Speed up of Fresnel transforms for Digital holography using pre-computation

Nitesh Pandey, Bryan M. Hennelly, Damien P. Kelly and Thomas J. Naughton

Abstract— **We show how the common Fresnel reconstruction of digital holograms can be speeded up on ordinary computers by precomputing the two chirp factors for a given detector array size and then calling these values from memory during the reconstruction. The speedup in time is shown for various hologram sizes. We also run the same algorithm on a Nvidia GPU using Matlab.**

Index Terms—**Digital holography, Optics, imaging, FFT**

I. INTRODUCTION

Digital holography[1] is a fast growing field with applications in Microscopy[2], Metrology[3] and 3-D information processing[4,5] and display[6] to name a few. In digital holography, the holograms are recorded electronically by a CCD target. The real image can be reconstructed from the digitally sampled hologram by numerically propagating the wavefield back to the plane of the object using the theory of Fresnel diffraction. Reconstructions based on the Fresnel transform are widely used for objects for large hologram sizes because they traditionally employ the fast Fourier Transform (FFT) algorithm [7] which reduces the computations required for a NxN matrix from $O(N^2)$ to $O(N\log N)$ steps. The memory consumption and the speed of the FFT makes it a highly important and useful algorithm. As CCD sensor sizes increase in terms of pixel numbers and density, the computational complexity of the reconstruction also increases. Here we aim to show how the reconstruction using the Fresnel transform can be simplified using pre computed chirp and phase factors.

II. FRESNEL TRANSFORM

Consider the Fresnel transform below which describes the relationship between the wavefield at 2 planes $h(x,y)$ and $\Gamma(\xi,\eta)$ separated by a distance d

$$\Gamma(\xi,\eta)=\frac{ia}{\lambda d}\exp\left[-i\frac{\pi}{\lambda d}\left(\xi^2+\eta^2\right)\right]\int_{-\infty}^{\infty}\int_{-\infty}^{\infty}h(x,y)\times\exp\left[\frac{-i\pi}{\lambda d}\left(x^2+y^2\right)\right]\times\exp\left[\frac{i2\pi}{\lambda d}(x\xi+y\eta)\right]dxdy$$

where λ is the wavelength, d is the distance and a is the amplitude. In Digital holography, we deal with discrete representations of this transform since the hologram is discretized at the CCD into NxM samples at intervals of Δx and Δy in the x and y directions. Δx and Δy being the pixel pitches of the sensor. Direct discreteization of the Fresnel integral gives the following

$$\Gamma(m,n)=\exp\left[\frac{-i\pi}{\lambda d}\left(m^2\Delta\xi^2+n^2\Delta\eta^2\right)\right]\times\sum_{k=0}^{M-1}\sum_{l=0}^{N-1}(k,l)\exp\left[\frac{-i\pi}{\lambda d}\left(k^2\Delta x^2+l^2\Delta y^2\right)\right]\exp\left[i2\pi\left(\frac{km}{M}+\frac{ln}{N}\right)\right]$$

Here $\Gamma(\xi,\eta)$ is the matrix of NxM points which describes the amplitude and phase distribution of the real image and $\Delta\xi$ and $\Delta\eta$ are the pixel pitches in the reconstructed images. For a thorough investigation of the resulting numerical algorithm, and the range of d over which it is useful, the reader can consult [8]. We assume we have available to us, a hologram of NxM size recorded with us of an object placed at a distance d from the camera. To calculate the real image field from this hologram, the following method is used.

   i) The first complex chirp is calculated and multiplied by the original hologram. To calculate the chirp NxM computations are needed. This can be speeded up using Vector multiplications (for ex in Matlab) but the number of computations remain the same.

   ii) A Discrete fourier transform of this matrix is taken. This is done using the FFT algorithm.

   iii) The resultant FFT is multiplied with the second chirp factor which must be computed in the same manner as the first.

In order to speed up the Fresnel reconstruction, we note that the input chirp is independent of the input hologram field. The second chirp and the constant factors are also independent of the input. If we precalculate a range of these matrices for different distances and load the corresponding data from memory during reconstruction, we can save $2NxM$ computations. This is advantageous for real time high

resolution digital holographic systems. The time take to load a pre-calculated matrix is significantly less than the time it takes for 'on the fly' calculation. The computation reduces to 2 vector multiplications and the FFT algorithm. The algorithm now becomes

    i) Load the two chirp values for the reconstruction distance d and multiply first chirp by hologram

    ii) Take FFT

    iii) Multiply by second chirp.

We further note that for the case of human visualization of reconstructed holograms, which is of sole importance in holographic 3D-TV[9] and in applications such as holographic endoscopes[10], the phase factors (step iii) can be neglected. This offers an additional speedup on large matrices. To test the improvement in reconstruction time, we used sections of a sample hologram of a macroscopic object of height 3cm recorded in an in-line like geometry on a 1392x1040 CCD ,pixel size 6.45µm (AVT Dolphin) using a 785nm laser. The average times taken for reconstruction and speed up achieved on a computer (Intel Pentium 4 3.0Ghz CPU 1Gb RAM), is shown in the table below.

Hologram size	No loading(time in seconds)	With loading (time in seconds)	Speed up
100x100	0.0613	0.04731	1.30x
200x200	0.1027	0.07023	1.46x
500x500	0.4390	0.26449	1.66x
1000x1000	1.7360	0.99297	1.74x
2000x2000	7.1904	4.22684	1.70x

A large number of these chirp matrices which cover a large distance can be stored permanently in memory and can be accessed by the program whenever the reconstruction is demanded for a particular distance. Recently the use of graphics cards for General purpose computing is becoming popular. Algorithms designed to exploit the parallel many-core capability of the GPU offer a significant speedup (5x-20x) over CPUs. GPGPU as it is called has already been used by a few groups to show the speed of reconstruction of holograms on the GPU architecture [11,12]. Here we show how the reconstruction and preloading works on a NVIDIA Geforce graphics card on a AMD Athlon 64 X2, 2.31Ghz, 2 Gb RAM . We use the Jacket[13] engine for Matlab to run our code on the GPU. The results are shown in Table 2

Hologram size	No loading(time in seconds)	With loading (time in seconds)	Speedup
100x100	0.0629	0.0555	1.13x
200x200	0.0646	0.0522	1.23x
500x500	0.0854	0.0672	1.27x
1000x1000	0.1819	0.0775	2.34x
2000x2000	0.5336	0.2514	2.12x

III. CONCLUSIONS

We have shown the benefits of using a table of precomputed chirp matrix and phase factors on the speed of digital hologram reconstruction on normal CPUs and GPUs. The speedup improves with larger matrices and occurs due to the fact that loading large data from memory takes very little time compared to calculation 'on the fly'. In this paper we have limited our investigation to the direct method (see Equation 2) of computing the Fresnel transforms which is made up of two chirp multiplications and a FFT algorithm [8]. Other methods of reconstruction also require calculation of chirp data which is independent of the input hologram. The convolution approach for example [8] is based on the description of the Fresnel Transform as a chirp multiplication in the Fresnel domain. This method requires calculation of a chirp and two FFTs. Thus precalculating will lead to a time saving of half as that in the direct fresnel approach. The slightly more accurate method based on the Fresnel-Kirchoff transformation requires FFT followed by multiplication by a chirp like function followed by a second FFT. Since this chirp like function (which reduces to a chirp in the paraxial Fresnel approximation) is is independent of the input hologram field, its recalculation will also result in considerable time savings.

REFERENCES

G[1] U. Schnars and W. Juptner, "Digital recording and numerical reconstruction of holograms," Meas. Sci. Technol. 13, 85–101 (2002).

[2] P. Marquet, B. Rappaz, P. J. Magistretti, E. Cuche, Y. Emery, T. Colomb, and C. Depeursinge, "Digital holographic microscopy: a noninvasive contrast imaging technique allowing quantitative visualization of living cells with subwavelength axial accuracy," Opt. Lett. 30, 468-470 (2005).

[3] C. Wagner, W. Osten, S. Seebacher, "Direct shape measurement by digital wavefront reconstruction and multiwavelength contouring," Opt. Eng. 39 79-85 (2000).

[4] B. Javidi and E. Tajahuerce, "Three-dimensional object recognition by use of digital holography," Opt. Lett. 25, 28-30 (2000)

[5] Yann Frauel, Thomas J. Naughton, Osamu Matoba, Enrique Tajahuerce, and Bahram Javidi, "Three-dimensional imaging and processing using computational holographic imaging," Proceedings of the IEEE, vol. 94, no. 3, pp. 636-653, March 2006.

[6] Digital Holography and Three-Dimensional Display Principles and Applications, Poon, Ting-Chung (Ed.) 2006, XIII, 430 p.

[7] The Fast Fourier Transform, Brigham, E.O, New York: Prentice-Hall, (2002)

[8] D. Mas, J. Garcia, C. Ferriera, L.M. Bernardo, "Fast algorithms for free space diffraction patterns calculation," Opt. Comm. 164,233245,(1999).

[9]L. Onural and H. Ozaktas, "Signal processing issues in diffraction and holographic 3DTV," in Proc. EURASIP 13th European Signal Processing Conference (2005).

[10] S. Schedin, G. Pedrini, H. J. Tiziani, and A. K. Aggarwal, "Comparative study of various endoscopes for pulsed digital holographic interferometry," Appl. Opt. 40, 2692-2697 (2001).

[11] L. Ahrenberg, P. Benzie, M. Magnor, and J. Watson, "Computer generated holography using parallel commodity graphics hardware," Opt. Express 14, 7636-7641 (2006).

[12] Tomoyoshi Shimobaba, Yoshikuni Sato, Junya Miura, Mai Takenouchi, and Tomoyoshi Ito, "Real-time digital holographic microscopy using the graphic processing unit," Opt. Express 16, 11776-11781 (2008)

[13] www. accelereyes.com