Efficient compression of Fresnel fields for Internet transmission of three-dimensional images

Thomas J. Naughton,[1] John B. Mc Donald,[1] Bahram Javidi[2]

[1] *Department of Computer Science, National University of Ireland, Maynooth, County Kildare, Ireland*

[2] *Department of Electrical and Computer Engineering, University of Connecticut, U-157, Storrs, CT 06269, USA*

Abstract. We compress phase-shift digital holograms (whole Fresnel fields) for the transmission of three-dimensional images. For real-time networking applications, the time required to compress can be as critical as compression rate. We achieve lossy compression through quantization of both the real and imaginary streams, followed by a bit packing operation. Compression losses in the reconstructed objects were quantified. We define a speedup metric that combines space gains due to compression with temporal overheads due to compression routine and transmission serialization. We empirically verify transmission speedup due to compression, using a special-purpose Internet-based networking application.

OCIS codes: 90.1760 Computer holography, 100.6890 Three-dimensional image processing, 999.9999 Data compression, 100.2000 Digital image processing, 100.0100 Image processing

1. Introduction

Digital holography[1–9] is one of several possible techniques for three-dimensional (3D) imaging.[10] Many existing 3D imaging and processing techniques are based on the explicit combination of several 2D perspectives (or light stripes, etc.) through digital image processing. Multiple perspectives of a 3D object can be combined optically, in parallel, and stored together as a single complex-valued digital hologram. Digital holography has seen renewed interest with the recent development of megapixel digital sensors with high spatial resolution and dynamic range. Their digital nature means that these holograms are in a suitable form for processing or transmission.

We record in-line digital holograms, recover the whole Fresnel field using a technique called phase-shift interferometry[3,6,8] (PSI), and introduce a third step, that of digital compression and decompression.[11] Each Fresnel field encodes multiple views of the object from a small range of angles. Different perspectives of the object can be reconstructed by extracting appropriate regions[12,13] from the field and applying a numerical propagation technique.[7–9] Real-time optical reconstruction techniques have also been demonstrated.[14,15]

Our digital Fresnel fields have dimensions 2028×2044 pixels and in their native format store 8 bytes of amplitude information and 8 bytes of phase information for each pixel. We would like to compress[16] these fields for more efficient storage and transmission. Compression of Fresnel fields (and digital holograms in general) differs to image compression principally because our fields store 3D information in complex-valued pixels, and secondly because of the inherent speckle content which gives them a white-noise appearance. It is not a straightforward procedure to remove the holographic speckle because it actually carries 3D information. The noisy appearance of digital Fresnel fields, and digital holograms in general, causes lossless data compression techniques (such as Lempel-Ziv-Welch, Huffman, and Burrows-Wheeler) to perform poorly.[11] The use of lossy compression techniques seems essential. Digital hologram compression techniques based on Fourier-domain processing have been demonstrated.[11,17] These block-based techniques tend to introduce localized noise at the boundaries of nonoverlapping blocks. A wavelet-based technique might be more effective. Ding et al.[18] perform compression of digital holograms through wavelet decomposition and selection of the principal wavelet basis components appropriate for

their pattern recognition application. Liebling et al.[19] have developed a wavelet-based reconstruction technique for digital holograms. A course-scale reconstruction or some wavelet-domain quantization in their scheme could also form the basis for a noise removal and/or compression technique.

In this paper, we apply quantization directly to the complex-valued pixels. Quantization and phase quantization have been applied successfully to Fourier and holographic data in the past.[11,17,20,21] We apply a two-stage compression technique based on complex-domain quantization and bit packing. This introduces a third reason why compression of Fresnel fields (and digital holograms in general) differs to compression of digital images; a change locally in a Fresnel field will, in theory, affect the whole reconstructed object. We are not interested in how compression noise affects the decompressed Fresnel field itself, only how compression noise affects subsequent object reconstruction. In this paper, we use a reconstructed-object-plane RMS metric to quantify the quality of our decompressed Fresnel fields.

Compression will permit Fresnel fields to be stored more efficiently. In terms of their transmission, however, there is at least one other property that should be evaluated when comparing compression strategies. We need to know the time it takes, relative to the transmission time, to compress and uncompress the field in order to decide on a compression mechanism for transmission. In particular, it might not even be advantageous to compress the data prior to transmission if the latency caused by the compression routine is significant relative to the average uncompressed transmission time. We consider the case where it is not possible to compress the data in advance, for example in a real-time imaging and transmission application. We use a

measure called speedup to quantify the effectiveness of our compression technique in terms of both space and time resources. Our data is obtained using a special-purpose freely-accessible Internet application that, through the integration of compression and transmission routines into a single application, was able to reliably measure compression time relative to transmission time.

In Sect. 2, we describe how the fully-complex Fresnel fields are captured using PSI. The networking system is detailed in Sect. 3, and the compression algorithm and compression performance in Sect. 4. Finally, in Sect. 5, we present the results of speedup experiments performed with an implementation of the networking system.

2. Phase-shift digital holography

We record Fresnel fields with an optical system based on a Mach-Zehnder interferometer (see Fig. 1). A linearly polarized Argon ion (514.5 nm) laser beam is expanded and collimated, and divided into object and reference beams. The object beam illuminates a reference object placed at a distance of approximately $d = 350\,\text{mm}$ from a 10-bit 2028×2044 pixel Kodak Megaplus CCD camera. Let $U_0(x, y)$ be the complex amplitude distribution immediately in front of the 3D object. The linearly polarized reference beam passes through half-wave plate RP_1 and quarter-wave plate RP_2. This beam can be phase-modulated by rotating the two retardation plates. Through permutation of the fast and slow axes of the plates we can achieve phase shifts of 0, $-\pi/2$, $-\pi$, and $-3\pi/2$. The reference beam combines with the light diffracted from the object and forms an interference pattern in the plane of the camera. At each of the four phase shifts we record an interferogram. We use these four real-valued im-

ages to compute the camera-plane complex field by PSI.[3,8] We call the camera-plane complex field the Fresnel field, and denote it $H_0(x, y)$. Fresnel fields captured using this architecture have themselves been referred to as digital holograms,[9,22–24] given a generalized definition of the term hologram.

A Fresnel field $H_0(x, y)$ contains sufficient amplitude and phase information to reconstruct the complex field $U(x, y, z)$ in a plane in the object beam at any distance z from the camera. This can be calculated from the Fresnel approximation[13] as

$$U(x, y, z) = \frac{-\mathrm{i}}{\lambda z} \exp\left(\mathrm{i}\frac{2\pi}{\lambda} z\right) H_0(x, y) \star \exp\left[\mathrm{i}\pi \frac{(x^2 + y^2)}{\lambda z}\right] \quad , \tag{1}$$

where λ is the wavelength of the illumination and $\star$ denotes a convolution operation. At $z = d$, and ignoring errors in digital propagation due to discrete space (pixelation) and rounding, the discrete reconstruction $U(x, y, z)$ closely approximates the physical continuous field $U_0(x, y)$.

Furthermore, as with conventional holography,[12,13] a windowed subset of the Fresnel field can be used to reconstruct a particular view of the object. As the window explores the field a different angle of view of the object can be reconstructed. The range of viewing angles is determined by the ratio of the window size to the full CCD sensor dimensions. Our CCD sensor has approximate dimensions of 18.5×18.5 mm and so a 1024×1024 pixel window has a maximum lateral shift of 9 mm across the face of the sensor. With an object positioned $d = 350$ mm from the camera, viewing angles in the range $\pm 0.74°$ are permitted. Smaller windows will permit a larger range of viewing angles at the expense of image quality at each viewpoint.

3. Network

We can evaluate compression algorithms in terms of both space and time resource usage by using a measure called speedup. Speedup s is defined as $s = P_{\mathrm{u}}/P_{\mathrm{c}}$ where P_{u} is the time required to process and transmit the uncompressed Fresnel field and P_{c} is the time required to process, and transmit the compressed field. In order to measure speedup of a compression system that resides over a public wide-area network we have found that the following three requirements should be met. Firstly, due to the temporal fluctuations in bandwidth over wide-area networks, it is necessary to average over a large number of timing measurements. Secondly, in order to accurately measure compression and decompression times, the compression routines should be removed from their controlled prototyping environment and executed in a real-world setting. Finally, both the networking software and compression software should be integrated so that meaningful conclusions can be drawn from the relative performance of the transmission and compression components of the system.

We have constructed an Internet-based Fresnel field compression application in order to measure reliably and accurately the interaction between compression times and transmission times. This client-server application and associated compression algorithms were written with Java™ (Sun Microsystems, Inc). This allowed us to develop a platform-independent environment for experimentation over the Internet. Platform-independence ensures that the system supports any architecture that runs a Java virtual machine, and is suitable for heterogeneous environments (the server needs no knowledge of a client's computer architecture or operating system to com-

municate). As such, we ensure as much as possible the repeatability and relevance of our results for a wide range of Internet set-ups.

An overview of the operation of the networking application is shown in Fig. 2. Multiple clients, through their user interfaces, access the server and request particular views of 3D objects stored as Fresnel fields. The server responds by providing the appropriate window of pixels, and the clients reconstruct views of the 3D objects locally. To build the communication component for the client-server application we used Java's remote method invocation (RMI) facilities. RMI allows applications running on different machines to communicate with each other in an efficient and transparent manner. Central to providing this ease of communication is Java serialization. The term serialization refers to the packaging of volatile data-structures into persistent bit-streams which can then be written to permanent storage or transmitted across a communications link. Our client-server system functions over any network (local, wide, wireless) that supports IP (Internet Protocol).

The internal operation of the clients and server from Fig. 2 are shown in Fig. 3. A request for a particular view of a particular 3D object is passed from the user interface, through the client, to the server (stage 0 in Fig. 3). The server extracts the appropriate window from the Fresnel field stored on disk (stage 1), and formats the field for transmission (stage 2). Java is an object-oriented language. As such, Fresnel fields are stored as generalized hologram objects, in a data format that allows efficient manipulation by specialized complex-valued digital hologram processing algorithms, and complete with the functionality to read/write them from/to disk, display them, and reconstruct their 3D objects. Formatting is required to streamline the hologram

object for transmission, by removing all functionality and converting the field data into two compact 1D arrays of real and imaginary values. The field is compressed (stage 3) to a stream of bytes as explained in Sect. 4. The particular compression algorithm and compression parameters to be employed by the server are specified by the client at stage 0.

In stages 4 and 5 the compressed Fresnel field data is serialized and transmitted. The server collects timing information (time to read from disk, format, and compress) and transmits this with the field. The server responds to the client through a dedicated communication channel that is set up through RMI, and closed immediately afterwards. The client, on receipt of the field data, performs a deserialization operation (stage 6) and decompresses the byte stream as explained in Sect. 4. The unformatting operation converts the separate real and imaginary arrays of field data into a hologram object that has the functionality to be propagated numerically and/or displayed as an intensity image at the user interface (stages 7 through 10). Once again, timings are taken by the client for each of its operations. The client-side interface of the timing application is shown in Fig. 4. Our Internet-based Fresnel field/hologram compression and timing application is accessible online.[25]

4. Compression

In our experiments, the Fresnel field window was compressed by the server (stage 3 in Fig. 3) using a two-step process. The field data was first quantized at a particular resolution and then compressed using a bit packing technique. Each pixel of the field data required two data values (real, imaginary). Quantization levels were chosen to be

symmetrical about zero; as a result b bits encode $(2^b - 1)$ levels. For example, two bits encode levels $\{-1, 0, 1\}$, three bits encode levels $\{-1, -2/3, -1/3, 0, 1/3, 2/3, 1\}$, and so on. By reducing the number of values available to each pixel we make it possible to reduce the number of bits required to describe it.

The first step consisted of a rescale and quantization operation on the field window. It is defined for individual pixels as

$$H_0'(x, y) = \text{round}\left[H_0(x, y)\sigma^{-1}\beta\right] \tag{2}$$

and is applied to each $x \in [1, N_x], y \in [1, N_y]$, where $\beta = 2^{(b-1)} - 1$, b represents the number of bits per real and imaginary value, and

$$\begin{aligned} \sigma \;\; = \;\; & \max\{|\min\left[\text{Im}(H_0)\right]|, |\max\left[\text{Im}(H_0)\right]|, \\ & |\min\left[\text{Re}(H_0)\right]|, |\max\left[\text{Re}(H_0)\right]|\}. \end{aligned} \tag{3}$$

Here, N_x and N_y are the number of samples in the x and y directions, respectively, $\max(\cdot)$ returns the maximum scalar in its argument(s), $|\cdot|$ denotes absolute value, and $\text{round}(\alpha)$ is defined as $\lfloor \alpha + 0.5 \rfloor$. After quantization, each real and imaginary value will be an integer in the range $[-\beta, \beta]$.

The actual data reduction was performed in the second step, where the appropriate b bits were extracted from each value. The parity bit and low-order $(b-1)$ bits from each quantized real and imaginary value were accumulated in a bit buffer and packed into bytes. A Fresnel field window of $N \times N$ pixels requires exactly $\lceil (2N^2 \times b)/8 \rceil$ bytes. The algorithm to perform both steps has a worst-case bN^2 time and $(N^2 + 1)$ space requirement, neglecting constant overhead factors. The field data is also decompressed in linear time, with respect to the number of pixels. Each block of b bits

is extracted from the byte stream and used to reconstitute, alternately, signed real and imaginary values. Although it will not affect the quality of object reconstruction, if desired (for example, for a calculation of compression-induced error), each value could then be rescaled back to the $[-1, 1]$ range by dividing by β.

In our networking system, a Fresnel field H_0 is compressed and then decompressed as H_0', and an object U_0' reconstructed by numerical propagation. To gauge the quality of the compressed reconstruction we compare U_0' with a reconstruction U_0 from an uncompressed Fresnel field (where the real and imaginary values in both fields were in the $[-1, 1]$ range). The quality of the reconstruction from the compressed field was evaluated in terms of normalized RMS (NRMS) difference, calculated from

$$\begin{aligned} D &= \left(\sum_{m=0}^{N_x-1} \sum_{n=0}^{N_y-1} \left[|U_0(m,n)|^2 - |U_0'(m,n)|^2 \right]^2 \right. \\ &\quad \left. \times \left\{ \sum_{m=0}^{N_x-1} \sum_{n=0}^{N_y-1} \left[|U_0(m,n)|^2 \right]^2 \right\}^{-1} \right)^{1/2}, \end{aligned} \tag{4}$$

where (m, n) are discrete spatial coordinates in the reconstruction plane. In order to lessen the effects of speckle noise we examine only intensity in the reconstruction plane and apply a subsampling (spatial integration) operation. Here, $n \times n$ subsampling means that nonoverlapping blocks of $n \times n$ intensity values are integrated to a single value.

Experimental results with a 3D die object are presented. It had approximate dimensions of 5 mm × 5 mm × 5 mm, and was positioned 323 mm from the camera. Figure 5 shows a plot of NRMS difference against number of bits per data value in the Fresnel field of the die, for various degrees of subsampling. Figure 6 shows the reconstructed object intensity for the die object for selected quantization resolutions.

Note that quantization at 4 bits (with 4×4 subsampling) reveals little visible loss in reconstruction quality, and (from Fig. 5) a small NRMS error of 0.056. Compression rate r is defined as

$$r = \frac{\text{uncompressed size}}{\text{compressed size}} \ . \tag{5}$$

Quantization at 4 bits (a reduction from 8 bytes for each real and imaginary value) corresponds to a compression rate of 16.

5. Timing experiments

For our experiments, the server was deployed on a Sun UltraSPARC 10 workstation at National University of Ireland, Maynooth, Ireland, and accessed by a client on a 2 GHz Pentium 4 personal computer at University of Connecticut, USA. For several degrees of quantization, and several Fresnel field window sizes, the client made requests to the server. Forty trials were recorded for each quantization and each window size. We conducted these trials overnight and at weekends during the period 2002.07.13 to 2002.07.25. The means and standard deviations for one such window size (512×512-pixels), and for selected quantizations, are given in Table 1. The complete timing data for window sizes from 8×8 pixels to 512×512 pixels, and for quantizations from 8 bits to 2 bits is available online.[25] Here, transmission time is the sum of the times required to transmit the request from client to server, transmit the field data, and both serialization and deserialization operations. The 3D objects were not reconstructed numerically in the experiments; the complex-valued field was simply displayed as an intensity image. All of the timing experiments were conducted with a 1024×1024-pixel Fresnel field of the die object, but are valid for any 16 byte-per-pixel complex-valued

digital Fresnel field of this size.

A refined speedup metric is defined that avoids the bias effects that would be introduced by including file read, formatting, and imaging operations. These operations are independent of compression strategy and significant in comparison to transmission time. The refined speedup s' is calculated from

$$s' = \frac{\overline{t_{\mathrm{u}}}}{(\overline{c} + \overline{t_{\mathrm{c}}} + \overline{d})}\,, \tag{6}$$

where t_{u} and t_{c} are the uncompressed and compressed transmission times, respectively, c and d are the times to compress and decompress, respectively, and $\bar{\cdot}$ denotes the mean of 40 trials. The timing data used to calculate speedup for two of the window sizes (full speedup data available online[25]) is given in Table 2. A plot of speedup against compression, for several window sizes, is shown in Fig. 7. For windows of size 64×64 pixels, or greater, there is significant speedup (over 2.5) for quantizations of 8 bits or lower. This speedup rises to over 20 for 512×512-pixel windows.

6. Conclusion

Digital hologram (or Fresnel field) compression results[11] undeniably make a case for applying compression strategies to the storage of digital holograms. However, in order to be useful for a real-time 3D object capture, transmission, and reconstruction system, the compression strategies must be shown to admit efficient algorithms that make it advantageous to spend time compressing and decompressing rather than simply transmitting the original data. We have defined a speedup metric that combines gains and losses, in both space and time, due to compression and believe that all

compression algorithms for real-time or time-critical applications could be evaluated in this way.

Our compression technique is based on quantization of real and imaginary components followed by a bit packing operation. The algorithm is efficient (linear in both space and time with respect to the number of holographic pixels) and has been ported for experimentation to a special-purpose Internet-based Fresnel field compression application. Lossy compression of complex-valued digital Fresnel fields through quantization at 4 bits results in a compression rate of 16, for low NRMS errors in the reconstructed object intensity of less than 0.06 (with 4×4 subsampling). Incorporating the compression/decompression delays into the corresponding transmission timings we still observe a speedup of over 9 for window sizes of 512×512 pixels and greater. As a benchmark, we can expect an average compression rate of 4.66 with lossless techniques.[11] Our Internet-based compression application and full timing results are accessible online.[25]

Acknowledgements

The authors wish to thank Yann Frauel and Enrique Tajahuerce for the use of their Fresnel field data.

References

1. J. W. Goodman and R. W. Lawrence, "Digital image formation from electronically detected holograms," *Appl. Phys. Lett.* **11**(2), 77–79, 1967.
2. L. P. Yaroslavsky and N. S. Merzlyakov, *Methods of Digital Holography*, Consultants Bureau, New York, 1980.
3. J. H. Bruning, D. R. Herriott, J. E. Gallagher, D. P. Rosenfeld, A. D. White, and D. J. Brangaccio, "Digital wavefront measuring interferometer for testing optical surfaces and lenses," *Appl. Opt.* **13**, 2693–2703, 1974.
4. U. Schnars and W. P. O. Jüptner, "Direct recording of holograms by a CCD target and numerical reconstruction," *Appl. Opt.* **33**, 179–181, 1994.
5. T.-C. Poon and A. Korpel, "Optical transfer function of an acousto-optic heterodyning image processor," *Opt. Lett.* **4**, 317–319, 1979.
6. J. Schwider, B. Burow, K. E. Elsner, J. Grzanna, and R. Spolaczyk, "Digital wavefront measuring interferometry: some systematic error sources," *Appl. Opt.* **22**, 3421–3432, 1983.
7. L. Onural and P. D. Scott, "Digital decoding of in-line holograms," *Opt. Eng.* **26**, 1124–1132, 1987.
8. I. Yamaguchi and T. Zhang, "Phase-shifting digital holography," *Opt. Lett.* **22**, 1268–1270, 1997.
9. B. Javidi and E. Tajahuerce, "Three-dimensional object recognition by use of digital holography," *Opt. Lett.* **25**, 610–612, 2000.
10. B. Javidi and F. Okano, eds., *Three-Dimensional Television, Video, and Display*

Technologies, Springer, Berlin, 2002.

11. T. J. Naughton, Y. Frauel, B. Javidi, and E. Tajahuerce, "Compression of digital holograms for three-dimensional object reconstruction and recognition," *Appl. Opt.* **41**, 4124–4132, 2002.

12. H. J. Caulfield, ed., *Handbook of Optical Holography*, Academic Press, New York, 1979.

13. J. W. Goodman, *Introduction to Fourier Optics*, McGraw-Hill, New York, second ed., 1996.

14. M. Sutkowski and M. Kujawinska, "Application of liquid crystal (LC) devices for optoelectronic reconstruction of digitally stored holograms," *Optics and Lasers in Engineering* **33**, 191–201, 2000.

15. O. Matoba, T. J. Naughton, Y. Frauel, N. Bertaux, and B. Javidi, "Real-time three-dimensional object reconstruction by use of a phase-encoded digital hologram," *Appl. Opt.* **41**, 6187–6192, 2002.

16. M. Rabbani, ed., *Selected Papers on Image Coding and Compression*, SPIE Milestone Series vol. MS48, SPIE Press, Bellingham, Washington, 1992.

17. T. Nomura, A. Okazaki, M. Kameda, Y. Morimoto, and B. Javidi, "Digital holographic data reconstruction with data compression," *Algorithms and Systems for Optical Information Processing V*, Proceedings of SPIE vol. 4471, 235–242, (San Diego, California), 2001.

18. L. Ding, Y. Yan, Q. Xue, and G. Jin, "Wavelet packet compression for volume holographic image recognition," *Opt. Comm.* **216**, 105–113, 2003.

19. M. Liebling, T. Blu, and M. Unser, "Fresnelets: new multiresolution wavelet bases for digital holography," *IEEE Transactions on Image Processing* **12**, 29–43, 2003.

20. J. W. Goodman and A. M. Silvestri, "Some effects of Fourier domain phase quantization," *IBM J. Res. Develop.* **14**, 478–484, 1970.

21. W. J. Dallas and A. W. Lohmann, "Phase quantization in holograms," *Appl. Opt.* **11**, 192–194, 1972.

22. E. Tajahuerce, O. Matoba, and B. Javidi, "Shift-invariant three-dimensional object recognition by means of digital holography," *Appl. Opt.* **40**, 3877–3886, 2001.

23. Y. Frauel, E. Tajahuerce, M.-A. Castro, and B. Javidi, "Distortion-tolerant three-dimensional object recognition with digital holography," *Appl. Opt.* **40**, 3887–3893, 2001.

24. Y. Frauel and B. Javidi, "Neural network for three-dimensional object recognition based on digital holography," *Opt. Lett.* **26**, 1478–1480, 2001.

25. http://hologram.cs.may.ie/online/jao2003/hologram.html .

List of Figure Captions

Fig. 1 Experimental setup for PSI: BE, beam expander; BS, beam splitter; RP, retardation plate; M, mirror.

Fig. 2 Illustration of the network-independent multiple-client system; U.I., user interface.

Fig. 3 Internal operation of (a) server and (b) client processes.

Fig. 4 Screenshot of client-side of timings application. Top row (l to r): full Fresnel field with window indicated, uncompressed Fresnel field window data, uncompressed timings, and uncompressed reconstruction. Bottom row (l to r): control panel, compressed window, compressed timing information, and compressed reconstruction.

Fig. 5 NRMS difference in the reconstructed intensity plotted against number of bits in each of the Fresnel field's real and imaginary values, for various degrees of subsampling.

Fig. 6 Reconstructed views (with 4×4 subsampling) from a 1024×1024-pixel window from the Fresnel field stored with different quantization resolutions: (a) no quantization, (b) 4 bits, (c) 3 bits, (d) 2 bits of resolution in each real and imaginary value.

Fig. 7 Speedup as a function of increasing compression, for various Fresnel field window sizes.

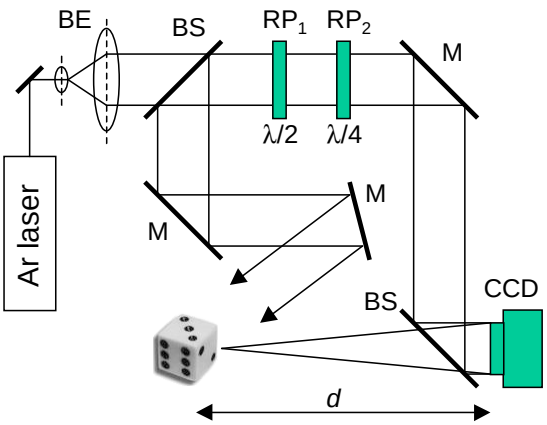

Fig. 1. Experimental setup for PSI: BE, beam expander; BS, beam splitter; RP, retardation plate; M, mirror. tjnF1.eps

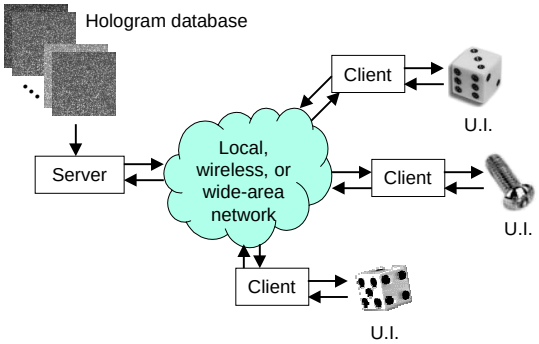

Fig. 2. Illustration of the network-independent multiple-client system; U.I., user interface. tjnF2.eps

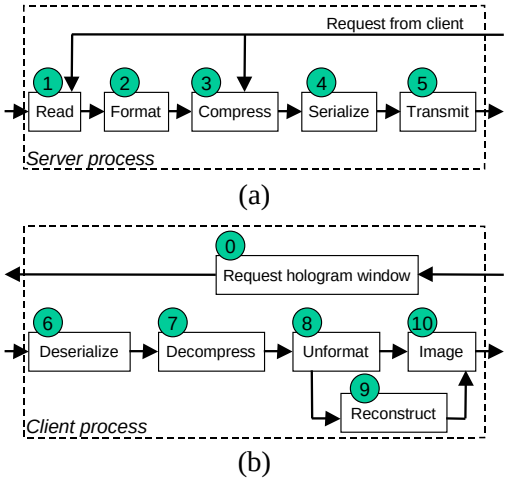

Fig. 3. Internal operation of (a) server and (b) client processes. tjnF3.eps

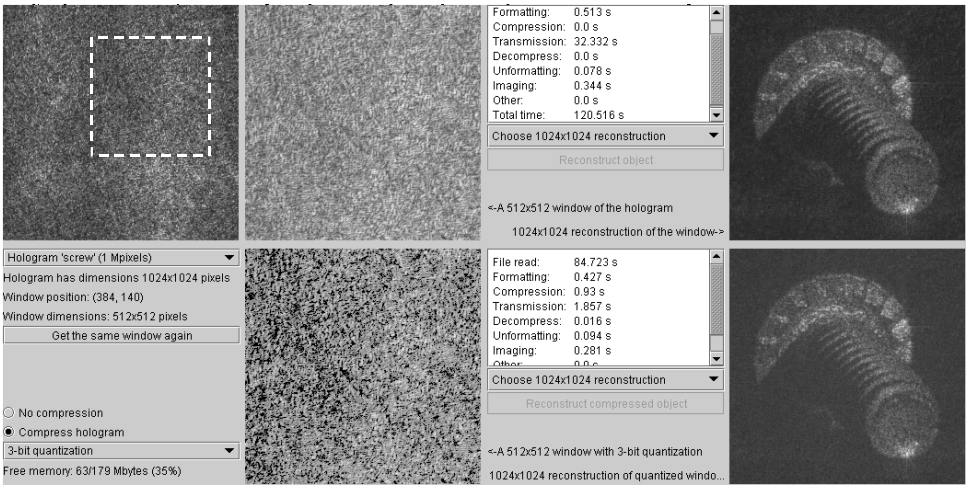

Fig. 4. Screenshot of client-side of timings application. Top row (l to r): full Fresnel field with window indicated, uncompressed Fresnel field window data, uncompressed timings, and uncompressed reconstruction. Bottom row (l to r): control panel, compressed window, compressed timing information, and compressed reconstruction. tjnF4.eps

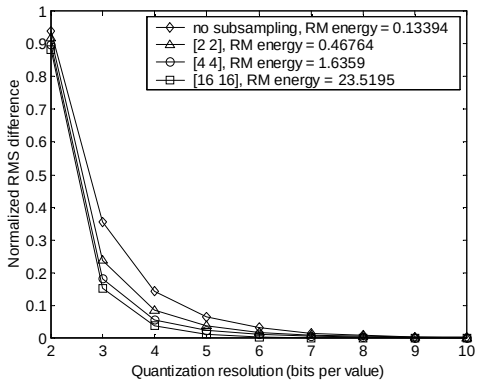

Fig. 5. NRMS difference in the reconstructed intensity plotted against number of bits in each of the Fresnel field's real and imaginary values, for various degrees of subsampling. tjnF5.eps

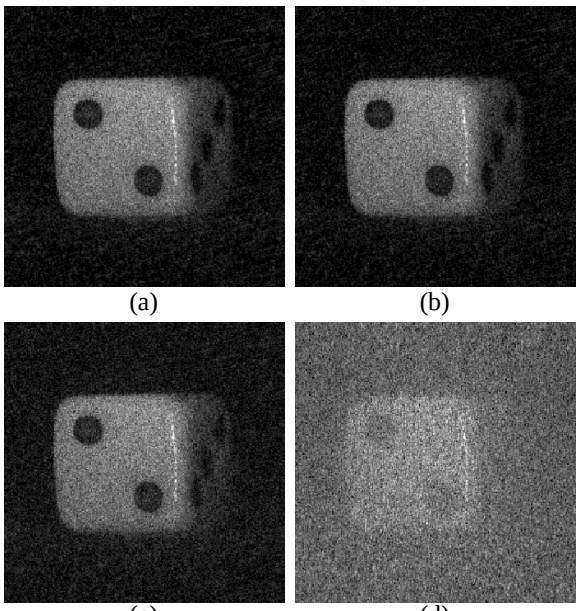

Fig. 6. Reconstructed views (with 4×4 subsampling) from a 1024×1024-pixel window from the Fresnel field stored with different quantization resolutions: (a) no quantization, (b) 4 bits, (c) 3 bits, (d) 2 bits of resolution in each real and imaginary value. tjnF6.eps

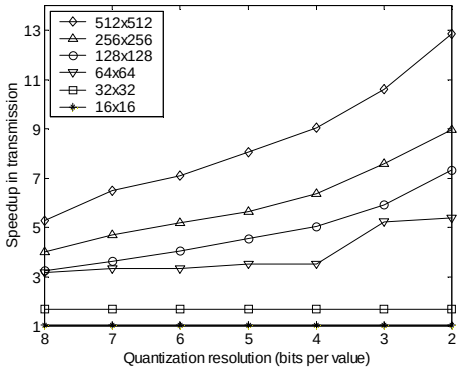

Fig. 7. Speedup as a function of increasing compression, for various Fresnel field window sizes. tjnF7.eps

Table 1. Timings for Fresnel field windows of side length 512 pixels[a]

		No quant-	Quantization[c]			
Stage	Measure[b]	ization	8	6	4	2
Read	μ	88.132	88.095	88.316	88.112	88.033
	σ	1.75	1.75	1.834	1.556	1.579
Format	μ	0.426	0.303	0.283	0.336	0.389
	σ	0.132	0.046	0.015	0.094	0.129
Compress	μ	0	0.937	0.897	0.881	0.782
	σ	0	0.127	0.071	0.112	0.044
Transmit	μ	36.433	5.883	4.182	3.104	2.008
	σ	14.7	1.71	1.237	0.84	0.503
Decompress	μ	0	0.081	0.048	0.039	0.043
	σ	0	0.113	0.007	0.015	0.013
Unformat	μ	0.086	0.083	0.066	0.064	0.067
	σ	0.046	0.046	0.019	0.016	0.021
Image	μ	0.418	0.415	0.42	0.416	0.341
	σ	0.033	0.034	0.029	0.023	0.031
Total time	μ	125.493	95.794	94.209	92.948	91.66
	σ	15.044	2.637	2.385	1.978	1.759

[a]Measured in seconds and rounded up to the nearest 1 ms.

[b]Means and standard deviations of 40 trials.

[c]Number of bits of resolution in each real and imaginary value.

Table 2. Timings used to calculate speedup

Win-dow	Quant-ization[a]	Compress[b] (ms)		Transmit[b] (ms)		Decompress[b] (ms)		Speed-up
		μ	σ	μ	σ	μ	σ	
512	None	0	0	36432.57	14699.4	0	0	1
512	8	936.75	126.32	5882.85	1709.98	80.45	112.5	5.28
512	7	909.25	95.77	4668.03	1346.59	45.7	10.76	6.48
512	6	896.9	70.24	4181.5	1236.7	47.88	6.26	7.11
512	5	899.93	76.77	3568.28	921.8	47.33	16	8.07
512	4	880.5	111.12	3103.08	839.95	38.68	14.67	9.06
512	3	868.2	77.98	2519.73	678.01	44.58	13.11	10.61
512	2	781.75	43.1	2007.9	502.91	42.13	12.38	12.87
64	None	0	0	1239.77	237.44	0	0	1
64	8	10.8	2.38	380.6	105.36	0.5	3.13	3.16
64	7	10.4	1.07	363.2	43.79	1	4.36	3.31
64	6	11.1	2.71	358.95	103.54	1	4.36	3.34
64	5	10.58	1.75	341.63	15.27	1.5	5.27	3.51
64	4	11.35	3.04	339.43	15.93	1	4.36	3.52
64	3	10.33	0.88	225.08	13.99	1.5	5.27	5.23
64	2	10.38	1.58	220.63	17.62	0.01	0.01	5.37

[a]Number of bits of resolution in each real and imaginary value.

[b]μ and σ calculated over 40 trials, rounded up to the nearest 0.01 ms.