

# Experimental demonstration of holographic three-dimensional light shaping using a Gerchberg-Saxton algorithm

Graeme Whyte and Johannes Courtial

Department of Physics & Astronomy, University of Glasgow, Glasgow, Scotland, UK

E-mail: g.whyte@physics.gla.ac.uk

**Abstract.** We use a three-dimensional Gerchberg-Saxton algorithm [G. Shabtay, Opt. Commun. **226**, 33 (2003)] to calculate the Fourier-space representation of physically realizable light beams with arbitrarily shaped three-dimensional intensity distributions. From this representation we extract a phase-hologram pattern that allows us to create such light beams experimentally. We show several examples of experimentally shaped light beams.

PACS numbers: 41.85.Ct Beam shaping, beam splitting , 41.85.Ew Beam profile, beam intensity , 42.40.Eq Holographic optical elements; holographic gratings , 42.40.Jv Computer-generated holograms

## 1. Introduction

A monochromatic light beam propagating in free space is a three-dimensional (3D) field and its intensity distribution forms 3D patterns. At the same time, the field cross-section in any one transverse plane completely determines the field everywhere else. This dependency of the entire beam on its cross-section in a single plane forms the basis of beam propagation methods (for example [1]). It also restricts the possible 3D shapes a monochromatic light beam's intensity can take.

Many methods allow shaping of light beams in 2D, ranging from passing the beam through a shaped aperture to algorithms for calculating phase-hologram patterns that shape the beam in a subsequent plane (for example [2]). Shaping light beams into non-trivial but restricted 3D shapes is possible by utilising phenomena such as the Talbot effect [3, 4] or spiral-type beams [5, 6]. It is also possible to shape a light beam into 3D configurations of bright spots [7], whereby the spots can be shaped individually [8]; these techniques are important in the field of holographic optical tweezers. 3D beam shaping into arbitrary shapes has been demonstrated using computationally intensive direct search methods‡ [9], but continues to be a challenge.

A recent algorithm [10] finds beams whose shape is an approximation within physically realisable limits to any arbitrary 3D target shape. We apply this algorithm to calculate examples of light beams shaped in 3D. From the beams' Fourier-space representations we calculate phase-only hologram patterns that allow us to create the beams experimentally.

This paper is organised as follows. Section 2 reviews the Gerchberg-Saxton (GS) algorithm [2], which allows light shaping in one plane and forms the basis of what is to follow. Section 3 discusses the 3D generalisation of the GS algorithm [10] that we use to calculate shaped light beams. Some of the details of our implementation of the 3DGS algorithm can be found in section 4. Finally, section 5 describes our experiment for the generation of shaped light.

## 2. Gerchberg-Saxton algorithm

The Gerchberg-Saxton (GS) algorithm [2] is an iterative method originally developed for recovering the phase of an electron or light beam from its intensity distributions in two transverse planes. It can also be applied to shape a light beam, specifically to calculate the phase pattern which light at one plane would require to form an almost diffraction-limited approximation to any desired intensity pattern at a second plane. As the phase pattern required in the first plane can be imprinted onto the light beam with a phase hologram, this allows two-dimensional (2D) holographic light shaping. If the requirements in the two planes cannot be met simultaneously within what is allowed by the laws of diffraction, the GS algorithm finds a useful compromise to reconcile

‡ Direct search methods allow shaping not only of the light intensity, to which we are restricted here, but also of the phase. They are therefore more versatile than the technique we discuss here.

these conflicting requirements [11]. For numerical simplicity, the second plane is usually chosen to be the far field of the first plane; mathematically, the fields in the two planes are then Fourier transforms of each other. Here we restrict ourselves to this case.

We write the intensity distribution of the unshaped light beam in the first plane – the hologram plane – as  $I_H$  and the target intensity distribution in the Fourier plane as  $I_T$ . Both  $I_H$  and  $I_T$  are functions of  $x$  and  $y$ , which are stored in the computer as 2D arrays of real-valued numbers. The light fields in the two planes are represented by 2D arrays of complex numbers,  $u_n$  in the hologram plane and  $\tilde{u}_n$  in the Fourier plane; the subscript  $n$  indicates the iteration number. The initial phase distribution in the hologram plane,  $\varphi_0^H$ , is set to any arbitrary distribution, often uniform, i.e.

$$\varphi_0^H \equiv 0. \quad (1)$$

One iteration of the algorithm, which calculates an improved phase distribution in the hologram plane,  $\varphi_n^H$ , from the previously calculated phase distribution,  $\varphi_{n-1}^H$ , then progresses as follows:

$$u_n^H = \sqrt{I_H} \exp(i\varphi_{n-1}^H). \quad (2)$$

$$\varphi_n^T = \arg(\text{FFT}(u_n^H)) \quad (3)$$

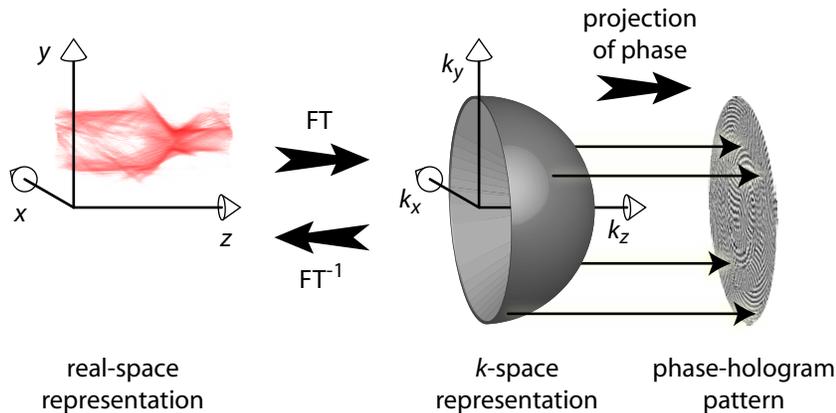
$$u_n^T = \sqrt{I_T} \exp(i\varphi_n^T) \quad (4)$$

$$\varphi_n^H = \arg(\text{FFT}^{-1}(u_n^T)) \quad (5)$$

Equation (2) calculates the initial field in the hologram plane, from which equation (3) then calculates the phase distribution in the target plane,  $\varphi_n^T$ . It uses a Fast Fourier transform (FFT), or more generally any discrete Fourier transform. Equation (4) combines this phase distribution in the target plane with the target intensity,  $I_T$ , giving the field  $u_n^T$ , from which equation (5) then calculates the corresponding phase distribution in the hologram plane,  $\varphi_n^H$ . Over a number of iterations, the actual intensity in the target plane,  $I_n = |u_n^T|^2$ , converges to an almost diffraction-limited approximation of the desired intensity there,  $I_T$ ;  $\varphi_n^H$  is the corresponding phase-hologram pattern needed to produce this field in the target plane.

### 3. Gerchberg-Saxton algorithm in three dimensions

The fields in the two planes in the GS algorithm can also be seen as a field in a real-space plane ( $x, y$ ) and its  $k$ -space ( $k_x, k_y$ ) representation. This allowed Shabtay [10] to adapt the GS algorithm to 3D: instead of dealing with fields in two planes, the algorithm deals with fields in two volumes. One field is a 3D real-space representation of the part of the beam that is to be shaped, the other is its 3D  $k$ -space representation, which needs to be consistent with the beam's wavenumber spectrum. The two fields are 3D Fourier-transform pairs (figure 1). Note that the 2D Gerchberg-Saxton algorithm usually shapes light in the Fourier plane, while the real-space plane contains restrictions such as the intensity profile of the unshaped beam, whereas the 3D GS algorithm shapes light in the real-space volume while the Fourier volume contains the physical constraints.



**Figure 1.** Schematic representation of a monochromatic light beam in real space (left) and  $k$  space (right). In  $k$  space, a monochromatic light beam is restricted to the surface of a hemisphere of radius  $k_0 = 2\pi/\lambda$ . A projection of the phase on this hemisphere into a plane (see section 5.1) provides a phase-hologram pattern suitable for beam shaping.

As discussed in the introduction, the field in a single plane completely describes a monochromatic light beam. This is reflected by the fact that the  $k$ -space representation of such a beam is confined to a surface. In monochromatic light of wavelength  $\lambda$ , the wave-vector component in the direction of propagation,  $k_z$ , is related to the values of  $k_x$  and  $k_y$  through the equation

$$k_0^2 = k_x^2 + k_y^2 + k_z^2, \quad (6)$$

where  $k_0 = 2\pi/\lambda$ . This forms a sphere of radius  $k_0$  in 3D  $k$  space – an Ewald sphere [10]. We consider here light moving in the positive  $z$  direction only, hence the wave vectors are limited to the half of the sphere with  $k_z > 0$  (figure 1).

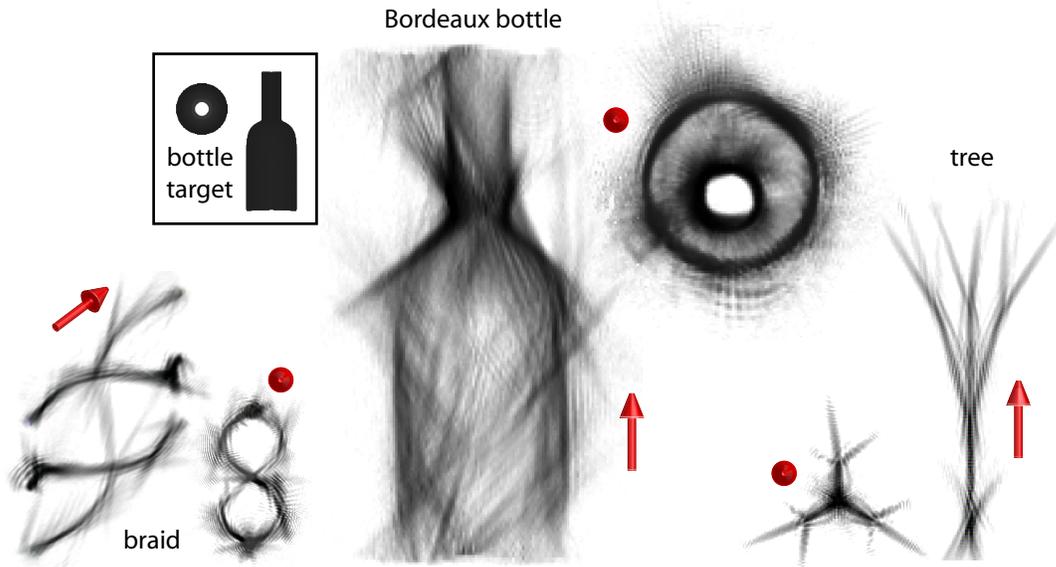
We have written a C++ implementation of the 3D GS algorithm. Figure 2 shows examples of light shaped using this program. The algorithm can clearly produce intensity distributions in which the target shapes can be recognized. When, like in figure 2, the intensity is shown inverted, the intensity distributions sometimes look like some form of three-dimensional expressionist charcoal drawings.

#### 4. Details of our implementation of the 3D Gerchberg-Saxton algorithm

In this section we explore some of the details of the implementation of the algorithm. We use the error [12]

$$\epsilon_n = \sqrt{\frac{\sum_{x,y,z} |\sqrt{I_n} - \sqrt{I_T}|^2}{\sum_{x,y,z} I_T}} \quad (7)$$

to quantify the similarity between the target intensity distribution,  $I_T(x, y, z)$ , and the intensity distribution the algorithm produces after  $n$  iterations,  $I_n(x, y, z)$ .



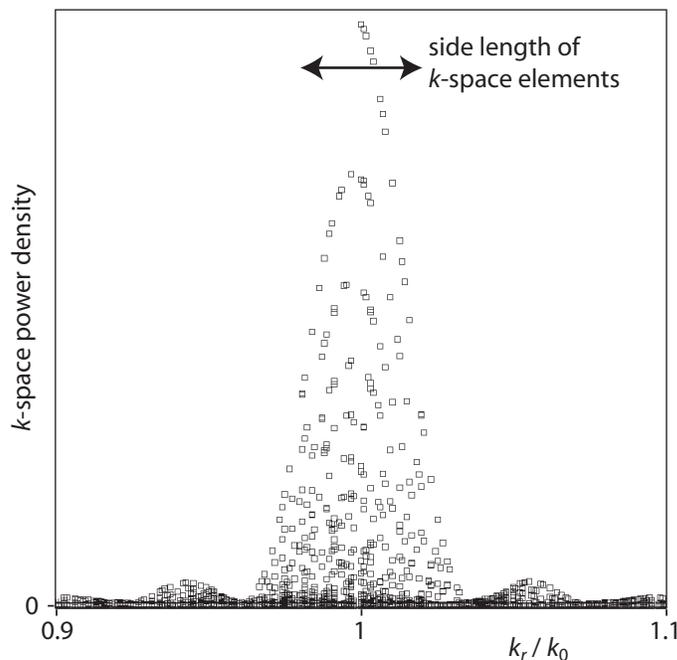
**Figure 2.** Examples of 3D light intensities shaped by the 3D Gerchberg-Saxton algorithm. High intensities are shown dark. Each light beam is shown from two directions; the propagation direction of the beam is indicated by a red arrow (the three smaller views look into the beam). In the case of the Bordeaux-bottle beam, two corresponding views of the target intensity distribution are also shown. The beams were calculated for a  $k$ -space cone angle  $\alpha = 90^\circ$  (see figure 5).

#### 4.1. Thickness of $k$ -space sphere

A purely monochromatic beam restricts the allowed  $k$ -space elements to a spherical surface. In our numerical simulations 3D  $k$  space is discretised, and it is not immediately obvious how to represent the spherical surface that corresponds to a monochromatic light beam in this discrete space. At the same time, the beam’s real-space representation is restricted to a cuboid volume of finite size, which implies that the light’s wavelength cannot be represented exactly.

In an effort to understand this better we investigate here one aspect associated with the discrete nature of a light beam whose 3D light field and  $k$ -space distribution are represented on cubic grids of points. Figure 3 shows the radial cross section of the discrete- $k$ -space representation of a simulated monochromatic light beam. This  $k$ -space representation was calculated by taking the three-dimensional discrete Fourier transform of the beam’s spatial representation over a discrete 3D volume. The spatial representation was calculated using a standard beam-propagation algorithm§ [1], starting with the field cross-section of a Gaussian beam close to the beam waist and propagating it into 63 further, equally-spaced, planes. In addition to a peak centered at  $k_0 = 2\pi/\lambda$ , the radial profile of the resulting discrete- $k$ -space-hemisphere distributions

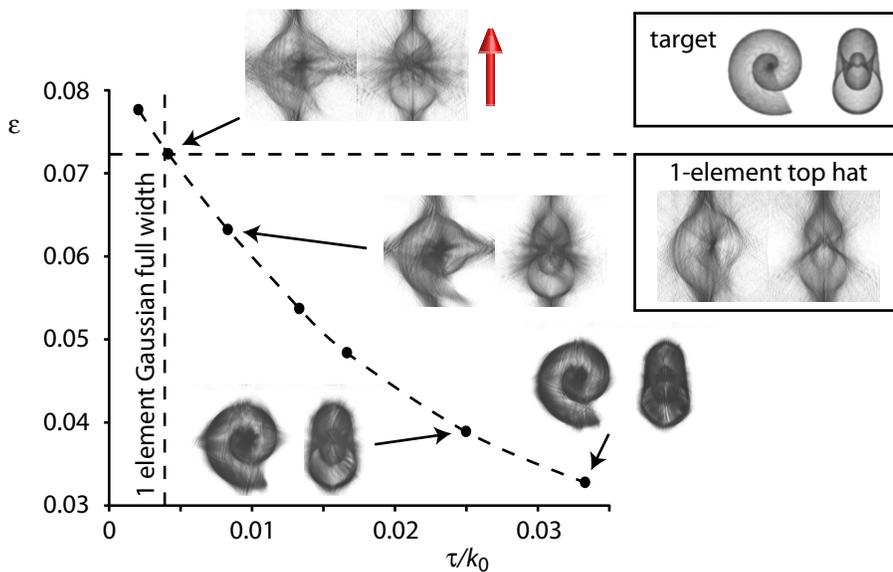
§ Instead of using the quadratic approximation used in ref. [1], we use the exact relationship, equation (6), to calculate  $k_z$  from  $k_x$  and  $k_y$ . We also use an absorbing boundary [13] (which is later removed) to avoid effects due to the periodic boundary conditions implicit in the discrete-Fourier-transform-based beam-propagation algorithm.



**Figure 3.** Example of the representation of a monochromatic light beam in discrete  $k$  space. The plot shows the powers in all discrete- $k$ -space elements with  $0.9k_0 \leq k_r \leq 1.1k_0$  (where  $k_0 = 2\pi/\lambda$ ) as a function of the radial wave number,  $k_r$ , calculated for a tightly focussed monochromatically Gaussian light beam. The discrete  $k$ -space representation was found by numerically calculating the light beam on a  $64 \times 64 \times 64$  grid representing a cube of side length  $25\lambda$  (where  $\lambda$  is the wavelength of the light), which was then 3D-Fourier-transformed. Whereas in continuous  $k$  space all the power would be at  $k_r = k_0$ , in discrete  $k$  space the power is distributed within a few  $k$ -space-element widths of  $k_0$ . Because the edges of the represented cube act like hard-edged slit apertures in the  $x$ ,  $y$  and  $z$  directions, the light field's  $k$ -space representation is that of continuous  $k$ -space, convolved with sinc functions in the  $k_x$ ,  $k_y$  and  $k_z$  directions (it is, of course, also discrete). As the beam used in this example propagates mainly in the positive  $z$  direction, the radial  $k$ -space structure of the power density broadly reflects the  $\text{sinc}^2$ -type structure in the  $k_z$  direction.

has other distinct features. We believe these additional features to be due to the beam being represented only in a cube and the represented beam therefore having a top-hat profile in the  $x$ ,  $y$  and  $z$  directions, which in turn leads to the  $k$ -space distribution being widened, more precisely being convolved with a sinc function in the  $k_x$ ,  $k_y$  and  $k_z$  directions.

In our simulations we use perhaps the simplest form of the  $k$ -space hemisphere: for each represented pair of  $k_x$  and  $k_y$  values, the power in the discrete  $k_z$  value closest to  $(k_0^2 - k_x^2 - k_y^2)^{1/2}$  is set to one, that of all the other  $k_z$  values to zero. Such a  $k$ -space hemisphere is exactly one element thick in the  $k_z$  direction; the radial power profile would be a top hat of width 1 element. This is clearly different from the arguably ‘natural’ representation shown in figure 3, but it works well enough to produce good experimental results (see below) and simplifies the extraction of phase-hologram patterns (see section 5.1). However, it is clearly an aspect of our implementation that could benefit from

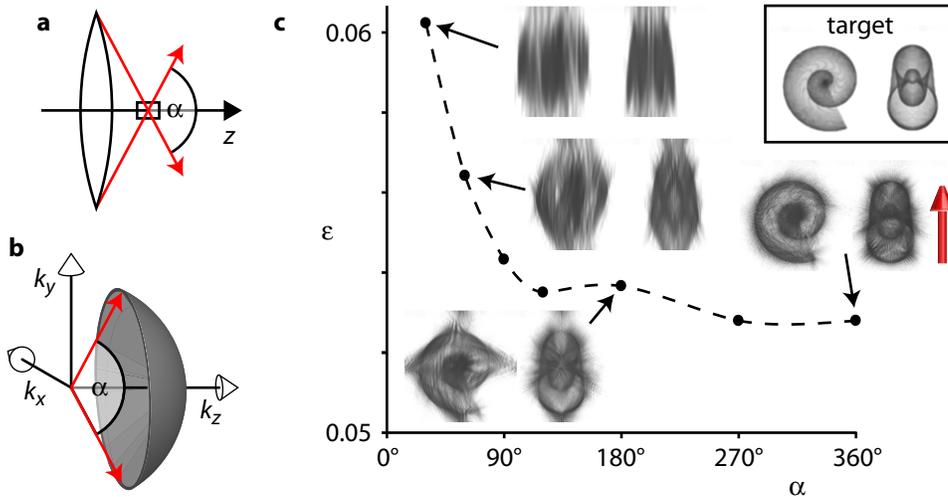


**Figure 4.** Effect of the thickness of the  $k$ -space hemisphere. The points show the error,  $\varepsilon$ , corresponding to light beams shaped with the 3D GS algorithm using  $k$ -space hemispheres of various Gaussian half-thicknesses,  $\tau$ . More precisely, the power in each  $k$ -space element with  $k_z > 0$  is set according to its distance from the origin according to the equation  $p(k) = \exp(-(k - k_0)^2 / (2\tau^2))$ , where  $k_0 = 2\pi/\lambda$  is the radius of the hemisphere; the power of  $k$ -space elements with  $k_z \leq 0$  is set to zero. The light beams corresponding to some of the points are also shown, each from two different perspectives; the propagation direction (red arrow) is always in the vertical direction. Also shown are the target shape (top inset on the right) and the light beam resulting from a single-element top-hat radial profile (lower inset on the right), which has an error comparable to a  $\tau = 0.0041$ , which for the grid size used ( $128 \times 128 \times 128$ ) corresponds to a Gaussian full width of approximately 1 element.

further investigation.

It is worth discussing very briefly the case of thicker hemispheres. Figure 4 shows results from the 3D GS algorithm with a  $k$ -space hemisphere with a Gaussian radial power profile of variable width. It can be seen that an increase in the thickness of the hemisphere leads to lower errors. The reason for this is that a thicker  $k$ -space hemisphere implies more non-zero  $k$ -space elements whose phase the algorithm can alter. Physically, a  $k$ -space hemisphere of non-zero thickness corresponds to the presence of a range of wavelengths<sup>||</sup> and therefore polychromatic light; the represented field is a snapshot of the light field at one particular instant. At this instant the plane-wave components of all colours have the relative phase given in the  $k$ -space representation; as different colours correspond to different frequencies, different colours change their relative phase immediately afterwards. The time evolution of the instantaneous field in polychromatic light can be controlled by an extended GS algorithm that incorporates a time dimension

<sup>||</sup> Note that the  $k$ -space components with  $k_r \neq k_0$  are not evanescent waves as they correspond to purely real values of  $k_z$ . By definition, all points in  $k$  space represent travelling waves. To represent evanescent waves, the Fourier transform has to be combined with a Laplace transform.



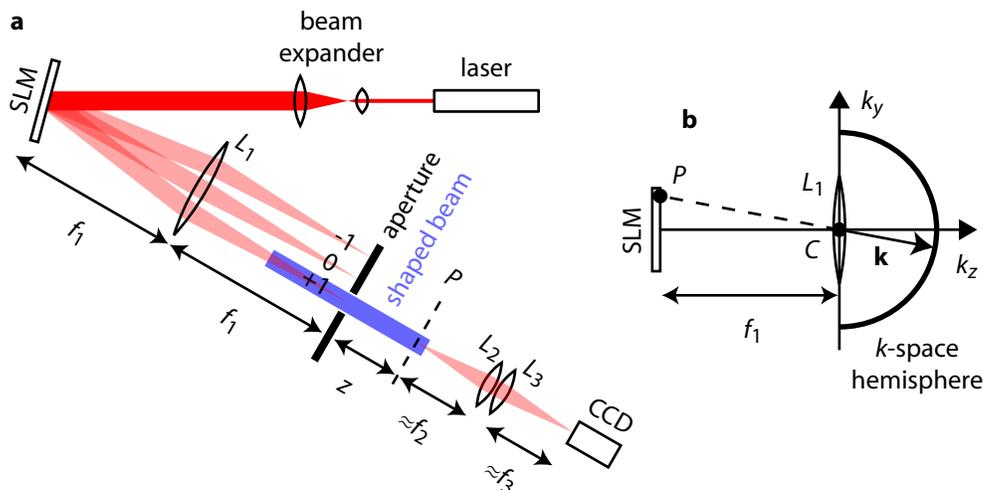
**Figure 5.** Geometry of the cone of light-ray directions and its effect on 3D light shaping. The cone of light angles is determined by the geometry of the setup, in the simplest case the aperture radius of the last lens in the system and its distance from the position where the light is to be shaped (a). The cone angle,  $\alpha$ , is also the angle of the  $k$ -space-sphere segment (b). The larger the cone angle, the smaller the error  $\epsilon$  between the desired and calculated intensity structure (c). The errors in c were calculated for the example of a shell-shaped target intensity, shown in the box in the top right corner. Also shown are the calculated intensity structures corresponding to some of the data points.

added to the real-space representation of the field and a frequency dimension added to the  $k$ -space representation. Experimental realisation requires that the relative phase between the plane-wave components of *all* colours can be controlled. This can be seen as an extension of the shaping of the time-resolved field of short pulses, which has previously been demonstrated experimentally [14].

#### 4.2. Influence of numerical aperture

Any experimenter realizing 3D light shaping at some point has to make an implicit or explicit choice about the range of directions of the plane-wave components that superpose in the target volume. This is a choice of the numerical aperture (NA) of the system and it depends on the solid angle spanned by the directions of light rays reaching the target volume. Usually the light-ray directions form a cone of angle  $\alpha$  (figure 5a), which in the simplest case is determined by the size of the aperture of the last optical component and the distance between that component and the space where the beam is to be shaped. As each  $k$ -space point corresponds to parallel light rays with a specific direction, which is given by the gradient of the phase, such a cone of light-ray directions restricts non-zero  $k$ -space values to the section of the  $k$ -space hemisphere that lies within an angle  $\alpha/2$  of the  $k_z$  axis (figure 5b).

Figure 5c demonstrates the effect of varying the angular size of the  $k$ -space-sphere segment on the resulting intensity distributions. It can be seen that a larger cone



**Figure 6.** (a) Schematic of the experimental set-up. A collimated HeNe-laser beam illuminates a phase-only spatial light modulator (SLM) [15] before passing through a Fourier lens ( $L_1$ ). The SLM displays a phase pattern that shapes the light beam in a volume around the SLM's Fourier plane. An additional blazed phase grating displayed on the SLM directs the shaped beam into the grating's +1st diffraction order; the other orders – caused by imperfections in the SLM's phase response – are filtered out by a Fourier-plane aperture. The volume in which the light beam is shaped is shown in blue. A CCD, with the help of imaging lenses  $L_2$  and  $L_3$ , records the intensity in a number of planes across this volume. In our experiment  $f_1 = 600\text{mm}$ ,  $f_2 = 1350\text{mm}$  and  $f_3 = 200\text{mm}$ . (b) Geometry of the correspondence between position on the SLM and on the  $k$ -space hemisphere. Each point light source  $P$  in the front focal plane of the Fourier lens  $L_1$  gives rise to a uniform plane wave whose  $\mathbf{k}$  vector is parallel to the line from  $P$  to  $C$ , the center of  $L_1$ .

angle allows the possibility of generating a 3D light intensity that resembles the target intensity more closely. The reason for this is that the larger range of values of  $k_x$ ,  $k_y$  and  $k_z$  in such a beam widens the range of structure sizes that can be present in the beam.

#### 4.3. Simulation parameters and procedure

Our particular implementation of the 3D GS algorithm, running on a dual 2.5 GHz Pentium Xeon desktop computer, takes several days to converge using a resolution of  $256 \times 256 \times 256$ . For this reason we initially test the generation of new shapes using a lower resolution of  $64 \times 64 \times 64$ . In addition to each iteration taking less time at this lower resolution, the algorithm requires fewer iterations to converge.

## 5. Experiment

We have created examples of light beams with a shaped 3D intensity structure experimentally, specifically some of the beams shown in figure 2. The experimental setup is outlined in figure 6a. It creates the shaped beam around the Fourier plane of

a phase hologram, which was calculated from the  $k$ -space representation of the beam found by the 3D GS algorithm (see section 5.1).

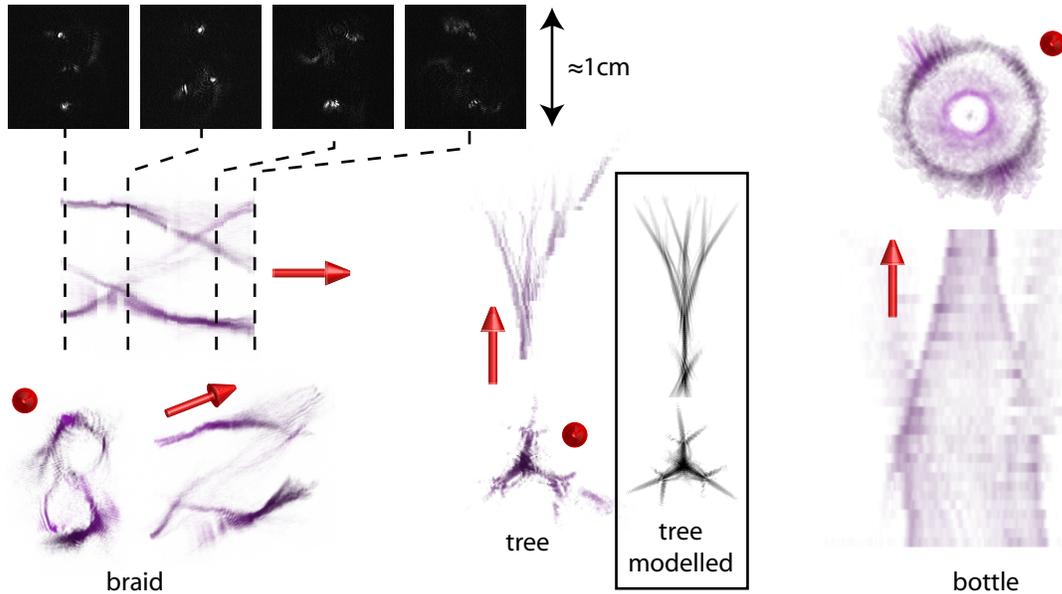
Our hologram patterns were calculated for beams whose  $k$ -space representations have a relatively large cone angle of  $\alpha = 90^\circ$ . With our phase hologram, which has a height of approximately 20mm, this cone angle could be realised by using a Fourier lens with a very short focal length of approximately 10mm; this would create the shaped beam in a volume of approximate size  $0.1\text{mm} \times 0.1\text{mm} \times 0.1\text{mm}$ . However, in our experiment we use a Fourier lens with a much longer focal length of  $f_1 = 600\text{mm}$ , which corresponds to a significantly smaller cone angle of  $\alpha \approx 2^\circ$ . Our setup can be seen as a combination of a  $f = 10\text{mm}$  Fourier lens that shapes the intensity in a cubic target volume of side length  $\approx 0.1\text{mm}$ , and two more lenses that image this target volume: a lens with  $f = -10\text{mm}$  in the same plane as the 10mm Fourier lens, and the 600mm Fourier lens we actually use. Because of the imaging characteristics of this lens pair, the image of the original target volume gets stretched to a size of approximately  $6\text{mm} \times 6\text{mm} \times 500\text{mm}$ .

In our experiment, an expanded beam from a HeNe laser was reflected off a computer-controlled phase hologram in the form of a phase-only spatial light modulator (SLM) [15]. To deal with imperfections in the SLM, a blazed diffraction grating was added to the phase-hologram pattern. This resulted in the desired beam travelling in the direction of the additional grating's first diffraction order; imperfections in the phase response of the SLM led to additional diffraction orders, which were filtered out by the aperture in the hologram's Fourier plane. We used a lens pair to image a plane at a variable distance  $z$  behind the aperture onto a CCD. The transverse magnification was slightly less than one. Intensity cross-sections corresponding to different planes taken across the shaped volume were later combined into volume data, which were visualized with bespoke 3D-viewer software based on the *VolPack* volume rendering library [16, 17].

### 5.1. Calculation of the phase-hologram pattern

The 3D GS algorithm finds phase values for the different points of a monochromatic beam's  $k$ -space hemisphere. Points on the  $k$ -space sphere correspond to infinite uniform plane waves; the direction of the  $k$  vector is the direction of the phase-front normal. Experimentally, a finite uniform plane wave can be created from a point light source in the front focal plane of a lens, whereby the position of the point light source determines the direction of the phase-front normal. Each point in a phase hologram that is placed in the front focal plane of a lens therefore corresponds to a point on the  $k$ -space hemisphere and controls its phase.

Figure 6b shows the geometry of the correspondence between points on the SLM and on the  $k$ -space hemisphere. In order to calculate the exact position of the point on the SLM that controls a given point on the  $k$ -space hemisphere, the point on the hemisphere should be projected through the center of the lens into the SLM plane. However, we find that a simpler parallel projection as indicated in figure 1, which gives



**Figure 7.** Experimental results. Left: three different projections of the experimentally recorded 3D intensity distribution of one period of braided bright lines (purple = brightest, white = darkest). Like in figure 2, red arrows indicate the propagation direction of the light beam. The intensity was collected over a volume of size  $\approx 10\text{mm}$  in the transverse directions and  $\approx 590\text{mm}$  in the longitudinal direction. The top row shows four (out of 60) experimentally recorded intensity cross-sections, which were later combined in the computer into volume data. Centre: two projections of the 3D intensity distribution over a volume of approximate size  $10\text{mm} \times 10\text{mm} \times 510\text{mm}$  of an experimentally generated light beam in the shape of a tree, together with modelled results (boxed). Right: two projections of the shoulder region of a bottle-shaped beam (different from the one shown in figure 2) over a volume of approximate size  $10\text{mm} \times 10\text{mm} \times 320\text{mm}$ . Additional multimedia material includes Quicktime movies showing animated rotation of the 3D intensity distributions measured for the braid beam (1.2MByte), the tree beam (584KByte) and the bottle beam (1.3MByte).

a phase-hologram pattern that is flipped and distorted, works very well. The distortions are small for points representing plane waves travelling at small angles with respect to the  $z$  direction, as is the case in our experiment. We calculate the phase-hologram pattern as the phase of the sum over all  $k$ -space elements with the same values of  $k_x$  and  $k_y$ . In the limit of an infinitely thin  $k$ -space hemisphere, this parallel projection is equivalent to calculating the phase cross-section of the 2D Fourier transform of the shaped beam in the Fourier plane (i.e. at  $z = 0$ ).

## 5.2. Results

Figure 7 shows some of our experimental results. The experimental patterns do not exactly match those calculated by the 3D GS algorithm. We believe this is mostly due to residual astigmatism in the optical system, imperfections in the SLM and imperfect alignment of the intensity data from different planes to 3D volume data. In any case, the results clearly demonstrate that the experiment works.

## 6. Conclusions

We have used an algorithm first described in reference [10] to calculate physically realizable light beams with a shaped intensity distribution, and we have created examples of such light beams in an experiment using a phase hologram calculated from the beam's Fourier-space representation. We are currently considering the use of such light beams in optical tweezers and atom optics.

## Acknowledgments

We would like to thank Aidan Arnold, Jonathan Leach, Miles Padgett and Patrik Öhberg for useful discussions. GW is supported by an EPSRC studentship. JC is a Royal Society University Research Fellow.

## References

- [1] E. A. Sziklas and A. E. Siegman. Mode calculations in unstable resonators with flowing saturable gain. 2: Fast fourier transform method. *Appl. Opt.*, 14:1874–1889, 1975.
- [2] 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.
- [3] Rafael Piestun, Boris Spektor, and Joseph Shamir. Unconventional light distributions in three-dimensional domains. *J. Mod. Opt.*, 43:1495–1507, 1996.
- [4] Z. Bouchal and J. Kyvalský. Controllable 3d spatial localization of light fields synthesized by non-diffracting modes. *J. Mod. Opt.*, 51:157–176, 2004.
- [5] E. Abramochkin and V. Volostnikov. Generation of spiral-type laser beams. *Opt. Commun.*, 141:59–64, 1997.
- [6] Rafael Piestun and Joseph Shamir. Control of wave-front propagation with diffractive elements. *Opt. Lett.*, 19:771–773, 1994.
- [7] J. Liesener, M. Reicherter, T. Haist, and H. J. Tiziani. Multi-functional optical tweezers using computer-generated holograms. *Opt. Commun.*, 185:77–82, 2000.
- [8] J. E. Curtis, B. A. Koss, and D. G. Grier. Dynamic holographic optical tweezers. *Opt. Commun.*, 207:169–175, 2002.
- [9] M. A. Seldowitz, J. P. Allebach, and D. W. Sweeney. Synthesis of digital holograms by direct binary search. *Appl. Opt.*, 26:2788–2798, 1987.
- [10] Gal Shabtay. Three-dimensional beam forming and ewald's surfaces. *Opt. Commun.*, 226:33–37, 2003.
- [11] V. Soifer, V. Kotlyar, and L. Doskolovich. *Iterative Methods for Diffractive Optical Elements Computation*. Taylor & Francis Ltd, London, 1997.
- [12] V. Soifer, V. Kotlyar, and L. Doskolovich. *Iterative Methods for Diffractive Optical Elements Computation*, page 14. Taylor & Francis Ltd, London, 1997.
- [13] R. L. Higdon. Absorbing boundary condition for difference approximations to the multi-dimensional wave equation. *Math. Comput.*, 47:437–459, 1986.
- [14] Joshua C. Vaughan, T. Feurer, and Keith A. Nelson. Automated spatiotemporal diffraction of ultrashort laser pulses. *Opt. Lett.*, 28:2408–2410, 2003.
- [15] LC-R 2500. HOLOEYE Photonics AG, www.holoeye.com, 2004.
- [16] Philippe Lacroute and Marc Levoy. Fast volume rendering using a shear-warp factorization of the viewing transformation. In *Proc. SIGGRAPH '94*, pages 451–458, 1994.
- [17] Volpack. <http://graphics.stanford.edu/software/volpack/>.