Hardware architecture for full analytical Fraunhofer computer-generated holograms

Zhi-Yong Pang,[a,†] Zong-Xi Xu,[a,†] Yi Xiong,[a,†] Biao Chen,[a] Hui-Min Dai,[a] Shao-Ji Jiang,[a] and Jian-Wen Dong[a,*]
[a]Sun Yat-sen University, School of Physics and Engineering, Guangzhou 510275, China

1 **Abstract.** Hardware architecture of parallel computation is proposed for generating Fraunhofer computer-
2 generated holograms (CGHs). A pipeline-based integrated circuit architecture is realized by employing the modified Fraunhofer analytical formulism, which is large scale and enables all components to be concurrently operated. The architecture of the CGH contains five modules to calculate initial parameters of amplitude, amplitude compensation, phases, and phase compensation, respectively. The precalculator of amplitude is fully adopted considering the "reusable design" concept. Each complex operation type (such as square arithmetic) is reused only once by means of a multichannel selector. The implemented hardware calculates an 800×600 pixels hologram in parallel using 39,319 logic elements, 21,074 registers, and 12,651 memory bits in an Altera field-programmable gate array environment with stable operation at 50 MHz. Experimental results demonstrate that the quality of the images reconstructed from the hardware-generated hologram can be comparable to that of a software implementation. Moreover, the calculation speed is approximately 100 times faster than that of a personal computer with an Intel i5-3230M 2.6 GHz CPU for a triangular object.
[DOI: 10.1117/1.OE.54.0.XXXXXX]

Keywords: computer-generated hologram; field-programmable gate array; Fraunhofer; real time.

Paper 150569 received May 6, 2015; accepted for publication Aug. 5, 2015

1 Introduction

As a technique for three-dimensional (3-D) images, holography enables storing and restoring 3-D information. Since it was first proposed by Gabor in 1948,[1] holography has been a well-studied research field. Computer-generated hologram (CGH)[2] is a promising technology for displaying true 3-D images. Digital 3-D object data stored in the computer can be transferred to an encoding picture, which can optically recreate all depth cues of the 3-D objects.[3] However, CGH for 3-D objects requires an enormous number of calculations. To address this issue, researchers have developed fast algorithms to accelerate calculations.[4,5] Some methods handle 3-D objects as a combination of individual self-luminous points. The object wave distribution on the hologram is calculated for each point and then superimposed. This requires a large number of points to achieve a solid effect and an enormous memory size for sampling.[5] Other methods handle 3-D objects as a combination of planar segments (polygons). This method significantly decreases the computation time of CGHs, because the number of polygons required to form the surface is much smaller than the number of point sources used in the point-based method. Recently, a triangle-mesh-based algorithm has been proposed to further reduce computational time.[6,7]

Although CGHs have a long history, few CGH techniques have been proposed for computing, especially in the aspects of application-specific integrated circuit or field-programmable gate array (FPGA) hardware. Several researchers have used high-class hardware to accelerate the computational speed. For example, an MIT group has developed a special purpose 3
computational board for holographic rendering.[8] The group has achieved a record speed for holographic rendering that was 50 times faster than that of typical workstations of that time. More recently, researchers have presented parallel algorithms based on commodity graphic processing units.[9–11] These approaches can accelerate CGH calculations in a cost-effective way.

FPGA is an alternative way to accelerate the computation speed. Since 1992, a Chiba group has developed another special purpose hardware architecture, so-called holographic reconstruction,[12–18] which achieves a remarkable performance. In addition, Seo et al. have proposed improved hardware architectures[9–21] to demonstrate a better performance. Both architectures have handled 3-D objects as a combination of individual self-luminous points, which is originated from the same equations as phase CGH. In our previous work, we have derived a set of analytical formulism without the use of fast Fourier transform to describe the diffraction fields of 3-D true-life scenes.[22] The speed can be somewhat accelerated by using CUDA algorithms. This paper will extend such a prominent method to FPGA architecture, by modifying the analytical Fourier transform formulism. This will propose a pipeline-based large-scale integrated circuit architecture to enable all components to concurrently operate. The paper is organized as follows. Section 2 reviews the polygon-based CGH that is based on full analytical holographic computations. Some formulisms are modified for maximum parallel computation. The corresponding hardware structure is proposed in Sec. 3. Section 4 describes the hardware implementation with FPGA from Altera. Section 5 will conclude the paper.

[*] Address all correspondence to: Jian-Wen Dong, E-mail: dongjwen@mail.sysu.edu.cn

[†] These authors contributed equally to this work.

0091-3286/2015/$25.00 © 2015 SPIE

2 Polygon-Based Fraunhofer Computer-Generated Holograms

In basic computer graphics, 3-D surface models are represented by polygons, such as triangles, as shown in Fig. 1(a). In triangle-based methods, each triangle (blue, for example) has two sets of spatial coordinates, one is the local coordinates given by (x_l, y_l, z_l) and the other is the global coordinates given by (x_g, y_g, z_g). Such two coordinate systems have a relationship connected by a rotation matrix, yielding,

$$\begin{bmatrix} x_l \\ y_l \\ z_l \end{bmatrix} = \begin{pmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{pmatrix} \begin{bmatrix} x_g - x_c \\ y_g - y_c \\ z_g - z_c \end{bmatrix} \quad \text{and}$$
$$\begin{bmatrix} x_g \\ y_g \\ z_g \end{bmatrix} = \begin{pmatrix} r'_{11} & r'_{12} & r'_{13} \\ r'_{21} & r'_{22} & r'_{23} \\ r'_{31} & r'_{32} & r'_{33} \end{pmatrix} \begin{bmatrix} x_l \\ y_l \\ z_l \end{bmatrix} + \begin{bmatrix} x_c \\ y_c \\ z_c \end{bmatrix}, \tag{1}$$

where (x_c, y_c, z_c) is the mass center of the triangle in the global coordinate system, and z_l on the triangular facet is 0. Consider the diffraction of the monochromatic plane wave by the finite triangular shape aperture in an infinite opaque screen, as shown in Fig. 1(b). By using the Rayleigh–Sommerfeld diffraction integral, the diffraction field of the blue triangle on the hologram can be simplified as[6,7,22]

$$\mathcal{O}_H(x_H, y_H) = \frac{\exp[ik(z_c + r_0)]}{i\lambda r_0} \mathcal{F}[\mathcal{O}_O(x_l, y_l)], \tag{2}$$

where $k = 2\pi/\lambda$ and λ is the wavelength, and $r_0 = \sqrt{(x_H - x_c)^2 + (y_H - y_c)^2 + z_c^2}$ is the distance between the triangular mass center and the hologram pixel. $\mathcal{F}[\mathcal{O}_O(x_l, y_l)]$ stands for Fourier transform of the blue triangle, evaluated at frequencies $(x'_H/\lambda r_0, y'_H/\lambda r_0)$. By applying affine transformation for two-dimensional Fourier transform, we have the following explicit expression:

$$\mathcal{F}[\mathcal{O}_O(x_l, y_l)] = (a_{22}a_{11} - a_{12}a_{21}) \exp\left(-i2\pi \frac{a_{13}x'_H + a_{23}y'_H}{\lambda r_0}\right) \times \mathcal{F}_\Delta\left(\frac{a_{11}x'_H + a_{21}y'_H}{\lambda r_0}, \frac{a_{12}x'_H + a_{22}y'_H}{\lambda r_0}\right), \tag{3}$$

where $a_{11} = x_l^2 - x_l^1$, $a_{12} = x_l^3 - x_l^2$, $a_{13} = x_l^1$, $a_{21} = y_l^2 - y_l^1$, $a_{22} = y_l^3 - y_l^2$, $a_{23} = y_l^1$, $x'_H = r'_{11}(x_H - x_c) + r'_{21}(y_H - y_c) - r'_{31}z_c - r'_{31}r_0$, $y'_H = r'_{12}(x_H - x_c) + r'_{22}(y_H - y_c) - r'_{32}z_c - r'_{32}r_0$. The last term $\mathcal{F}_\Delta(\cdot)$ in Eq. (3) is the Fourier transform of a right triangle with vertices at the points of (0, 0), (1, 0), (1,1), which can be written in the compact form of $\mathcal{F}_\Delta(u, v) = |F_\Delta| e^{i\varphi}$ using Euler's formulism, yielding,

$$\mathcal{F}_\Delta(u, v) = \begin{cases} \dfrac{1}{2} & u = v = 0 \\ \dfrac{\sqrt{2 + 4\pi^2 v^2 - 2\cos(2\pi v) - 4\pi v \sin(2\pi v)}}{(2\pi v)^2} e^{i \arctan\left(\frac{\sin(2\pi v) - 2\pi v}{1 - \cos(2\pi v)}\right)} & v \neq 0, u = 0 \\ \dfrac{\sqrt{2 + 4\pi^2 u^2 - 2\cos(2\pi u) - 4\pi u \sin(2\pi u)}}{(2\pi u)^2} e^{i \arctan\left(\frac{2\pi u \cos(2\pi u) - \sin(2\pi u)}{\cos(2\pi u) + 2\pi u \sin(2\pi u) - 1}\right)} & u \neq 0, v = 0 \\ \dfrac{\sqrt{2 + 4\pi^2 v^2 - 2\cos(2\pi v) - 4\pi v \sin(2\pi v)}}{(2\pi v)^2} e^{i \arctan\left(\frac{2\pi v - \sin(2\pi v)}{1 - \cos(2\pi v)}\right)} & u = -v \neq 0 \\ \dfrac{\sqrt{(u+v)^2 + u^2 + v^2 - 2v(u+v)\cos(2\pi u) + 2uv\cos[2\pi(u+v)] - 2(u+v)u\cos(2\pi v)}}{(2\pi)^2 uv(u+v)} & \\ \quad \times e^{i \arctan\left(\frac{u \sin[2\pi(u+v)] - (u+v)\sin(2\pi u)}{(u+v)\cos(2\pi u) - v - u\cos[2\pi(u+v)]}\right)} & \text{else} \end{cases}, \tag{4}$$

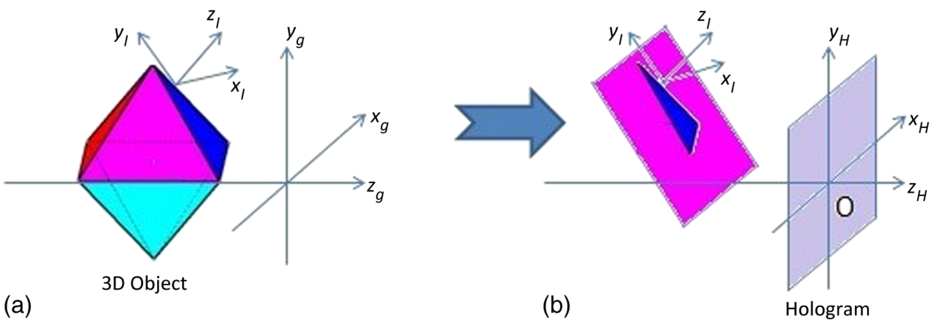


Fig. 1 (a) Local coordinate and global coordinate of triangle-mesh object and (b) diffraction by one of the triangles.

where the magnitude and the phase of $\mathcal{F}_\Delta(u,v)$ are dependent on the spatial frequency pair of (u,v). In this way, the diffraction field of the blue triangle on the hologram can be simplified as

$$O_H(x_H, y_H) = Ae^{iP}, \tag{5}$$

where there is one amplitude calculator $A = |F_\Delta| a_{22}a_{11} - a_{12}a_{21}/\lambda r_0$ and one phase calculator $P = k(z_c + r_0) - 2\pi/\lambda r_0(a_{13}x'_H + a_{23}y'_H) + \varphi - \pi/2$. This is very important because it enables not only the direct generation of the hologram pixel by pixel without fast Fourier transform, but also execution of the wave optics theory on the parallel FPGA platform. As the diffraction field emitted from each triangle of the 3-D model can be explicitly calculated on the hologram plane, the whole object field distributions can be obtained through their linear superpositions of the diffraction field of each triangle.

3 Proposed Hardware Architecture

To accelerate the polygon-based CGH computation, we consider a parallel algorithm based on FPGA. We propose a simple polygon-based CGH equation that can be fully performed in parallel when implemented in hardware.

3.1 *Sub-Module-Separated Computer-Generated Hologram Equation*

The most widely used technique in digital design is the pipeline technique that requires the breakdown of the sequential process into suboperations. We, therefore, separate Eq. (5) into two components. One is the amplitude calculator,

$$A = \text{amp} \times \text{amp\_}cp, \tag{6}$$

where $\text{amp} = |F_\Delta|$, $\text{amp\_}cp = a_{22}a_{11} - a_{12}a_{21}/\lambda r_0$, and the other is the phase calculator,

$$P = \text{pha} + \text{pha\_}cp, \tag{7}$$

where $\text{pha1} = r_0/\lambda$, $\text{pha2} = a_{13}Gx + a_{23}Gy$, $\text{pha\_}cp = \text{pha1} - \text{pha2}$, $\text{pha} = \phi$. Then the hologram can be obtained by the real part of Eq. (5) by the linear superposition principle,

$$\text{CGH}(j,i) = \text{CGH}(j,i) + \text{CGHtemp}_{j,i}, \tag{8}$$

where $\text{CGHtemp}_{j,i} = A \times \cos(2\pi P)$.

3.2 *Computer-Generated Hologram Processor*

Next, we propose the hardware architecture based on Eqs. (7) to (9). As shown in Fig. 2, the architecture consists of five modules for the calculation of initial parameters (ipc), amplitude (amp), amplitude compensation (amp_cp), phases (pha), and phase compensation (pha_cp), respectively.

3.2.1 *Initial parameter calculator hardware architecture*

The initial parameter calculator is primarily used to calculate the parameter values of r_0, u, and v, which correspond to Eqs. (9) and (10), respectively. Equation (9) requires that $|x_H - x_c| \ll z_c$, $|y_H - y_c| \ll z_c$ is typically satisfied. The initial parameters are implemented in the hardware components, as shown in Fig. 3. All arithmetic components are connected with timing registers to enable a fully pipelined operation. In the figure, the timing registers are depicted with rectangular boxes, in which the intermediate variables are shown.

$$\begin{aligned} r_0 &= \sqrt{(x_H - x_c)^2 + (y_H - y_c)^2 + z_c^2} \\ &= abs(z_c + (x_H - x_c)^2 + (y_H - y_c)^2/2z_c), \end{aligned} \tag{9}$$

$$u = \frac{a_{11}x'_H + a_{21}y'_H}{\lambda r_0}, \qquad v = \frac{a_{12}x'_H + a_{22}y'_H}{\lambda r_0}. \tag{10}$$

3.2.2 *Hardware architecture of amp and pha*

The amp and pha calculators are primarily used to calculate the amplitude and phase of $\mathcal{F}_\Delta(u,v)$, which correspond to Eqs. (4) to (7). The proposed hardware architecture consists of one precalculator, one phase calculator, and one amplitude calculator, as shown in Figs. 4 to 6, respectively. The precalculator is fully adopted considering the "reusable design" concept. Each complex operation type (such as square arithmetic) is reused only once by means of a multichannel selector (mux). Tables 1 and 2 show the phase selected values and amplitude selected values for different u and v. All arithmetic components are connected with timing registers to enable a fully pipelined operation. The proposed architecture provides the advantages of high throughput and low power consumption.

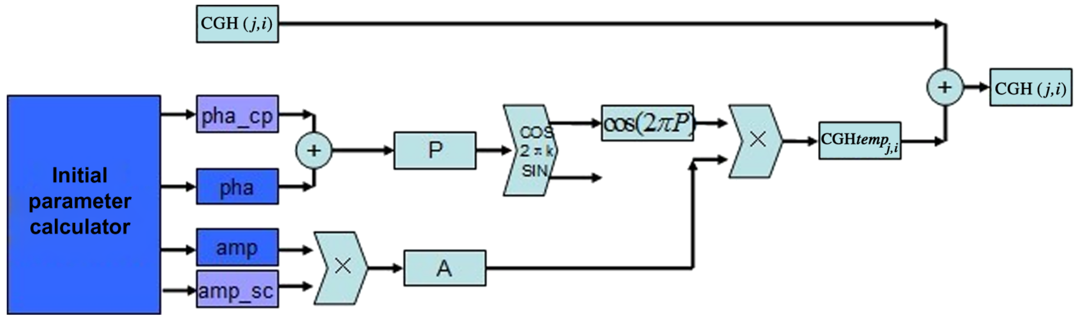
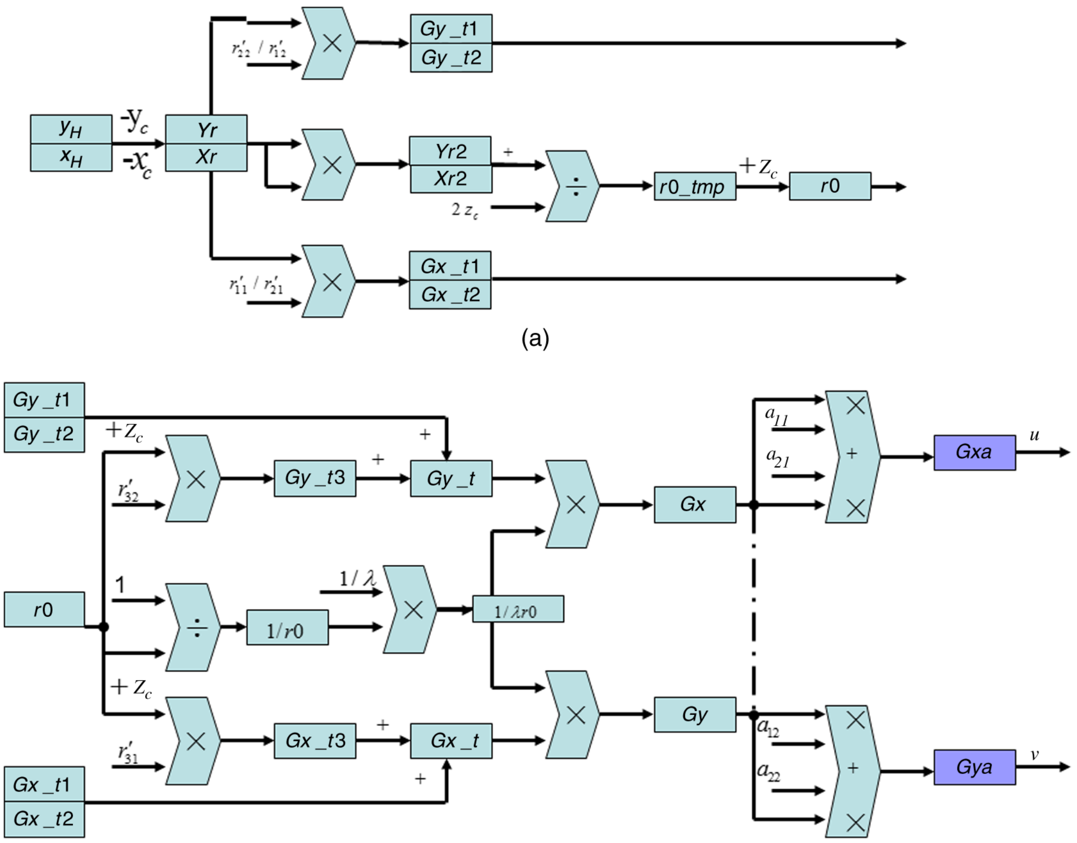


Fig. 2 Computer-generated hologram (CGH) processor hardware architecture.



(a)

(b)

Fig. 3 Architecture of initial parameter calculator: (a) *r*0 calculator and (b) *u* and *v* calculator.

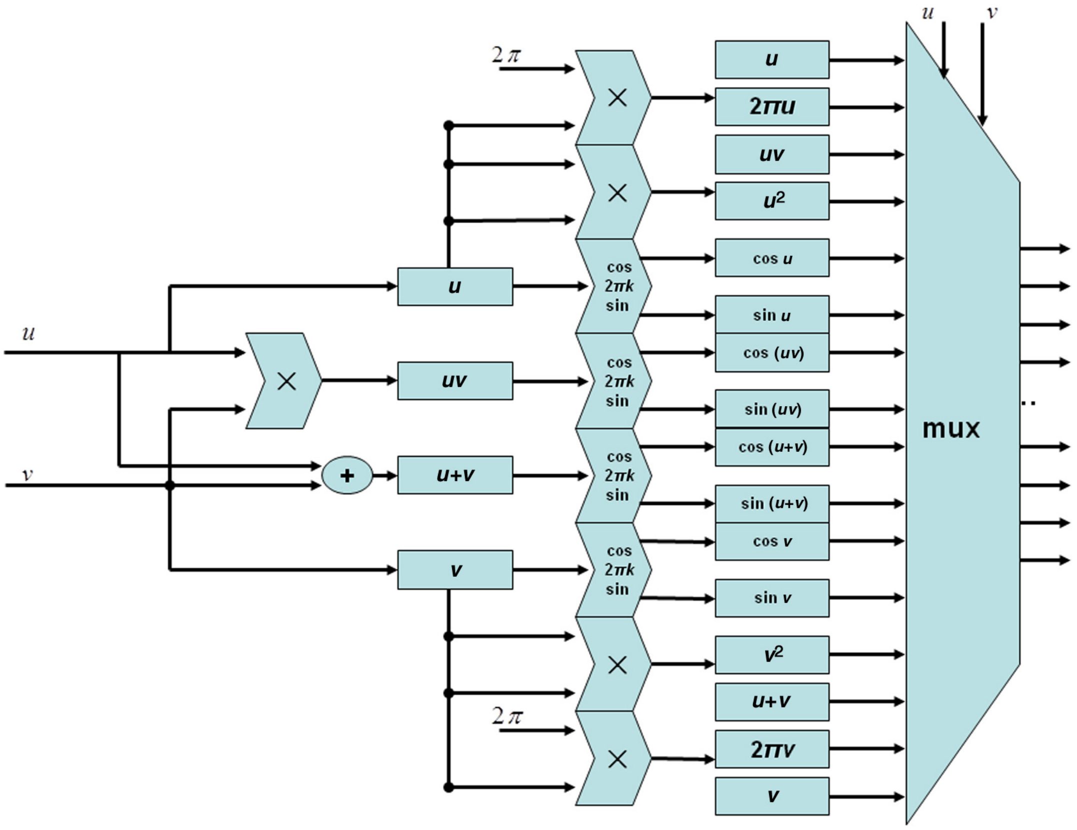


Fig. 4 Pipelined architecture of precalculator.

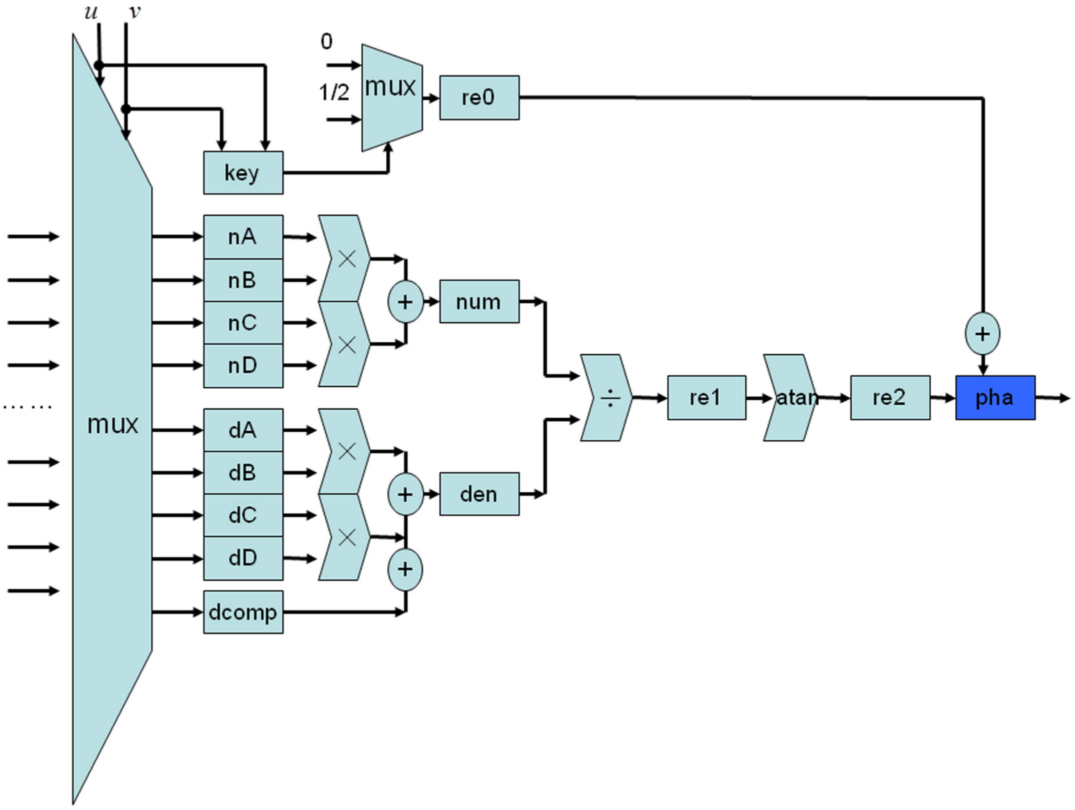


Fig. 5 Pipelined architecture of phase calculator.

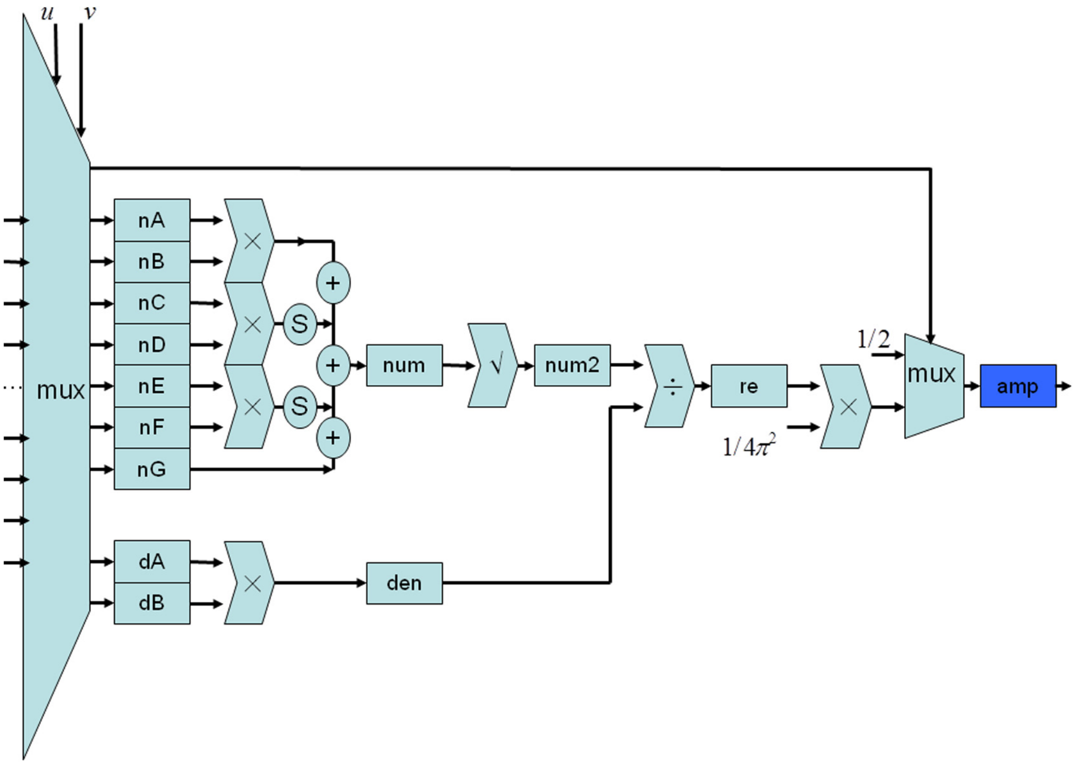


Fig. 6 Pipelined architecture of amplitude calculator.

Table 1 Phase select variable values.

	Phase select				
mux	$u = v = 0$	$u = 0$	$v = 0$	$u = -v$	$u! = v$
nA	0	1	$2\pi u$	$2\pi v$	u
nB	0	sin $2\pi v$	cos $2\pi u$	1	sin $2\pi(u + v)$
nC	0	1	1	1	$u + v$
nD	0	$2\pi v$	sin $2\pi u$	sin $2\pi v$	sin$2\pi u$
dA	1	1	1	1	$u + v$
dB	1	1	cos $2\pi u$	1	cos $2\pi u$
dC	0	1	$\sim 2\pi u$	1	u
dD	0	cos $2\pi v$	sin $2\pi u$	cos $2\pi v$	cos $2\pi(u + v)$
dcomp	0	0	1	0	v
key	0	0	0	0	1

Table 2 Amplitude select variable values.

	Amplitude select				
mux	$u = v = 0$	$u = 0$	$v = 0$	$u = -v$	$u! = v$
nA	2π	$2\pi v$	$2\pi u$	$2\pi v$	$2uv$
nB	2π	$2\pi v$	$2\pi u$	$2\pi v$	cos $2\pi(u + v)$
nC	0	1	1	1	v^2
nD	0	cos $2\pi v$	cos $2\pi u$	cos $2\pi v$	cos $2\pi u$
nE	0	$2\pi v$	$2\pi u$	$2\pi v$	u^2
nF	0	sin $2\pi v$	sin $2\pi u$	sin $2\pi v$	cos $2\pi v$
nG	0	1	1	1	v^2
dA	1	1	1	1	uv
dB	2	v^2	u^2	v^2	$u + v$

3.2.3 *Hardware architecture of amp_cp and pha_cp*

The amp_cp and pha_cp calculators are primarily used to calculate the CGH amplitude and phase compensation values. Such compensation is mostly because of the interferences between the diffraction fields of the same edge that belong to two conjoint triangles. The compensation can smooth the edges and has little effect on the reconstruction positions of the macroscopical triangle. The proposed hardware architecture is shown in Fig. 7. All arithmetic components are connected with timing registers to enable a fully pipelined operation.

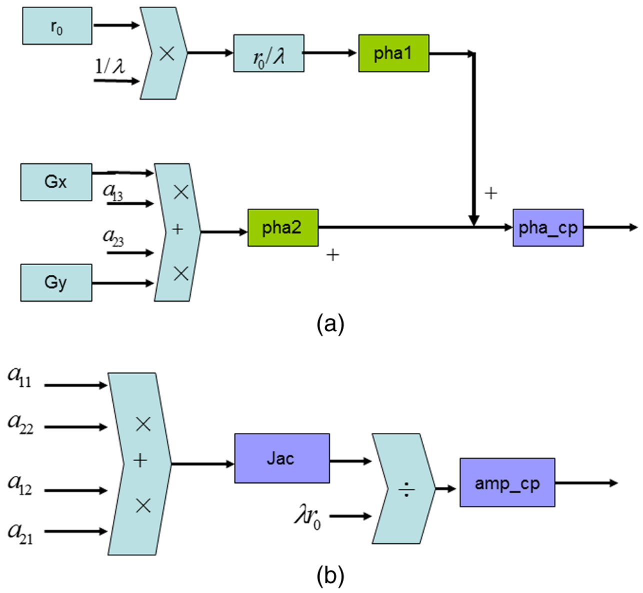


Fig. 7 Pipelined architecture of amplitude and phase compensation calculators, (a) pha_cp calculator and (b) amp_cp.

Table 3 Experimental environments and variable values.

Item	specification
Hologram size	800 × 600 pixels
Pixel pitch (p)	10.4 μm
Wavelength (λ)	632.8 nm
Target FPGA	cyclone IV EP4CE115

Table 4 Resource utilization.

Item	resource
Total logic elements	39,319/114,480 (34%)
Total combinational functions	37,181/114,480 (32%)
Dedicated logic register	21,074/114,480 (18%)
Total memory bits	12,651/3,981,312 (<1%)

Table 5 Hologram construction times for a few different triangular models.

Number of triangles	1	8	120
CPU (ms)[6]	1484	—	—
CUDA GPU (ms)[22]	—	—	40
Proposed CPU (ms)	856	7406	101322
Proposed hardware (1 processor) (ms)	9.612	~76.896	~1153.44
Proposed hardware (N processors) (ms)	9.612($N = 1$)	~9.612($N = 8$)	~39($N = 30$)

4 Hardware Implementations and Experiments

The proposed architecture described in the previous section was implemented with Verilog in Altera FPGA environments. Accordingly, Quartus II 13.1 and ModelSim 6.5e

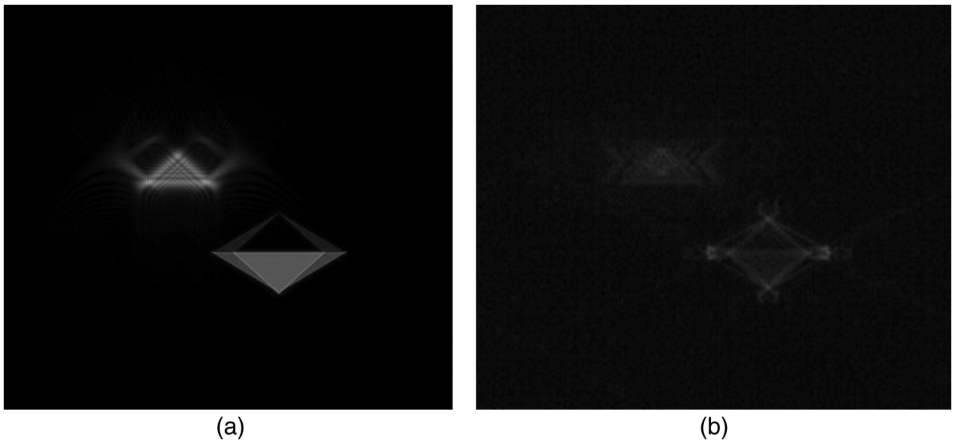


Fig. 8 Reconstruction images from (a) software and (b) proposed hardware.

were used for the Verilog design and simulation, respectively. The experimental environments and parameter values used in our implementation are shown in Table 3. For the target FPGA chip, we used Cyclone IV EP4CE115 (speed grade 5) from Altera, which contains 114,480 logic elements. The CGH resolution was 800 × 600 pixels, and the pixel size was 10.4 μm.

Table 4 outlines the resources used in the design for the CGH with the resolution of 800 × 600 pixels. For the CGH 4 processor, 39,319 logic elements, 21,074 registers, and 12,651 memory bits were used. Results illustrate that it can stably operate at a maximum clock frequency of 50 MHz.

Table 5 lists the calculation time for the different models used. We compared the calculation time of the proposed hardware with those of a typical personal computer, the latter of which included an Intel i5-3230M 2.60 GHz CPU, 8.0 GB of memory, Microsoft Windows8 64 bits, and Software vs 2012. The time of one-triangle object calculation is 9.612 ms for the proposed hardware, while it is 856 ms for the personal computer. The calculation speed of the proposed hardware was around 100 times less. If there are N parallel processors, a higher calculation speed may be achieved.

To be sure of the correction of the proposed hardware implementation, we compared the reconstructed images from both the hardware and software. Figure 8 shows the example images of eight triangles of the 3-D diamond model. It can be seen that the reconstruction imagings are consistent with each other. In the future, we may extend this FPGA method to various kinds of polygon-based CGHs.

5 Conclusion

This paper proposed a hardware scheme for generating CGHs based on a full analytical algorithm. It can be implemented by a fully pipelined hardware. It consists of five calculators of initial parameters, amplitude, amplitude compensation, phases, and phase compensation. All modules have been designed in pipeline to enable a fully pipelined calculation scheme. The hardware was implemented with FPGAs with Altera intellectual property cores. Experimental results demonstrated that the hardware implementation is superior to that of the CPU by achieving a speed about 100 times faster than that of the serial CPU algorithm when a one-triangle object is employed as an example. The reconstruction images between both the hardware and software are consistent with each other. The current hardware implementation is relatively straightforward; nevertheless, it can be optimized. In the future, we will improve the CGH processing power to enable 3-D reconstruction in real time.

Acknowledgments

This work is supported by grants from the Natural Science Foundation of China (No. 61235002).

References

1. D. Gabor, “A new microscope principle,” *Nature* **161**, 777–778 (1948). 5
2. C. Slinger, C. Cameron, and M. Stanley, “Computer-generated holography as a generic display technology,” *Computer* **38**(8), 46–53 (2005).
3. J. W. Goodman, *Introduction to Fourier Optics*, Roberts & Company 6 Publishers (2004).
4. M. Lucente, “Interactive computation of holograms using a look-up table,” *J. Electron. Imaging* **2**(1), 28–34 (1993).
5. 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).
6. L. Ahrenberg et al., “Computer-generated holograms from three dimensional meshes using an analytic light transport model,” *Appl. Opt.* **47**(10), 1567–1574 (2008).
7. H. Kim, J. Hahn, and B. Lee, “Mathematical modeling of triangle-mesh-modeled three-dimensional surface objects for digital holography,” *Appl. Opt.* **47**(19), D117–D127 (2008).
8. J. Watlington et al., “A hardware architecture for rapid generation of electro-holographic fringe patterns,” *Proc. SPIE* **2406**, 172–183 (1995).
9. C. Petz and M. Magnor, “Fast hologram synthesis for 3D geometry models using graphics hardware,” *Proc. SPIE* **5005**, 266–275 (2003).
10. V. M. Bove, Jr. et al., “Real-time holographic video images with commodity PC hardware,” *Proc. SPIE* **5664**, 255–262 (2005).
11. L. Ahrenberg et al., “Computer-generated holography using parallel commodity graphics hardware,” *Opt. Express* **14**(17), 7636–7641 (2006).
12. T. Yabe, T. Ito, and M. Okazaki, “Holography machine HORN-1 for computer-aided retrieval of virtual,” *Jpn. J. Appl. Phys.* **32**, L1359 (1993).
13. T. Ito et al., “Special-purpose computer HORN-1 for reconstruction of virtual image in three dimensions,” *Comp. Phys. Commun.* **82**, 104–110 (1994).
14. T. Ito et al., “Special-purpose computer for holography HORN-2,” *Comp. Phys. Commun.* **93**, 13–20 (1996).
15. T. Shimobaba et al., “Special-purpose computer for holography HORN-3 with PLD technology,” *Comp. Phys. Commun.* **130**, 75–82 (2000).
16. T. Shimobaba, S. Hishinuma, and T. Ito, “Special-purpose computer for holography HORN-4 with recurrence algorithm,” *Comp. Phys. Commun.* **148**, 160–170 (2002).
17. T. Ito et al., “A special-purpose computer HORN-5 for a real-time electroholography,” *Opt. Express* **13**, 1923–1932 (2005).
18. T. Shimobaba and T. Ito, “An efficient computational method suitable for hardware of computer-generated hologram with phase computation by addition,” *Comp. Phys. Commun.* **138**, 44–52 (2001).
19. Y.-H. Seo et al., “An architecture of a high-speed digital hologram generator based on FPGA,” *J. Syst. Archit.* **56**, 27–37 (2010)
20. Y.-H. Seo et al., “Cell-based hardware architecture for full-parallel generation algorithm of digital holograms,” *Opt. Express* **19**, 8750–8761 (2011).

21. Y.-H. Seo et al., "Hardware architecture of high-performance digital hologram generator on the basis of a pixel-by-pixel calculation scheme," *Appl. Opt.* **51**, 4003–4012 (2012).
22. Y.-Z. Liu et al., "High-speed full analytical holographic computations for true-life scenes," *Opt. Express* **18**, 3345 (2010).
7 23. Y.-Z. Liu et al., "Fraunhofer computer-generated hologram for diffused 3D scene in Fresnel region," *Opt. Lett.* **36**, 2128 (2011).
24. J. K. Chung and M. H. Tsai, *Three-Dimensional Holographic Imaging*, Wiley (2002).

Zhi-Yong Pang is a lecturer in the School of Physics and Engineering, Sun Yat-Sen University. He received his PhD from the School of Information Science and Technology of the same university in 2013. His current research areas include image processing and integrated circuit design.

Zong-Xi Xu received his master's degree from the School of Physics and Engineering, Sun Yat-Sen University in 2014. His current research includes computer-generated holography.

Yi Xiong received his bachelor's degree from the School of Physics and Engineering, Sun Yat-Sen University in 2013. He is a postgraduate student of Sun Yat-Sen University. His current research areas include image processing and integrated circuit design.

Biao Chen received his bachelor's degree from the School of Physics and Engineering, Sun Yat-Sen University in 2013. His current research areas include image processing and integrated circuit design.

Hui-Min Dai received his master's degree from the School of Physics and Engineering, Sun Yat-Sen University in 2013. His current research areas include image processing and integrated circuit design.

Shao-Ji Jiang is a professor in the School of Physics and Engineering, Sun Yat-Sen University. His current research areas include holographic three-dimensional display and nanophotonics.

Jian-Wen Dong is a professor in the School of Physics and Engineering, Sun Yat-Sen University. His current research areas include holographic three-dimensional display and nanophotonics.

Queries

1. Please provide department name in affiliation.

2. Please clarify whether the word “formulism” can be changed as “formalism” here and throughout the paper.

3. Please define MIT at first occurrence.

4. Logic elements has been changed from 9,319 to 39,319 to match the abstract. Please check.

5. This query was generated by an automatic reference checking system. These references 8, 9, 10, and 12 could not be located in the databases used by the system. While the reference may be correct, we ask that you check it so we can provide as many links to the referenced articles as possible.

6. Please provide publisher location names in Refs. [3 and 24].

7. Please cite references 23 and 24 in the text.