

Real-time computation of diffraction fields for pixelated spatial light modulators

G. Bora Esmer*

Marmara University, Dept. of Electrical and Electronics Engineering, Istanbul, TR-34722, Turkey

*bora.esmer@marmara.edu.tr

Abstract: Fast calculation of diffraction field from a three-dimensional object is an important problem in holographic three-dimensional television. As a result of this, several fast algorithms can be found in the literature, but most them omit pixelated structure of the display device used in the reconstruction process. We propose a fast algorithm for diffraction field calculation from a three-dimensional object for a pixelated display device. Real-time calculations can also be achieved when the proposed algorithm is run on a graphical processing unit. Performance assessment of the algorithm is obtained by the computation time of the diffraction field and the error on the reconstructed object. The proposed algorithm employs a precomputed look-up-table which is formed by one-dimensional kernels. Each kernel in the look-up-table denotes a diffraction field on the display device from a point light source at a specific depth along longitudinal axis. Allocated memory space by the precomputed look-up-table is optimized according to the sampling policy of the depth parameter. Look-up-table formed by uniformly sampled depth parameter provides the same error on the reconstructed object with less number of kernels. Also, optical experiments are conducted and successful results are obtained.

© 2015 Optical Society of America

OCIS codes: (090.1760) Computer holography; (090.2870) Holographic display; (090.5694) Real-time holography.

References and links

1. V. Toal, *Introduction to Holography* (CRC Press Taylor and Francis Group, US, 2012)
2. J.W. Goodman, *Introduction to Fourier Optics* (Mc-Graw-Hill, 1996)
3. M. Born and E. Wolf, *Principles of Optics: Electromagnetic theory of Propagation, Interference and Diffraction of Light* (Cambridge University Press, 1980)
4. G. Saxby, *Practical Holography*, 3rd edition (Taylor and Francis, 2003).
5. M. Lucente, *Diffraction-specific fringe computation for electro-holography*, Ph.D. thesis, (Massachusetts Institute of Technology, Cambridge, MA USA, 1994).
6. S.A. Benton and V.M. Bove Jr, *Holographic Imaging* (Wiley-Interscience, 2008).
7. L. Yaroslavsky, *Digital Holography and Digital Image Processing: Principles, Methods, Algorithms* (Springer, 2013).
8. T. Shimobaba, H. Nakayama, N. Masuda, and T. Ito, "Rapid calculation algorithm of Fresnel computer-generated-hologram using look-up table and wavefront-recording plane methods for three-dimensional display," *Opt. Express*, **18**(19), 19504–19509 (2010).
9. T. Shimobaba, Y. Sato, J. Miura, M. Takenouchi, and T. Ito, "Real-time digital holographic microscopy using the graphic processing unit," *Opt. Express*, **16**(16), 11776–11781 (2008).
10. S.-C. Kim and E.-S. Kim, "Effective generation of digital holograms of three-dimensional objects using a novel look-up table method," *Appl. Opt.*, **47**(19), D55–D62 (2008).

11. S.-C. Kim and E.-S. Kim, "Fast computation of hologram patterns of a 3D object using run-length encoding and novel look-up table methods," *Appl. Opt.*, **48**(6), 1030–1041 (2009).
12. H. Kang, T. Yamaguchi, H. Yoshikawa, S.-C. Kim, and E.-S. Kim, "Acceleration method of computing a compensated phase-added stereogram on a graphic processing unit," *Appl. Opt.*, **47**(31), 5784–5789 (2008).
13. H. Kang, F. Yaras, and L. Onural, "Graphics processing unit accelerated computation of digital holograms," *Appl. Opt.*, **48**(34), H137–H143 (2009).
14. K. Murano, T. Shimobaba, A. Sugiyama, N. Takada, T. Kakue, M. Oikawa, and T. Ito, "Fast computation of computer-generated hologram using Xeon Phi coprocessor," *Comput. Phys. Commun.*, **185**(10), 2742–2757 (2014).
15. C. Chang, J. Xia, and Y. Jiang, "Holographic Image Projection on Tilted Planes by Phase-Only Computer Generated Hologram using Fractional Fourier Transform," *J. Disp. Technol.*, **10**(2), 107–113 (2014).
16. B.J. Jackin, H. Miyata, T. Baba, T. Ohkawa, K. Ootsu, T. Yokota, Y. Hayasaki, and T. Yatagai "A decomposition method for fast calculation of large scale CGH on distributed machines," *Laser Applications to Chemical, Security and Environmental Analysis*, Seattle, Washington United States July 13-17, (2014).
17. D. Leseberg and C. Frère, "Computer generated holograms of 3D objects composed of tilted planar segments," *Appl. Opt.*, **27**, 3020–3024 (1988).
18. T. Tommasi and B. Bianco, "Computer-generated holograms of tilted planes by a spatial frequency approach," *J. Opt. Soc. Am. A*, **10**, 299–305 (1993).
19. N. Delen and B. Hooker, "Free-space beam propagation between arbitrarily oriented planes based on full diffraction theory: a fast Fourier transform approach," *J. Opt. Soc. Am. A*, **15**, 857–867 (1998).
20. G.B. Esmer, "Computation of holographic patterns between tilted planes," M.S. thesis, (Bilkent University, Dept. of Electrical and Electronics Engineering, Ankara, Turkey, 2004).
21. K. Matsushima, "Computer generated holograms for three-dimensional surface objects with shade and texture," *Appl. Opt.*, **44**(22), 4607–4614 (2005).
22. K. Yamamoto, T. Senoh, R. Oi, , and T. Kurita, "8k4k-size computer generated hologram for 3-d visual system using rendering technology," 4th International Universal Communication Symposium (IUCS), (2010).
23. L. Ahrenberg, P. Benzie, M. Magnor, and J. Watson, "Computer generated holograms from three dimensional meshes using an analytic light transport model," *Appl. Opt.*, **47**(10), 1567–1574 (2008).
24. H. Kim, J. Hahn, and B. Lee, "Mathematical modelling of triangle-mesh-modelled three-dimensional surface objects for digital holography," *Appl. Opt.*, **47**(19), D117–D127 (2008).
25. Y.-Z. Liu, J.-W. Dong, B.-C. Chen, H.-X. He, and H.-Z. Wang, "High-speed full analytical holographic computations for true-life scenes," *Opt. Express*, **18**(4), 3345–3351 (2010).
26. W. Lee, D. Im, J. Paek, J. Hahn, and H. Kim, "Semi-analytic texturing algorithm for polygon computer-generated holograms," *Opt. Express*, **22**(25), 31180-31191 (2014).
27. T. Haist, M. Schönleber, and H.J. Tiziani, "Computer-generated holograms from 3D-objects written on twisted-nematic liquid crystal displays," *Opt. Commun.*, **140**, 299–308 (1997).
28. L. Yu and L. Cai, "Iterative algorithm with a constraint condition for numerical reconstruction of a three-dimensional object from its hologram," *J. Opt. Soc. Am. A*, **18**(5), 1033–1045 (2001).
29. J. Rosen and G. Brooker, "Digital spatially incoherent Fresnel holography," *Opt. Lett.*, **32**(8), 912–914 (2007).
30. R.P. Muffoletto, J.M. Tyler, and J.E. Tohtline, "Shifted Fresnel diffraction for computational holography," *Opt. Express*, **15**(9), 5631–5640 (2007).
31. D. Abookasis and J. Rosen, "Computer-generated holograms of three-dimensional objects synthesize from their multiple angular viewpoints," *J. Opt. Soc. Am. A*, **20**(8), 1537–1545 (2003).
32. M. Kovachev, R. Ilieva, P. Benzie, G.B. Esmer, L. Onural, J. Watson, and T. Reyhan, *Three-Dimensional Television: Capture, Transmission, Display*, chapter 15: Holographic 3DTV Displays Using Spatial Light Modulators, 529–556 (Springer-Verlag Berlin Heidelberg, 2008).
33. V. Katkovnik, J. Astola, and K. Egiazarian, "Discrete diffraction transform for propagation, reconstruction, and design of wavefield distributions," *Appl. Opt.*, **47**(19), 3481–3493 (2008).
34. V. Katkovnik, A. Migukin, and J. Astola, "Backward discrete wave field propagation modelling as an inverse problem: toward reconstruction of wave field distributions," *Appl. Opt.*, **48**(18), 3407–3423 (2009).
35. G.B. Esmer, "Fast computation of Fresnel diffraction field of a three dimensional object for a pixelated optical device," *Appl. Opt.*, **52**(1), A18A25 (2013).
36. G.B. Esmer, "Performance assessment of a fast and accurate scalar optical diffraction computation algorithm," *3D Research*, **4**(1), (2013).
37. G.B. Esmer, "Algorithms for Fast Calculation of Scalar Optical Diffraction Field on a Pixelated Display Device", *IEEE AFRICON 2013, Mauritius*, 9-12 September (2013).
38. HoloEye, "PLUTO Phase Only Spatial Light Modulator (Reflective)," <http://holoeye.com/spatial-light-modulators/slm-pluto-phase-only/>
39. W. Gander, and W. Gautschi, "Adaptive Quadrature Revisited", *BIT*, **40**(1), 84-101, (2000).

1. Introduction

Holography is a three-dimensional (3D) visualisation technique that provides the most natural visualization, because it is based on capturing and regenerating the diffracted optical waves from a 3D object [1–6]. When capturing process is performed by an optical setup, we should be careful about stabilizing the assembled optical equipments. Even if the capturing equipments are shifted in quarter wavelength of the light which is used for illumination, hologram of the object can be spoiled. However in computer generated holography (CGH), there is no such a stabilization problem. In CGH, diffraction field of a 3D object can be calculated by using numerical methods and signal processing techniques [1–3, 7]. Then, calculated diffraction field can be printed on a holographic film for still 3D reconstructions or used in driving a dynamic display device to obtain moving 3D reconstructions as in holographic 3D television (H3DTV).

Calculation of diffracted fields from 3D objects should be tailored according to methods used in generation of the objects. Several methods can be employed for that purpose. One of them is based on having a set of points which are distributed over the space. 3D objects formed by a set of points can be called as point cloud object. Then, superposition of the diffraction fields emitted by each point will give the diffraction field of the object [5, 8–13]. Another method used for generation of a 3D object is based on combining small planar patches. Again, superposition of diffracted fields from each patch will give the diffraction field of the object [17–26]. Yet another method used in forming a 3D object is based on having multiple two-dimensional (2D) cross-sections of the object along longitudinal axis. Then, superposition of diffracted fields from those 2D cross-sections will be taken as the diffraction field of the 3D object [27–31]

Fast calculation of the diffraction field from a 3D object can be achieved by employing fast methods such as fast Fourier transform (FFT), look-up-table (LUT) and segmentation of diffracted field emitted from each elementary building block of 3D object. In [8, 22], FFT is used for decreasing the calculation time of diffraction field between two planes. Precomputed LUTs are also useful to obtain fast calculations [5, 8, 10, 11, 15]. By segmenting the diffraction field of a point light source, fast calculation of the diffracted field from 3D object can be achieved [12, 13]. Further improvements on the computation time can be obtained by running the diffraction field calculation algorithms in parallel. Graphical processing units (GPUs) are one of the most convenient hardware for that purpose [9, 13, 14, 16].

Diffraction field of a 3D object should be calculated rapidly to obtain H3DTV. Hence, the computational complexity of diffraction field calculation process should be decreased and in the meantime quality of the reconstructed object should be improved as well. The diffraction field calculation algorithms presented in [32–34] provide improvements on the quality of the reconstructed objects by taking into consideration the pixelated structure of the spatial light modulators (SLMs) which are used as dynamic display devices in reconstruction process, but their computational complexities are higher than the fast algorithms mentioned above. As a result of this, real-time calculation of diffracted field from a 3D object may not be achieved when the algorithms presented in [32–34] are employed in H3DTV.

The algorithms presented in [35–37] can be candidates for overcoming computation time and quality problems in H3DTV. Although, those algorithms provide improvements on quality of the reconstructed object by taking into consideration the pixelated structure of the SLM, they may not be fast enough to achieve real-time calculations. In this work, we proposed an algorithm utilizing a LUT which is optimized for parallel processing on a GPU to achieve real-time calculations. Precomputed LUT is formed by one-dimensional (1D) kernels and each kernel is calculated according to the pixel structure of the employed SLM. Several numerical and optical experiments are conducted and successful reconstructions are obtained.

2. Calculation of diffraction pattern used in driving SLM with pixelated structure

Diffraction field of a 3D object can be calculated by using numerical analysis methods and signal processing techniques, thus it is possible to obtain holographic patterns in computer environment. Calculation of the diffraction field depends on the method used in generation of the 3D object. If the object is formed by a set of points which are distributed over the space, then superposition of diffraction fields emitted by those points will be taken as the diffraction field of the object. In this work, 3D objects are formed as point clouds, because it is one of the simplest method to obtain 3D object in computer environment.

The diffraction field of a point cloud object over a planar surface can be calculated as

$$\psi(\mathbf{r}_0) = \sum_{l=1}^L \psi(\mathbf{r}_l) h_F(\mathbf{r}_0 - \mathbf{r}_l) \quad (1)$$

where $\psi(\mathbf{r}_0)$ and $\psi(\mathbf{r}_l)$ are the diffraction field over SLM and diffraction field at l^{th} point of 3D object, respectively. Surface of SLM is spanned by the position vector $\mathbf{r}_0 = [x, y, 0]$ and locations of the point that forms 3D object are shown by $\mathbf{r}_l = [x_l, y_l, z_l]$. The variable $h_F(\mathbf{r})$ stands for the diffraction field emitted by a point light source over SLM under Fresnel approximation,

$$h_F(\mathbf{r}) = \frac{e^{jkz}}{j\lambda z} e^{j\frac{k}{2z}(x^2+y^2)} \quad (2)$$

where $\mathbf{r} = [x, y, z]$, k is the wave number and λ is the wavelength of the light source used in illumination.

The calculated diffraction field over SLM plane will be used for driving SLM. Then, SLM will be illuminated by a plane wave and reflected wave will provide reconstruction of an optical replica of the 3D object. Commercially available SLMs may have square pixels with high filling factors as 93% [38]. Therefore, there will be no significant difference on the calculated diffraction field when filling factor is taken as 100%. An illustration of the pixel structure of the simulated SLM can be seen in Fig. 1. To improve the quality of the reconstructed object,

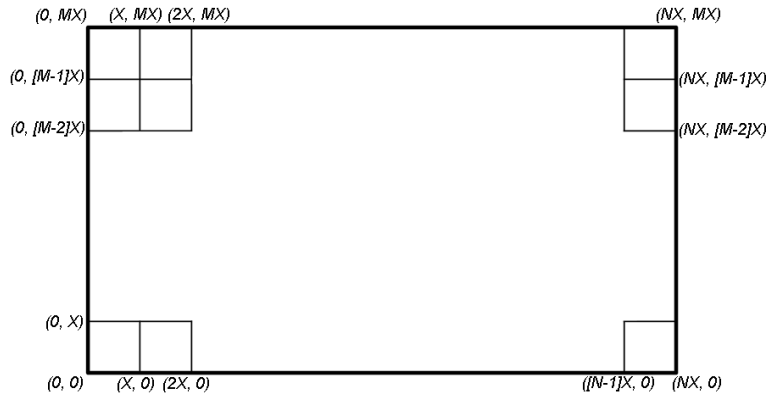


Fig. 1. Pixel structure of the simulated SLM.

pixelated structure of the SLM should be taken into consideration. As a result of this, surface integration over each pixel of SLM should be evaluated. We assume that each pixel of SLM will be constant over pixel area. Then, the diffraction field over SLM can be calculated as

$$\psi_{2D,z=0}(n, m) = \int_{x_n}^{x_{n+1}} \int_{y_m}^{y_{m+1}} \sum_l \psi(\mathbf{r}_l) h_F(\mathbf{r}_0 - \mathbf{r}_l) dx dy \quad (3)$$

where n and m stand for indices of SLM along x - and y -axes, respectively. It is also possible to represent Eq. (3) by using 2D kernels $\mathbf{K}_{\alpha_l, 2D}$,

$$\psi_{2D, z=0} = \sum_{l=1}^L P(\mathbf{r}_l) \mathbf{K}_{\alpha_l, 2D} \quad (4)$$

where $P(\mathbf{r}_l) = -j\psi(\mathbf{r}_l)e^{jkz_l}$ and $\alpha_l = \frac{1}{\sqrt{\lambda z_l}}$. Each 2D kernel can be decomposed into 1D kernels,

$$\mathbf{K}_{\alpha_l, 2D} = (\mathbf{K}_{\alpha_l, 1D}^{x_l})^T \mathbf{K}_{\alpha_l, 1D}^{y_l} \quad (5)$$

where x_l and y_l refer to locations of l^{th} point of 3D object along x - and y -axes, respectively. Each 1D kernel $\mathbf{K}_{\alpha_l, 1D}$ can be represented as

$$\mathbf{K}_{\alpha_l, 1D} = [K_{\alpha_l, 1D}(1) \quad K_{\alpha_l, 1D}(2) \quad \cdots \quad K_{\alpha_l, 1D}(N)] \quad (6)$$

and its elements can be calculated as

$$K_{\alpha_l, 1D}(n) = C(\zeta_{l, n+1}) + jS(\zeta_{l, n+1}) - C(\zeta_{l, n}) - jS(\zeta_{l, n}) \quad (7)$$

where $\zeta_{l, n} = \frac{x_n - x_l}{\sqrt{\lambda z_l}}$. The operators $C(\cdot)$ and $S(\cdot)$ stand for cosine and sine Fresnel integrals, respectively [2, 3] and they are calculated as

$$\begin{aligned} C(w) &= \int_0^w \cos\left(\frac{\pi}{2} \tau^2\right) d\tau \\ S(w) &= \int_0^w \sin\left(\frac{\pi}{2} \tau^2\right) d\tau. \end{aligned} \quad (8)$$

Numerical evaluation of cosine and sine Fresnel integrals given in Eq. (8) are obtained by using adaptive Lobatto quadrature [39]

In standard algorithm for calculating the diffraction field from a point cloud object, we have to compute cosine and sine Fresnel integrals for each object points [35]. Therefore, calculation of diffraction field will be too long to achieve real-time application. To decrease the calculation time, we propose an algorithm that utilizes a precomputed LUT to calculate $\mathbf{K}_{\alpha_l, 2D}$ rapidly.

3. Proposed algorithm for fast computation of CGH

Computation time of diffraction field calculation can be decreased when 2D kernel $\mathbf{K}_{\alpha_l, 2D}$ is obtained without evaluating sine and cosine Fresnel integrals. The proposed algorithm utilizes a precomputed LUT and the diffraction field over the SLM can be calculated as

$$\hat{\psi}_{2D, z=0} = \sum_{l=1}^L P(\mathbf{r}_l) \hat{\mathbf{K}}_{\alpha_l, 2D} \quad (9)$$

where $\hat{\psi}_{2D, z=0}$ denotes estimated diffraction field of the 3D object over SLM and $\hat{\mathbf{K}}_{\alpha_l, 2D}$ is a 2D kernel which denotes the diffraction field of l^{th} point of the object over SLM. 2D kernel $\hat{\mathbf{K}}_{\alpha_l, 2D}$ is calculated as in Eq. (5) from 1D kernels fetched from LUT. Each 1D kernel in LUT denotes a diffraction field over SLM for a specific depth along longitudinal axis. 1D kernels are fetched from LUT in a way that minimize depth difference between object points and 1D kernels. Better estimation of the diffraction field can be obtained when the number of 1D kernels in the LUT is

increased, but having more kernels in LUT may result in occupying infeasible memory space. Therefore, optimization of allocated memory space for LUT is obtained by performing different sampling policies on depth parameter. First sampling policy is based on uniform sampling of the depth parameter. In the second policy, uniform sampling of $\alpha_l = \frac{1}{\sqrt{\lambda z_l}}$ is performed, hence there will be non-uniform sampling of depth along longitudinal axis.

4. Simulation results

Proposed diffraction field calculation algorithm is tested and evaluated under several scenarios. To give an insight to the reader, only a couple of simulation results are presented in this paper. Performance of the algorithm is evaluated by computation time of the diffraction field and the normalized mean square error (NMSE) on the reconstructed object. NMSE on the reconstructed object is calculated as

$$NMSE = \frac{\sum_{n=1}^N \sum_{m=1}^M |\hat{\psi}_{2D,z=z_0}(n,m) - \psi_{2D,z=z_0}(n,m)|}{\sum_{n=1}^N \sum_{m=1}^M |\psi_{2D,z=z_0}(n,m)|} \quad (10)$$

where $\psi_{2D,z=z_0}(n,m)$ and $\hat{\psi}_{2D,z=z_0}(n,m)$ denote reconstructed objects at $z = z_0$ plane from the diffraction field calculated by the standard and the proposed algorithms, respectively.

The setup used in the simulations can be seen in Fig. 2. The 3D object employed in the

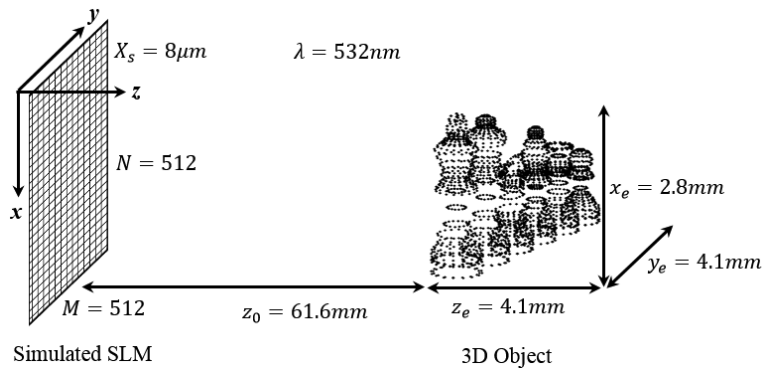


Fig. 2. An illustration of simulated optical setup. Simulated SLM has N and M pixels along x - and y -axes, respectively. The distance between each pixel is indicated by X_s . The distance between 3D object and SLM is denoted as z_0 . The volume occupied by 3D object defined by x_e , y_e and z_e .

simulation is formed by 3144 points. The volume occupied by the 3D object is defined by $x_e = 2.8\text{mm}$, $y_e = 4.1\text{mm}$ and $z_e = 4.1\text{mm}$. The distance between the 3D object and the SLM is set to $z_0 = 61.6\text{mm}$. Simulated SLM has 100% fill factor and pitch distance X_s is taken as $8\mu\text{m}$. Furthermore, it has 512 pixels along both x - and y -axes. Also, wavelength of the coherent light source used for illumination purpose is taken as 532nm .

The proposed diffraction field calculation algorithm run on a computer system which has i5-2500 CPU at 3.3GHz, 4GB RAM and 64 bit Windows 7 OS. Algorithms are implemented by MATLAB. The proposed algorithm can be run in parallel to obtain further improvements on the computation time. Moreover, proposed algorithm is implemented by Visual C++ and CUDA libraries, and run on a GPU to achieve real-time application.

In the reconstruction process, calculated diffraction fields are used for driving SLM, but most of the SLMs available on the market have a pixelated structure. When the pixelated structure of SLM is omitted in calculation of diffraction field, focused and unfocused sections of a 3D object may not be distinguished easily as shown in Fig. 3(a) and the quality of the reconstructed object can be degraded. On the other hand, the difference between focused and unfocused sections of object could be easily seen in the reconstructed object from the diffraction field obtained by the proposed algorithm as given in Fig. 3(b). Furthermore, optical reconstructions are very similar

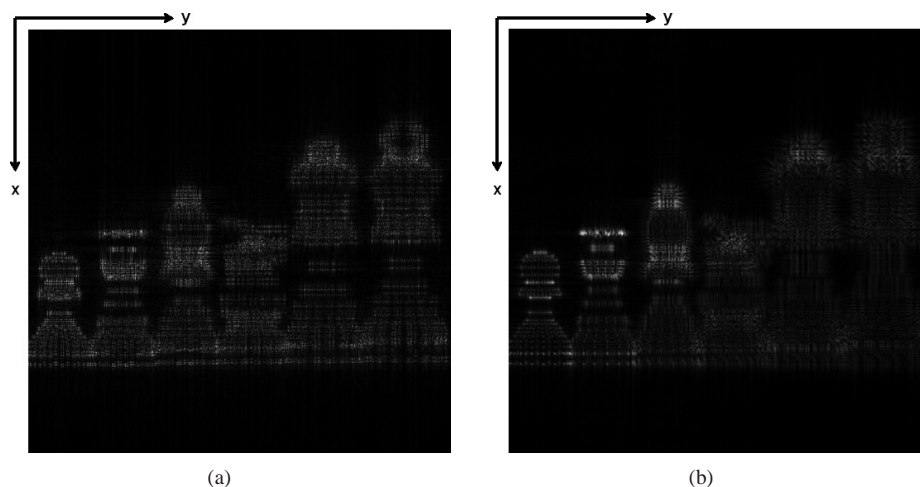


Fig. 3. 3D object is formed by six small parts as shown in Fig. 2 and each part is located at different depth along longitudinal axis. The part which is at the leftmost is reconstructed. (a) Magnitude of the reconstructed object from the diffraction field calculated without taking into consideration the pixelated structure of SLM and (b) from the diffraction field by the proposed algorithm.

to the numerical reconstruction obtained by the proposed algorithm and those results can be seen in Fig. 4.

When standard algorithm is used in calculation of diffraction field for the scenario illustrated in Fig. 2, we need 2701.10 seconds to get the results. On the other hand, when the proposed algorithm is used in the calculations significant decrement on the computation time is achieved and the diffraction field is calculated within 8.15 seconds. Also, the proposed algorithm can be run in parallel and further improvements on the computation time can be obtained. However, there will be negligible amount of deviation on the reconstructed object. The deviation on the reconstructed object is caused by the estimation errors on the calculated diffraction field. Summary of those results can be seen in Table 1

Increasing the number of kernels in LUT improves NMSE performance of the algorithm without changing the computation time, but size of the allocated memory is increased and available memory space is limited by the employed hardware. As a result of this, the size of LUT can be taken as another constraint on the proposed algorithm. Each kernel in LUT refers to diffracted pattern from a point light source at a specific depth along longitudinal axis. Therefore, the sampling policy along longitudinal axis is another important issue that affects NMSE performance of the algorithm. We implement two sampling policies along longitudinal axis in preparation of LUT. First one is uniform sampling of $\alpha_l = \frac{1}{\sqrt{\lambda z_l}}$, thus there will be a non-uniform sampling along longitudinal axis. The other one is uniform sampling along longitudinal

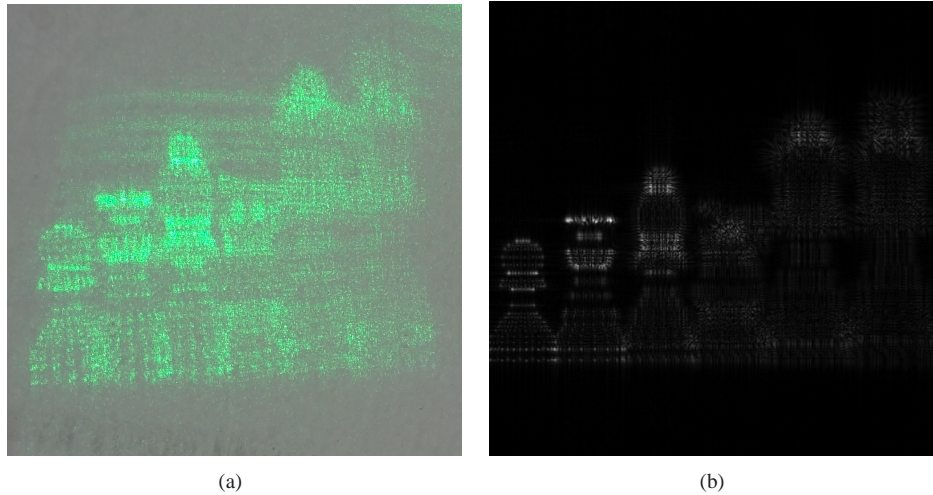


Fig. 4. (a) Optical reconstruction of a 3D object. (b) Numerical reconstruction of the object shown in (a).

Table 1. Performance of the proposed algorithm based on LUT according to NMSE and computation time. 1D kernels of LUT are calculated according to uniform sampling of α_l . When parallel computation is employed, further improvements on computation time can be achieved.

3D Object: 3144 points $\lambda = 532nm, N=M=512, X_s = 8\mu m$	Computation Time (sec)	NMSE
Standard Algorithm	2701.10	-
LUT (with 125 diffraction kernels)	8.15	0.080
LUT (with 125 diffraction kernels) Parallel run over 4 cores	7.08	0.080
LUT (with 125 diffraction kernels) Parallel run over GTX-680	0.08	0.080

axis. Table 2 and Table 3 summarize NMSE performances of the proposed algorithm according to implemented sampling policies. As it can be seen from Tables 2 and Table 3, uniform sampling along longitudinal axis provides better NMSE with the same number of 1D kernels.

Although, there are small numerical errors on reconstructed objects from the diffraction field obtained by the proposed algorithm, the difference between the reconstructed objects obtained by the standard algorithm and the proposed algorithm may not be observable visually. This result can be seen in Fig. 5. Main reason of the numerical error is the unfocused sections of the reconstructed object. This result can be observed from Fig. 6.

The proposed algorithm is also tested by optical setups. An assembled optical setup for reconstruction of 3D objects from the calculated diffraction patterns can be seen in Fig. 7. Captured optical reconstructions of two different 3D objects are given in Fig. 8. Furthermore, optical reconstructions of a 3D object which has more depth than the ones shown in Fig. 8 at different depths along longitudinal axis can be seen in Fig. 9. Also, proposed diffraction field calculation algorithm is tested under red and blue laser illuminations. Captured optical reconstructions for red, green and blue lasers are shown in Fig. 10.

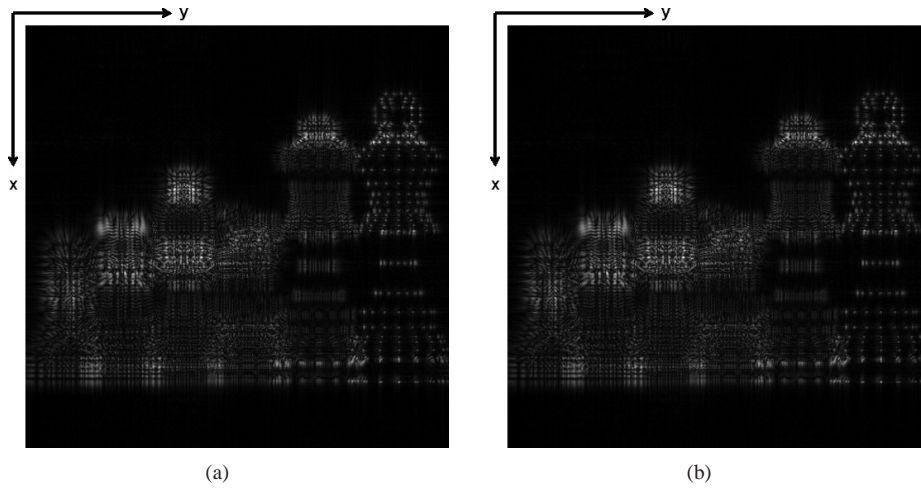


Fig. 5. (a) Magnitude of the reconstructed object at $z = z_0$ from the diffraction pattern calculated by standard algorithm (b) by the proposed algorithm.

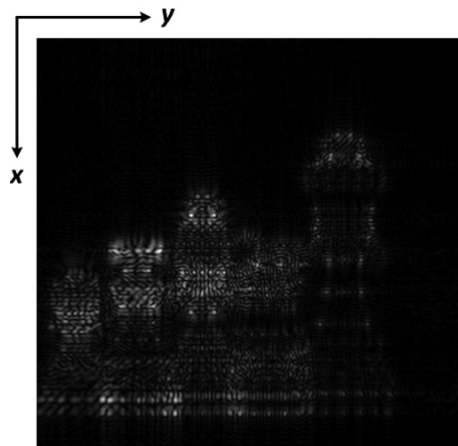


Fig. 6. Magnitude of the difference between the reconstructed objects given in Fig. 5. Please note that, image is scaled linearly from 0 to 255, thus the insignificant differences may become visible.

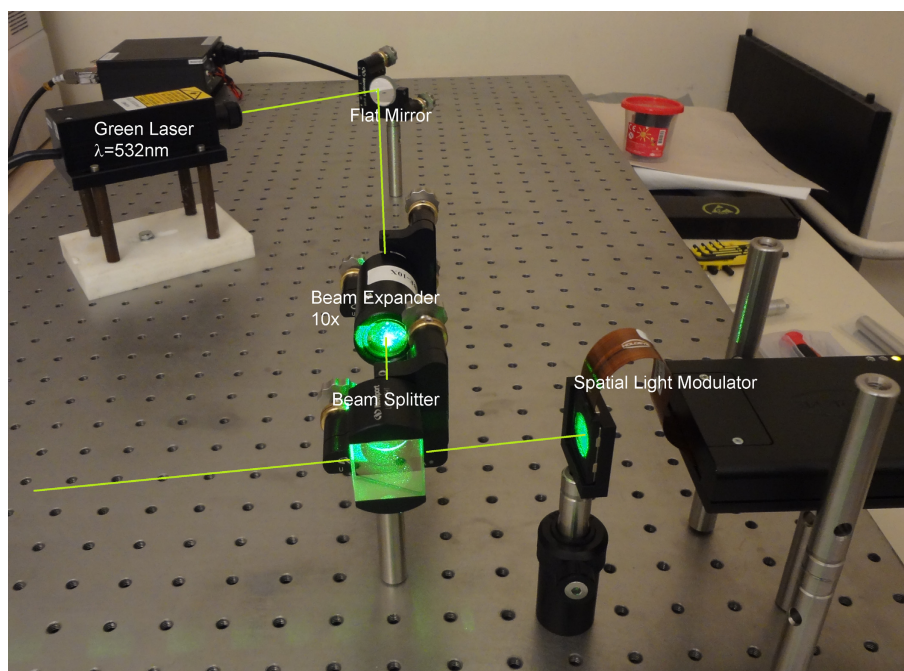


Fig. 7. Assembled optical setup for reconstruction of 3D objects when green laser is used for illumination.

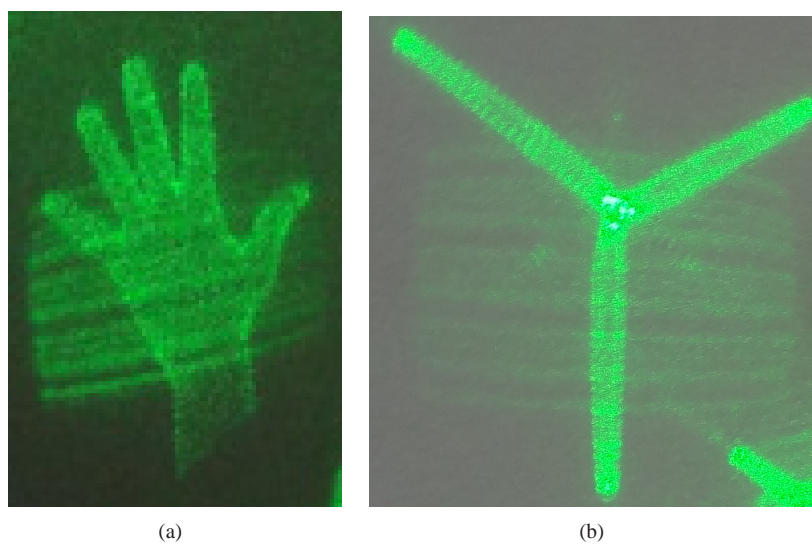


Fig. 8. (a) Optically reconstructed hand shaped 3D object. (b) Another reconstructed 3D object in propeller shape with three blades.

Table 2. Performance of the proposed algorithm according to number of kernels used in LUT, NMSE and allocated memory space. LUT is formed by uniform sampling of α_l parameter. Each element in 1D kernels are represented by four Bytes.

3D Object: 3144 points $\lambda = 532nm, N=M=512, X_s = 8\mu m$	Number of 1D Kernels	NMSE	Memory Space (KB)
	83	0.127	332
	92	0.114	368
	103	0.104	412
	118	0.088	472
	137	0.077	548
	165	0.062	660
	206	0.051	824
	274	0.038	1096
	411	0.025	1644
821	0.013	3284	

Table 3. Performance of the proposed algorithm according to number of kernels used in LUT, NMSE and allocated memory space. LUT is formed by uniform sampling of depth parameter along longitudinal axis. Each element in 1D kernels are represented by four Bytes

3D Object: 3144 points $\lambda = 532nm, N=M=512, X_s = 8\mu m$	Number of 1D Kernels	NMSE	Memory Space (KB)
	83	0.068	332
	92	0.061	368
	103	0.054	412
	118	0.048	472
	137	0.039	548
	165	0.034	660
	206	0.026	824
	274	0.020	1096
	411	0.014	1644
821	0.006	3284	

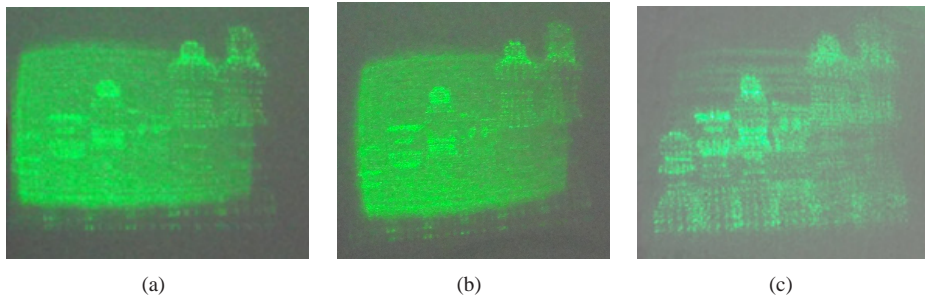


Fig. 9. Optically reconstructed 3D object which has more depth than the ones illustrated in Fig. 8. (a) Front piece is reconstructed. (b) Middle pieces are reconstructed. (c) Rear piece is reconstructed.

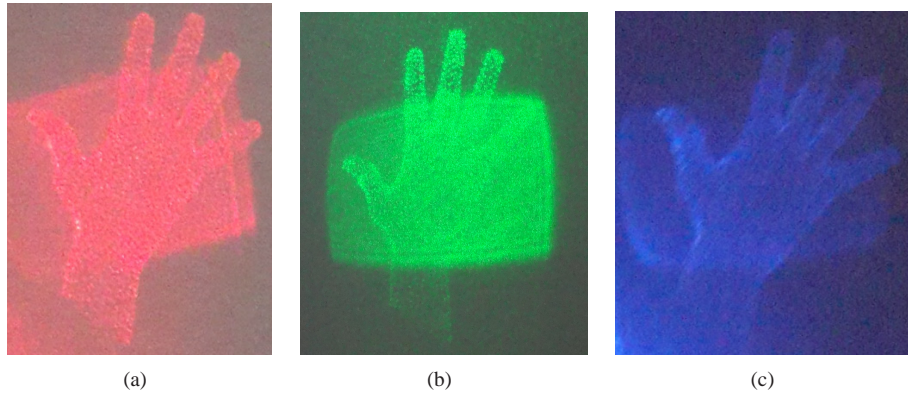


Fig. 10. Optically reconstructed hand shaped 3D object (a) for red laser with $\lambda = 637nm$, (b) for green laser with $\lambda = 532nm$, (c) for blue laser with $\lambda = 473nm$.

5. Conclusion

Fast calculation of diffraction field from 3D objects and improving the quality of the reconstructed objects are two challenging problems in H3DTV. As a solution to those problems, we propose a diffraction field calculation algorithm which provides fast calculation of diffraction field from a point cloud object over an SLM which has square pixels with 100% filling factor. When the proposed algorithm run in parallel on a GPU, real-time computations can be achieved. Both numerical and optical experiments are conducted for red, green and blue lasers and successful results are obtained. Similar reconstructions are observed in both numerical simulations and optical experiments. Two different LUT generation methods are used for optimizing the performance of the proposed algorithm according to NMSE and allocated memory space. First method is based on uniform sampling of the depth parameter along longitudinal axis. Second method uses non-uniform sampling of depth parameter. LUT calculated by uniform sampling of depth parameter gives better NMSE with the same number of kernels. Hence, physically available memory space can be used more effectively when uniform sampling policy is employed in calculation of LUT.

Acknowledgments

This work was supported by The Scientific and Technological Research Council of Turkey project under grant EEEAG-112E220.