



University  
of Glasgow

Leach, J. et al. (2006) Interactive approach to optical tweezers control. *Applied Optics*, 45 (5). pp. 897-903. ISSN 0003-6935

<http://eprints.gla.ac.uk/67211/>

Deposited on: 18<sup>th</sup> July 2012

# Interactive approach to optical tweezers control

Jonathan Leach, Kurt Wulff, Gavin Sinclair, Pamela Jordan, Johannes Courtial, Laura Thomson, Graham Gibson, Kayode Karunwi, Jon Cooper, Zsolt John Laczik, and Miles Padgett

We have developed software with an interactive user interface that can be used to generate phase holograms for use with spatial light modulators. The program utilizes different hologram design techniques, allowing the user to select an appropriate algorithm. The program can be used to generate multiple beams and can be used for beam steering. We see a major application of the program to be in optical tweezers to control the position, number, and type of optical traps. © 2006 Optical Society of America

*OCIS codes:* 090.1970, 140.7010, 170.4520.

## 1. Introduction

Optical tweezers<sup>1</sup> have been implemented for many years. They use a high-magnification objective lens to tightly focus light, producing a region of high electric field. Dielectric particles are then attracted to this region through the gradient force such that microscopic particles can be trapped. In initial experiments, these particles are dielectric spheres, but the trapping of biological media has also been achieved.<sup>2–5</sup> The trapping of metal particles is also possible by use of the scattering force as the trapping mechanism.<sup>6,7</sup>

Holographic optical tweezers<sup>8,9</sup> use a diffractive optical element to shape the light into multiple foci so that multiple particles can be trapped. With this technique, complex arrangements of particles can be built, as well as dynamically rotated or scaled. In the past, holographic optical tweezers schemes have implemented static, one-off computer-generated phase masks that create the desired beam, but the inability

to update or quickly modify the design limits their functionality. The development and commercial availability of spatial light modulators (SLMs) over the past 3 years has revolutionized holographic optical tweezers, greatly expanding their range of applications. Indeed, it is possible to buy a stand-alone holographic optical tweezers system with an integrated SLM.<sup>10</sup> SLMs are computer-controlled holograms that can be updated at video frame rates. For holographic optical tweezers, they make possible interactive control of the traps.<sup>11–13</sup>

The drive to fully integrate SLMs with optical tweezers places increasing demand on computer and hologram-generation technologies. Effective and efficient hologram design has therefore become crucial to many optical tweezers schemes. Selecting the appropriate technique can allow the user real-time control of optical traps.

For our uses, we have developed a LabVIEW program that integrates a number of existing hologram design algorithms for different tweezers applications. These algorithms range from the use of the phase distribution of the superposition of the fields behind different lenses and wedges<sup>14</sup> to a Gerchberg–Saxton algorithm for three-dimensional light shaping.<sup>15</sup> We have also incorporated a direct binary search algorithm<sup>16,17</sup> to generate a hologram sequence that will build complex three-dimensional structures from much simpler starting configurations. Each technique has advantages appropriate for specific applications. In each case there is a trade-off in speed of calculation of the hologram with complexity of design. Here we describe our software and application to optical tweezers. Note that a similar system for inter-

---

J. Leach (j.leach@physics.gla.ac.uk), G. Sinclair, P. Jordan, J. Courtial, L. Thomson, G. Gibson, K. Karunwi, and M. Padgett are with the Department of Physics and Astronomy, University of Glasgow, Glasgow G12 8QQ, UK. K. Wulff is with the Center for Biologically Inspired Materials and Material Systems, Duke University, Durham, North Carolina 27708. J. Cooper is with the Department of Electronics and Electrical Engineering, University of Glasgow, Glasgow G12 8LT, UK. Z. J. Laczik is with the Department of Engineering Science, University of Oxford, Oxford OX1 3PJ, UK.

Received 11 May 2005; revised 30 June 2005; accepted 5 September 2005.

0003-6935/06/050897-07\$15.00/0

© 2006 Optical Society of America

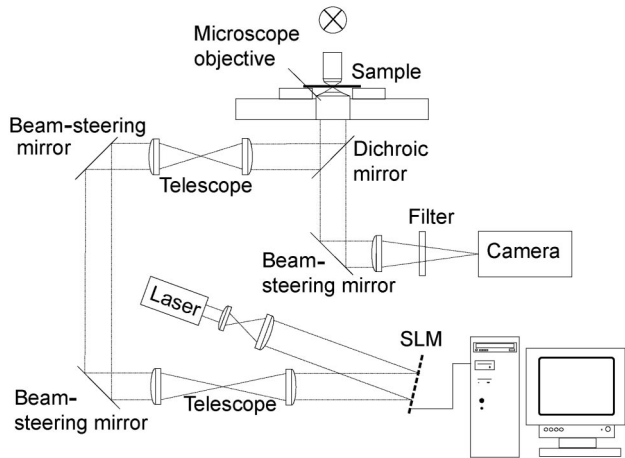


Fig. 1. Schematic diagram of a holographic optical tweezers system. The plane of the SLM is imaged into the pupil plane of the microscope objective.

active hologram design based on using a virtual pen has recently been reported.<sup>18</sup>

## 2. Program Overview

We have developed an interactive LabVIEW program that can be integrated within holographic optical tweezers, such as the system in Fig. 1. The program can be used to generate holograms such that multiple optical traps can be created and individually controlled. In addition, it is possible to overlay a live video image of the objects being manipulated onto a schematic showing the trap positions. This allows the user to use the computer mouse or a joystick as an interactive optical hand. The program is available from the Glasgow University Optics web page, <http://www.physics.gla.ac.uk/Optics/projects/tweezers/>.

Our program consist of three main parts. The first of these is the interface, which is shown in Fig. 2. This is where the user specifies the number of traps and the trap coordinates. For example, the simplest configuration would be to generate one trap with coordinates  $x$ ,  $y$ ,  $z$  relative to the focal point of the microscope objective. Complex arrangements with multiple trap positions can be specified; however, the trapping efficiency falls off with increasing displacement in  $x$ ,  $y$ , and  $z$ .<sup>19</sup> In addition, one is limited by the spatial resolution of the SLM, and there is a trade-off between trap complexity and hologram calculation time.

The interface contains an area that represents the trapping plane of the optical tweezers. Cursors can be placed within this area with the computer mouse to the positions of the desired traps. Clicking and dragging allows the user to change the coordinates of each trap, and traps can be added or deleted with the appropriate controls. The coordinates of the traps can also be entered numerically via the keyboard. It is also possible to display a suitably scaled camera image of the trapping plane as the background of this

The interface for SLM Control

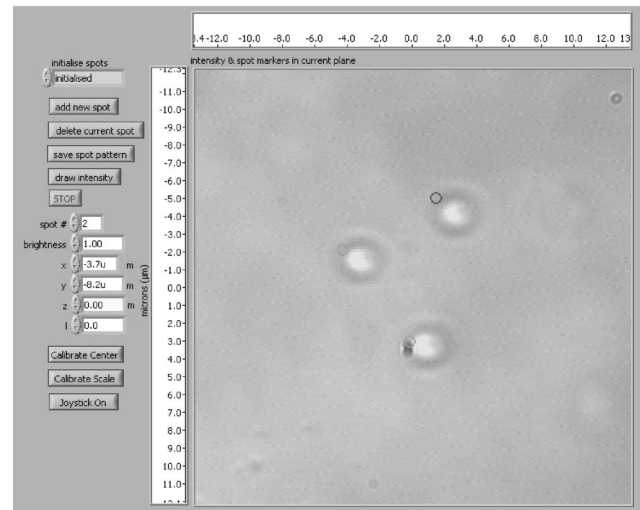


Fig. 2. User interface for the software. The circular markers indicate the trap positions. These can be moved with a mouse or joystick.

area. This gives the user visual feedback on the positions of the optical traps.

The coordinates that are generated with the interface are then used by the calculation engine to generate the corresponding hologram. Within the engine, one can specify the algorithm with which to calculate the hologram. There are four algorithms to choose from that we refer to as gratings and lenses,<sup>14</sup> Gerchberg–Saxton two-dimensional,<sup>20</sup> optimized gratings and lenses,<sup>11</sup> and direct binary search.<sup>16,17</sup> These are explained in more detail in Section 3. The hologram design is then passed to the final part of the program, which displays the calculated hologram pattern on the SLM.

## 3. Algorithm Summary

In this section we briefly describe the algorithms that are currently implemented within our program. An optical beam can be represented by a complex field, which can be written as

$$u = a \exp(i\phi). \quad (1)$$

The modulus  $a$  is the amplitude of the field and the argument  $\phi$  is the phase. SLMs can act as phase holograms, which have transmission  $t(x_h, y_h) = \exp[i\phi_h(x_h, y_h)]$  and which therefore modify only the phase of the beam. This is advantageous because the transmitted intensity is not reduced and therefore light throughput is maximized. When the beam given by Eq. (1) is incident on a SLM, the complex amplitude profile of the beam after passing through the SLM will be  $u_h = a \exp[i(\phi + \phi_h)]$ .

The algorithms that we implement are used to generate this phase pattern  $\phi_h$ . The phase pattern is designed such that when the modified beam  $u_h$  is focused by the objective lens, the light intensity distribution in the image space (at the focus of the ob-

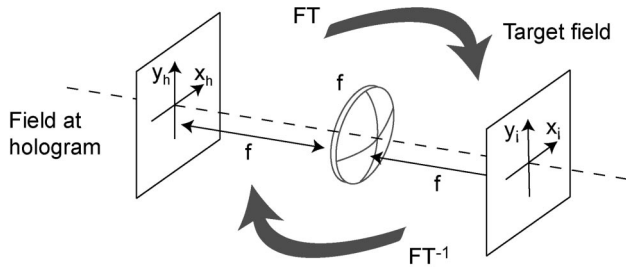


Fig. 3. Fourier relationship between the plane of the hologram and the image space of the objective lens. One can easily move between the two planes by using Fourier transforms (FT) and inverse Fourier transforms (FT<sup>-1</sup>).

jective lens) is the one desired, for example, multiple focused spots at prescribed positions.

For some of the design techniques we make use of the fact that there is a Fourier transform relationship between the complex field  $u_h$  in the pupil plane of the objective lens, onto which the plane of the SLM is imaged, and the complex field  $u_i$  at the focus of the lens in image space:

$$u_i(x_i, y_i) = \mathcal{F}\{u_h(x_h, y_h)\}. \quad (2)$$

The image space field can therefore be thought of as a scaled version of the far-field diffraction pattern that would be produced by the beam leaving the SLM. Clearly, if one of  $u_h$  or  $u_i$  is known, it is easy to calculate the other unknown field by taking the forward or inverse Fourier transform of the known field (Fig. 3).

Our program is a tool that calculates the phase-only hologram for a desired far-field intensity pattern. It is designed to accept a target intensity specification  $I_t(x_i, y_i) = |u_t(x_i, y_i)|^2$  for the image space of the objective lens. From this specification it then calculates a phase  $\phi_h(x_h, y_h)$  such that when it is displayed on the SLM the desired intensity distribution (within experimental error) is produced in the tweezers system at the focus of the objective lens.

#### A. Gratings and Lenses Algorithm

Of all the algorithms that we implement, the gratings and lenses algorithm is the simplest and computationally the fastest. It is a method that allows the user to generate holograms that will produce single or multiple independent foci.<sup>14</sup> The algorithm combines the phase of basic optical components: gratings, which result in lateral shifts, and lenses, which in a similar way produce axial shifts.

A single beam with planar phase fronts at the plane of the hologram, i.e.,  $\phi_h = \text{constant}$ , corresponds to a single focused spot in the image space. It is simple to move this point around by altering the phase at the hologram plane. For example, a spot in image space that is displaced laterally from the optic axis corresponds to a beam in the hologram plane that has an inclined phase front. This is equivalent to passing the light through a (small-angle) prism that

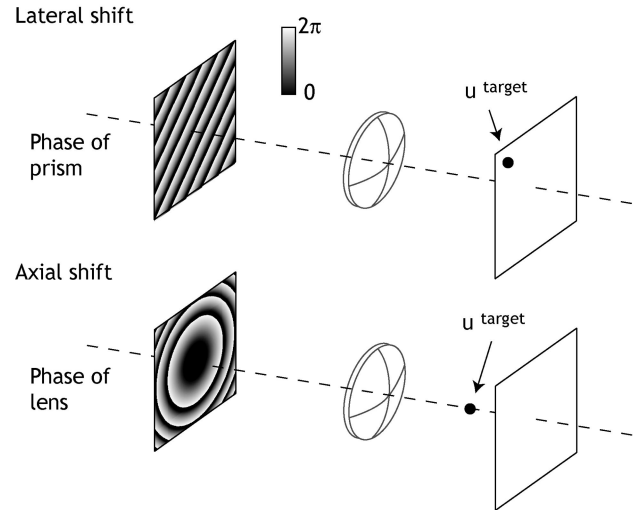


Fig. 4. Phase required for lateral and axial shifts.

introduces a linearly increasing phase delay  $\phi_{prism}$  across the beam. The phase at the hologram plane that is required to produce a lateral shift ( $\Delta x, \Delta y$ ) in the position of the focused spot is given by

$$\phi_{prism}(x_h, y_h) = a(\Delta x x_h + \Delta y y_h), \quad (3)$$

where  $a$  depends on the imaging characteristics and wavelength. Similarly, the phase that is required to introduce an axial shift in image space is given by

$$\phi_{lens}(x_h, y_h) = -\frac{k}{2f}(x_h^2 + y_h^2), \quad (4)$$

where  $f$  is a function of the axial shift distance and  $k$  is the wave vector of the light. This is equivalent to passing the light through an additional lens of focal length  $f$ . These two processes are clearly indicated in Fig. 4.

If the target intensity in image space is a focused spot that is displaced both laterally and axially, the phase of the beam at the hologram plane is required to be the (modulo  $2\pi$ ) sum of  $\phi_{prism}$  and  $\phi_{lens}$ :

$$\phi_h = (\phi_{prism} + \phi_{lens}) \bmod 2\pi. \quad (5)$$

To produce such a beam, these phases are calculated and added, and then the corresponding hologram is displayed on the SLM.

This simple method, to produce one point of high intensity at any arbitrary position can be easily extended to allow multiple beams to be generated and independently controlled. For  $N$  focused target spots there are  $N$  corresponding field components,  $u_{h,n}$ , at the plane of the hologram. Each of these field components can be calculated using the method described above. Adding these complex field components at the hologram plane gives the total field,  $u_h$ , that will produce the focused spots of light at the target positions. Taking the argument of this total field gives the

phase,  $\phi_h$ , that will be displayed on the SLM:

$$\phi_h = \arg\left(\sum_{n=1}^N u_{h,n}\right). \quad (6)$$

It should be noted that  $\phi_h$  in Eq. (6) is only an approximate solution to the phase of the desired beam. In particular, the magnitude of  $u_h$  is replaced by a constant and the relative phases of the optical traps are fixed. However, for many optical tweezers applications the above technique is a suitable and effective method of hologram calculation, and in practice the resulting optical traps are of sufficient quality for many applications.

#### B. Gerchberg–Saxton Two-Dimensional Algorithm

The second algorithm that is implemented to calculate holograms is the Fourier-transform-based Gerchberg–Saxton algorithm.<sup>20</sup> This algorithm can be used to calculate the phase required in the hologram plane to produce a predefined intensity distribution in the focal plane of the objective lens. The algorithm can be used to generate multiple focused spots of light or even arbitrary two-dimensional intensity distributions. Although there is a restriction to shaping light only in the focal plane, it is useful to look at the algorithm in some detail because it will be the starting point for the three-dimensional case below. In contrast to the gratings and lenses algorithm, where the phase between the traps is fixed, this algorithm takes advantage of the phase freedom and iteratively optimizes the focal-plane intensity distribution by varying both these phase values and  $\phi_h$ .

The target intensity distribution is defined in the image space as  $I_t(x_i, y_i)$ , and our aim is to find  $\phi_h(x_h, y_h)$  such that  $(\mathcal{F}\{\exp(i\phi_h)\})^2 = I_t$ . To find the desired phase  $\phi_h$ , we start at the hologram plane with a random phase  $\phi_r$  and constant unit amplitude. Therefore, for the first iteration step, we set

$$u_{h,1} = \exp(i\phi_r). \quad (7)$$

For the first and subsequent iterations, we propagate the field in the hologram plane to the image space by taking its Fourier transform, i.e., for the  $l$ th iteration:

$$u_{i,l} = \mathcal{F}\{u_{h,l}\}. \quad (8)$$

We then modify the image-space field; we keep the phase but replace the intensity with the target intensity:

$$\phi_{i,l} = \arg(u_{i,l}), \quad (9)$$

$$u_{i,l}' = \sqrt{I_t} \exp(i\phi_{i,l}). \quad (10)$$

The resulting field is then propagated back to the hologram plane by taking its inverse Fourier transform:

$$u_{h,l}' = \mathcal{F}^{-1}\{u_{i,l}'\}. \quad (11)$$

One iteration is finally completed by again keeping the phase but replacing the intensity with uniform constant intensity:

$$\phi_{h,l+1} = \arg(u_{h,l}'), \quad (12)$$

$$u_{h,l+1} = \exp(i\phi_{h,l+1}). \quad (13)$$

Note that it is possible, and sometimes desirable, to replace the intensity with a Gaussian profile. Here, an array of Gaussian traps is produced in the far field.

After a few iterations, the algorithm converges such that the argument of the field at the hologram plane is the phase that is required to produce the target field in image space,

$$\phi_h = \arg(u_h). \quad (14)$$

#### C. Optimized Gratings and Lenses Algorithm

The original Gerchberg–Saxton algorithm was extended so that three-dimensional arrays of traps could be generated.<sup>11,21</sup> It was recognized that the field in any plane in the image space can be calculated by applying a kernel to the field at the hologram plane and then calculating the Fourier transform. The kernel in this case corresponded to the phase profile that shifts the plane of interest to the focal plane of the objective lens, i.e., to the Fourier plane. As in the 2D case, this algorithm makes use of the phase freedom in image space.

We have developed a further extension to the above where the kernel includes the grating calculation as well as the lens calculation. The initial field in the hologram plane is a uniform intensity with the phase distribution in the hologram plane as calculated by the wedges and lenses algorithm:

$$u_{h,l}(x_h, y_h) = \exp\left\{i \arg\left[\sum_{p=1}^N k_{h,p}(x_h, y_h)\right]\right\}, \quad (15)$$

where  $k_{h,p}(x_h, y_h)$  is the field in the hologram plane that corresponds to a bright focus at the  $p$ th target position; this field will be referred to here as the kernel corresponding to the  $p$ th target position. We then calculate the field at each target position in turn. This can be done efficiently in the following way. First we multiply  $u_{h,l}(x_h, y_h)$  by  $k_{h,p}^*(x_h, y_h)$ ; this shifts target position  $p$  into the center of the image volume. Then the field at target position  $p$ —now the center of the image volume—is the field at the center of the Fourier transform, which can be calculated simply by integrating (in the computer, summing) over the entire field–kernel combination:

$$u_{i,l,p} = \iint dx_h dy_h u_{h,l}(x_h, y_h) k_{h,p}^*(x_h, y_h). \quad (16)$$

Because this is a Gerchberg–Saxton algorithm, we then replace the intensity at the target positions by the desired intensities (we assume here that the intensity of all the traps is the same, specifically 1) and calculate the corresponding field in the hologram plane. This field in the hologram plane is still going to be simply the sum of the fields corresponding to bright foci at the target positions, except that the fields at the target positions now have phases  $\arg(u_{i,l,p})$ . For the start of the next iteration of the algorithm the intensity needs to be set to a uniform distribution across the hologram, so the field at the hologram plane for the next iteration is simply

$$u_{h,l+1}(x_h, y_h) = \exp\left(i \arg\left\{\sum_{p=1}^N k_{h,p}(x_h, y_h) \times \exp[i \arg(u_{i,l,p})]\right\}\right). \quad (17)$$

This field converges after a number of iterations; its phase distribution is the desired phase-hologram pattern.

Note that this algorithm is essentially a wedges and lenses algorithm in which the relative phases of the traps are iteratively optimized using a Gerchberg–Saxton-type algorithm. This algorithm performs slightly better than the wedges and lenses algorithm when the target intensity is an array of traps with some degree of symmetry, for example, points arranged in the corners of a square.

#### D. Direct Binary Search Algorithm

The direct binary search (DBS) technique to hologram design is a powerful algorithm that allows the generation of arbitrary light distributions in three dimensions.<sup>16,17</sup> This is an iterative algorithm where changes to the phase at the hologram plane are assessed in the image space.

Starting at the hologram plane, we begin with a two-dimensional array of random phase values,  $\phi_h = \phi_r$ . This is the starting point for the hologram. This starting field at the hologram corresponds to some light distribution  $u_i$  in the image space, which can be compared to the desired target field  $u_t$ . There is no restriction on the target field; for example, it can be an array of spots in three dimensions or a line in the focal plane of the objective lens. By comparing the two fields, one can define an error function that is an indicator of the performance of the hologram. As we are comparing two fields, it is possible to define the error function on the basis of any property of the field, the intensity, amplitude, phase, or any combination. A common choice of error function,  $\varepsilon$ , is the difference between the modulus of the amplitudes of the two fields at the target positions:

$$\varepsilon = \sum_{n=1}^N \left| |u_{t,n}| - |u_{i,n}| \right|. \quad (18)$$

The next step of the algorithm is to change the phase of a random pixel at the hologram plane. This

change will manifest itself as a change to the field in image space. The corresponding new error value can be compared to the existing error to assess whether the change in phase resulted in a field closer to the target field than before. If yes, the pixel change is accepted; otherwise the pixel phase is returned to its previous value.

This is one iteration of the DBS and it is repeated many times. However, due to the iterative approach where each pixel is addressed in turn, in general the calculation time for DBS is much longer than for the other algorithms. A hologram of dimensions  $(M_x, M_y)$  pixels requires  $M_x \times M_y$  iterations before each pixel has had a chance of being altered, and it takes a number of visits to each pixel until a satisfactory hologram is found. The consequence of this is that the calculation time is very long; for example, depending on the error function, a  $512 \times 512$  hologram can take minutes to calculate.

#### E. Decreasing the Hologram Calculation Time

For particular applications, it is desirable to have interactive control of multiple optical traps. Currently, the limiting factor in interactive control is the calculation of the required holograms. The update rate of commercial SLMs is between 20 and 60 Hz (Hamamatsu and HoloEye), whereas the calculation rate of a hologram of  $512 \times 512$  on a Pentium 4 dual-processor machine using the gratings and lenses algorithm is between 3 and 5 Hz. Clearly, optimizing the algorithms to generate holograms more efficiently is desirable.

The first step to increasing the calculation rate is to perform checks on the trap positions to determine whether it is necessary to recalculate a new hologram. As an example, let us assume there are three traps (target points) in the image space. The hologram required to generate this field can be calculated using the gratings and lenses algorithm. From Eq. (6), the algorithm first calculates each of the  $u_{h,n}$  field components in the hologram plane corresponding to each target point. The phase of the required beam is the argument of the sum of all the fields. If one of the trap positions was to move, it was necessary to generate a new hologram, but it was not necessary to recalculate all the  $u_{h,n}$  field components. The only field component that has to be recalculated is the one that corresponds to the new trap position. Performing a check on the trap position before summing the fields clearly saves computation time and speeds up the calculation rate. This is particularly important as the number of traps increases. This technique is implemented within the gratings and lenses and the optimized gratings and lenses algorithms.

The second procedure that we use saves computation time by not calculating a full 2D Fourier transform when field values are only required at discrete points rather than in a plane. This is discussed in the previous section. Using the Fourier transform approach, the field at a single point in image space would be calculated as

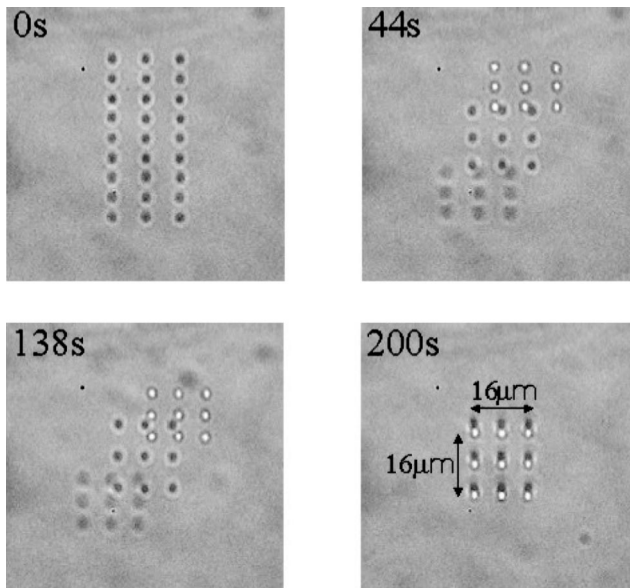


Fig. 5. Using our software to generate a  $3 \times 3 \times 3$  array of  $1 \mu\text{m}$  silica spheres from a two-dimensional grid of  $3 \times 9$  trapped spheres.

$$\begin{aligned}
 u_i(x_i, y_i) &= \mathcal{F}\{u_h\} \\
 &= \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} u_h(x_h, y_h) \exp\left\{i\left[\frac{k}{f}(x_h^2 + y_h^2) + a(x_h x_i + y_h y_i)\right]\right\} dx_h dy_h. \quad (19)
 \end{aligned}$$

For a pixelated hologram implemented with a SLM, Eq. (19) simplifies as

$$u_i(x_i, y_i) = \sum_{p=1}^{M_x} \sum_{q=1}^{M_y} u_{i,p,q}(x_i, y_i), \quad (20)$$

where  $u_{i,p,q}(x_i, y_i)$  are the contributions from each pixel in the hologram plane to the total field at  $(x_i, y_i)$ . If the field can be assumed to be uniform over a single pixel,  $u_{i,p,q}(x_i, y_i)$  will be sinc functions corresponding to the Fourier transform of a single square pixel positioned at  $(x_h, y_h)$ , where  $(x_h, y_h)$  is the center coordinate of the pixel with index  $(p, q)$ . If the pixels are small and the point  $(x_i, y_i)$  is sufficiently close to the optic axis, the magnitude of the sinc function can be ignored and only the pixel phases have to be taken into account in the summation. In practice, the calculation speed can be further improved by setting up lookup tables to avoid the recalculation of the pixel contributions. This allows us to calculate the field at a particular point without beam propagation and the calculation of a full Fourier transform and greatly reduces the overall calculation time.

#### 4. Example

Figure 5 shows an example of this technique. A  $3 \times 3 \times 3$  array of  $1 \mu\text{m}$  silica spheres is generated from an initial two-dimensional grid of  $3 \times 9$  trapped

spheres. The coordinates of the trap positions had been calculated beforehand and then loaded in sequence. The optimized gratings and lenses algorithm was used to generate the corresponding hologram. There were 50 intermediate steps between the starting and finishing positions, and the time between each was 4 s.

#### 5. Conclusions

Our software provides a simple user-friendly interface that can be integrated within holographic optical tweezers. It gives the user three-dimensional control of trap positions by using simple cursors controlled by the computer mouse or joystick—or in terms of pre-defined trajectories. We believe that our software has the potential to become a standard part of holographic optical tweezers.

One of the most powerful features of our software is that one can load in a series of coordinates corresponding to trap positions. The engine will then calculate the required holograms, and these can be displayed on the spatial light modulator. With this technique, complex three-dimensional arrays of trapped particles can be generated starting from much simpler trap positions, for example, lines of traps in two dimensions.

#### References

1. A. Ashkin, J. M. Dziedzic, J. E. Bjorkholm, and S. Chu, "Observation of a single-beam gradient force optical trap for dielectric particles," *Opt. Lett.* **11**, 288–290 (1986).
2. A. Ashkin and J. M. Dziedzic, "Optical trapping and manipulation of viruses and bacteria," *Science* **235**, 1517–1520 (1987).
3. S. Sato, M. Ishigure, and H. Inaba, "Optical trapping and rotational manipulation of microscopic particles and biological cells using higher-order mode Nd:YAG laser beams," *Electron. Lett.* **27**, 1831–1832 (1991).
4. J. E. Molloy, J. E. Burns, J. C. Sparrow, R. T. Tregear, J. Kendrick-Jones, and D. C. S. White, "Single-molecule mechanics of heavy-meromyosin and s1 interacting with rabbit or drosophila actins using optical tweezers," *Biophys. J.* **68**, S298–S305 (1995).
5. S. Sato and H. Inaba, "Optical trapping and manipulation of microscopic particles and biological cells by laser beams," *Opt. Quantum Electron.* **28**, 1–16 (1996).
6. M. Edidin, S. C. Kuo, and P. Sheetz, "Lateral movements of membrane glycoproteins restricted to dynamic cytoplasmic barriers," *Science* **29**, 1379–1382 (1991).
7. A. T. O'Neil and M. J. Padgett, "Three-dimensional optical confinement of micron-sized metal particles and the decoupling of the spin and orbital angular momentum within an optical spanner," *Opt. Commun.* **185**, 139–143 (2000).
8. E. Dufresne and D. Grier, "Optical tweezer arrays and optical substrates created with diffractive optics," *Rev. Sci. Instrum.* **69**, 1974–1977 (1998).
9. 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.* **24**, 608–610 (1999).
10. Arryx Inc., BioRyx 200, <http://www.arryx.com/>.
11. J. E. Curtis, B. A. Koss, and D. G. Grier, "Dynamic holographic optical tweezers," *Opt. Commun.* **207**, 169–175 (2002).
12. V. Bingelyte, J. Leach, J. Courtial, and M. J. Padgett, "Optically controlled three-dimensional rotation of microscopic objects," *Appl. Phys. Lett.* **82**, 829–831 (2003).

13. J. Leach, G. Sinclair, P. Jordan, M. J. Padgett, J. Courtial, J. Cooper, and Z. J. Laczik, "3D manipulation of particles into crystal structures using holographic optical tweezers," *Opt. Express* **12**, 220–226 (2004).
14. J. Liesener, M. Reicherter, T. Haist, and H. J. Tiziani, "Multi-functional optical tweezers using computer-generated holograms," *Opt. Commun.* **185**, 77–82 (2000).
15. G. Sinclair, J. Leach, P. Jordan, G. Gibson, E. Yao, Z. J. Laczik, M. J. Padgett, and J. Courtial, "Interactive application in holographic optical tweezers of a multi-plane Gerchberg–Saxton algorithm for three-dimensional light shaping," *Opt. Express* **12**, 1665–1670 (2004).
16. M. A. Seldowitz, J. P. Allebach, and D. W. Sweeney, "Synthesis of digital holograms by direct binary search," *Appl. Opt.* **26**, 2788–2798 (1987).
17. Z. J. Laczik, "3D beam shaping using diffractive optical elements," in *Laser Beam Shaping III*, F. M. Dickey, S. C. Holswade, and D. L. Shealy, eds., *Proc. SPIE* **4770**, 104–111 (2002).
18. X. Xun, X. Chang, and R. W. Cohn, "System for demonstrating arbitrary multispot beam steering from spatial light modulators," *Opt. Express* **12**, 260–268 (2004).
19. G. Sinclair, P. Jordan, J. Leach, M. J. Padgett, and J. Cooper, "Defining the trapping limits of holographical optical tweezers," *J. Mod. Opt.* **51**, 409–414 (2004).
20. R. W. Gerchberg and W. O. Saxton, "A practical algorithm for the determination of the phase from image and diffraction plane pictures," *Optik* **35**, 237–246 (1972).
21. R. Piestun and J. Shamir, "Generalized propagation-invariant wave fields," *J. Opt. Soc. Am. A* **15**, 3039–3044 (1998).