# Fast time-division color electroholography using a multiple-graphics processing unit cluster system with a single spatial light modulator

**Hiromitsu Araki[1], Naoki Takada[2,*], Shohei Ikawa[3], Hiroaki Niwase[1], Yuki Maeda[1], Masato Fujiwara[1], Hirotaka Nakayama[4], Minoru Oikawa[2], Takashi Kakue[5], Tomoyoshi Shimobaba[5], and Tomoyoshi Ito[5]**

[1]*Graduate School of Integrated Arts and Sciences, Kochi University, Kochi, 780-8520, Japan*
[2]*Science Department, Natural Sciences Cluster, Research and Education Faculty, Kochi University, Kochi 780-8520, Japan*
[3]*Faculty of Science, Kochi University, Kochi 780-8520, Japan*
[4]*Center for Computational Astrophysics, National Astronomical Observatory of Japan, Mitaka-shi 181-8588, Japan*
[5]*Graduate School of Engineering, Chiba University, Chiba 263-8522, Japan*
**Corresponding author: ntakada@is.kochi-u.ac.jp*
Received August 28, 2017; accepted October 20, 2017; posted online November 14, 2017

We demonstrate fast time-division color electroholography using a multiple-graphics-processing-unit (GPU) cluster system with a spatial light modulator and a controller to switch the color of the reconstructing light. The controller comprises a universal serial bus module to drive the liquid crystal optical shutters. By using the controller, the computer-generated hologram (CGH) display node of the multiple-GPU cluster system synchronizes the display of the CGH with the color switching of the reconstructing light. Fast time-division color electroholography at 20 fps is realized for a three-dimensional object comprising 21,000 points per color when 13 GPUs are used in a multiple-GPU cluster system.
*OCIS codes: 090.1705, 090.5694, 090.1760.*
doi: 10.3788/COL201715.120902.

Holography[1] is the ideal three-dimensional (3D) display technology because it can record and appropriately reconstruct a 3D object. A computer-generated hologram (CGH)[2] is digitally generated holographic interference patterns between the scattered light from a 3D object and reconstructing light. Electroholography using CGH can reconstruct 3D movies using a spatial light modulator (SLM). Therefore, electroholography has the potential to ultimately realize a 3D television (TV)[3,4]. However, the practical use of real-time electroholography is limited by the complexity of the CGH calculations and the necessity of high-performance computational power[5].

A graphics processing unit (GPU) offers high performance at low cost. A GPU code for general numerical calculations can be created with the provided software development kit (SDK). Fast CGH calculations using GPUs have been demonstrated[6–20].

Color electroholography is indispensable for realizing a 3D TV. Color electroholography using three SLMs for red, green, and blue (RGB) -colored lights has been reported[21–23]. However, SLMs are expensive, and a system containing three SLMs is very large. Color electroholography using a single SLM has also been investigated. The space-division method[24], depth-division method[25,26], and time-division method[27–30] have been proposed. In the space-division method and the depth-division method, there is the advantage that flicker does not occurred. However, it is not easy for the space-division method to reconstruct clear color 3D images comprising many object points, because a CGH comprises three parts for the RGB-colored reconstructing lights. The reconstructed 3D image by the depth-division method is slightly contaminated by unwanted diffracted images. While flicker occurs for time-division color electroholography using the proposed system, the resolution power of the reconstructed 3D image is high.

Figure 1 shows the outline of time-division electroholography. At each frame of a color-reconstructed 3D movie, three CGHs for the RGB-colored lights are displayed on an SLM at regular time intervals. The 3D color image is generated by time-division multiplexing of the RGB-colored 3D images reconstructed from the CGHs.

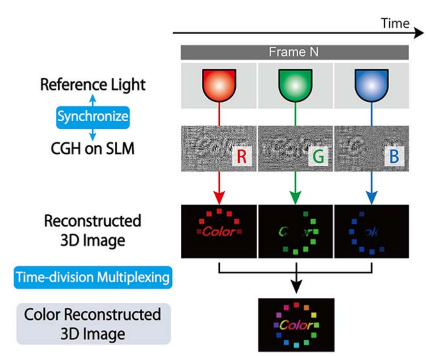


Fig. 1. Outline of time-division color electroholography.

 

However, the wavelength of the reconstructing light is used in the CGH calculation. Therefore, the color of the reconstructing light has to be switched in synchronicity with the three CGHs displaying RGB-colored lights at regular intervals. Furthermore, the cost of color electroholography is three times that of monochromatic electroholography. Hence, it is difficult to realize real-time time-division color electroholography.

Previously[30], we tried to overcome these difficulties using a single GPU and a universal serial bus (USB) module to synchronize the reconstructing light. A one-chip RGB light-emitting diode (LED) was used as the reconstructing light. Using the synchronizing controller with the USB module is an effective technique for displaying real-time time-division color electroholography. However, the computational power of the system was insufficient for real-time time-division color electroholography. We also reported the use of a multiple-GPU (multiGPU) cluster system with a single SLM and an InfiniBand network for real-time monochromatic electroholography[20]. The system has high-performance computational power of floating point arithmetic and high scalability. Therefore, it is desirable to adapt the multiGPU cluster system to real-time time-division color electroholography.

In this work, we propose fast time-division color electroholography using a multiGPU cluster system with a single SLM.

In the CGH calculation, we used a simple algorithm to calculate an in-line hologram from a 3D object expressed by a point cloud. The light intensity of each point on the hologram is given by the following equation[6]:

$$I(x_h, y_h, 0) = \sum_{i=1}^{N_p} A_i \cos\left\{\frac{\pi}{\lambda z_i}[(x_h - x_i)^2 + (y_h - y_i)^2]\right\}, \quad (1)$$

where $I\ (x_h,\ y_h,\ 0)$ is the light intensity of the point $(x_h,\ y_h,\ 0)$ on the hologram, $(x_i,\ y_i,\ z_i)$ is the coordinate of the $i$th point on the 3D object, $A_i$ is the amplitude of the object point, $N_p$ is the number of object points of the 3D model, and $\lambda$ is the wavelength of the reconstructing light. Eq. (1) is obtained by using the Fresnel approximation.

In Eq. (1), the number of hologram points is equal to the resolution ($H \times W$) of an SLM, where $H$ and $W$ are the height and width of the display resolution, respectively. The CGH calculation becomes prohibitively large because the computational complexity of Eq. (1) is $O(N_p H W)$. Furthermore, color electroholography requires three CGHs for RGB-colored reconstructing lights because of the related wavelength [Eq. (1)].

Figure 2 shows the proposed multiGPU cluster system for fast time-division color electroholography. This system is composed of a CGH display node (personal computer, PC 0) and $N$ CGH calculation nodes. The CGH display node (PC 0) has a single GPU (GPU 0), and each CGH calculation node has three GPUs. In the system,

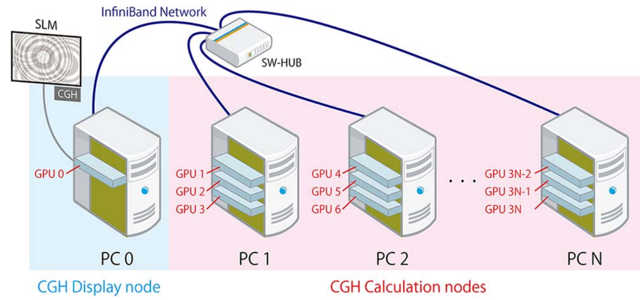


Fig. 2. Proposed multiGPU cluster system for fast time-division color electroholography.

the number of GPUs is $3N + 1$ in total. The CGH calculation nodes calculate all of the CGHs for the reconstructed color 3D movie in parallel. The CGH display node receives the calculated CGHs from the CGH calculation nodes and displays the CGHs on the SLM. The data coordinates of the 3D object points of all frames in the original 3D movie are stored on the CGH display node (PC 0). The CGH display node also acts as the network file system server.

Figure 3 shows the timing chart of fast time-division color electroholography using the proposed multiGPU cluster system. In Fig. 3, Frame $N$(R), Frame $N$(G), and Frame $N$(B) show the CGH calculations for RGB-colored reconstructing lights at Frame $N$ of the original 3D movie, respectively. In the CGH calculation nodes, each PC calculates the CGHs for RGB-colored reconstructing lights at each frame of the original 3D movie.

The computation for time-division color electroholography using the proposed multiGPU cluster system proceeds as follows:

Step 1: PC 1 calculates Frame 1. In PC 1, GPU 1, GPU 2, and GPU 3 calculate the CGHs for RGB-colored reconstructing lights at Frame 1 of the original 3D movie, respectively. Similarly, the other CGH calculation nodes

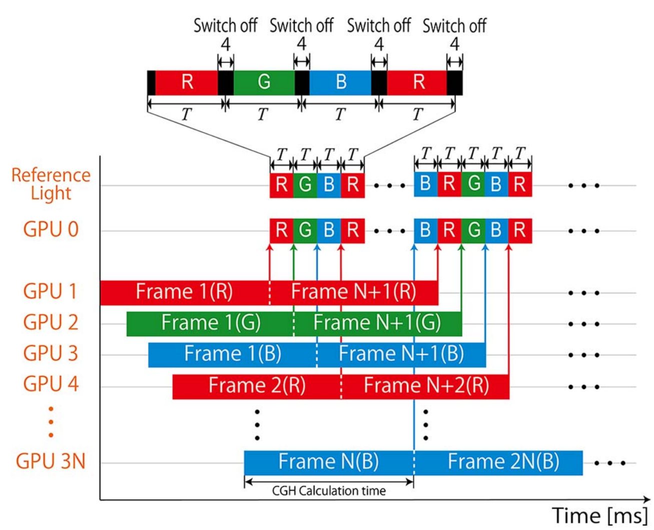


Fig. 3. Timing chart of fast time-division color electroholography using the proposed multiGPU cluster system.

calculate the CGHs for RGB-colored reconstructing lights at the frames from Frame 2 to Frame $N$.

Step 2: After the completion of the CGH calculations for RGB-colored reconstructing lights at Frame 1, PC 1 sends the respective calculated CGHs to the CGH display node PC 0. Similarly, the other CGH calculation nodes send the respective calculated CGHs for RGB-colored reconstructing lights at Frame 2 to Frame $N$ to the CGH display node PC 0 in turn.

Step 3: The CGH display node PC 0 receives the respective calculated CGHs for RGB-colored reconstructing lights at Frame 1 from PC 1, and GPU 0 displays the respective calculated CGHs for RGB-colored reconstructing lights at Frame 1, in turn, on the SLM for a constant time $T$. Similarly, the CGH display node PC 0 receives the respective calculated CGHs for RGB-colored reconstructing lights from Frame 2 to Frame $N$ from PC 2 to PC $N$, and GPU 0 displays the respective calculated CGHs for RGB-colored reconstructing lights at each frame in turn on the SLM for a constant time $T$.

After step 3, the CGHs for RGB-colored reconstructing lights at Frame $N + 1$ to $2N$ are calculated and displayed in turn on the SLM of the CGH display node PC 0. The process is repeated until the last frame of the 3D movie is reached.

We investigated the CGH image data transfer time between the CGH display node and each CGH calculation node. The CGH image data size is 32 bits $\times$ 1920 pixels $\times$ 1024 pixels $\approx$ 62.9 Mbit, when 32 bits is estimated as the CGH image data size per pixel. A resolution of 1920 pixels $\times$ 1024 pixels is used to apply the optimized CGH calculation algorithm to the proposed system[12]. The CGH image data transfer time is 63 ms if the Gigabit Ethernet (1 Gbps) is used as the computer network. The CGH image data transfer time exceeds the vertical synchronizing interval of 16.7 ms when the SLM refresh rate is 60 Hz. Therefore, a faster high-speed network is needed to display the calculated CGH on an SLM within 16.7 ms; thus, we used an InfiniBand QDR (40 Gbps). The CGH image data transfer time between the CGH display node and each CGH calculation node is 1.6 ms when InfiniBand QDR is used as the computer network of the proposed system.

Figure 4 shows the optical setup in the proposed system, in which we used a single transmissive liquid crystal display (LCD) panel extracted from a projector (Epson Corp. EMP-TW1000) as an SLM. The LCD panel is used to modulate the amplitude of light. The specifications of the LCD panel are as follows: pixel interval = 8.5 μm, resolution = 1920 × 1080, refresh rate = 60 Hz, and size = 16 mm × 9 mm. The projector with LCD panels is connected to GPU 0 on the CGH display node. We used three laser lights with wavelengths of 625, 525, and 470 nm as RGB-colored reconstructing lights for color electroholography, respectively. Each laser light is converted into a parallel light by an objective lens and a collimator lens. The parallel light is made incident on the LCD panel.

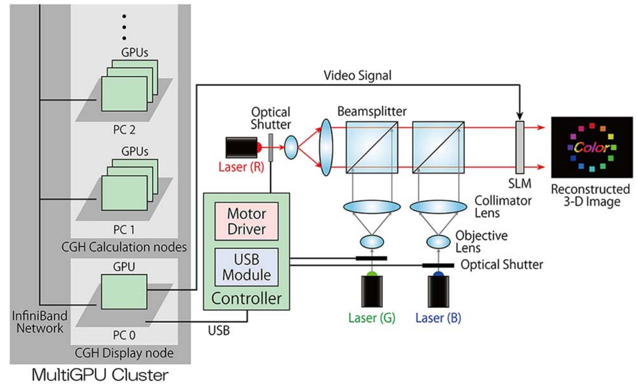


Fig. 4. Optical setup for fast color electroholography in the proposed system.

Time-division color electroholography requires synchronization between the color switching of the reconstructing light and the suitable CGH display for the color of the reconstructing light on an LCD panel. We developed the color-switching controller comprising a USB module (Future Technology Devices International Ltd., FT245RL) and three motor driver integrated circuits (TOSHIBA, TB6643KQ). We used liquid-crystal-based optical shutters [LC-Tec Display AB, X-FOS (G2)] to switch the color of the reconstructing light. By sending the color-switching signal from the CPU of the CGH display node to the developed controller via USB, the controller can open and close three optical shutters of three laser lights with the wavelengths of 625, 525, and 470 nm. This indicates that the CGH display node can change the color of the reconstructing light.

The refresh rate of the LCD panel used is 60 Hz. We used a vertical synchronizing signal ($V_{sync}$) that is sent from GPU 0 on the CGH display node to the LCD panel as the synchronizing signal for the system. In Fig. 3, the regular time interval $T$, when each of the calculated CGHs is displayed on the LCD panel, is set to the vertical synchronizing cycle of $V_{sync}$. By using $V_{sync}$ and the color-switching controller, the CPU of the CGH display node can synchronize the color switching of the reconstructing light with the display of the suitable CGH on the LCD panel when the CGH calculation time/$3N$ is less than the vertical synchronizing cycle of $V_{sync}$.

We evaluated the proposed time-division color electroholography using a five-node multiGPU cluster system. The system is comprised of a CGH display node and four CGH calculation nodes. The CGH display node had a GPU, and each CGH calculation node had three GPUs. Therefore, the total number of GPUs was 13. In each node of the system, the PC was equipped with an Intel Core i7 4770 (clock speed: 3.4 GHz, quad-core), and the operating system was Linux (Cent OS 7.1). We used the NVIDIA GeForce GTX TITAN X as the GPU and InfiniBand QDR as the network. We used CUDA 7.0 SDK[31] for GPU programming, Open GL as the graphics application programming interface (API), and Open MPI v1.8.7 as

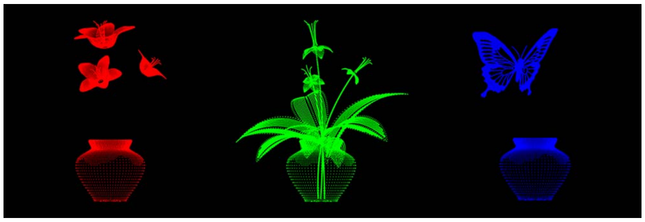

Fig. 5. RGB-colored 3D objects for the original color 3D movie; (left) the red-colored 3D object comprising 19,168 points; (center) the green-colored 3D object comprising 27,012 points; (right) the blue-colored 3D object comprising 18,228 points.

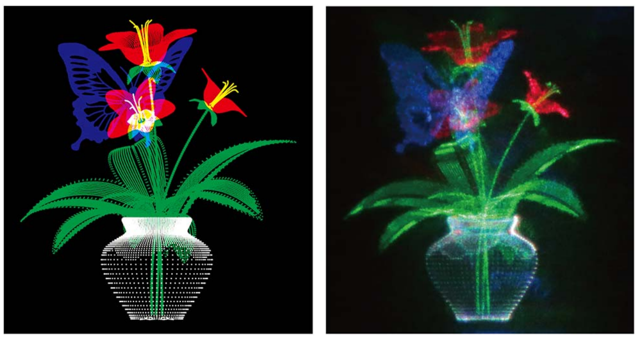

Fig. 6. (left) Snapshots of the original color 3D movie and (right) the reconstructed 3D movie using the proposed method (video 1).

the message passing interface (MPI) library. As shown in Fig. 3, the reconstructing light was switched off for 4 ms when the color of the reconstructing light was switched.

Figure 5 shows the images of three original 3D objects that are used for the original color 3D movie "flower and butterfly" here for the RGB-colored reconstructing lights, respectively. The original 3D objects for the RGB-colored reconstructing lights comprised 19,168, 27,012, and 18,228 points, respectively. Figure 6 (left) shows a snapshot of the original color 3D movie. The snapshot is overlapped with the images of three original 3D objects for the RGB-colored reconstructing lights. Figure 6 (right) shows the snapshot of the reconstructed color 3D movie using the proposed system. The distance between the original 3D objects and CGHs was 1.35 m.

In conclusion, fast time-division color electroholography is achieved with the proposed system in the color 3D movie "flower and butterfly" when the CGH calculation nodes have 12 GPUs in total.

This work was partially supported by the Japan Society for the Promotion of Science through a Grant-in-Aid for Scientific Research (C) under Grant No. 15K00153.

## References

1. D. Gabor, Nature **161**, 777 (1948).
2. G. Tricoles, Appl. Opt. **26**, 4351 (1987).
3. S. A. Benton and J. V. M. Bove, *Holographic Imaging* (Wiley, 2008).
4. P. St-Hilaire, S. A. Benton, M. Lucente, M. L. Jepsen, J. Kollin, H. Yoshikawa, and J. S. Underkoffler, Proc. SPIE **1212**, 172 (1990).
5. J.-S. Chen and D. Chu, Appl. Opt. **55**, A127 (2016).
6. N. Masuda, T. Ito, T. Tanaka, A. Shiraki, and T. Sugie, Opt. Express **14**, 603 (2006).
7. A. Shiraki, N. Takada, M. Niwa, Y. Ichihashi, T. Shimobaba, N. Masuda, and T. Ito, Opt. Express **17**, 16038 (2009).
8. Y. Pan, X. Xu, S. Solanki, X. Liang, R. B. A. Tanjung, C. Tan, and T.-C. Chong, Opt. Express **17**, 18543 (2009).
9. H. Kang, F. Yaraş, and L. Onural, Appl. Opt. **48**, H137 (2009).
10. P. Tsang, W. K. Cheung, T.-C. Poon, and C. Zhou, Opt. Express **19**, 15205 (2011).
11. J. Weng, T. Shimobaba, N. Okada, H. Nakayama, M. Oikawa, N. Masuda, and T. Ito, Opt. Express **20**, 4018 (2012).
12. N. Takada, T. Shimobaba, H. Nakayama, A. Shiraki, N. Okada, M. Oikawa, N. Masuda, and T. Ito, Appl. Opt. **51**, 7303 (2012).
13. Y. Pan, X. Xu, and X. Liang, Appl. Opt. **52**, 6562 (2013).
14. G. Li, K. Hong, J. Yeom, N. Chen, J.-H. Park, N. Kim, and B. Lee, Chin. Opt. Lett. **12**, 060016 (2014).
15. B. J. Jackin, H. Miyata, T. Ohkawa, K. Ootsu, T. Yokota, Y. Hayasaki, T. Yatagai, and T. Baba, Opt. Lett. **39**, 6867 (2014).
16. H. Niwase, N. Takada, H. Araki, H. Nakayama, A. Sugiyama, T. Kakue, T. Shimobaba, and T. Ito, Opt. Express **22**, 28052 (2014).
17. Y. Zhang, J. Liu, X. Li, and Y. Wang, Chin. Opt. Lett. **14**, 030901 (2016).
18. Z. Chen, X. Sang, Q. Lin, J. Li, X. Yu, X. Gao, B. Yan, C. Yu, W. Dou, and L. Xiao, Chin. Opt. Lett. **14**, 080901 (2016).
19. J. Song, C. Kim, H. Park, and J.-I. Park, Appl. Opt. **55**, 3681 (2016).
20. H. Niwase, N. Takada, H. Araki, Y. Maeda, M. Fujiwara, H. Nakayama, T. Kakue, T. Shimobaba, and T. Ito, Opt. Eng. **55**, 093108 (2016).
21. F. Yaraş, H. Kang, and L. Onural, Appl. Opt. **48**, H48 (2009).
22. H. Nakayama, N. Takada, Y. Ichihashi, S. Awazu, T. Shimobaba, N. Masuda, and T. Ito, Appl. Opt. **49**, 5993 (2010).
23. J. Roh, K. Kim, E. Moon, S. Kim, B. Yang, J. Hahn, and H. Kim, Opt. Express **25**, 14774 (2017).
24. T. Ito, T. Shimobaba, H. Godo, and M. Horiuchi, Opt. Lett. **27**, 1406 (2002).
25. M. Makowski, M. Sypek, and A. Kolodziejczyk, Opt. Express **16**, 11618 (2008).
26. M. Makowski, M. Sypek, I. Ducin, A. Fajst, A. Siemion, J. Suszek, and A. Kolodziejczyk, Opt. Express **17**, 20840 (2009).
27. T. Shimobaba and T. Ito, Opt. Rev. **10**, 339 (2003).
28. M. Oikawa, T. Shimobaba, T. Yoda, H. Nakayama, A. Shiraki, N. Masuda, and T. Ito, Opt. Express **19**, 12008 (2011).
29. H. Sasaki, K. Yamamoto, K. Wakunami, Y. Ichihashi, R. Oi, and T. Senoh, Sci. Rep. **4**, 1 (2014).
30. H. Araki, N. Takada, H. Niwase, S. Ikawa, M. Fujiwara, H. Nakayama, T. Kakue, T. Shimobaba, and T. Ito, Appl. Opt. **54**, 10029 (2015).
31. NVIDIA, "CUDA C programming guide ver. 7.0," https://developer.nvidia.com/cuda-toolkit-70 (2015).