

Interactive Control of Holographic Optical Traps with Fast Hologram Generation

Sun-Uk Hwang, Yun-Hui Park, and Yong-Gu Lee*

Abstract—Holographic optical tweezers (HOTs) make use of computer generated holograms to manipulate multiple optical traps. Uptill now, various algorithms have been proposed to calculate the hologram patterns. However the update rate of such holograms for a spatial light modulator (SLM) is often restricted to around 10Hz due to the slow computation in computer, limiting fast manipulation of optical traps in real-time. In this paper, we present the method and control software for interactive control of holographic optical traps with fast hologram generation by increasing the update rate of holograms. The developed software is written in Visual C++ platform and is capable of computing complex holograms with an update rate of 55 Hz, regardless of the number of traps. For experimental demonstrations, we show three dimensional, real-time manipulations of 5 μm polystyrene beads.

I. INTRODUCTION

SINGLE-BEAM optical traps (or optical tweezers) have become an important tool for manipulating and probing micro/nano scale objects in biological [1], [2] and physical researches [3], [4] due to their non-physical contact property and relatively small forces ($\sim 1\text{-}100\text{pN}$). This technique uses a highly focused laser beam to trap and manipulate the objects with help of the forces exerted by the intensity gradient of the focused laser beam.

As the demand for multiple traps has been raised, single-beam optical traps are advanced to allow simultaneous manipulation of multiple objects. One of the methods for trapping multiple objects is by the use of holographic optical tweezers (HOTs) [5]. Unlike the scanning optical tweezers that employ a time-sharing approach to make multiple optical traps using galvano mirrors [6], piezo stages [7], and acousto-optic deflectors (AOD) [8], HOTs make use of diffractive optical elements, i.e., spatial light modulator (SLM) to shape the laser beam into continuous multiple foci, by diffracting the laser beam incident on SLM. By modifying the hologram pattern being displayed on SLM, the trapping geometries can be easily controlled without any mechanical movements in optical setup [9]. However, this method needs

fast computation of new hologram patterns if the trapping geometries need to be changed. To this end, commercial SLMs have been highly advanced in terms of update rates and resolution. Nowadays, SLMs show update rates of 60-200 Hz that exceeds normal update rates of video microscopy of around 30 Hz. However, due to the computation burden on the computer, update rate of computed hologram for SLMs is often limited. This is due to the fact that the relation between the required hologram and the reconstructed trap patterns is usually nonlinear and non-analytic requiring sophisticated iterative algorithms [10].

So far, there are various iterative and non-iterative algorithms developed for the computation of holograms [10] – [12]. Due to the slow convergence of iterative approaches, non-iterative approaches are frequently used to enhance computation speed, by using the direct superposition of holograms. Leach et al. suggested gratings and lenses algorithm for interactive control of optical traps and presented a control software based on Labview [13]. In addition, Montes-Usategui et al. proposed random mask encoding method and presented a control software based on Java [14]. However, in both of the studies, update rates of hologram are limited to around 10 Hz with central processing unit (CPU) based computation. This slow update rate can apply discrete motions to the trapped particles [15] and they might escape away easily, if the traps move a little faster. Also, this requires careful selection of step size to use HOTs as a force transducer due to the varying trap stiffness with the step size [15].

As one of the solutions for fast hologram generation, graphic processing unit (GPU) based computation was proposed [16], [17]. Reicherter et al. showed an update rate of 13 Hz for 100 doughnut traps using consumer graphics card [17]. However, GPU based programming is relatively complex requiring the knowledge of computer graphics and great effort is needed to implement for the researchers who are unfamiliar with that. Furthermore, implementations are usually non-portable. Another solution is to improve the algorithms and optimize the code with CPU based computation. CPU based computation is slower than GPU based computation due to the low peak performance of CPU [17]. However, in most research applications, CPU based computation is prevalent because it is flexible and most research tools are designed to interface with CPU for mathematical computation. Nevertheless, in our knowledge, no one has yet reported the computation speed comparable to a video update rate of 30 Hz or above using CPU based computation.

Manuscript received June 30, 2009. This work was supported by the institute of Medical System Engineering (iMSE) in GIST, Korea.

S.-U. Hwang is with the Department of Mechatronics, Gwangju Institute of Science and Technology (GIST), 1 Oryong-dong, Buk-gu, Gwangju, 500-712, Korea. (e-mail: suhwang@gist.ac.kr)

Y.-H. Park is with the Department of Mechatronics, GIST. (e-mail: yhpark@gist.ac.kr)

Y.-G. Lee is with the Department of Mechatronics, GIST. (e-mail: lygu@gist.ac.kr)

* To whom correspondence should be made.

In this paper, we present the method and control software for the interactive control of holographic optical traps implying CPU based computation. The developed software is written in Visual C++ platform and is capable of computing complex holograms with the update rate of 55 Hz regardless of the number of traps, using an Intel Pentium IV processor at 3.0 GHz. Implying our method experimentally, we show three dimensional manipulations of micro particles in real-time.

II. EXPERIMENTAL SETUP

Our optical trapping system is built on a custom-made inverted, bright-field microscope geometry as shown in Fig. 1. For optical trapping, a continuous-wave, p-polarized, NIR fiber laser (IPG Photonics YLM-10, 1050 nm, 10 W, TEM₀₀) is used. The telescope composed of lenses L1 and L2 are used to broaden the incident laser beam of 5.0 mm in diameter by 2 times, and the broadened beam is then delivered to fill the SLM (Hamamatsu X10468-3). Used SLM is a reflective phase modulator with high precision phase controls providing the light utilization efficiency of 90%.

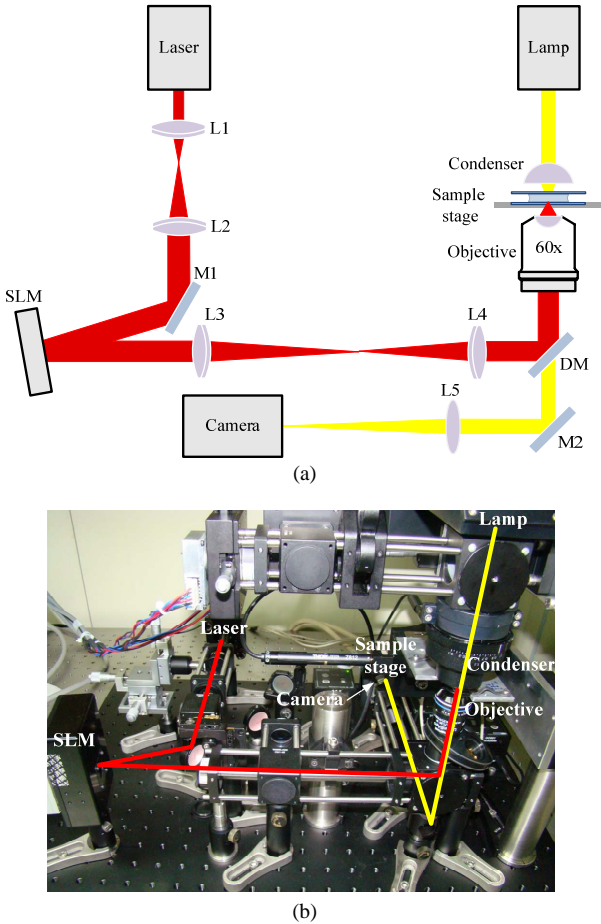


Fig. 1. Experimental setup. (a) schematic diagram, Dichroic mirror: DM, Lens: L (b) actual experimental setup

It has the update rate of 60 Hz though DVI interface and the resolution of 800×600 pixels with a pixel pitch of 20 μm. The SLM is positioned at the Fourier plane of the objective back

aperture. The telescope formed by lenses L3 and L4 decrease diameter of the beam reflected by SLM to slightly overfill the objective lens. In addition, each lens is positioned to minimize vignetting of the incident beam at objective back aperture caused by the diffraction on the SLM. To form two telescopes, achromatic lens doublets are used to achieve best performance in focusing collimated laser beam, removing spherical aberration due to the large beam size. Then, the laser beam is delivered to the water immersion objective lens (Olympus UPLSAPO 60XW, 60x, 1.2 NA) by a dichroic mirror (DM), and is finally tightly focused onto the sample stage. A CCD camera (VRMagic, 640×480 pixels) is used for viewing the bright-field image of the sample in real time, which is imaged through the tube lens L6 (f: 175 mm).

III. HOLOGRAM GENERATION AND COMPUTATION SPEED ENHANCEMENT

A. Hologram Generation

As we describe earlier, fast computation of holograms in video real time is still a challenging task using CPU based computation. It is well known that there is a Fourier relationship between the plane of hologram on SLM and the plane of the traps [10]. The trap patterns at the focal plane can be reconstructed by using the Fourier transform of the hologram at the hologram plane. Similarly, the hologram at the hologram plane can be obtained by the inverse Fourier transform of the trap patterns. Based on the application purpose such as trap quality and speed, various algorithms are available to compute the hologram required to generate required trap patterns. Among them, we have used gratings and lenses algorithm [13] to compute phase-only hologram in real time because it is non-iterative direct approach and thus computationally fast. The algorithm employs optical components of gratings (or equivalently prisms) and lenses to determine the inverse Fourier transform of required trap patterns. Gratings make lateral shift of the traps, while the lenses can make axial shift of the traps. The phase at the hologram plane required to generate the lateral shift ($\Delta x, \Delta y$) of the trap is represented as [13],

$$\phi_{prism}(x_h, y_h) = \frac{2\pi}{\lambda f}(x_h \Delta x + y_h \Delta y). \quad (1)$$

Where, x_h, y_h denote hologram plane coordinates, λ is the wavelength of the laser beam and f is the effective focal length of the objective lens.

If the trap moves along axial direction, the phase required to generate axial shift (Δz) of the traps is represented as [9],

$$\phi_{lens}(x_h, y_h) = \frac{2\pi z}{\lambda f^2}(x_h^2 + y_h^2). \quad (2)$$

Where, z is the desired axial displacement of the optical trap from the focal plane.

In addition, optical vortex with helical phase structure can be generated using Laguerre-Gaussian (LG) beam that can carry orbital angular momentum. The phase required to generate LG beam trap is given by [18],

$$\begin{aligned} \phi_{\text{vortex}}(x_h, y_h) \\ = -l \tan^{-1}(y_h / x_h) + \pi \theta[-L_p^{|l|}(2(x_h^2 + y_h^2) / w_0^2)]. \end{aligned} \quad (3)$$

Where, p and l denote radial mode and azimuthal mode respectively, $L_p^{|l|}(x)$ denotes generalized Laguerre polynomial, θ denotes unit step function and w_0 denotes the beam waist.

If the trap at the focal plane is displaced both laterally and axially, and has optical vortex, the phase at the hologram plane is sum of all terms in Eq.(1)-Eq.(3). Practically, when the phase is displayed on SLM, the phase value is first wrapped in the interval between 0 and 2π using modulus operation,

$$\begin{aligned} \phi_h(x_h, y_h) &= \phi_{\text{prism}} + \phi_{\text{lens}} + \phi_{\text{vortex}} \\ &= (\phi_{\text{prism}} + \phi_{\text{lens}} + \phi_{\text{vortex}}) \bmod 2\pi. \end{aligned} \quad (4)$$

And then conversion of the wrapped phase into the gray value (0~255) is performed. For this, the precise and linear phase modulation characteristic curve shown in Fig. 2, provided by the manufacturer, is applied.

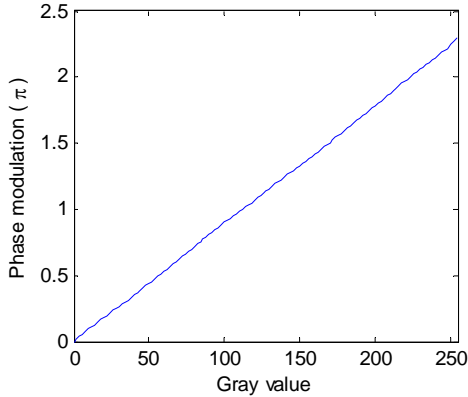


Fig. 2. Phase modulation characteristics of Hamamatsu X10468-3

For N optical traps, individual optical field of all traps $u_{h,n}$ at the hologram plane is computed using the phase in Eq. (4) with the unit field amplitude and is simply superpositioned to compute total field at the hologram plane [13]. Taking the argument of the total field, the phase ϕ_h that will be displayed on SLM can be determined using Eq. (5). In the implementation, the argument of total field is calculated using the sum of cosine and sine terms by as follows,

$$\begin{aligned} \phi_h &= \arg\left(\sum_{n=1}^N u_{h,n}\right) = \arg\left(\sum_{n=1}^N \exp(i\phi_{h,n})\right) \\ &= \tan^{-1}\left(\frac{\sum_{n=1}^N \sin(\phi_{h,n})}{\sum_{n=1}^N \cos(\phi_{h,n})}\right), \end{aligned} \quad (5)$$

where N denotes number of the trap and $\phi_{h,n}$ denotes the phase of n^{th} optical trap defined by Eq. (4). The equations so far describe a single iteration of gratings and lenses algorithm to compute the phase at a certain pixel (x_h, y_h) in the hologram plane. If the hologram dimension is $(M_x) \times (M_y)$, then the same process is repeated in a loop to compute the phase values for all the pixels within hologram plane to constitute the entire phase hologram.

In addition, in order to correct phase distortion due to the non-smooth surface of SLM chip, we performed wavefront correction using the correction image as shown in Fig. 3(a), provided by the manufacturer. Each pixel of this image has the phase represented as gray value (0~255) to correct the corresponding pixel of SLM chip. When the correction image is displayed on the SLM, the pattern of focal spot with aberration in Fig. 3(b) is corrected into the symmetric diffraction pattern as shown in Fig. 3(c). In the implementation, the phase from wavefront correction image is added to the Eq. (5).

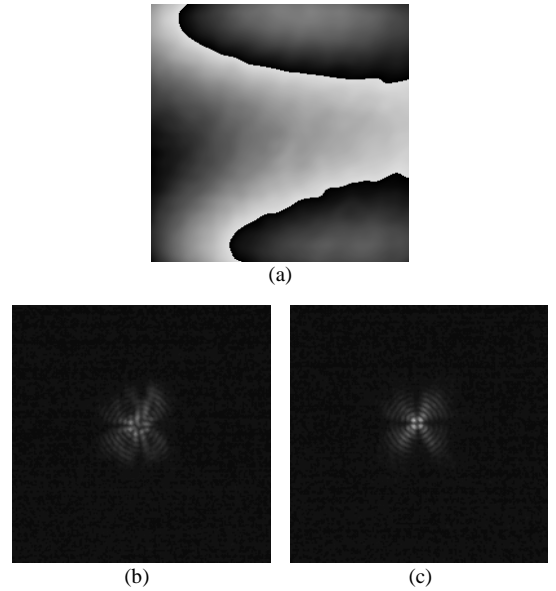


Fig. 3. SLM flatness correction. (a) phase holograms for wave front correction (512x512 pixels) (b) without correction, (c) with correction

Fig. 4 shows exemplary phase holograms to generate various optical traps, which are computed using gratings and lenses method with wavefront correction. Considering the diameter of incident beam on SLM, the used pixel area of SLM for hologram computation is limited to a region of

512×512 pixels within 800×600 pixels centered at the SLM chip, because the diameter of incident beam on SLM is 10 mm. Fig. 4(a)-(c) show the phase holograms for a single trap, dual traps, and triple traps configurations. Fig. 4(d) illustrates the phase hologram for two axially displaced traps. Fig. 4(e) shows the phase hologram for LG beam trap of mode $p = 3$ and $l = 3$.

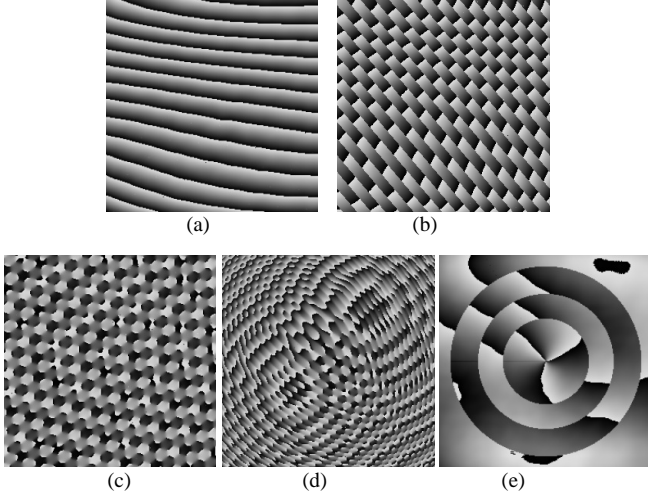


Fig. 4. Phase holograms to generate various optical traps (512×512 pixels). (a) single trap, (b) dual traps, (c) triple traps, (d) two axially displaced traps (e) single doughnut trap with LG beam of mode $p = 3$ and $l = 3$

B. Computation Speed Enhancement

For many applications, fast hologram generation is required for interactive control of the multiple optical traps. To enhance the computation speed, we used several tips. The first procedure is to use lookup table for trigonometric computation in Eq. (5). Trigonometric functions in standard “cmath” library are well optimized for accuracy and speed, but to compute the entire phase hologram, processing of these functions takes tremendous time resulting in the computation burden and deteriorating overall computation speed in case of CPU based computation. As shown in Eq. (5), the computation of phase at each pixel needs N times of sine and cosine operations and one time arc tangent operation excluding relatively non-time consuming operations such as addition, multiplication and division. Because the used hologram dimension is 512×512, the number of computation for evaluating sine and cosine terms account for 512×512× N making the computation speed worse. For this reason, decreasing the time for evaluating sine and cosine functions is required. As a good solution, the lookup table was used and it was generated using standard math library in advance in the memory. The lookup table can either be made regarding sine or can be made regarding cosine function since both the functions have just $\pi/2$ phase difference. The used lookup table can cover within $\pm 2\pi$ radians and the table size is set to 2048, which can be increased to enhance the accuracy of lookup table. For the table size of 2048, the error with sine and cosine functions in standard “cmath” library occurs at 4th

digit from the floating point. Considering the conversion of the phase into gray value, the error at 4th digit from the floating point is negligible. Using this method, the computation speed of sine and cosine functions could be enhanced up to seven times compared to the standard library.

Second procedure is by the incorporation of hologram binning e.g., CCD binning in CCD technology. Unlike CCD binning that combines adjacent pixels in a CCD during readout to make single representative pixel value, we define hologram binning as the process of assigning single pixel value to the adjacent pixels. Fig. 5 shows the example of 2×2 (quarter) hologram binning.

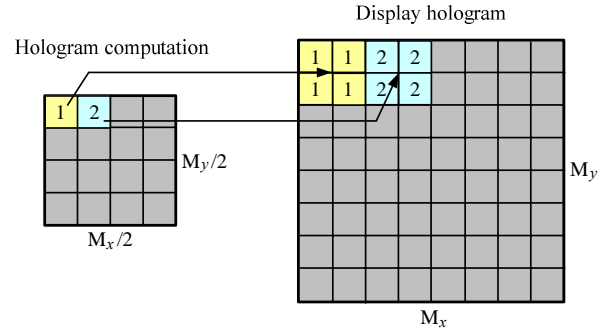


Fig. 5. 2×2 (quarter) hologram binning

In 2×2 hologram binning, the hologram dimension for computation of phase hologram is reduced to the quarter of the dimension of the used hologram displayed on SLM. And the phase value of a single pixel in the computed hologram fills neighboring 4 pixels in the display hologram. This approach decreases the computation time of the required hologram to 4 times. However, the number of pixel dimensions in the hologram computation may influence the quality of optical trap. For our optical setting, 2×2 hologram binning has worked quite well for the manipulation of trapped particles and there is not much difference in trap quality compared to full hologram as shown in Fig. 6.

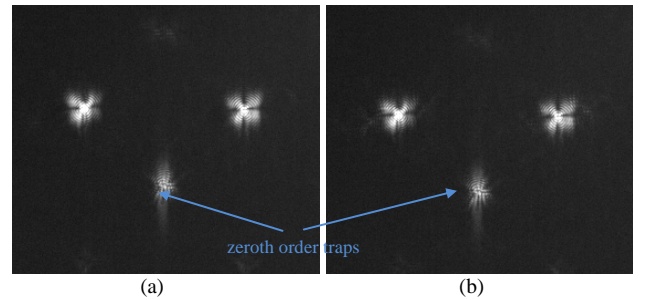


Fig. 6. Trap patterns with respect to the hologram binning mode using hologram dimension of 512×512 pixels. (a) full hologram, (b) 2×2 hologram binning

Third procedure is adaptive calculation of hologram only for the case when the trap is generated, deleted, and manipulated as suggested in [13]. Whenever the trap is generated, the phase of trap is calculated using Eq. (4) and is saved in the 512×512×4 bytes of individual memory for easy accessing. Then, the phases of newly generated traps are

added to the total phase and the resulting total phase is displayed on SLM to make corresponding trap configurations. If some traps are deleted from the current trap configuration, the phases of the deleted traps are subtracted from the total phase and the resulting total phase is displayed on SLM. If the positions of the traps are manipulated, the phases of only selected traps are recalculated and the resulting total phase is displayed on SLM. This prevents abrupt drop of hologram computation rate with an increased number of optical traps and can maintain constant hologram computation rate, regardless of the number of traps, if only one trap is manipulated at a time.

C. Computation Speed Enhancement

Based on the description of Sec. III, the control program was developed using Microsoft® Visual Studio 2003 in debug mode and test was done in release mode to enhance the computation speed of CPU. When the program is built in release mode, the compiler performs all available code optimizations to ensure that the outputted executables and libraries execute as efficiently as possible. The control software is composed of two classes for hologram computation and trap manipulations. First class is hologram computation class that produces the phase hologram for generating the arbitrary optical trap configurations. Second class is the mouse control class to handle several functionalities such as trap generation, trap deletion, trap selection, and trap manipulation.

Fig. 7 shows generation of optical traps using the developed control program. On the CCD screen, the five traps are generated and in each trap position, the circles are overlaid as trap avatars. All functionalities related to trap control are implemented using mouse click and drag for easy control of the optical traps. If a user selects the traps, the circles of selected traps are changed into different color and only the selected traps can be position controlled.

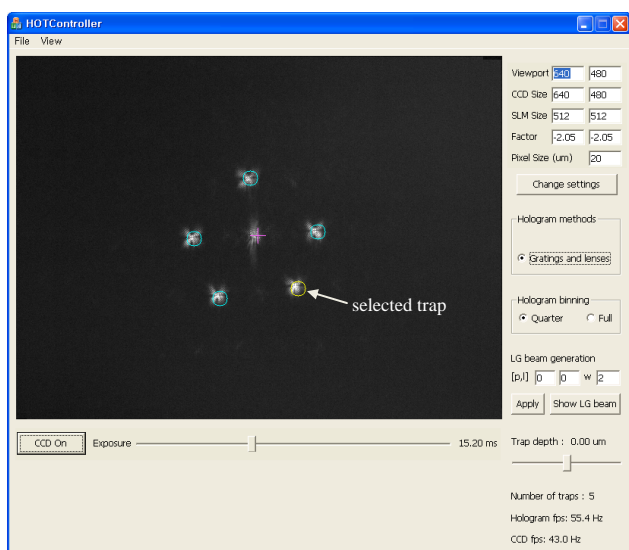


Fig. 7. Five optical traps generated using developed control program

The developed control software is capable of updating phase holograms with the update rate of 55 Hz using Intel Pentium IV processor at 3.0 GHz including CCD frame rendering run at around 40 Hz. The test was done using 512×512 hologram dimensions at 2×2 binning mode and the update rate of the hologram was nearly not influenced by the number of traps. The developed control software can generate up to two hundred traps in a system equipped with 2 GB RAM.

IV. EXPERIMENTAL RESULTS

To evaluate the performance of the discussed three methods for hologram computation speed enhancement, we conducted comparative analysis based on the developed program. Fig. 8 illustrates the comparison of hologram computation rates for the four cases denoted with different markers. The test was performed for the hologram dimensions of 512×512 pixels with a range of 1 to 50 numbers of optical traps. Circle and triangle markers indicate the results for the full binning and quarter binning with standard “cmath” library, respectively. Square marker denotes the case when the lookup table is used instead of standard “cmath” library in quarter binning. Cross marker denotes the case when the adaptive calculation is applied with quarter binning and lookup table methods. As can be seen in Fig. 8, quarter binning can greatly enhance the computation speed that is around four times at the single trap compared to the full binning. In addition, lookup table method can provide additional increment of ~12 Hz resulting in ~55 Hz computation speed in quarter binning.

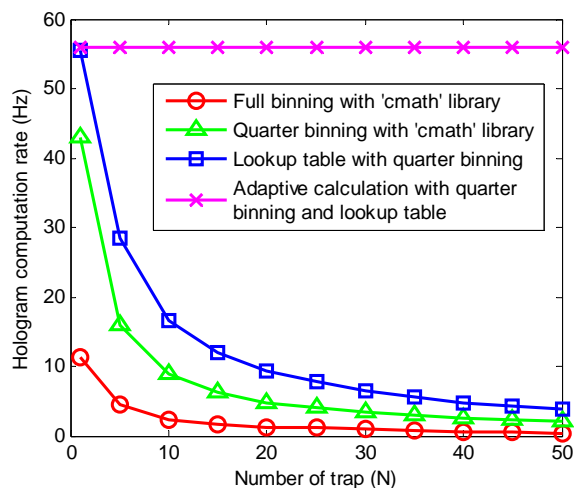


Fig. 8. Comparison of the hologram computation rate

However, this method could not show good performance as much as ~7 times increase in the computation rate as we tested in the environment computing only sine and cosine functions. We think this is due to the fact that the hologram computation in the developed program has more number of computations than simple lookup table test environment, slowing down the CPU based computation. Also, other operations in the program, i.e. microscopic frame grabbing

and visualization, competingly share the computing resources, lowering the performance of lookup table method.

As the number of traps are increasing, number of computations are also increased, which causes the sudden drop in hologram computation rate. This problem is alleviated with adaptive calculation method denoted as cross marker. By applying the adaptive calculation method with quarter binning and lookup table, we could get constant hologram computation rate around 55 Hz, regardless of the number of trap, if only one trap is manipulated at a time.

Using the developed control program, we conducted the optical trapping using 5 μm polystyrene beads. Fig. 9 illustrates the phase hologram (a) and trapped beads (b) for the optical trapping of five 5 μm polystyrene beads. In this experiment, three trapped beads are axially controlled while two beads remain at the focal plane to show three dimensional manipulations.

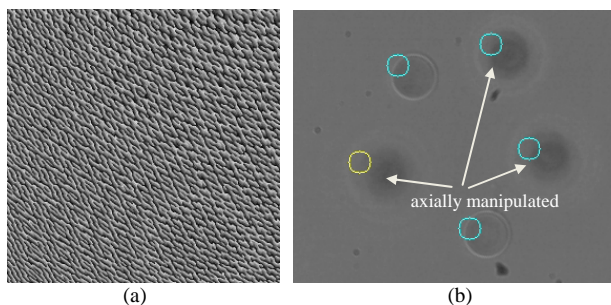


Fig. 9. Optical trapping of five 5 μm beads (a) Phase holograms for generating five trap configurations (b) Optically trapped five 5 μm beads

In addition, Fig. 10 shows consecutive images of the optical rotation of three trapped beads using single LG beam of mode $p = 0$ and $l = 10$. The trapped particles were rotated in the counter-clockwise direction and the rotation speed of trapped beads was around 5 Hz at 800 mW of source laser power.

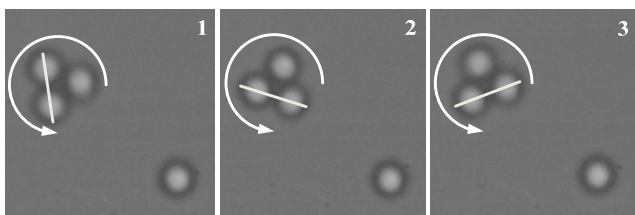


Fig. 10. Rotation of trapped three 5 μm beads using LG beam ($p = 0$, $l = 10$)

V. CONCLUSIONS

In this paper, we have presented the method and control software for interactive control of holographic optical traps using the gratings and lenses algorithm with CPU based computation. Specifically, lookup table, hologram binning and adaptive hologram calculation were applied and these could enhance the computation speed up to 55 Hz in release mode. The developed control software is capable of updating

phase holograms with the update rate of 55 Hz regardless the number of traps using Intel Pentium IV CPU at 3.0 GHz, including CCD frame rendering run at around 40 Hz. In the experimental demonstrations, we showed three dimensional manipulations of micro particles in real-time.

ACKNOWLEDGMENT

This work was supported by the institute of Medical System Engineering (iMSE) in GIST, Korea.

REFERENCES

- [1] K. Svoboda and S. M. Block, "Biological applications of optical forces," *Ann. Rev. Biophys. Biomol. Struct.*, vol. 23, pp. 247–285, 1994.
- [2] M. J. Lang, C. L. Asbury, J. W. Shaevitz, and S. M. Block, "An automated two-dimensional optical force clamp for single molecule studies," *BioPhys. J.*, vol. 83, pp. 491–501, 2002.
- [3] D. G. Grier, "Optical tweezers in colloid and interface science," *Curr. Opin. Colloid Interface Sci.*, vol. 2, pp. 264–270, 1997.
- [4] D. McGloin, "Optical tweezers: 20 years on," *Phil. Trans. R. Soc. A*, vol. 364, pp. 3521–3537, 2006.
- [5] D. G. Grier and Y. Roichman, "Holographic optical trapping," *Appl. Opt.*, vol. 45, pp. 880–887, 2006.
- [6] K. Sasaki, M. Koshioka, H. Misawa, N. Kitamura, and H. Masuhara, "Pattern formation and flow control of fine particles by laser-scanning micromanipulation," *Opt. Lett.*, vol. 16, pp. 1463–1465, 1991.
- [7] C. Mio, T. Gong, A. Terray, and D. W. M. Marr, "Design of a scanning laser optical trap for multiparticle manipulation," *Rev. Sci. Instrum.*, vol. 71, pp. 2196–2200, 2000.
- [8] K. C. Vermeulena, J van Mamerena, G. J. M. Stienen, E. J. G. Peterman, G. J. L. Wuite, and C. F. Schmidt, "Calibrating bead displacements in optical tweezers using acousto-optic deflectors," *Rev. Sci. Instrum.*, vol. 77, 013704, 2006.
- [9] J. E. Curtis, B. A. Koss, and D. G. Grier, "Dynamic holographic optical tweezers," *Opt. Commun.*, vol. 207, pp. 169–175, 2002.
- [10] E. R. Dufresne, G. C. Spalding, M. T. Dearing, S. A. Sheets, and D. G. Grier, "Computer-generated holographic optical tweezer arrays," *Rev. Sci. Instrum.*, vol. 72, pp. 1810–1816, 2001.
- [11] R. W. Gerchberg and W. O. Saxton, "A practical algorithm for the determination of the phase from image and diffraction plane pictures," *Optik*, vol. 35, pp. 237–246, 1972.
- [12] M. Reicherter, T. Haist, E. U. Wagemann, and H. J. Tiziani, "Optical particle trapping with computer-generated holograms written on a liquid-crystal display," *Opt. Lett.*, vol. 24, pp. 608–610, 1999.
- [13] J. Leach, K. Wulff, G. Sinclair, P. Jordan, J. Courtial, L. Thomson, G. Gibson, K. Karunwi, J. Cooper, Z. J. Laczik, M. J. Padgett, "Interactive approach to optical tweezers control," *Appl. Opt.*, vol. 45, pp. 897–903, 2006.
- [14] M. Montes-Usategui, E. Pleguezuelos, J. Andilla, and E. Martín-Badosa, "Fast generation of holographic optical tweezers by random mask encoding of Fourier components," *Opt. Express*, vol. 14, pp. 2101–2107, 2006.
- [15] E. Eriksson, S. Keen, J. Leach, M. Goksör, and M. J. Padgett, "The effect of external forces on discrete motion within holographic optical tweezers," *Opt. Express*, vol. 15, pp. 18268–18274, 2007.
- [16] T. Haist, M. Reicherter, M. Wu, and L. Seifert, "Using graphics boards to compute holograms" *Comp. in Sci. & Eng.*, vol. 8, pp. 8–13, 2006.
- [17] M. Reicherter, S. Zwick, T. Haist, C. Kohler, H. Tiziani, and W. Osten, "Fast digital hologram generation and adaptive force measurement in liquid-crystal-display-based holographic tweezers," *Appl. Opt.*, vol. 45, pp. 888–896, 2006.
- [18] Y. Ohtake, T. Ando, N. Fukuchi, N. Matsumoto, H. Ito, and T. Hara, "Universal generation of higher-order multiringed Laguerre-Gaussian beams by using a spatial light modulator," *Opt. Lett.*, vol. 32, pp. 1411–1413, 2007.