732B, SPIE (2008).

[4] Sun, H., Song, B., Dong, H., Reid, B., Player, M. A., Watson, J., and Zhao, M., "Visualization of fast-moving cells in vivo using digital holographic video microscopy," *J. Biomed. Opt.* **13**(1), 014007 (2008).

[5] Blanche, P.-A., Tay, S., Voorakaranam, R., Saint-Hilaire, P., Christenson, C., Gu, T., Lin, W., Flores, D., Wang, P., Yamamoto, M., Thomas, J., Norwood, R., and Peyghambarian, N., "An updatable holographic display for 3d visualization," *J. Disp. Technol.* **4**, 424–430 (Dec. 2008).

[6] Darakis, E., Naughton, T. J., Soraghan, J. J., and Javidi, B., "Measurement of compression defects in phase-shifting digital holographic data," **6311**, 63110B, SPIE (2006).

[7] Darakis, E. and Soraghan, J. J., "Compression of interference patterns with application to phase-shifting digital holography," *Appl. Opt.* **45**(11), 2437–2443 (2006).

[8] Darakis, E., Naughton, T. J., and Soraghan, J. J., "Compression defects in different reconstructions from phase-shifting digital holographic data," *Appl. Opt.* **46**(21), 4579–4586 (2007).

[9] Darakis, E., Singh, V. R., Asundi, A. K., and Soraghan, J. J., "Phase measurement errors due to holographic interferograms compression," **6616**, 661612–11, SPIE (2007).

[10] Pereira, F. and Ebrahimi, T., [*The MPEG-4 Book*], Prentice Hall, Upper Saddle River, NJ (2002).

[11] "http://sourceforge.net/projects/videoio/," (2009).

[12] "http://www.ffmpeg.org/," (2009).

[13] "http://www.xvid.org/," (2009).