

Analysis of computational complexity in holographic lens-less projection

A. Kowalczyk,* M. Bieda, M. Makowski, I. Ducin, K. Kakarenko, J. Suszek, and A. Sobczyk

Faculty of Physics, Warsaw University of Technology, Koszykowa 75, 00-662 Warszawa

Received August 05, 2014; accepted September 14, 2014; published September 30, 2014

Abstract—In lens-less projection the illuminating wavefront must be modified, so that after propagation on a certain distance it would form an image on the screen. Formed images will be displayed with a frame rate minimum of 20fps in the case of an animated movie. Thus, the speed of computing is extremely important. Computing might be performed by the central processing unit (CPU) or the graphics processing unit (GPU). Reaching a compromise between computational complexity (as well as the time of computing) and the quality of an obtained image is needed. This leads to a need for a proper iteration number of an algorithm, its parameters and size of computing matrices.

In order to achieve lens-less projection, a Fourier hologram must be displayed on a Spatial Light Modulator (SLM). The hologram has to be computed from the input image which then would be displayed on the screen. To reduce calculation time, computing can be made by the GPU. The GPU together with Compute Unified Device Architecture (CUDA) is able to improve the speed of calculations significantly by parallel computing of the entire calculation matrix [1-4]. To increase calculation speed in digital holography an FPGA (Field Programmable Gate Array) can be also used [5].

In order to achieve a good quality of displayed images with minimal speckle noise, specific algorithms must be used, for example the pixel separation algorithm [6], which is based on interleaving pixels in the projection screen so that the neighbouring object points are formed in different periods of time, and thus the interference between them is minimised.

This paper shows computational complexity for different algorithm parameters and time of calculations on GPUs and CPU.

The scheme of an algorithm is presented in Fig. 1. Three colours are split into three channels and processed separately (a). In the next step a proper frame is selected from the image to apply a pixel separation method [7-8]. In step c) the image is placed in the centre of a computational matrix with dimensions M_x and M_y . Then, the Fresnel hologram is calculated by the Iterative Fourier transform algorithm (IFTA) [3] with I_{it} iterations (d). In step e) a part of the computed hologram is placed into the

output image in a proper position because different light wavelengths are illuminating different areas of the SLM.

Finally, holograms are multiplied by divergent lens transmittance with the focal length depending on the distance to the screen. Holograms also are processed with linear transformation and gamma correction. That processing is needed to achieve the best contrast in projection on an SLM. Thus, three RGB (red-green-blue) channels are merged into one image which is then displayed on the SLM.

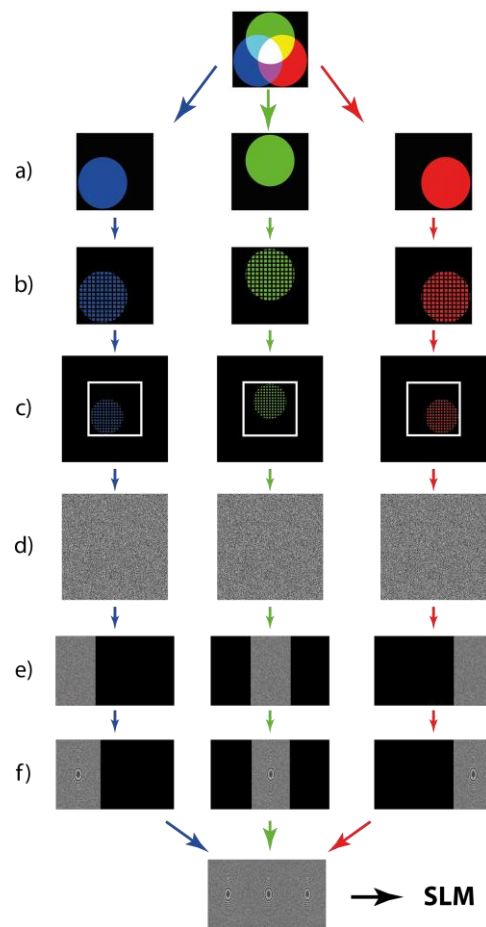


Fig. 1. Scheme of the algorithm.

* E-mail: akowalczyk@if.pw.edu.pl

The tested hardware is presented in Table 1. On the CPU there were used 4 threads for calculations.

Table 1. Tested hardware.

		Cores	Memory
GPU	nVidia GeForce GTX590 (desktop)	1024	1.5 GB
	nVidia GeForce GT755M (laptop)	384	2 GB
	nVidia NVS 5200M (laptop)	96	1 GB
CPU	Intel Core i7-3720QM 2.6GHz	4	8 GB

Computational complexity was tested by the measurement of time needed to compute a single frame for different parameter values.

In Table 2 the time of calculation for different calculation matrix sizes is presented with the IFTA iterations number equal to three. Figure 2 presents the graph which shows the time of calculation in the function of pixels number.

Table 2. Time of calculation for different calculation matrices.

Matrix dim. $M_x \times M_y$	Num. of pixels [Mpx]	Time of computation [ms]			
		GTX590	GT755M	NVS5200M	Intel i7
4096x4096	16.78	158	401	907	46440
2048x2048	4.19	48	95	253	14580
1024x2048	2.10	29	51	133	9620
1920x1080	2.07	42	72	210	-
1024x1024	1.05	19	30	73	7200
512x1024	0.52	13	18	43	6410
512x512	0.26	11	12	28	5550

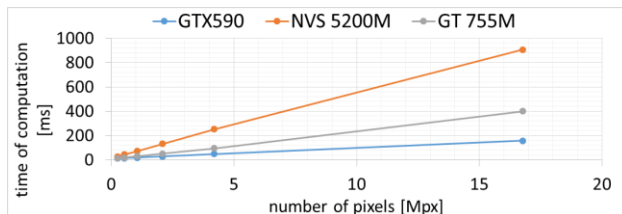


Fig. 2. Time of computation in the function of pixels number.

Table 3 and Fig. 3 present the time of calculation for different IFTA iterations with matrix dimensions equal to 2048px by 2048px.

Table 3. Time of calculation for different IFTA iterations.

	Time of computation [ms]			
	GTX590	GT755M	NVS5200M	Intel i7
$I_{it} = 1$	21	34	83	8960
$I_{it} = 2$	35	65	168	11830
$I_{it} = 3$	48	95	253	14580
$I_{it} = 4$	61	126	338	16860

This results in parallel calculations on the GPU which are much faster than calculations on the CPU. To calculate a Fourier hologram, the IFTA algorithm uses a Fast Fourier Transform (FFT) of the whole matrix many times. We use an IFTA with two FFTs in a single iteration of the algorithm. Although other computations are made, the time needed for an FFT algorithm dominates the whole processing time of a hologram. Parallel computing on the GPU allows to reduce the time needed to compute FFT. This leads to the conclusion that parallel computing must be used to calculate the hologram in real time.

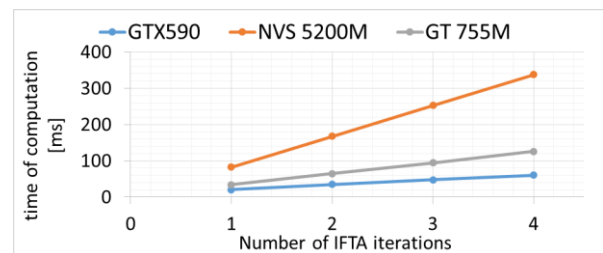


Fig. 3. Time of calculation in the function of IFTA iterations.

The time of calculations is proportional to the number of processing pixels and to the IFTA iterations when computational matrix sizes are the power of 2 (2^k , where k is an integer).

A typical SLM can refresh the displayed frame 60 times per second. It means that a single frame is displayed in 16.5 ms. Standard movies have a frame rate of 24 fps. In this case a single frame is displayed in 41.5 ms. If those two numbers are compared with Table 2 and Table 3 one can see that for many parameter combinations of computation time it is sufficient to saturate the SLM refresh rate or to display moving images at the standard movie speed.

Calculations can be made on matrices greater and smaller than the size of an SLM matrix. If a computational matrix is smaller, the computed pattern is placed in the middle of an SLM and if it is greater, only the central part of a computed hologram is displayed on an SLM.

Figure 4 presents projected images with different parameters. The input image has a resolution of 2048x2048px and it is proportionally scaled when the computational matrix is smaller. All images are an integration of 60 frames during 1 second exposure on the CMOS matrix of a Canon EOS 650D. Images are displayed and registered on the setup presented in [9].

The images in Fig. 4a and Fig. 4b have a similar quality but the time of calculation of the first image is 48ms and the second 19ms. The calculation time of 19ms determines the ability to display animated movie with 53 frames per second.

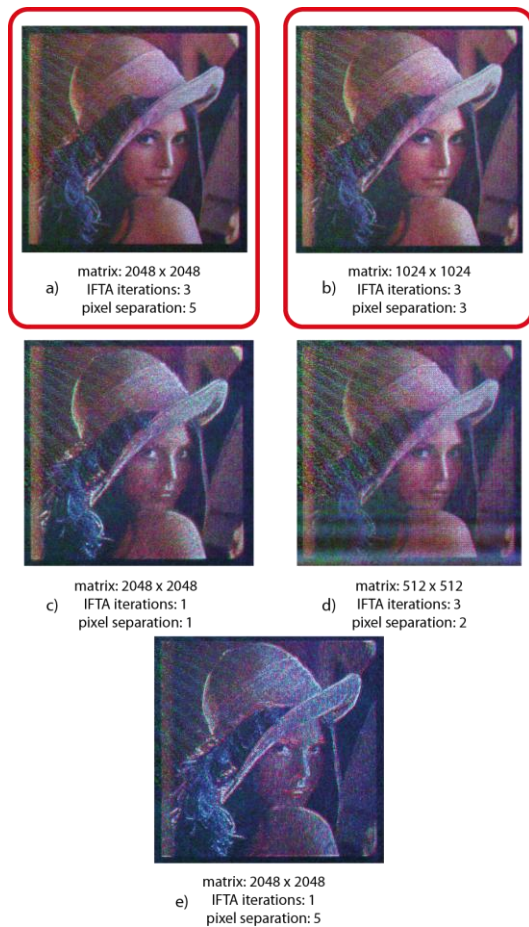


Fig. 4. Projected images.

Consumer electronic devices tend to have smaller dimensions, consume less current and have increased computing power. Thus in near future mobile devices can have enough computing power to display hologram projected images.

In the case of high frame rate movies, faster processing is needed. Within the next few years this kind of processor will be accessed by everybody.

Future effort should be taken to investigate this method of image projection.

This work was supported by the Polish National Center for Research and Development grant LIDER/013/469/L-4/12/NCBR/2013.

References

- [1] S. W. Keckler, W. J. Dally, B. Khailany, M. Garland, D. Glasco, *IEEE Micro*, **31**(5), 7 (2011).
- [2] T. Shimobaba, T. Ito, N. Masuda, Y. Ichihashi, N. Takada, *Opt. Expr.* **18**, 9955 (2010).
- [3] A. Kowalczyk, M. Bieda, M. Makowski, M. Sypek, A. Kolodziejczyk, *Appl. Opt.* **52**, 4743 (2013).
- [4] C.-J. Cheng, W.-J. Hwang, C.-T. Chen, X.-J. Lai, *J. Display Technol.* **10**, 272 (2014).
- [5] Y.-H. Seo, H.-J. Choi, J.-S. Yoo, D.-W. Kim, *J. Systems Architecture* **56**, 27 (2009).
- [6] M. Makowski, *Opt. Expr.* **21**, 29205 (2013).
- [7] T. Utsugi, M. Yamaguchi, *Opt. Expr.* **22**, 17193 (2014).
- [8] F. Wyrowski, O. Bryngdahl, *J. Opt. Soc. Am. A* **5**, 1058 (1988).
- [9] M. Makowski, I. Ducin, K. Kakarenko, J. Suszek, M. Sypek, A. Kolodziejczyk, *Opt. Expr.* **20**, 25130 (2012).