Inverse Techniques: Problems in Optics and Gas Sensing

Laura C. Thomson

BSc Hons

Submitted in fulfilment of the requirements

for the Degree of PhD

Department of Physics

Faculty of Physical Sciences

University of Glasgow

March 1, 2010

Abstract

In this thesis, two, seemingly different, classes of problems are discussed: locating gas sources from downwind gas concentration measurements and designing diffractive optics (i.e. computer generated holograms), which on illumination will produce a desired light beam in the far field. The similarity between these problems is that they are both "inverse problems" and we discuss the use of inverse techniques to solve them.

In many instances within science, it is possible to calculate accurately a set of consequences which result from defined events. In most cases, however, it is mathematically impossible to analytically calculate the unique set of events which led to the observed consequences. Such problems are termed "inverse problems". Taking the example of gas dispersion, one sees that a known source leads to a calculable set of downwind concentrations. However, given a single concentration measurement it is impossible to distinguish a specific source and location from a larger, more distant source that would have given the same measured concentration. This is an example of the same consequence resulting from two, or more, different events.

Key to solving inverse problems are iterative algorithms which randomly trial different possible events to find those which best describe the observed consequences.

Such algorithms use a search method to postulate possible events, apply a forward model to calculate the anticipated consequences and then use a cost function to compare the postulated with the known consequences. The process is iterated until the optimum value of the cost function is found, at which point the current set of postulated events are taken to be the best estimate of the real events. In this thesis I apply similar iterative algorithms to solve the two classes of problem.

The current demand on the world's oil resources have encouraged the development of new prospecting techniques. LightTouch is one such solution which is discussed in this thesis and was developed with Shell Global Solutions. LightTouch uses the fact that oil reserves, through microseepages, leak hydrocarbons to their surface. Detection of these hydrocarbons can indicate the presence of oil reserves. LightTouch measures Ethane to sub-part-per-billion sensitivity at multiple positions across a survey area. Locating the source of the Ethane from the sparse downwind concentration measurements is an inverse problem and we deploy algorithms of the type discussed above to locate the Ethane sources. The algorithm is written in LabView and the software, Recon, is currently used by Shell Global Solutions to solve this problem. In appendix B the Recon user interface is shown.

We investigate both the impact of choice of cost function (chapter 3) and forward model (chapter 4), which in this inverse problem is a gas dispersion model, on the algorithm's ability to locate the gas sources. We find that the choice of cost function is more important to the success of the algorithm than the choice of forward model.

Optical tweezers trap and manipulate particles with light beams. In order to manipulate the particles in a desired way it is necessary for the shape and position

of the light beam to be controlled. One way of achieving a desired light beam is to use a spatial light modulator (SLM) which displays a phase pattern (referred to as a computer generated hologram), off which the light is diffracted. Calculating the phase pattern which will result in the desired light beam is an inverse problem and is referred to as holographic light shaping. The forward model in this case is a Fourier transform. In this thesis we use an algorithm similar to that used to solve the gas location problem and the Gerchberg-Saxton algorithm to calculate phase patterns with applications in optical tweezers.

Within an optical tweezers system the highest trap resolution (the smallest distance between neighboring traps) that can be achieved is conventionally dictated by the diffraction limit. In this thesis we investigate two possible ways of beating the diffraction limit: superresolution and evanescent waves.

In chapter 5 we investigate the application of inverse techniques to calculating phase patterns which produce superresolution optical traps. We calculate theoretically the improvements to both relative trap stiffness and trap resolution using the superresolution optical traps. Although both are improved it comes at a cost to trap strength. In chapter 7 we simulate evanescent wave fields and demonstrate shaping three dimensional evanescent optical traps. Similar light shaping techniques are used in chapter 6 to shape light beams which after being disturbed will self-reconstruct.

Papers

1. *Momentum paradox in a vortex core*, Roberta Zambrini, Laura C. Thomson, Stephen M. Barnett, and Miles Padgett, Journal of Modern Optics 52 (2005), 1135-1144.

2. *Interactive approach to optical tweezers control*, Jonathan Leach, Kurt Wulf, Gavin Sinclair, Pamela Jordan, Johannes Courtial, Laura Thomson, Graham Gibson, Kayode Karunwi, Jon Cooper, Zsolt John Laczik, and Miles Padgett, Applied Optics 45 (2006), 897-903.

3. *An improved algorithm for locating a gas source using inverse methods*, Laura C. Thomson, Bill Hirst, Graham Gibson, Steve Gillespie, Philip Jonathan, Kenneth D. Skeldon, and Miles J. Padgett, Atmospheric Environment 41 (2006), 1128-1134.

4. *Simulation of superresolution holography for optical tweezers*, Laura C. Thomson, Yannick Boissel, Graeme Whyte, Eric Yao, and Johannes Courtial, in New Journal of Physics , (2008), 10, 023015

5. *Holographic shaping of generalized self-reconstructing light beams*, Laura C. Thomson and Johannes Courtial, in Optics Communications, (2008), 281,

1217-1221

6. *Simulated holographic three-dimensional intensity shaping of evanescent-wave fields*, Laura C. Thomson, Graeme Whyte, Michael Mazilu, and Johannes Courtial, in Journal of the Optical Society of America B, (2008), 25, 849-853

Author's Declaration

I hereby declare that this thesis is the result of my own work, except where explicit reference is made to the work of others, and has not been presented in any previous application for a degree at this or any other institution.

. . . .

Acknowledgements

I would like to take this opportunity to express sincere gratitude to my supervisors Johannes Courtial, Bill Hirst and Miles Padgett for all of the technical expertise, ideas, time and patience they have shared with me. Without their input I would not be capable of producing this thesis.

I have been fortunate enough to benefit from working with a variety of individuals from the Optics Group, Shell Global Solutions and Surrey University and would like to thank them for the support and help they provided me with.

Thank you to my family; Mum, Dad, Louise, Gran and Simon. Without your encouragement I would not have been inspired to complete this thesis.

Contents

Chapter 1

Inverse problems

1.1 Introduction

Most physical problems can be described in terms of events and consequences. Given the occurrence of an event we can calculate the resulting consequences using some known function f,

$$\text{consequence} = f(\text{event}). \tag{1.1}$$

This relationship is termed a forward problem and is generally well understood. However, in many problems in science we do not always necessarily wish to calculate consequences from events rather calculate events that caused some measured consequences. This could be achieved using an inverse relationship,

$$\text{event} = f^{-1}(\text{consequence}). \tag{1.2}$$

Unfortunately for many problems the inverse relationship, unlike the forward problem, is not well understood and cannot be solved directly. Such a problem is termed an inverse problem. Essentially an inverse problem is one in which, given

some known event, we can easily calculate the resulting consequences, however, for some physical reason given some measured consequences we cannot directly calculate the events that resulted in these consequences.

Inverse problems are solved by finding the events that result in consequences closest to those that have actually been observed. Algorithms that solve inverse problems are commonly divided into three components: search algorithm, forward model and cost function. Each component is necessary to solve the problem.

The reason that the forward relationship can be solved but the inverse relationship cannot is dependent on the problem. Reasons for this in the physical sciences range from problems which are fundamentally difficult to solve for mathematical reasons to those where the forward relationship has been studied previously and more rigorously so is just better understood and it is simply more efficient to solve using inverse techniques.

Inverse problems can be divided into three types: backward or retrospective problems, coefficient inverse problems and boundary inverse problems [1]. Coefficient inverse problems are concerned with estimating parameters in an equation. Boundary inverse problems aim to find certain boundary conditions related to a function. Finally, backward problems aim to find the initial conditions of a system. It is coefficient inverse problems that we are mainly concerned with and solve in the coming chapters.

In the remainder of this chapter I will summarise two of the inverse problems I have been working on. These problems are discussed in greater detail in the coming chapters.

1.2 Locating gas sources from down wind concentration measurements

Locating gas sources from down-wind gas-concentration measurements is an example of an inverse problem. Many gas-dispersion models have been developed that aim to predict gas concentrations resulting from a known source dispersing into the atmosphere. Therefore given a known gas source and knowledge of the wind field we can easily calculate the resulting down wind gas concentration using a dispersion model. This is the forward model in this problem. However, many applications require the inverse of this, locating gas sources from down-wind gas-concentration measurements, to be solved. The inverse is ambiguous: a small source close to the measurement location can give the same concentration as a large source further away. It is this ambiguity that means that locating gas sources from known down-wind concentration measurements is an inverse problem.

By making multiple measurements at various locations under different wind conditions and applying inverse dispersion techniques, unknown gas sources can be determined from downwind concentration measurements and knowledge of the wind field. Such an approach has been applied to estimating and locating gas sources within agricultural settings [2] and land-fills [3]. We apply a similar approach to locating oil reserves [4, 5].

This is a relevant and important problem. The current demand on the world's resources of oil and the increasing difficulty in locating new oil and gas reserves has sparked the development of many novel oil prospecting techniques. Many of

these techniques aim to survey a large area to find areas worthy of a more detailed survey. One such technology is LightTouch [4], which aims to locate oil reserves from downwind gas concentration measurements. This is achieved by solving an inverse problem.

It is well known that through microseepages oil reserves leak hydrocarbons to the surface [6, 7]. We can use the presence of hydrocarbons in the atmosphere as an indication that an oil reserve exists in the vicinity of the measurement. An obvious approach to locating oil reserves in this way would be to make multiple measurements over the area, and where the largest concentration was measured would be considered to be the location of the reserve. Unfortunately, given that we make measurements from the back of a car, that each measurement takes hours, and the survey area can be of the order of 20km $\times$ 20km, for such a large area it could take months to survey. Instead we make multiple measurements of gas concentrations at various locations along with measurements of the wind field and solve the problem as an inverse problem to locate the unknown gas source. The basic algorithm we apply is described in chapter 2, and in chapters 3 and 4 we discuss how we optimise the algorithm.

In chapter 2 we describe the algorithm that we have developed to locate gas sources from down wind concentration measurements. This algorithm is currently used by Shell Global Solutions for this application.

The basic structure of this algorithm was already implemented when I started my PhD. My role was to add the cost functions, the annealing methods used and take part in discussions about the direction of this work.

In chapter 3 we apply the algorithm described in chapter 2 and show that we can locate a known gas source in a desert environment. The known source is a cylinder release of Ethane. We compare the use of three different cost functions in the algorithm. The cost function is a measure of the quality of the gas source proposed by the search algorithm and essentially determines the optimum source. All three of our cost functions locate the known gas source. For our application in this environment it is very difficult to fully understand the error in the measurements. For this reason we investigate whether adding Gaussian noise and concentration offsets to our data set effects the performance of our algorithm. For our application we show that the most successful cost function in response to the addition of noise is based on minimising the total emissions of the proposed source.

The work described in chapter 3 was done in collaboration with the following people: Graham Gibson, Steve Gillespie and kenneth D. Skeldon who gathered the data used in this chapter; Bill Hirst, Philip Jonathan and Miles J. Padgett provided useful assistance in the analysis techniques and cost functions used. My own role was to develop the software used, develop the algorithm, take part in discussions about the work and produce the results.

In chapter 4 we use simple optimisation techniques to try to improve the Gaussian dispersion model used in our algorithm. We use a data set where gas concentration measurements were made downwind of a known cylinder release. We apply multiplicative correction factors to the Gaussian plume model and optimise these such that given the known source the calculated concentrations are closer to those actually measured. We use this optimised Gaussian plume model in our algorithm to

investigate whether this results in an improved location of the known source. We show that in actual fact the Gaussian plume and optimised Gaussian plume model yield similar results. Again it is the cost function that has greatest bearing on the results.

The work described in chapter 4 was done in collaboration with the following people: Claire Patterson, Kenneth D. Skeldon and Graham Gibson collected the data analysed, Miles J. Padgett provided supervision of the work. My role in this work was to write and develop the optimisation routine used and carry out the analysis.

1.3 Light shaping as an inverse problem

Optical tweezers use foci of light to trap, move and manipulate microscopic dielectric particles. If the refractive index of the particle being trapped is greater than that of the surrounding medium the particle will be attracted to, and held in, the beam focus. In order to manipulate and move the particle in a controlled manner the beam foci must be manipulated in a corresponding fashion.

Optical tweezers are just one example of an application where an optical beam of a desired shape is required. The shape of light beams can be manipulated using any one of a number of optical elements: prisms and lenses for example. These elements obviously limit the shape of the beam that can be achieved and also the way in which it can be manipulated. Another less limiting way in which a beam can be manipulated is the application of holographic phase masks. In most modern light

shaping systems the phase mask is applied to a beam using a spatial light modulator (SLM). This is a device that contains an array of liquid crystals, each of which is addressable under computer control. The phase pattern applied to the SLM can be changed interactively and hence used not only to shape a desired beam but also change the shape of the beam in almost real time. However, in order to achieve this we must calculate the hologram that will result in the desired beam.

In most holographic light shaping set-ups the plane that contains the shaped light lies in the focal plane of a Fourier lens. Therefore given a known phase hologram and hologram illumination we can calculate the beam in the Fourier plane using a Fourier transform. In this particular problem this is the forward problem, however we wish to solve the inverse of this. It could be considered that calculating the hologram required to give a desired beam in the Fourier plane could be achieved by inverse Fourier transforming the desired beam. However, this would normally result in both a calculation of a phase hologram and the hologram illumination. Unfortunately, there is a constraint on the calculation: we specify the hologram illumination and it is only the phase hologram we need calculate. Therefore the phase hologram calculated by inverse Fourier transform would not result in the desired beam when illuminated by the specified hologram illumination. This is the physical reason that we must solve this problem as an inverse problem.

Various algorithms can be applied to solve this inverse problem. One example of an algorithm used to calculate the phase hologram that can be applied to produce a desired intensity distribution is the Gerchberg-Saxton algorithm [8]. We apply both the direct search and Gerchberg-Saxton algorithm in chapters 5, 6 and 7 to solve

light shaping problems.

Algorithms used to solve inverse problems are, like the Gerchberg-Saxton and direct search algorithms, iterative. The direct search algorithm is the simplest and is based on a brute force trial and error principle. Initially a random phase hologram is generated and combined with the set hologram illumination to give the hologram field. This field is then Fourier transformed to give the resulting field in the Fourier plane. The desired beam is compared to the resulting beam numerically using a cost function. This corresponds to an iteration. In all subsequent iterations a change is made to a single pixel in the phase hologram. The change is retained if the resulting value of the cost function is less than the value calculated during the previous successful iteration. We also utilise a version of this algorithm to solve the problem of locating a gas source from downwind gas concentration measurements, this version of the algorithm is described in chapter 2.

In chapter 5 we show, using computer simulations, that we can improve the optical trapping characteristics of foci using superresolution algorithms. We use both a superresolution direct search and Gerchberg algorithm to improve both trap resolution and relative trap stiffness.

The work described in chapter 5 was done in collaboration with the following people: Yannick Boissel coded the Gerchberg superresolution model; Eric Yao and Graeme Whyte provided useful ideas on the work; Johannes Courtial had the idea for the work and carried out some of the analysis. My own role was to develop the direct search superresolution model and calculate and compare the results from the algorithms.

In chapter 6 we discuss an algorithm that we have developed which can be used to arbitrarily shape self-reconstructing light beams. We show examples of beams shaped using the algorithm. Self-reconstructing light beams are any light beam which after being disturbed by an object, such as a particle in an optical trap, will reconstruct itself on further propagation.

The work described in chapter 6 was done in collaboration with Johannes Courtial who had the original idea for the work and provided useful supervision. My role in this work was to develop the self-reconstructing model and produce the results.

In chapter 7 we demonstrate simulating 3D intensity shaping of evanescent wave fields. Our approach to this is to use light shaping algorithms, we use a version of the direct search algorithm, to shape the evanescent wave fields.

The work described in chapter 7 was done in collaboration with Michael Mazilu and Johannes Courtial. The theoretical work and the analysis was carried out by Michael Mazilu and Johannes Courtial. My own role was to write the model, take part in discussions about the work and produce some of the results.

Chapter 2

An algorithm to locate gas sources from downwind concentration measurements

2.1 Introduction

We have described in chapter 1 that locating gas sources from downwind gas concentration measurements is an inverse problem. We aim to solve this problem to locate oil reserves. Oil reserves leak hydrocarbons to their surface, we measure ethane as an indication of the presence of an oil reserve. The measurements of ethane concentration are made using a highly sensitive atmospheric ethane sensor based on laser diode absorption spectroscopy described fully by [9]. Along with the gas measurements our algorithm requires knowledge of the wind field at the time of measurement, we measure the wind field using a 3D ultra-sonic anemometer. In this chapter we describe the algorithm that uses these measurements to locate the unknown gas source.

In all inverse problems the aim is to infer the unknown state from measured consequences of that state. In the case of gas dispersion, the unknown state is the

gas source distribution of strengths and locations; and the measured consequences are the gas concentrations for the associated wind conditions and measurement locations. Our aim is to find the source distribution that will generate predicted concentrations closest to those actually measured. This requires three components to the algorithm: a search algorithm, forward model and cost function. The forward model is unique to the problem and it is this that we will describe first.

The algorithm described in this chapter was implemented by us in software written in Labview titled Recon. This software was used to produce the results discussed in chapters 3 and 4 and is currently used by Shell Global Solutions to locate gas sources in the field.

The approach to oil and gas prospecting we describe in this thesis is targeted at frontier exploration - where large areas must be screened prior to more detailed investigation. Other examples of such novel techniques include searching satellite and airborne images for subtle signs of oil on the sea surface [10] and stimulation of florescence in these oil films by airborne lasers [11]. Currently more detailed oil and gas exploration is carried out by seismic reflection [12]. Using this technology properties of the earth's subsurface, such as whether the survey area contains geological structures which could support hydrocarbons, can be estimated by the analysis of reflected seismic waves.

2.2 The forward model

Gas dispersion is a complicated phenomenon with numerous physical processes active at different length scales [13], in most experimental situations access to wind field data is limited, typically comprising a very small number of measurement locations. In common with much earlier work, [4] and [14], we employ a simple Gaussian plume dispersion model which can be implemented using a single location wind speed and direction measurement. Within our numerical model we divide the survey area (in our trials in the Middle East the survey area was of km dimension) into a grid, with each cell, j, containing a central diffuse source of unknown strength S_j. Each of these M sources gives rise to a downwind gas concentration approximated by a Gaussian plume model [14], see figure 2.1, and the predicted concentration is the sum over all the gas sources in the grid of cells. For the i^{th} measurement of mass concentration (units of kg/m^3) we have,

$$C_i = \sum_{j=1}^{M} \frac{S_j}{\pi V_i \sigma_{w_i} \sigma_{h_i}} \exp\left(-\frac{1}{2}\left(\frac{\Delta_{w_i}}{\sigma_{w_i}}\right)^2\right) \exp\left(-\frac{1}{2}\left(\frac{\Delta_{h_i}}{\sigma_{h_i}}\right)^2\right) \tag{2.1}$$

where V_i is the wind velocity (m/s) at the time and location of the i^{th} measurement, Δ_{w_i} and Δ_{h_i} (m) are the offset from the plume centre, σ_{w_i} is the width of the Gaussian plume and σ_{h_i} is the height in the plume. Both σ_{w_i} and σ_{h_i} are length measurements and are calculated using the angular standard deviations of the wind direction in both the horizontal, θ_{w_i}, and vertical, θ_{h_i}, directions respectively,

$$\sigma_{w_i} = \theta_{w_i} z + k \tag{2.2}$$

$$\sigma_{h_i} = \theta_{h_i}(z + k). \tag{2.3}$$

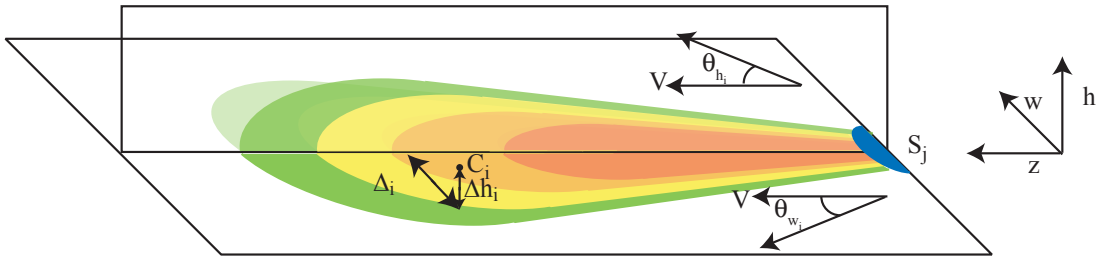


Figure 2.1: The Gaussian Plume Model. The source is shown by the blue semi-circle and the gas concentration it disperses is shown by the coloured plume. The parameters used in the model are also shown.

Where z is the distance from the source position, j, to the measurement location, i, and we have approximated the near field concentration effects of diffuse sources by adding a constant, k, to these plume width terms, where k is in units of cell size.

2.3 The search algorithm

There are many search algorithms that could be used to identify the combination of source strengths and locations that best predict the measured concentrations. We employ an iterative direct search algorithm [15] to postulate different source distributions. Within each iteration, a new postulated source distribution is generated by assigning a random source strength to a randomly selected cell, j. The predicted concentrations arising from the new set of postulated sources are then calculated using the Gaussian plume model. The new postulated source distribution is numerically assessed by evaluating the chosen cost function, E_n, giving a single numerical

value reflecting the closeness of the predicted concentrations to those actually measured. Typically, E_n would be defined such that a closer match between observed and predicted concentrations gives a lower value of E_n. The new postulated source value, S_j, is accepted or rejected depending on whether the corresponding value of E_n is reduced or not. Repeated iterations minimize the value of E_n and result in a source distribution that best accounts for the concentrations measured. χ^2 is an obvious candidate cost function, based solely on the predicted, C_{e_i}, and measured, C_{m_i}, concentration;

$$E_n = \frac{\chi^2}{N} = \frac{1}{N}\sum_{i=1}^{N}\frac{|C_{m_i} - C_{e_i}|^2}{\sigma_i^2}. \tag{2.4}$$

We note that an accurate determination of χ^2/N requires a correspondingly accurate knowledge of the error, σ_i^2, which encompasses both the noise associated with the measurement technique and errors in the forward model. In chapter 3 we will discuss how to estimate σ_i.

2.4 Simulated annealing

If we calculated the cost function over the full range of M source strengths, we would obtain an M dimensional landscape where each axis represented the range of emissions from the source in a particular cell. Ideally, the resulting cost function landscape would contain a clearly identified global minimum, corresponding to the optimum distribution of M source strengths and locations. However, typically, this global minimum will be surrounded by local minima, to which the algorithm will converge to. Figure 2.2 is a representation of a typical cost function landscape. We

avoid converging on a local minimum by refining the search algorithm to occasionally accept postulates that increase E_n, thereby allowing escape from a local minimum. The technique is referred to as simulated annealing [16] and was developed by analogy with thermodynamic cooling, [17]. The criterion for accepting an increase in the cost function is based by thermodynamic analogy on the Boltzman probability distribution,

$$R(0,1) < \exp\left(-\frac{E_n - E_{n-1}}{T_n}\right) \tag{2.5}$$

where $R(0,1)$ is a random number between zero and one, E_n is the value of the cost function for the new postulate, E_{n-1} is the value of the cost function at the last accepted change and T_n is the temperature or cooling parameter, which is reduced after each iteration. If $E_n < E_{n-1}$ that change is always accepted, however if $E_n > E_{n-1}$ the change is sometimes accepted dependent on whether $\exp(E_n - E_{n-1}/T)$ is greater than the random number, R. Whether or not this holds true depends on the value of T. The value of T is reduced after each iteration according to,

$$T_n = T_{n-1}(1 - \varepsilon). \tag{2.6}$$

The initial iterations, corresponding to high values of T, accept most changes whether E_n is improved or not. As T is reduced fewer changes are accepted. Since the parameter T is described as a temperature it is usual to describe the rate of cooling in terms of thermal mass. It is the parameter ε, which has a value between 0 and 1, that determines the rate of cooling. For our reconstructions of source strengths on a 16×16 grid we typically run upwards of 4 million iterations during which time the annealing parameter, T, is progressively reduced to zero.

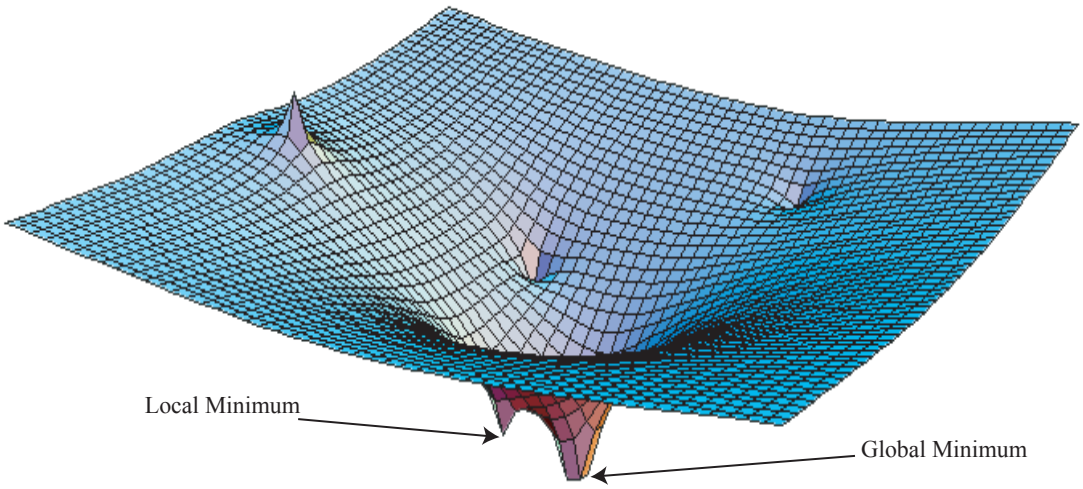


Figure 2.2: A representation of a cost function landscape. The global and a local minimum are highlighted by the arrows.

Noise annealing is another annealing method, that operates on the principle that by persistently adding different random noise distributions to the measured concentrations C_m the algorithm will consequently be displaced and knocked out of any local minima. Like simulated annealing we believe that any source distribution the algorithm persistently converges towards despite the constant addition of added noise it does so because it is the global minimum.

We add a random noise distribution, K, of standard deviation, σ_K, to C_m. After a particular number of iterations, $n = A$, K is recalculated with a reduced standard deviation,

$$\sigma_{K,n+A} = \sigma_{K,n=0}(1 - \varepsilon) \tag{2.7}$$

and again is added to C_m. By both methods of annealing we have increased confidence that the algorithm will converge to a global minimum.

2.5 Conclusions

We have described in this chapter the basic algorithm that we apply to locate gas sources from down-wind gas concentration measurements. This algorithms was implemented in LabView and the software is currently used by Shell Global Solutions.

We have described above a simple χ^2/N cost function. Such a function in most cases will result in an unrealistic reconstructed source distribution. In chapter 3 we apply the algorithm and discuss why χ^2/N results in unrealistic source distribution. We also demonstrate the cost functions we apply in practice and show the results of our algorithm using each cost function. Each cost function, although locating the source, results in a different looking source distribution.

The algorithm described above is also used in chapter 4. Having optimised the Gaussian plume model we compare the results of using this model and the non-optimised Gaussian plume model in the algorithm.

Chapter 3

Comparison of cost functions used to locate gas sources from downwind concentration measurements

Locating gas sources from downwind gas concentration measurements is an inverse problem. Having made measurements of both gas concentration and wind data we can apply our algorithm, described in chapter 2, to solve this problem. We use a random search algorithm with simulated annealing to generate candidate source distributions. These are assessed numerically using a cost function. It is the cost function that greatly influences the resulting optimum source distribution. In this chapter we investigate the affect of three different cost functions on the converged solution. The work described here was published in reference [5].

3.1 Introduction

In chapter 2 we have discussed the algorithm we use to locates gas sources from down-wind concentration measurements. The algorithm consists of three components, a search algorithm, forward model and a cost function. The search algorithm postulates a source distribution, the forward model calculates the resulting concentrations and the cost function numerically assesses the postulated source distribution by quantifying the degree to which the postulated source distribution accounts for the measured gas concentrations. In this chapter we discuss three different cost functions with differing regularisation terms and present results from applying each. We assess the robustness of these and the differing regularisation terms by the progressive addition of random noise and systematic offsets to the concentration data. We show that for our application, the best reconstructions are obtained by using a multiplicative regularisation parameter defined to minimise the total gas emissions.

The reconstructions shown in this chapter were produced using data collected from a single calibrated gas source at a known location. This survey site was a flat desert location meaning that the wind field could be assumed consistent over the area of interest. The source comprised a continuously weighed cylinder of ethane with a regulated release rate of 5 Kg/hr. We used a two hundred meter diameter network of perforated pipes to distribute the gas to simulate a diffuse area source. The gas concentrations were measured at a height of 5m above the ground collocated with the anemometer at seven downwind locations. These locations were 2 to 5km downwind of the release in a North West direction. Gas concentration and wind measurements were made for a period of at least fifteen minutes, before moving to

the next location. The whole measurement sequence was completed over four hours from noon i.e. during the most stable wind conditions. Within our algorithm we divide the 8km x 8km survey area into a grid of 16 x 16.

3.2 Cost functions

To evaluate how accurately we have estimated the source distribution we use a cost function , E_n. An obvious choice of cost function would be χ^2,

$$E_n = \frac{\chi^2}{N} = \frac{1}{N}\sum_{i=1}^{N}\frac{|C_{m_i} - C_{e_i}|^2}{\sigma_i^2}. \tag{3.1}$$

The χ^2 cost function described in equation 3.1 is based solely on the measured data and although this is a useful indicator of the closeness of fit of the measured and estimated concentrations a minimisation of this function generally leads to an unrealistic distribution of sources. By simply minimising equation 3.1 any noise in the data has the ability to dominate the minimisation of the function which can lead to an unrealistic distribution. By simply minimising χ^2 we often see a halo effect. Where sources are placed around the outskirts of our survey area. It is possible that the correct sized source placed far enough away can account for any of the measurements recorded by fitting the data alone we see this effect. The map in figure 3.1 shows the result of our algorithm (the calculation of the source distribution) used with a χ^2 minimisation, the data set used contains concentration measurements downwind of a known source. In this figure colour indicates source strength, the scale is shown on the figure. Although the cylinder is approximately located this minimisation gives rise to strong sources at positions distant from the source location which in this case

is simply wrong. We can note at this point that although we do not show the results in this thesis we have experimented with using our algorithm to successfully locate multiple known sources. The data set in this case was simulated.

In order to avoid encountering this problem it is usual to add another term to the merit function,

$$E = \frac{\chi^2}{N} + \lambda f(S). \tag{3.2}$$

This additional term $f(S)$ is called a regularisation term which is a function of the source distribution. The term $f(S)$ is generally based on some knowledge of what we expect the source distribution to look like before we begin the reconstruction. This type of knowledge is termed *a priori*. The parameter λ is a weighting factor that dictates the respective influence that both the concentration data fitting χ^2/N and the prior source knowledge term $f(S)$ will have on the reconstruction.

The regularisation term $f(S)$ is chosen to favour a source distribution that corresponds to physically realisable source values and behaviours. $f(S)$ can be chosen for a range of reasons. Some may be specific to the particular inverse problem that is being solved and some may be based on a statistical belief. One obvious choice of $f(S)$ specific to this and many other problems would be to enforce positivity. We know that there does not exist some form of massive gas sink inhaling the gas that we are trying to measure which would numerically correspond to a negative source. This is in fact something we do not need to include in our cost function as it is already strictly enforced in the algorithm as when making random changes to the sources we only consider positive values as a possible replacement.

One regularisation term that we use is based on geophysical expectation; we ex-

pect that the microseepage process responsible for oil reserves leaking hydrocarbons into the atmosphere gives rise to a limited supply of gas. We therefore choose a cost function where we aim to reduce the magnitude of the postulated source distributions,

$$E(Flux) = \frac{\chi^2}{N}\lambda\sum_{j=1}^{M}(S_j + \frac{1}{\lambda}). \tag{3.3}$$

The form that $E(Flux)$ takes is different to that suggested in equation 3.2. The $E(Flux)$ cost function is multiplicative not additive like the others, however, the function contains the same parameters χ^2, λ and $f(s)$.

Another distinctly different and widely used cost function we have chosen to implement is based on minimising the entropy [17] of the source distribution,

$$E(Maxent) = \frac{\chi^2}{N} + \lambda\sum_{j=1}^{M} S_j \ln\left(\frac{S_j}{\sum_{j=1}^{M} S_j}\right) \tag{3.4}$$

The entropy of any distribution refers to its disorder. By minimising the entropy of the source distribution we are encouraging the sources to be of the same value meaning that any distinguishing feature that forms in the source distribution we take it to do so because it exists.

The third and final cost function that we have implemented encourages the source distribution to take a similar form to that of $E(Maxent)$. The regularisation parameter in this function

$$E(Smooth) = \frac{\chi^2}{N} + \lambda\sum_{j=1}^{M}\frac{(S_j - < S_j >_8)^2}{\alpha + (S_j - < S_j >_8)^2} \tag{3.5}$$

compares the source S_j in grid box j to the average source in the 8 grid boxes surrounding S_j this is denoted by $< S_j >_8$. The parameter α has a similar effect

on the function to that of λ but also prevents singularities in the cost function. Although both $E(Maxent)$ and $E(Smooth)$ encourage a smooth distribution of sources $E(Smooth)$ is sensitive to the interchange of one or more sources, this same sensitivity does not apply to $E(Maxent)$.

In all three cost functions discussed in equations 3.3, 3.4 and 3.5 we must define λ so that it sets the correct balance between χ^2/N and the regularisation term. It is discussed in [17] that λ should be chosen such that χ^2/N falls within a statistical range, $1\pm\sqrt{2/N}$. Obviously to achieve values of χ^2/N within the range it is necessary to have an accurate knowledge of the standard deviation σ. Unfortunately we lack understanding of the true value of σ. We can estimate the appropriate value of σ by simply running an initial χ^2/N reconstruction for a given data set and from this reconstruction calculating the standard deviation of the residuals, effectively normalising χ^2/N to unity. We can then use the value of the standard deviation of the residuals as the standard deviation σ used in the calculation of χ^2/N. The acquisition of σ in this way allows us to achieve values of χ^2/N within the appropriate statistical range $1 \pm \sqrt{2/N}$.

We must determine the optimum value of λ for each cost function; the value of λ that results in the reconstruction closest to the known source distribution. Figure 3.2 shows reconstructions using the known data set for each of the three cost functions discussed above for increasing values of λ. The value of λ is increased from 0 to a value that results in an increase of χ^2/N by a factor $1 + \sqrt{2/N}$.

When using the optimised λ values, all three cost functions approximately locate the cylinder release. However, the smoothing cost function results in the source

being, incorrectly, spread over adjacent cells; a result of the averaging function.

3.3 Reconstructions with added noise

We are able to test the robustness of each cost function by modifying our controlled release data to include Gaussian noise added to the measured concentrations. In figure 3.3 we see that even adding up to 1.2ppb of noise to the intrinsic sensor noise of 0.2ppb does not prevent reasonable reconstruction of the gas source. This significant level of noise immunity arises because multiple measurements under similar wind directions are taken at each measurement location. As might be expected, we see that using the smoothing regularization spreads the reconstructed source over the neighbouring grid cells. On visual inspection of figure 3.3 we can see that the Maxent and Flux Minimiser regularisation functions perform similarly with the latter finding the best overall reconstruction. The best result is achieved by adding 1.2ppb of noise and using the Flux cost function, see figure 3.3.

3.4 Reconstructions with concentration offset

We also want the reconstruction results to be robust to offsets in measured concentrations. This is particularly relevant because the exact background ethane concentration in the atmosphere is uncertain and subject to daily variation. In order to test robustness to offsets, we add a concentration offset to the data collected from the controlled release and calculate the reconstructions using each of the cost functions. Figure 3.4 shows reconstructions for each of the three cost functions with

progressively increasing offsets added to the concentration measurements. From the reconstructions shown in figure 3.4 we draw similar conclusions to those made in relation to the addition of Gaussian noise, namely the flux minimzer cost function is the most robust.

3.5 Conclusion

In this chapter we have shown that we can locate (and estimate the release rate of) a source of ethane dispersing in a desert environment by inverting the gas dispersion process. We have used the algorithm described in chapter 2 in conjunction with simulated annealing to evaluate the performance of three cost functions. Despite the addition of Gaussian noise or concentration offsets to the data our algorithm continues to locate the gas source with each of our three cost functions. For our application the cost function minimising the total source emissions is the most robust to the experimentally relevant effects of increased noise or concentration offsets in the measured data.

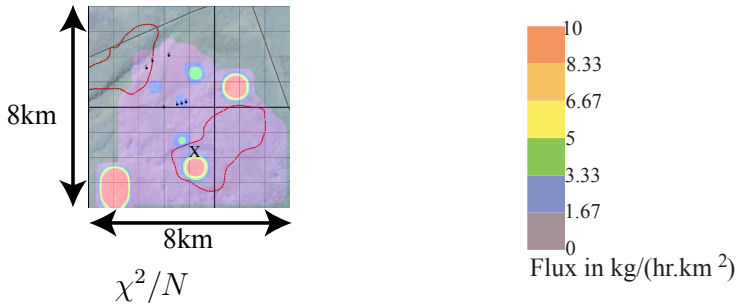


Figure 3.1: Result of the algorithm used with χ^2/N. The cylinder release area is shown with colour indicating source strength, with the scale shown. X marks the known location of the cylinder. It can been seen from the map that the cylinder is located by the algorithm, however the algorithm also places sources at other locations. This is a result of the χ^2 cost function.

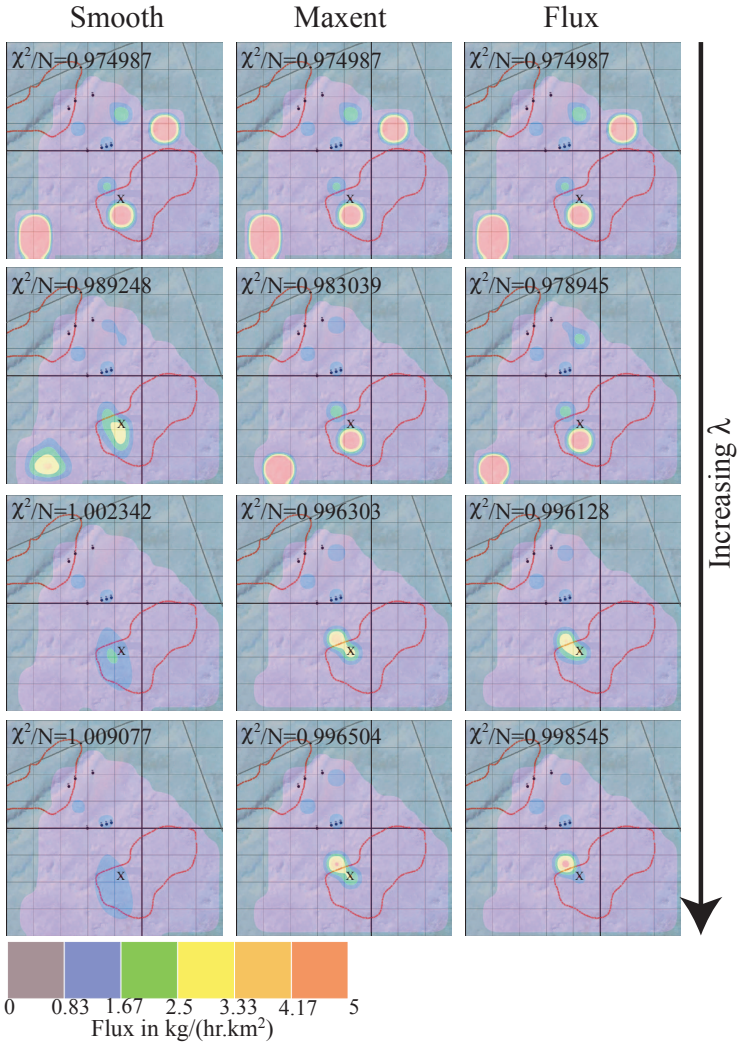


Figure 3.2: Reconstructions shown for each cost function for a range of regularisation, λ from $\lambda = 0$ (the χ^2/N solution) to a value which increases χ^2/N by a factor $1 + \sqrt{2/N}$. The location of the release is marked with an x. All three cost functions approximately locate the cylinder, reconstructions using the optimum value of λ for each cost function are shown in the third row of the figure.

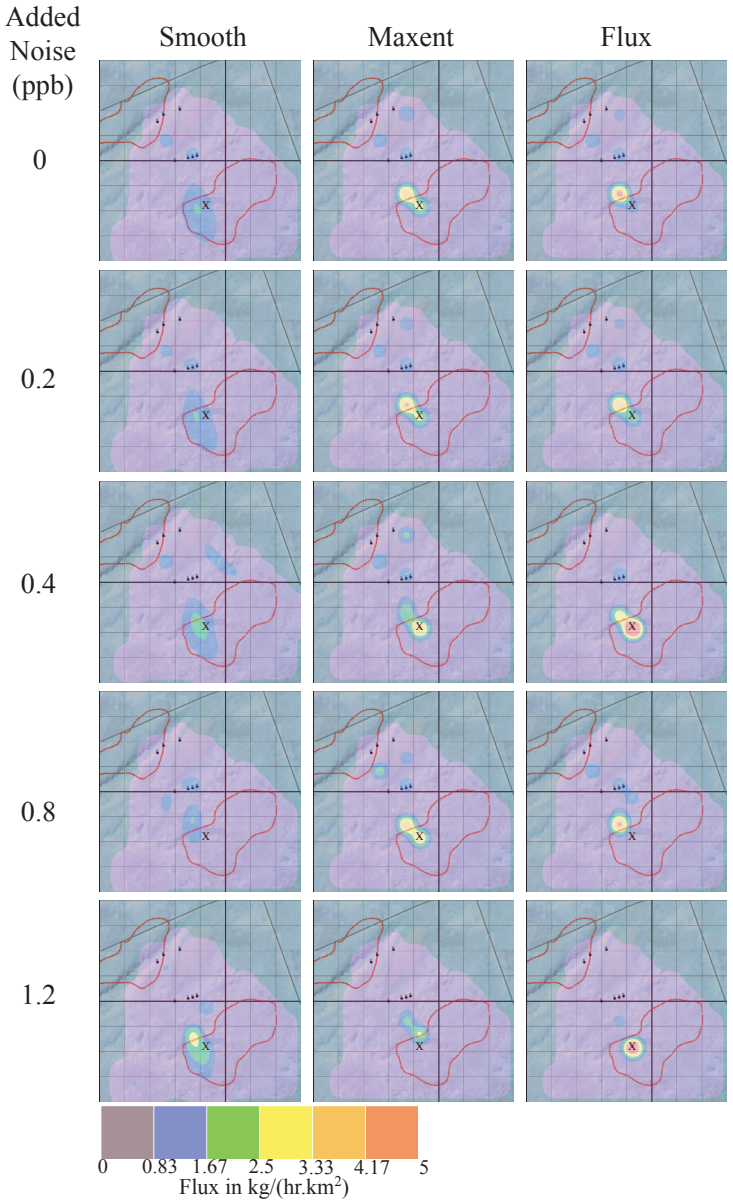


Figure 3.3: Reconstructions for each cost function with increasing added noise, colour indicates ethane flux at source. The first row shows the reconstructions based on the original data. Subsequent rows show reconstructions for each of the cost functions as increasing amounts of random noise are added to the concentration data. The location of the release is marked with an x. The reconstructions produced using the flux cost function are most robust against added noise.

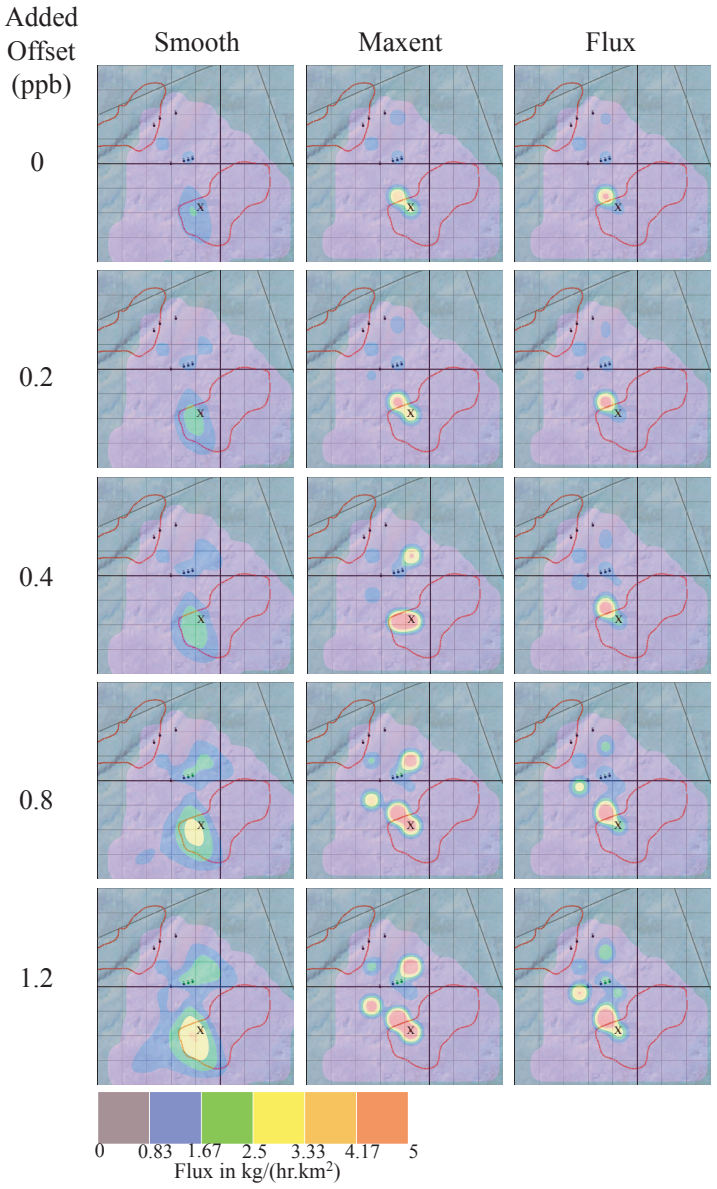


Figure 3.4: Reconstructions for each cost function with increasing added concentration offsets, colour indicates ethane flux at source. The first row shows the reconstructions based on the original data. Subsequent rows show reconstructions for each of the cost functions as increasing amounts of offset are added to the concentration data. The location of the release is marked with an x. From the reconstructions the flux cost function is most robust in response to concentration offset.

Chapter 4

Dispersion model optimisation

Many different gas dispersion models have been applied to calculate gas concentration downwind of a known source [18]. So that they can be applied in real field situations, many of these models are simplified to require measurement of only a small number of meteorological parameters. Their simplicity means that although they may provide a good estimate of the time averaged concentration they, by their very nature, have little predictive power for short term measurements. We investigate in this chapter whether a simple Gaussian plume dispersion model, like that described in chapter 2 and utilised in chapter 3, can be optimized to better describe the measurements within a particular field situation. The optimized model is then compared to a non optimized model: both are utilised in the algorithm described in chapter 2 and their resulting converged source distributions are compared for effectiveness at locating a known gas source. We find that although the model parameters may be set to give a better match to measured downwind concentrations, the overall performance of the algorithm is more dependent upon the extent of regularization rather then variations in dispersion model.

4.1 Introduction

Locating a gas source from a few measurements of down-wind gas concentration is an example of an inverse problem, this was previously discussed in chapter 1. High-speed and cheap computational power has meant that this and similar problems can be tackled numerically using various techniques to iteratively trial possible solutions, eventually finding a solution which adequately predicts the real measurements. In chapters 2 and 3 we discussed our algorithm which aims to solve such a problem by finding the distribution of gas sources which best describe a set of concentration measurements, such inverse techniques have been applied by [4] and [5].

In many inverse problems, the forward model is known precisely - like the Fourier transform used in the light shaping inverse problem described in chapter 1. However, for the location of gas sources a precise forward model would require full 3D knowledge of the wind field. In most field situations these measurements are not possible and data collection is restricted to a single point measurement of the concentration and wind speed and direction. Various simple gas dispersion models have been proposed using single-point, time-varying wind speed and direction, supplemented by additional local meteorological data to predict the shape of the down-wind gas plume [13, 19, 20]. Most of these models assume a gas plume, within which the gas concentration varies with a Gaussian cross section, centered on the average wind direction. We use a similar simple Gaussian plume model, described by equation 2.1 [14], within our algorithm and have shown in the previous chapter that it can be used to locate a gas source from downwind gas concentration measurements. However, from these and other observations, by ourselves and others, we know that like

most other gas dispersion models it does not accurately describe the concentration measurements made in the field. In this chapter we examine the extent to which optimizing the dispersion model for a set of measurements made downwind of a single known source can improve the performance of the algorithm we use to locate an unknown gas source from various downwind gas concentration measurements.

4.2 Measurements of gas concentration and wind data, used to evaluate the dispersion model and algorithm

Measurements for this work were made over a 3-day period at an open air site, approved for limited gas release. The gas source was a certified high pressure cylinder, feeding a calibrated regulator to deliver 1Kg/hour of ethane to a perforated network of pipes laid over an area of $4m^2$, the cylinder and pipes laid out on site are shown in the picture in figure 4.1. The wind was measured in the same location using an ultrasonic anamometer, giving both horizontal and vertical wind speed to an accuracy of 0.1m/s with a data rate of 1Hz. The gas concentrations were measured to an accuracy of 0.1ppb using a lead-salt laser system operating at 3.4microns, performing wavelength modulation spectroscopy of atmospheric samples within a multi pass Heriot cell. The air, filtered to remove particulates, was drawn through the cell using a high speed pump such that the sample refresh rate was approximately 0.5Hz [21]. The gas concentration at a specific position and the corresponding wind

Figure 4.1: The cylinder and perforated pipes laid out to simulate a diffuse micro seepage.

measurements were logged to a common, time stamped, data file for subsequent analysis. We typically measured data for a period of 15 minutes at each position and at approximately 10 different positions on each of the three days. The various measurement locations are marked on a map of the survey area shown in figure 4.2.

4.3 Optimization of the Gaussian plume model

We have shown in [5] that the Gaussian plume model allows easily obtainable data to be used, specifically relating to meteorological data and we have succesfully utilised the Gaussian plume model within our algorithm (chapter 2). Despite this it is clear that the Gaussian plume model has a number of limitations. Firstly, in many applications it is the location of the gas source rather than its flux which is of primary

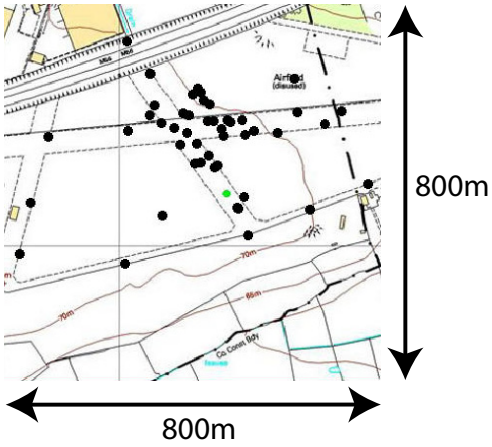


Figure 4.2: The survey area with the location of the cylinder (known source) shown in green and each measurement position shown in black.

importance. Factors such as calibration of the sensor and partial buoyancy of gas may additionally result in a small multiplicative error. Secondly, in estimating the height, σ_{h_i}, and width, σ_{w_i} of the Gaussian plume it is usual to define this by the fluctuation in wind direction over a chosen averaging time. This averaging time is often, somewhat arbitrarily, set as the anticipated transit time of the gas from the source to the detector. Thirdly, that the Gaussian plume model assumes a point source, hence predicting extremely high concentrations near the source, these can perturb the algorithm we apply. Finally, in some situations our measurement system may can contain systematic errors: there may be a slight alignment error in the compass setting, or perturbation in measured wind direction due to the ananometers local environment.

We use our experience of optimisation to try to overcome these limitations. We modify the Gaussian plume model described by equation 2.1 such that it contains

correction factors,

$$C_i = \frac{aS_j}{\pi V_i (b\sigma_{w_i})\sigma_{h_i}} \exp\left(-\frac{1}{2}\left(\frac{\Delta_{w_i}}{b\sigma_{w_i}}\right)^2\right) \exp\left(-\frac{1}{2}\left(\frac{\Delta_{h_i}}{\sigma_{h_i}}\right)^2\right). \tag{4.1}$$

Where a and b are multiplicative correction factors modifying the measured concentration and plume width. Both σ_{w_i} and σ_{h_i} are length measurements and are calculated using the angular standard deviations of the wind direction calculated over a one minute average in both the horizontal, θ_{w_i}, and vertical, θ_{h_i}, directions respectively,

$$\sigma_{w_i} = \theta_{w_i} z + k \tag{4.2}$$

$$\sigma_{h_i} = \theta_{h_i}(z + k). \tag{4.3}$$

where k describes the minimum source size, and therefore prevents singularities in the distribution of predicted concentrations.

In our situation, the vertical offset from the plume centre is zero and the horizontal offset, Δ_{w_i}, is calculated with respect to the plume centre whilst allowing for an additive angular correction c.

Within our plume optimisation model we position the release at it's known location and use the data we measured to simply optimise the correction parameters, a, b and c, such that the model best describes the concentrations we measured. The known release location is shown with respect to the measurement locations in figure 4.2. To ensure that the optimisation only considers likely combinations of correction parameters we set a max/min range of a, b and c. The optimisation algorithm we utilise here is an exhaustive search. This algorithm trials all possible combinations of a, b and c and calculates the predicted concentrations at the actual measurement

locations due to each combination using the modified Gaussian plume model (4.1). To evaluate the quality of each set of predicted concentrations and hence to evaluate the merit of each combination of a, b and c we use a cost function.

The cost function we use for this application is similar to χ^2 (equation 2.4), however we modified the function so that emphasis is placed on correct prediction of the low concentrations,

$$E^k = \frac{1}{N}\sum_{i=1}^{N}\frac{|C_{m_i} - C_i^k|^2}{(C_{m_i} + \sigma_i)^2}. \tag{4.4}$$

Where E^k is the value of the cost function due to combination k, N is the number of measurements made, C_{m_i} is the concentration measured at $i = 1, ..., N$ and σ_i is the estimated error.

We found that using this modified cost function to find the optimised values of a, b and c gave a more realistic set of correction values and a better match over a wider range of measured and predicted concentrations than a χ^2 cost function alone.

The bar charts in figure 4.3 compare the concentrations for each day's data. The measured concentrations are plotted in red, the concentrations calculated using a simple Gaussian plume algorithm where the algorithm only optimizes the release rate are plotted in green, and the concentrations where the Gaussian plume is optimized for the release rate, plume width and compass correction are plotted in blue. From the figure it is clear that by optimizing the wind direction and the plume width as well as the release rate that we are able to produce a better fit to the measured data. The fit can be assessed using the sum of the squared differences. The value calculated using the measured concentrations and the concentrations calculated with optimised release rate is 1.28×10^{-14}. The value calculated using the measured concentrations

and the concentrations calculated with optimised release rate, wind direction and plume width is 8.82×10^{-15}. The typical values of the optimisation parameters are $a = 0.36$, $b = 1.98$ and $c = 0.8$.

4.4 Application of the optimised Gaussian plume model within our algorithm

We have shown in the previous section that we can optimise the Gaussian plume model to predict the concentrations we have measured in the field given a known source and known wind conditions. We now investigate the effect this optimised dispersion model has when included in the algorithm used to locate gas sources from downwind gas concentrations.

We described the algorithm we use to locate an unknown gas source in chapter 2. We use the algorithm with the optimised Gaussian plume model - equation **??** with the values of a, b and c that led to the optimum value of the cost function. To evaluate whether the application of the optimised model leads to improved performance of the algorithm we compare the converged source distributions to those resulting when the non-optimised model is used in the algorithm.

In chapter 3 we described three cost functions which we can use within the algorithm. The first favours postulated source distributions with low total flux, $E(Flux)$, described by equation 3.3, another which aims to minimize the entropy of the postulated source distribution, $E(Maxent)$, described by equation 3.4 and the final cost function which favors a smooth postulated source distribution. Figure 4.4 show

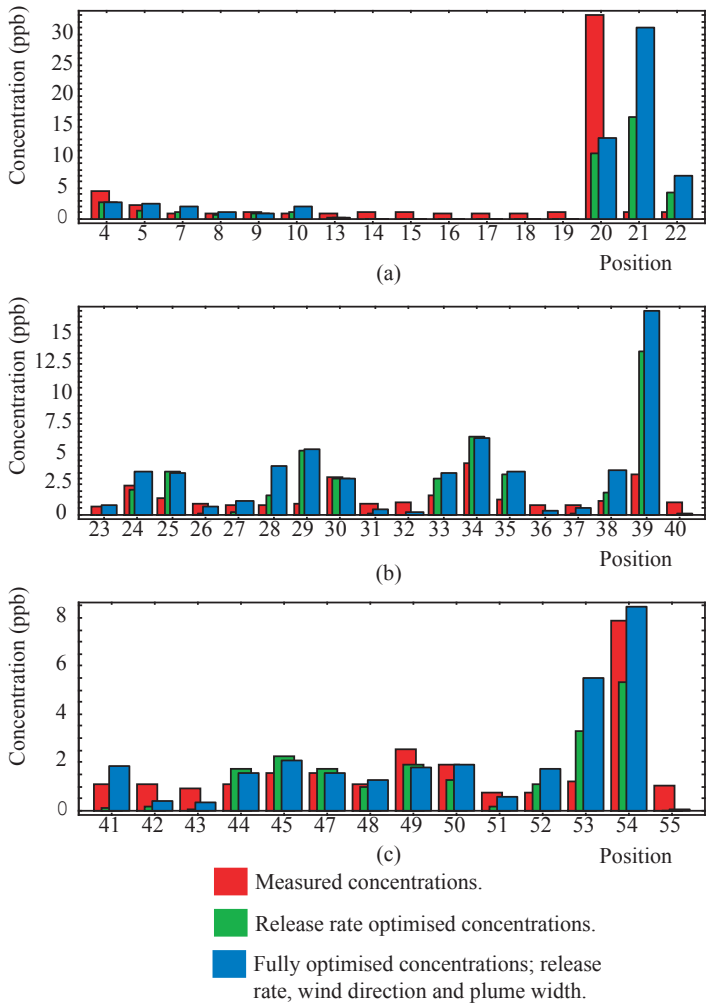


Figure 4.3: Both the measured and optimised concentrations averaged over each measurement are plotted. The measured concentrations are plotted in red, the concentrations calculated having used the algorithm to optimise only the release rate are plotted in green and the concentrations calculated having used the algorithm to optimise release rate, wind direction and plume width are plotted in blue. Plot (a) shows the concentrations corresponding to Monday's measurements, (b) the concentrations corresponding to Tuesday's measurement and (c) the concentrations corresponding to Wednesday's measurements. From the better fit of blue bars to red bars than green bars to red bars we can conclude that by optimising release rate, wind direction and plume width we can optimise the Gaussian plume model.

reconstructions using $E(Flux)$ as the cost function, using both the simple and optimized Gaussian plume dispersion models. In both cases values of λ are chosen so that the reconstructions correspond to a gradual fractional increase in the χ^2 component of the cost function compared to that obtained for $\lambda = 0$. We see that both dispersion models yield similar results, correctly identifying the known source location for a range of λ. Figure 4.4 shows a similar set of reconstructions obtained using $E(Maxent)$ as the cost function, again with little to choose between the two dispersion models.

4.5 Conclusions

We have shown, perhaps not surprisingly, that under a specific set of field conditions the simple Gaussian plume model can be optimized such that the predicted concentrations agree better with those actually measured. Although this specific optimization may not hold for all situations it is probable that in any field environment, a controlled release could be made in advance of real data being collected. The optimized plume model could then be used in subsequent investigations or monitoring of unknown gas sources.

More surprising is that when used within the algorithm, both the simple and optimized Gaussian plume model yield similar results. In both cases the accuracy to which the gas source is located depends less on the choice of dispersion model and more on the choice of cost function and the degree to which the prediction is regularized away from χ^2. This is demonstrated by the comparison of reconstruction

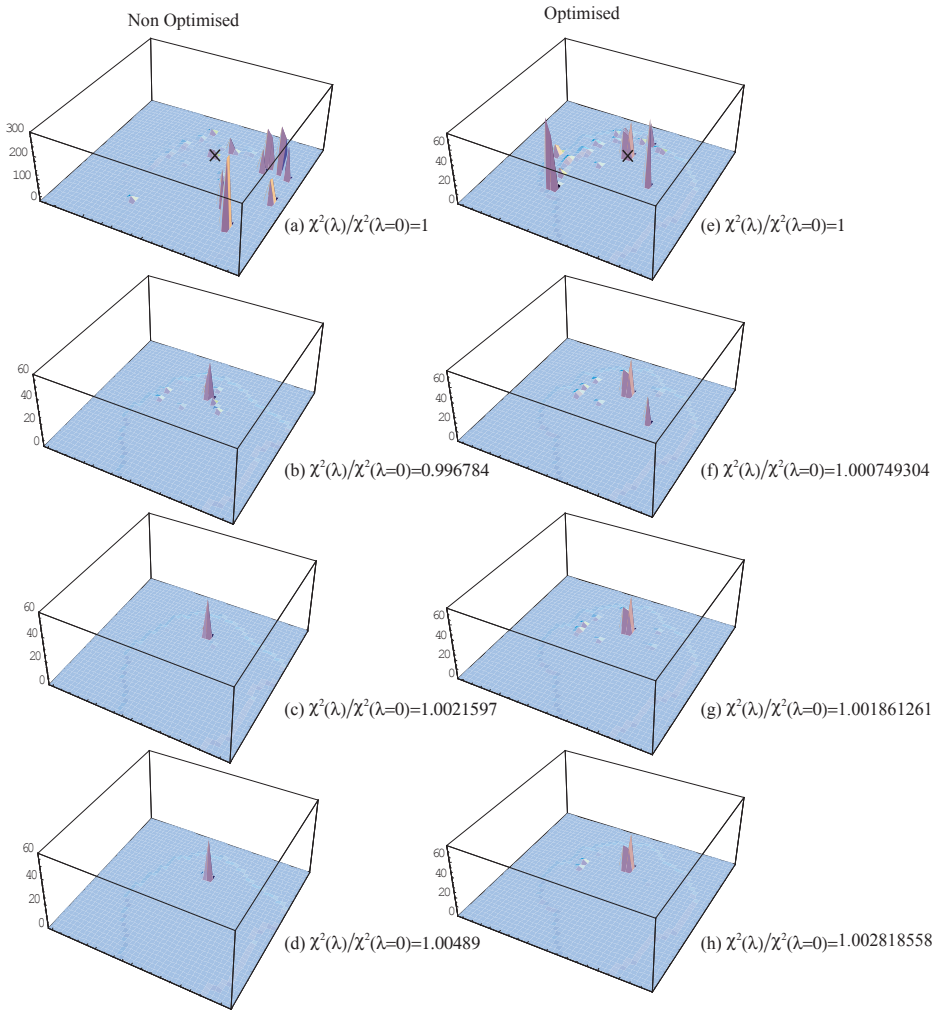


Figure 4.4: Comparison of reconstruction results produced using the algorithm with the optimised and non-optimised Gaussian plume models, the flux cost function is used. The results are shown for a range of regularisation, λ, values. X marks the location of the known source. It is clear that algorithms with both the optimised and non-optimised Gaussian plume models can be used to locate the source. The degree of regularisation has a greater bearing on the result than choice of dispersion model. The base of each graph represents an 800m $\times$ 800m survey area.

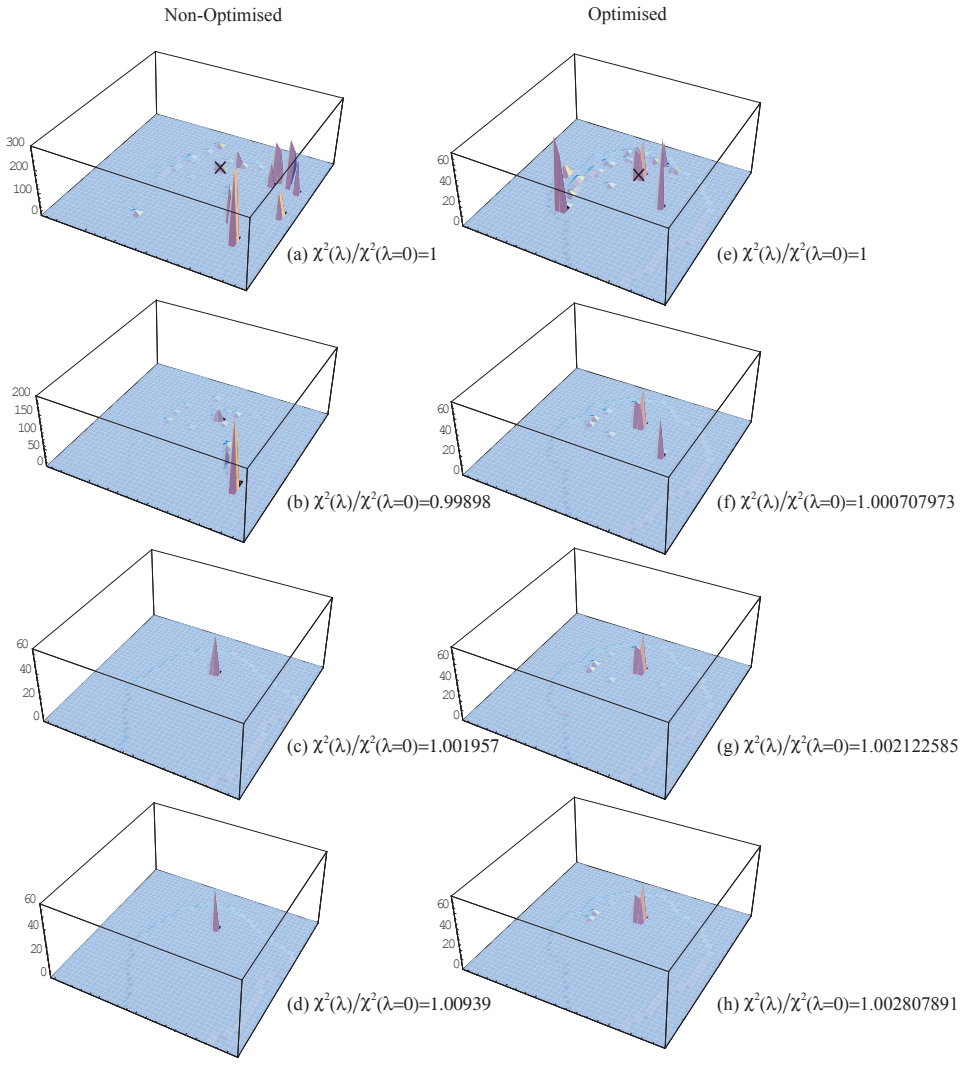


Figure 4.5: Comparison of reconstruction results produced using the algorithm with the optimised and non-optimised Gaussian plume models, the maxent cost function is used. The results are shown for a range of regularisation, λ, values. X marks the location of the known source. It is clear that algorithms with both the optimised and non-optimised Gaussian plume models can be used to locate the source. The degree of regularisation has a greater bearing on the result than choice of dispersion model. The base of each graph represents an 800m $\times$ 800m survey area.

results in figures 4.4 and 4.5. Suggesting that in both cases the errors between predicted and measured concentration are largely random in nature and therefore can be largely overcome by sufficient numbers of measurements and appropriate regularization of the cost function.

Whether more complex dispersion models, requiring more meteorological data, would give an improved performance of the algorithm cannot be ruled out, but it seems that the most important consideration is that the errors in the dispersion model be random.

Chapter 5

Superresolution optical tweezers

Optical tweezers use foci of light to trap and manipulate microscopic particles [22, 23]. It is the intensity distribution of each focus that defines its trapping characteristics. The trapping characteristics we are interested in here are trap resolution and relative trap stiffness. Improvement in both of these characteristics could be useful for the development of optical tweezers in many areas. In this chapter we show, using computer simulations, that by using superresolution algorithms we can produce foci with improved trapping characteristics. This work was published in reference [24].

5.1 Superresolution

Optical and acoustic signals are often band limited. Monochromatic light is one such signal band limited because the transverse k value is between $-\frac{2\pi}{\lambda}$ and $+\frac{2\pi}{\lambda}$. The diffraction limit suggests that the smallest focus size that can be achieved is approximately half the wavelength. This is also the smallest distance that can be achieved between neighbouring foci. Superresolution gives the ability to beat the

diffraction limit and shape a signal such that it locally contains features that have a higher spatial frequency than would normally be expected according to the frequency spectrum. This is achievable by leaving part of the signal unconstrained, which then allows the rest of the signal to be shaped to greater detail. Such signals have been described by Berry [25], who constructs a 1Hz frequency-bandwidth-limited sound signal that contains Beethoven's ninth symphony. The downside of this signal is that it is very quiet. Locally, the signal contains frequency components not contained in the signal as a whole, which Berry uses to shape the symphony. These frequency components are called superoscillations.

Superoscillations have been found to exist in optics, for example near the centre of an optical vortex [26] and close to a beam focus [27]. It is possible to shape superoscillations; in optics this has been achieved for example in the development and implementation of the Gerchberg algorithm [28, 29]. Such a light-shaping algorithm aims to find a hologram that will produce a desired light beam, and in the case of the Gerchberg algorithm the desired beam can contain structures smaller than allowed by the diffraction limit.

One area of optics where the application of superoscillations could have significant implications is microscopy. Much work has already been done in this area, minimizing the spot size of a light beam for use in microscopy [30]. More commonly used superresolution techniques in microscopy include 4Pi-microscopy [31] and STED microscopy [32]. Such techniques are reviewed in reference [33]. The resolution of traditional confocal laser scanning microscopes is limited by the size to which the excitation spot can be focussed. STED (stimulated emission depletion) microscopy

beats this limit by using a very short excitation pulse followed by a depletion pulse, which significantly reduces the size of the excited area and determines the resolution. It is the fluorescence of this reduced excited area which is then detected by the microscope.

Other areas where superresolution techniques have already been utilised include two-photon polymerisation [34] and for the manipulation of molecules [35].

The ability to produce an intensity distribution that contains features that beat the diffraction limit appears desirable for optical tweezers [22]. Materials exist that have a negative refractive index. A slab of such materials can amplify evanescent waves and thus resolve images to better than the diffraction limit [36]. Such materials have already been used in optical tweezers to produce superresolved optical traps. In chapter 7 we discuss the use of a light shaping algorithm to shape a three dimensional intensity distribution of evanescent waves. This intensity distribution contains superresolution optical traps.

The application of superresolution holography in optical tweezers has previously been suggested but never studied any further. We describe our implementation of two superresolution holography algorithms and study theoretically the effect superresolution has on trap resolution and relative trap stiffness.

5.2 Optical tweezers

Optical tweezers [22] use foci of light to trap, move and manipulate microscopic dielectric particles. Provided the refractive index of the particle being trapped is

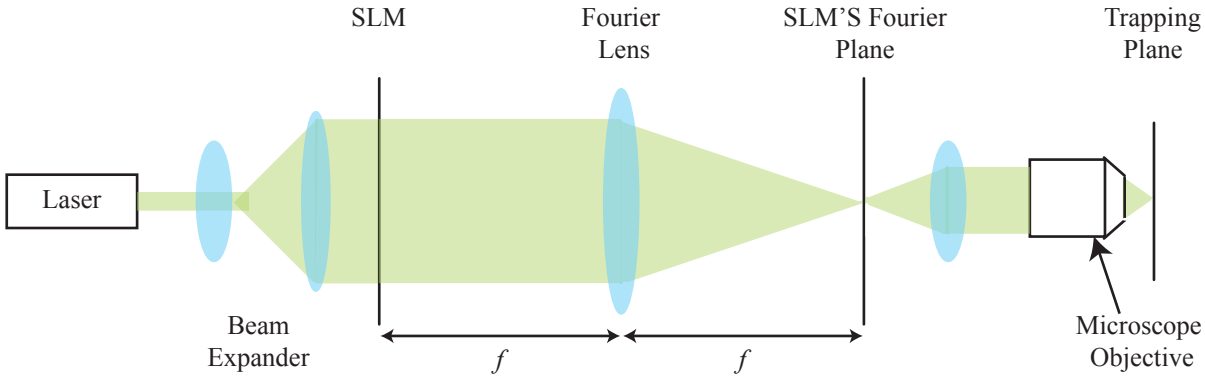


Figure 5.1: Typical layout for holographic optical tweezers. A laser beam is expanded before hitting a spatial light modulator (SLM). The SLM is shown here as operating in transmission, however most SLMs operate in reflection. The first lens acts like a Fourier lens and Fourier transforms the field in SLM plane onto the Fourier plane. The remaining optics are used to image the beam in the Fourier plane onto the trapping plane. It is in this, the trapping plane, where the foci produced are used as optical traps to manipulate particles.

greater than that of the surrounding medium the particle will be attracted to and held in the beam focus as a result of the gradient force [37, 38].

Simple arrangements of optical elements can be used to create beam foci suitable for optical trapping. A lens positioned in the centre of a laser beam will create a single focus and a lens array will create an array of foci; foci created in these ways can be shifted sideways using a prism or a mirror. Although the use of these optical elements results in foci that can be used as optical traps there is clearly a limit on how particles can be manipulated when trapped in this way. Holographic optical tweezers replace such optical elements with spatial light modulators. Figure 5.1 shows a typical experimental setup for holographic optical tweezers.

Recent development of spatial light modulator (SLM) technology has led to ex-

tensive improvements in the optical trap configurations that can be achieved and also significantly progressed the interactive use of optical tweezers. An SLM consists of an array of liquid crystals, each of which is addressable under computer control. The properties of each of the liquid crystals can be changed in almost real time to give the illuminating beam a specific phase (it is also possible for SLMs to intensity-modulate the beam). An SLM is the equivalent of a hologram and the phase pattern applied to the SLM to achieve intensity shaping can be calculated using any number of light-shaping algorithms [15, 8, 39]. Such light shaping algorithms numerically aim to find a phase array (hologram) that, when combined with the illumination intensity, will give the best approximation to a desired beam, usually in the SLM/hologram's Fourier plane. The desired beam can take any physically realisable form such as an approximation to a Bessel beam or even an array of multiple foci; examples of both beams have been proven to be useful in optical tweezers [40, 41]. Many of the light shaping algorithms can produce a phase array in almost real time given a change to the desired beam [42]. Such algorithms, and the fact that the phase hologram can be applied to an SLM, make this setup ideal for use in optical tweezers and even allow optical tweezers to work interactively.

Improvements in trap resolution and relative trap stiffness are likely to open up new possibilities in areas such as the manufacturing of permanent structures and the use of tweezers to make precision biological measurements. Trap resolution refers to the inverse of the minimum resolvable distance separating two optical traps that can be achieved and trap stiffness is a measure of how still a trap can hold a particle. Both of these characteristics are defined by the intensity distribution of the traps

and are what we chose to try to improve in this chapter. The trap resolution and stiffness that can be achieved in the traditional holographic optical tweezers setup, discussed here, is limited by the diffraction limit. We aim to beat these limits with the introduction of superresolution.

5.2.1 Superresolution optical tweezers

To implement superresolution into holographic optical tweezers no changes to the setup are necessary. It is however required that in order to shape (even just represent) superresolved structure we need to calculate the field in the Fourier plane at higher resolution than we would normally represent the field. This is achieved by zero-padding the field in the SLM plane, which corresponds to trigonometrically interpolating in the Fourier plane. The setup described in figure 5.1 can be used to describe superresolution optical tweezers.

In optical tweezers it would be normal to shape all of the beam in the Fourier plane. For example if the desired beam was a single bright spot surrounded by darkness it would be common to desire all of the power in the Fourier plane to make up the spot and any good algorithm would attempt to put all of the power in the Fourier plane within this spot. As was discussed in section 5.1, superresolution is achieved in part of a signal. In that case we were refering to an audio signal, in this case we are considering a signal which is a transverse light beam function which is frequency limited.

To implement superresolution into the light shaping algorithms we use – direct search [15] and Gerchberg-Saxton [8] algorithms – we restrict the algorithms such

that the optical traps are shaped within a central area of interest, A, of diameter a, in the Fourier plane. Within A we limit the amount of power to be a fraction, P, of the total power in the beam. Outwith A the algorithm does not attempt to directly influence the form the beam takes. By decreasing P we are increasing the superresolution strength, allowing improved shaping of the light within A and leading to improvements in both the trap resolution, d, and relative trap stiffness, $S_{x,y}$.

Without implementing this limitation to the Fourier plane that leads to superresolution light shaping, improvements can be made to the achievable trap resolution by altering the beam that illuminates the spatial light modulator (SLM). It is common to illuminate the SLM with a narrow Gaussian beam such that all of the power in the beam is utilised. This increase in power is of course transferred to the optical traps. However, when the SLM is illuminated with a narrow Gaussian beam the outer pixels which correspond to high spatial frequencies in the Fourier plane are not illuminated, leading to a reduction in the trap resolution that can be achieved. The alternative method leads to a decrease in power with better shaping, giving a valuable increase in trap resolution and relative trap stiffness. In this case the SLM is illuminated with a wide Gaussian beam such that the pixels on the outside of the SLM are also illuminated. However not all of the power in the illuminating beam hits the SLM leading to a reduction in power in the optical traps. We calculate our superresolution results using both a narrow and wide Gaussian beam to illuminate the SLM.

The geometry describing both the restriction we place on the Fourier plane to implement superresolution and both illumination options are shown in figure 5.2.

We implement this geometry into both the direct search algorithm (appendix C) and the Gerchberg algorithm.

5.3 Superresolution algorithms

We chose to adapt the direct search algorithm [15] and we also used the Gerchberg algorithm [28] to introduce superresolution by including the restrictions discussed in section 5.2.1.

5.3.1 Superresolution direct search

The direct search algorithm [15] repeatedly makes random changes to the phase in the hologram plane until it results in a light distribution in the Fourier plane that is the best approximation to the desired light beam. The changes are assessed using a cost function which compares the resulting light distribution to the desired light beam. An improvement is made when the current value of the cost function is less than the previous value. The best approximation is reached when no further changes result in a lower value of the cost function. We modify the direct search algorithm – described in appendix C – such that it contains the modifications required to shape superresolution optical traps.

The algorithm begins by initially generating a random phase distribution, $\phi_S^{n=0}(x_i, y_j)$, containing phases between 0 and 2π, that represents the SLM, where n is the current iteration number, x_i and y_j are the coordinates of the SLM pixels ($i, j = 1, ...N$ where $N \times N$ is the total number of pixels that represents the SLM). All subsequent

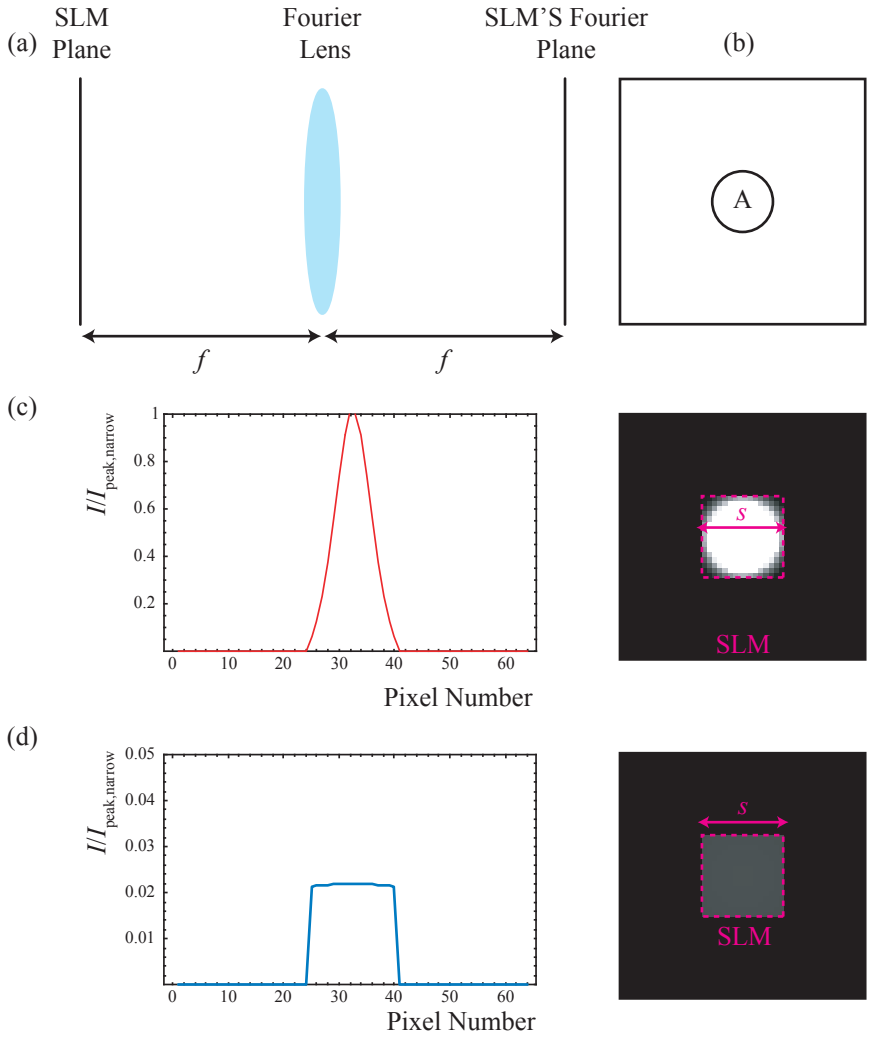


Figure 5.2: The geometry of the illumination of an SLM with the restrictions on the SLM's Fourier plane required for superresolution algorithm. (a) The Fourier relationship between the SLM and its Fourier plane utilised by light shaping algorithms. (b) The xy cross section of the Fourier plane of length L showing the central area of interest, A, of diameter a. (c) The illumination of the zero padded SLM by the narrow Gaussian beam. The SLM plane is shown with the narrow Gaussian beam illuminating the central area (highlighted by the pink dashed line) of length s. The plot shows, for the pixels that di-sect the beam at its peak, the intensity of the beam, I, as a fraction of its peak intensity, $I_{peak,narrow}$. (d) Shows the same as in (c) for illumination of the zero padded SLM by the wide Gaussian beam.

iterations begin by combining the phase with the illumination intensity, $I_S(x_i, y_j)$, to give the field in the SLM plane, $U_S^n = \sqrt{I_S(x_i, y_j)}\exp(i\phi_S^n(x_i, y_j))$. As is shown in figure 5.2, the field that represents the SLM is zero-padded (U_S^n is positioned in the centre of the larger $Y \times Y$ matrix which is 0 outside the area that represents the SLM) and is then Fourier-transformed to give the corresponding field in the Fourier plane, $U_F^n(x_k, y_l)$, where x_k and y_l $(k, l = 1, ..., Y)$ are the coordinates of the pixels in the Fourier plane which is represented by the $Y \times Y$ pixels. We use a cost function to evaluate how close the current intensity in the Fourier plane, $I_F^n(x_k, y_l) = |U_F^n(x_k, y_l)|^2$, is to the desired intensity in the Fourier plane $I_D(x_k, y_l)$. The cost function we use is the sum of the square of the difference between the actual and desired intensity at each pixel,

$$E^n = \sum_{(x_k, y_l \epsilon A)} (I_F^n(x_k, y_l) - I_D(x_k, y_l))^2. \tag{5.1}$$

As we are only interested in shaping within the area of interest, A, we only compare pixels within A. If E^n is less than the value of the cost function calculated during the previous successful iteration, E^{n-1}, we retain $\phi_S^n(x_i, y_j)$ and replace the value of a random pixel with a random value between 0 and 2π to give the phase distribution in the SLM, $\phi_S^{n+1}(x_i, y_j)$, to be used the next iteration, $n + 1$. However if $E^n > E^{n-1}$ we discard $\phi_S^n(x_i, y_j)$ and replace a random pixel in the previously accepted (successful) phase distribution that represents the SLM, $\phi_S^{n-1}(x_i, y_j)$, with a random value between 0 and 2π to give $\phi_S^{n+1}(x_i, y_j)$. It should be noted that if the current iteration is the first, $n = 0$, $\phi_S^n(x_i, y_j)$ is always retained. The next iteration can then begin in the same way as the previous. The algorithm continues to iterate until it converges; no more changes to the phase distribution that represents the SLM

improve the value of the cost function.

We scale the desired intensity distribution, $I_D(x_k, y_l)$, within A to be a fraction P of the total power in the beam, as is described by the following equation:

$$\sum_{(x_k, y_l \epsilon A)} I_D(x_k, y_l) = P \sum_{x_i, y_j} I_S(x_i, y_j). \tag{5.2}$$

A value of P should be chosen such that $0 < P < 1$. For a value of $P = 1$ no superresolution is present, the closer P is to 0, the greater the superresolution effect.

5.3.2 Superresolution Gerchberg algorithm

A superresolution algorithm was developed by Gerchberg [28] that can be used to shape a signal to better than the diffraction limit. The Gerchberg algorithm is a variation on the Gerchberg-Saxton algorithm [8] (see appendix A); during each iteration of the Gerchberg algorithm only part of the intensity distribution in the Fourier plane is replaced as opposed to the whole intensity distribution in the Gerchberg-Saxton algorithm.

The first iteration, $n = 0$, of the Gerchberg algorithm begins with an arbitrary phase distribution, $\phi_S^{n=0}(x_i, y_j)$, being calculated to represent the phase of the SLM, where x_i and y_j are the coordinates of the SLM pixels $(i, j = 1, ...N)$ and $N \times N$ is the total number of pixels that represent the SLM. This phase $\phi_S^{n=0}(x_i, y_j)$ is combined with the illumination intensity, $I_S(x_i, y_j)$, and in all subsequent iterations $(n > 0)$ it is the phase calculated during the previous iteration that is combined with $I_S(x_i, y_j)$ to give the field in the Fourier plane, $U_S^n = \sqrt{I_S(x_i, y_j)} \exp(i\phi_S^n(x_i, y_j))$, at the current iteration n. As is shown in figure 5.2, U_S^n is zero-padded (U_S^n is

positioned in the centre of the larger $Y \times Y$ matrix, which is 0 outside the area that represents the SLM) and is then Fourier-transformed to give the field in the Fourier plane, $U_F^n(x_k, y_l) = \sqrt{I_F^n(x_i, y_j)} \exp(i\phi_F^n(x_i, y_j))$, where x_k and y_l $(k, l = 1, ..., Y)$ are the indices of the pixels in the Fourier plane. As we are only interested in shaping the beam within A, the algorithm retains both the resulting phase in the entire Fourier plane and the resulting intensity outwith A. The resulting intensity within A is replaced with the desired intensity within A, rescaled such that the desired fraction of power in the beam, P, lies within A, and then combined with the retained intensity to give the replaced field in the Fourier plane, $I_{F,R}^n(x_k, y_l)$. The new field $U_{F,R}^n = \sqrt{I_{F,R}(x_k, y_l)} \exp(i\phi_F^n(x_k, y_l))$ is inverse-Fourier-transformed to give the field in the SLM plane. It is only the resulting phase, $\phi_S^{n+1}(x_i, y_j)$, in the SLM plane that is retained and that forms the start point of the next iteration. This process continues until the algorithm converges – repeated iterations no longer improve the resulting beam in the Fourier plane.

5.4 Resolution

Trap resolution is inversely related to the minimum distance separating two optical traps (foci). The minimum separation, d, between two traps is the distance at which the intensity distribution of the traps still satisfies the Rayleigh criterion [43] . For a given setup there is a limit on the trap separation that is normally achievable without the use of superresolution – we will refer to this limit as the resolution benchmark $d_{\text{benchmark}}$. In this section we show that by using the superresolution light shaping

algorithms discussed in section 5.3 we are able to improve the achievable trap separation to values less than the benchmark and that by increasing the superresolution strength (as $P \to 0$) we are able to further improve this. It is the intensity distribution of the beam that defines the trap resolution, hence we investigate this by numerically studying the intensity distribution of beams shaped by our algorithm. We restrict ourselves here to the superresolution direct search algorithm. Before describing how we calculate the trap separation benchmark, the trap separation and whether they are resolvable from the resulting beam we will first describe the desired beam we use in the algorithm.

As was discussed in sections 5.3.1 and 5.3.2, both the superresolution direct search algorithm and the superresolution Gerchberg algorithm, like their non-superresolution counterparts, require the user to specify a desired beam in the Fourier plane. These algorithms aim to find the hologram that gives the best approximation to the desired beam. To investigate trap separation we consider the simplest case of two optical traps separated in the x direction. The desired beam therefore contains two optical traps (represented in the algorithm by the only two non-zero values in the array representing the desired intensity distribution) which are normalised to contain a fraction P of the total beam power in the Fourier plane. Such a desired beam can only have a discrete separation between traps limited by the fact that the beam is described by an array containing a specific number of elements. An example of a desired beam used in the algorithms to calculate resolution is shown in figure 5.3.

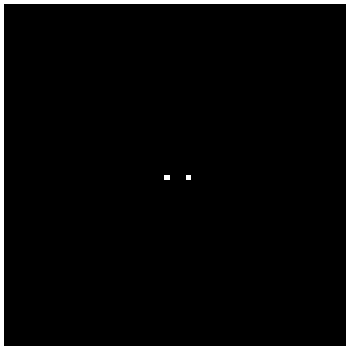


Figure 5.3: An example of a desired beam for calculating resolution. The trap separation (or inverse resolution) in this case is $d = 4$pixels

5.4.1 Calculating the resolution and Rayleigh criterion

The Rayleigh criterion [43] states that two intensity peaks are resolved if the ratio between the intensity of the smaller of the two peaks, $I_{\text{peak,min}}$, (for simplicity we will refer to this in this section as the peak intensity) and the intensity of the trough between the peaks, I_{trough}, meets the following criterion:

$$\frac{I_{\text{trough}}}{I_{\text{peak,min}}} \leq \frac{8}{\pi^2}. \tag{5.3}$$

Figure 5.4 (a) shows the intensity distribution (xy cross section) of a beam (propagating in the z direction) shaped using the direct search algorithm. It contains two intensity maxima, each of which can be considered an optical trap. The graph shown in figure 5.4 (b) is a plot of the intensity at each of the pixels that dissect the peaks at their maximum (in the x direction along the $y = 0$ axes as shown), ranging from a pixel to the left of the first peak to a pixel to the right of the second peak. To evaluate whether the traps satisfy the Rayleigh criterion (equation (5.3)), we locate the peaks and trough from the discrete intensity values at each pixel by fitting parabolas to the

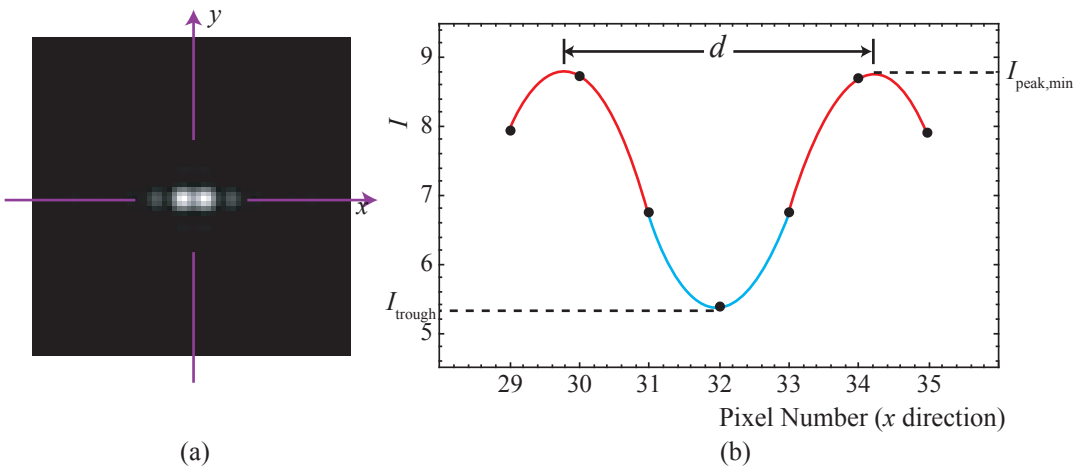


Figure 5.4: The intensity cross section of a beam containing two optical traps. The xy cross section of the beam, which is propagating in the z direction, is shown in (a). In (b) the intensity of the beam is plotted along a line that di-sects the optical traps through their peaks. Parabolas are fitted to the peaks (red) and the trough (blue) to give the actual location and intensity of the peaks and trough. This information is then used to calculate whether the optical traps satisfy the Rayleigh criterion; in this case the beam satisfies the Rayleigh criterion.

three points closest to each of the peaks and the trough, like in the example shown in figure 5.4 (b). Having fitted the parabolas we now achieve interpolated values for the separation between the peaks, d, the intensity of the trough, I_{trough}, and the intensity of the peak with the lowest intensity value, $I_{\text{peak,min}}$. If the condition in equation (5.3) holds true for the values of I_{trough} and $I_{\text{peak,min}}$, the traps are resolvable with a trap separation of d.

5.4.2 The resolution benchmark

To show that we can improve the trap resolution using light shaping algorithms that incorporate superresolution (section 5.3) we must first define a benchmark trap

separation to improve upon.

To create multiple peaks across the Fourier plane of the set-up that we use (as shown in figure 5.2) it is necessary to illuminate four different points of the SLM. The optimum separation of peaks in the intensity distribution in the Fourier plane that can be achieved in this way is to illuminate the outer most pixels of the SLM, essentially interfering the highest spatial frequency components. Such an SLM illumination, shown by beam (a) in figure 5.5, when Fourier transformed results in an array of intensity peaks in the Fourier plane, like beam (b) shown in figure 5.5. Having achieved the array of peaks it is then necessary to calculate their separation, d, which can be achieved using the method discussed in section 5.4.1.

The beams in figure 5.5 are calculated using the parameters we use to calculate our results. The benchmark resolution is then calculated (in the way discussed in section 5.4.1) in our case to be $d_{\text{benchmark}} = 4.6$pixels; the intensity across the peaks at their maximum is plotted in figure 5.5 with the parabolas fitted to the peaks and trough. When we run the direct search superresolution algorithm for the wide Gaussian illumination of the SLM and with $P = 1$ (no superresolution) we find that the algorithm results in a value of trap separation approximately equal to $d_{\text{benchmark}}$.

5.4.3 Improvements in resolution

We run the algorithm for various values of desired trap resolution to find the desired power fraction, P, within the area of interest, A, that results in a beam in the Fourier plane where the intensity at the minimum of the two peaks, I_{trough}, to

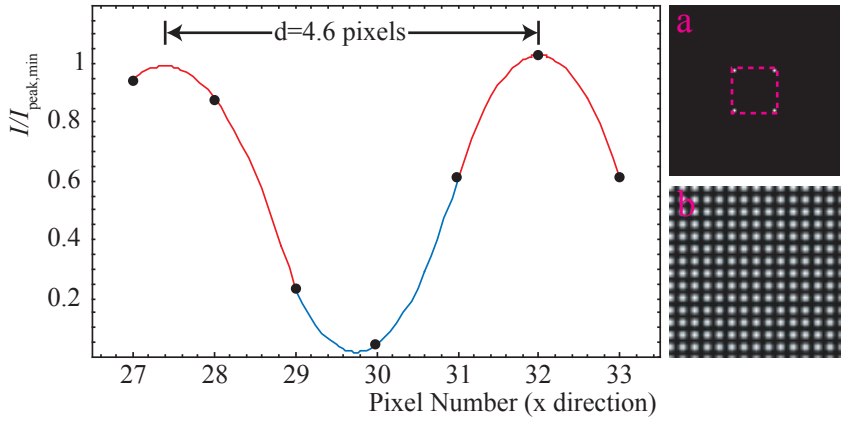


Figure 5.5: Calculating a benchmark trap separation for comparison. The beam cross sections represent (a) the illumination of the SLM and (b) the resulting beam in the Fourier plane. The dashed line in (a) represents the edge of the area that represents the SLM. In the graph the intensity of the peaks is plotted along the axis that dissect the peaks at their maximum.

the intensity at the trough, $I_{\text{peak,min}}$, just meets the Rayleigh criterion ($0.8 \leq I_{\text{trough}}/I_{\text{peak,min}} \leq 8/\pi^2$). Figure 5.6 shows the separation d between resolvable optical traps calculated using the direct search algorithm plotted as a function of the logarithm of the actual power in the area of interest p. The desired beam in each run of the algorithm is normalised such that it contains a desired power fraction P in the area of interest A. When the algorithm has converged the beam in the Fourier plane within A does not contain exactly the power fraction P we therefore calculate and use the actual power in the area of interest of the converged beam p in the results shown. For all beams calculated we find that $p < P$. The graph shows that a decrease in p leads to an improvement in d, although at a cost: the line fitted to the data has a slope of ≈ 0.18, which means that a reduction in power in A from 100% to 10% results in only an 18% reduction in d.

The data is plotted relative to the trap separation benchmark $d_{\text{benchmark}} = 4.6$ i.e. $d/d_{\text{benchmark}}$ is plotted. As all of the values of $d/d_{\text{benchmark}}$ plotted are less than 1, our results show that by using the superresolution direct search algorithm we are able to reduce d to less than the benchmark $d_{\text{benchmark}}$ which was set by illuminating the four corner pixels of the SLM. We must however highlight that when calculating the trap separation benchmark the trough intensity drops to 0, however we select the values plotted in figure 5.6 such that the trough intensity is only $\approx 81\%$ of the peak intensity. For this reason perhaps a more realistic comparison would be with the trap separation calculated using the direct search without superresolution ($P = 1$) d_{max} which is calculated to be 4.4 pixels, all trap separations plotted here are less than 4 pixels.

5.5 Relative trap stiffness

The trap stiffness [44] is the "spring constant" of the trap. The trap stiffness defines the ability of an optical trap to hold a trapped particle tight; the greater the trap stiffness the tighter the particle will be held.

Just like the trap resolution it is the intensity distribution of the optical trap that defines the trap stiffness. In the case of a point particle (which we consider here) the trap stiffness can be calculated as the second derivative of the intensity at the peak. Trap stiffness is proportional to the intensity in the trap. Therefore, if we consider two traps of equal peak intensity, the peak with narrower width will have a higher trap stiffness. It is this dependence on the intensity in the trap which makes

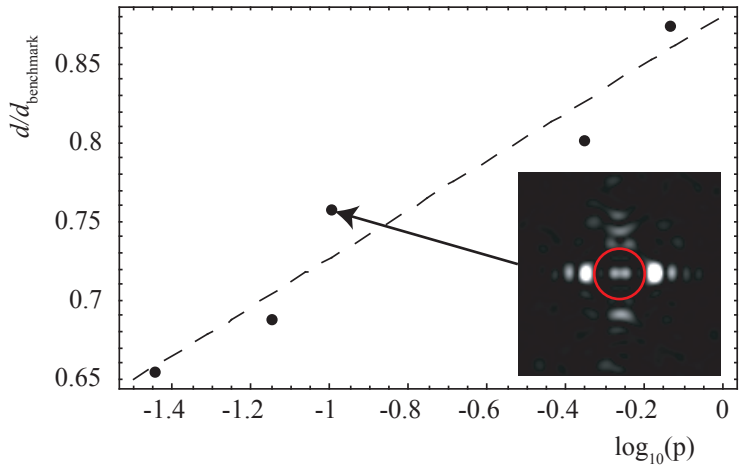


Figure 5.6: The trap separation, d, as a function of the logarithm (base 10) of the actual fraction of the power ration in the area of interest p. The trap resolution is plotted relative to the benchmark resolution, $d_{\text{benchmark}}$. All the values plotted here are for those light distributions that are just only just resolvable according to the Rayleigh criterion: the ratio of the lower of the peak intensities to the trough intensity is between 0.8 and $8/\pi^2$. All the data shown was calculated using the direct search algorithm with a wide Gaussian illumination. The dashed line is a linear fit to the data. The beam cross section inset is an example of the shaped light distribution in the Fourier plane (the beam shown corresponds to the point at $\log_{10}(p) = -1.15$) the red circle shown indicates the area of interest.

it necessary for us to calculate and compare the relative trap stiffness here; we find that using our algorithm with increased superresolution strength (decreasing P) the width of the peak does not decrease at the same rate as the peak intensity, therefore not improving the trap stiffness. The relative trap stiffness is the stiffness divided by the intensity at the peak of the optical trap, I_{peak}.

To show improvements in relative trap stiffness beyond the relative trap stiffness benchmark (the optimum value of relative trap stiffness normally achievable without the use of superresolution) using the superresolution algorithms, we use a desired beam in both the superresolution direct search algorithm and superresolution Gerchberg algorithm that consists of a single intensity maximum at its centre. This intensity maximum is normalised such that it contains the desired fraction P of the total power in the beam. This desired beam has optimum stiffness that can be achieved for the value of P; all possible power is in a single pixel surrounded by pixels of 0 value. We repeatedly run the algorithm for various values of P and show that as $P \to 0$ (increasing superresolution strength) the relative trap stiffness improves beyond the benchmark. Before we show the improvements we make with the introduction of superresolution, we discuss how we calculate relative trap stiffness and derive the relative trap stiffness benchmark.

5.5.1 Calculating relative trap stiffness

For an infinitely small trapped particle the relative trap stiffness can be calculated as the second derivative of the intensity at the peak of the optical trap divided by the peak intensity. To calculate the relative trap stiffness of an optical trap in a

beam shaped using one of our algorithms (like the beam shown in figure 5.7) we fit a parabola to the three pixel intensities closest to the trap. We fit parabolas, $I(x)$ and $I(y)$, to the pixels that dissect the optical trap at its peak in both the x and y directions. Figure 5.7 shows the parabola fitted to the pixel intensities that dissect an optical trap in the x direction. The relative trap stiffness in the x and y direction, S_x and S_y, can be calculated using the following equations:

$$S_x = \frac{1}{I_{peak}} \frac{\mathrm{d}^2 I}{\mathrm{d}x^2} \tag{5.4}$$

$$S_y = \frac{1}{I_{peak}} \frac{\mathrm{d}^2 I}{\mathrm{d}y^2}. \tag{5.5}$$

I_{peak} is the peak intensity of the optical trap – the actual intensity peak calculated from the fitted parabola. The values of both S_x and S_y are comparable; we use their average, $S_{x,y} = (S_x + S_y)/2$, as a measure of relative trap stiffness.

5.5.2 The relative trap stiffness benchmark

Without the use of superresolution algorithms the relative trap stiffness of an optical trap positioned at the centre of the Fourier plane for our wide and narrow illuminations would simply be, in each case, that of the Fourier transform of the illumination combined with a uniform phase distribution. We refer to this value as the relative trap stiffness benchmark. Figure 5.8 shows a plot of the intensities in the pixels that dissect an optical trap produced in this way as a function of their peak intensity for the trap resulting from each illumination. From this plot it is easy to see that with a wide Gaussian illumination the relative trap stiffness is much greater than that

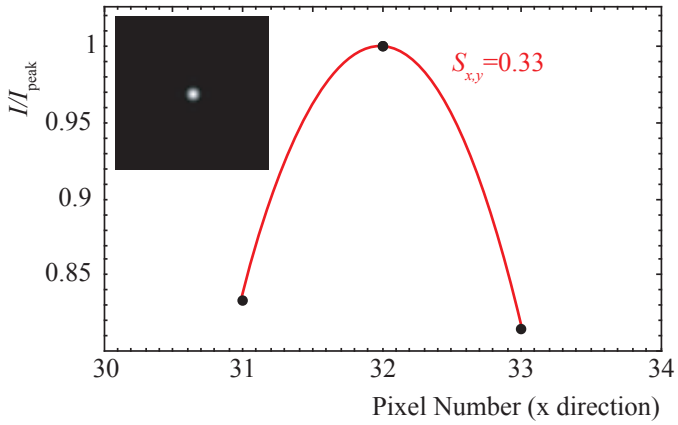


Figure 5.7: Calculating the relative trap stiffness by fitting a parabola to the intensities of the pixels that dissect the optical trap at its maximum. The cross-section of a beam shaped using the direct search superresolution algorithm is shown in the inset. The main plot shows the intensities in the pixels that dissect the optical trap in the x direction with the parabola fitted from which the relative trap stiffness is calculated.

produced using the narrow Gaussian illumination, as previously discussed in section 5.2.1. Using the method discussed in section 5.5.1 we calculate the benchmark for the wide Gaussian illumination to be 0.375pixel^{-2} and for the narrow Gaussian illumination to be 0.173pixel^{-2}.

5.5.3 Improvements in relative trap stiffness

By running both superresolution algorithms for values $0 < P \leq 1$ we find that by increasing the superresolution strength ($P \to 0$) the relative trap stiffness increases. Figure 5.9 shows the relative trap stiffness, $S_{x,y}$, plotted as a function of p, the actual fraction of power within the area of interest, A, for both illuminations and superresolution algorithms. From the results plotted here it is clear that the increase

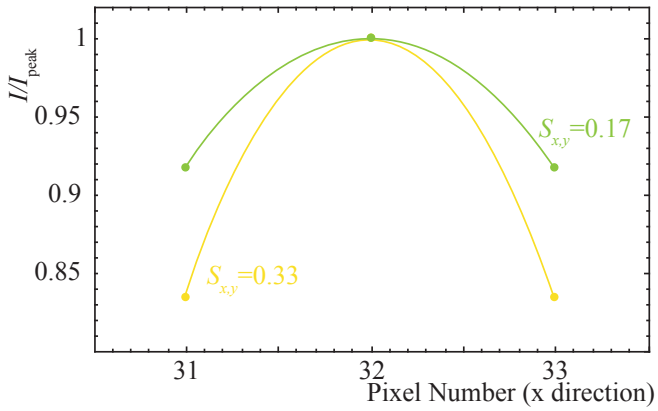


Figure 5.8: Calculating the relative trap stiffness benchmark by fitting parabolas to the pixel intensities that dissect the optical traps resulting from a Fourier transform of both the narrow (shown in green) and wide Gaussian illumination (shown in yellow) with a uniform phase distribution. In order to compare the relative trap stiffness for the trap resulting from both the narrow and wide illumination we plot intensity, I, divided by peak intensity, I_{peak}. From the plot we can see that with the wide Gaussian illumination the relative trap stiffness is much greater than that produced using the narrow Gaussian illumination.

in $S_{x,y}$ is greatest for the direct search superresolution algorithm with the narrow Gaussian illumination; reducing p by 90% leads to an improvement in $S_{x,y}$ by a factor of ≈ 1.7.

5.6 Z-trapping properties

We have so far been concerned with shaping the light in one transverse plane ($z = 0$) only. The result was one or more small foci in a disc of darkness in the $x - y$ plane, the dark disc is surrounded by brightness. Here we discuss the trapping properties of such light fields in the direction perpendicular to the trapping plane, z. When moving out of the $z = 0$ plane, the small foci will diffract very quickly, which suggests that the z-trapping properties of such fields are very good. On the other hand, the brightness surrounding the central dark disc might diffract into the center very quickly, spoiling the z-trapping properties. It is not obvious which of the two mechanisms dominates, and therefore what the z-trapping properties of our shaped light fields are.

We restrict ourselves here to demonstrating that it is possible to create situations in which the former mechanism dominates, corresponding to good z-trapping properties. This is also backed up by a recent investigation into the lifetime of superoscillations in quantum-mechanical wave functions [45]. We calculate the z-dependence of a specific light field. The light field we chose to investigate was the single peak calculated with the DS algorithm for illumination with a narrow-Gaussian-beam and a power ratio in the area of interest of $p = 0.029$. To calculate the z evolution of

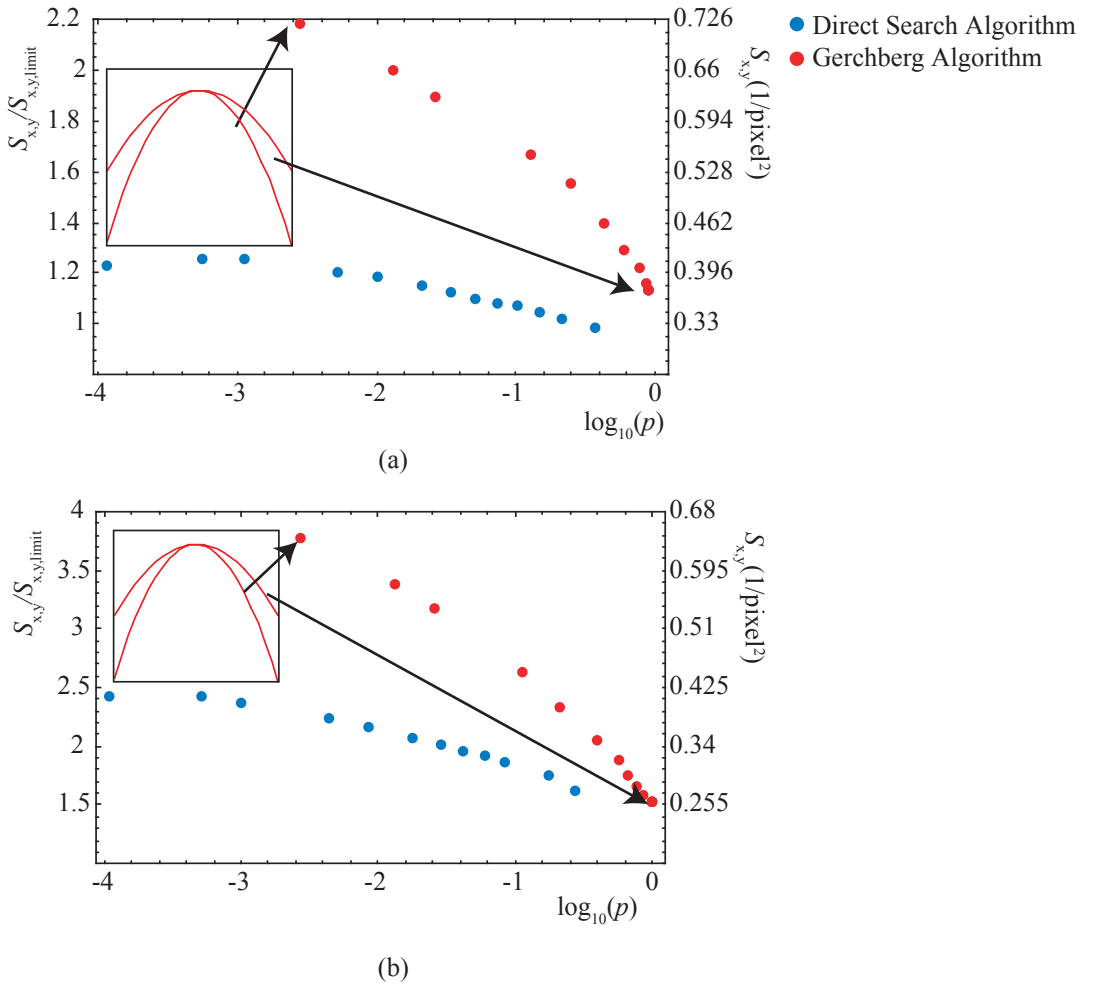


Figure 5.9: Relative trap stiffness, $S_{x,y}$, and relative trap stiffness with respect to the benchmark value $S_{x,y\text{benchmark}}$ plotted as a function of the power in the area of interest, p. $S_{x,y}$ is plotted for both the wide Gaussian illumination (a) and the narrow Gaussian illumination (b). In both cases the data is calculated using two superresolution algorithms: the direct search algorithm (red points) and Gerchberg algorithm (blue points). The direct search supperesolution algorithm with the narrow Gaussian illumination results in the greatest increase in relative trap stiffness.

the field we had to make a choice of λ and L; we chose $\lambda = 633$ nm and $L = 1$ mm. Figure 5.10 shows the resulting intensity both in the xy (transverse) plane (figure 7(c)) and in the yz (longitudinal) plane (figure 7(d)). It can clearly be seen that the central intensity peak is surrounded by darkness not just in the xy-plane, but also in the yz-plane.

Figure 5.10 also shows the corresponding intensity cross-sections of the narrow-Gaussian, flat-phase, benchmark beam (figures 5.10(a) and (b)). Comparison confirms that the focus of the superresolved beam is narrower in the xy-plane, and that it is significantly narrower in the yz-plane. This is reflected in the 2.2-fold and dramatic 14-fold increases in the relative trap stiffnesses in the transverse (xy) and longitudinal (z) directions, respectively.

5.7 Conclusions

In this chapter we have discussed our theoretical investigation of the application of superresolution holography to optical tweezers. We have found that in our scaled-down model system, this can improve optical tweezers in a number of ways: it can improve the relative stiffness of individual traps, and it can make it possible to bring neighboring traps closer together while maintaining the identities of the individual traps. Unfortunately, these improvements come at the cost of a significant loss of trap depth. We show that by increasing the amount of superresolution we can improve the relative trap stiffness by a factor of 1.7, this was achieved using the direct search algorithm with the narrow Gaussian illumination. We also show that the trap

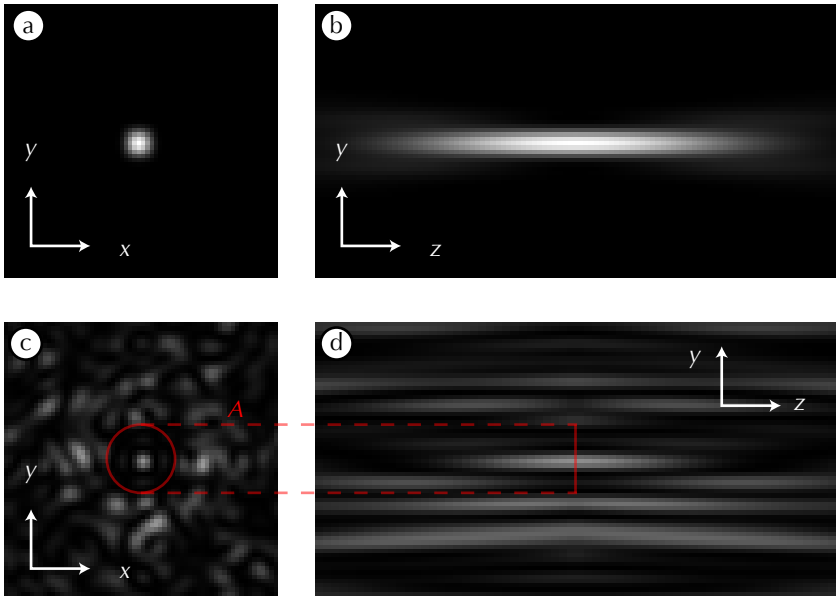


Figure 5.10: Intensity cross-sections in the xy- (left; i.e. (a) and (c)) and yz-planes (right; (b) and (d)) of a focus created without (top; (a) and (b)) and with (bottom; (c) and (d)) superresolution. xy cross-sections represent a 1mm×1mm area, yz cross-sections show an area of 1mm×10mm area (compressed in the z-direction). The non-superresolution beam is the flat-phase benchmark beam (section 6). The superresolved beam was calculated with the DS algorithm with $p = 0.029$ (the target power fraction was $P = 0.01$). Both were calculated for narrow-Gaussian illumination. Propagation of both beams was calculated for $\lambda = 633$ nm and $L = 1$mm. The normalized stiffnesses corresponding to the four frames are $S_{xy} = 0.24$ (a); $S_{xy} = 0.52$ (c), a 2.2-fold increase with respect to (a); $S_z = 0.000\,048$ (b); and $S_z = 0.000\,69$ (d), a 14-fold increase compared to (b).

separation can be improved by 18% by increasing the amount of superresolution.

We tried to make our results as useful as possible by formulating them in terms of improvement factors over benchmark values. However, these improvement factors are not universal: they do, for example, depend significantly on the intensity profile of the beam illuminating the SLM and on the size of the area of interest. The dependence on illumination and area of interest needs to be investigated in more detail.

Our algorithms in their simplest form are not fast enough to be used interactively in optical tweezers with moderate computing power, but they might form part of other algorithms, which we are planning to explore.

We are considering testing our predictions experimentally. We will need to overcome a number of difficulties, including aberrations in the optics, most likely the SLM, which will decrease the trap resolution, and light from the bright, and uninteresting, part of the beam being scattered into the interesting, but dark, part. We hope that it will be possible to overcome the latter problem, certainly in samples that inherently scatter little light. In order to overcome the former problem, we will be able to draw on expertise in our research group for correcting for such aberrations [46].

Chapter 6

Holographic shaping of generalised self-reconstructing light beams

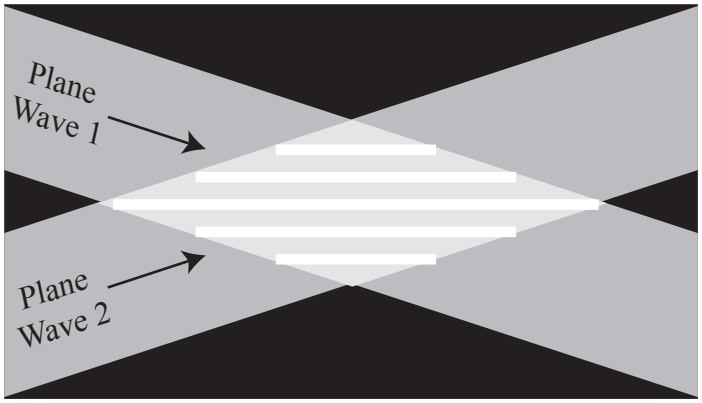
The term self-reconstructing light beam can be used to describe any light beam that, having been disturbed, will on further propagation reconstruct itself. Bessel beams [47] are examples of self-reconstructing light beams [48]. However, it is not well understood that many other light beams can exhibit this self-reconstructing property. In this chapter we present details of a light-shaping algorithm that can be used to arbitrarily shape self-reconstructing light beams, and we show examples of beams that have been shaped using the algorithm. The worked discussed in this chapter was published in this reference [49].

6.1 Self-reconstructing light beams

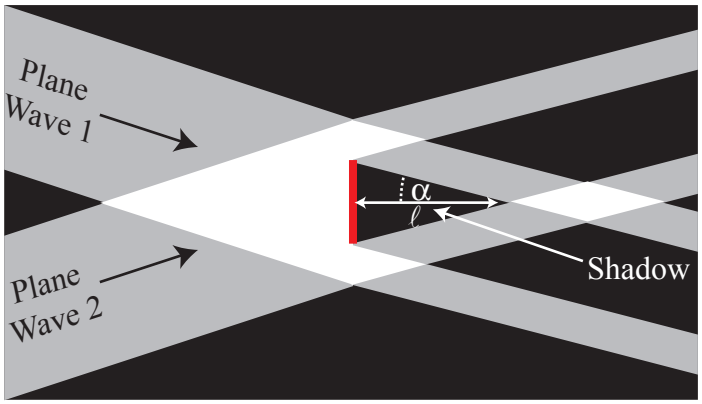
Bessel beams are useful for the optical trapping of 3-dimensional structures [40, 50] due to their self-reconstructing character; on further propagation a Bessel beam will reconstruct itself after any disturbance in the beam caused by a trapped particle.

The self-reconstructing property of Bessel beams is a result of their structure. Bessel beams are formed exclusively by multiple plane waves propagating at an angle to the beam axis. Provided the plane wave spectrum is symmetric in the region where the plane waves overlap a Bessel beam is formed (figure 6.1(a) shows the two-dimensional equivalent of a circularly symmetric Bessel beam). As a result, light that forms the centre of the beam in one transverse plane moves outwards on propagation. Likewise, some of the light that forms the outside of the beam in one transverse plane moves inwards to form the center of the beam on further propagation. Therefore any disturbance to the beam caused by an obstruction in one plane will move outwards in subsequent planes, allowing the beam to be reconstructed behind the obstruction. Figure 6.2 shows a square disturbance to a Bessel beam and the beam after further propagation where the disturbance has moved outwards.

Bessel beams are propagation-invariant; on propagation the beam remains unchanged. One example of a Bessel beam (a zero order Bessel beam, where $m = 0$, like that shown in figure 6.2) has a bright centre contained within concentric dark and bright rings. This bright centre corresponds to a line of brightness along the centre of the beam in the direction of propagation. As the intensity distribution of a beam defines the beam's trapping potential, this would correspond to a long trap which is not usually desirable in optical trapping. It is normally desirable in optical trapping to use an intensity distribution that contains bright points of light (optical traps) located at specific controllable positions in a dark volume. Although it has previously been shown that by using a superposition of Bessel beams it is possible to create trap configurations other than the single long trap (using counter-propagating Bessel



(a)



(b)

Figure 6.1: Two planes waves, shown in grey, travelling at an angle α to the z axis (direction of propagation). The area where the plane waves overlap and the beam is constructed is shown in white. (a) The formation of non-diffracting interference fringes (shown in white) is the two dimensional equivalent to a Bessel beam (formed by multiple plane waves from cylindrically symmetric directions). (b) An obstruction of radius r in the beam creates a shadow of length ℓ after which the beam is reconstructed again.

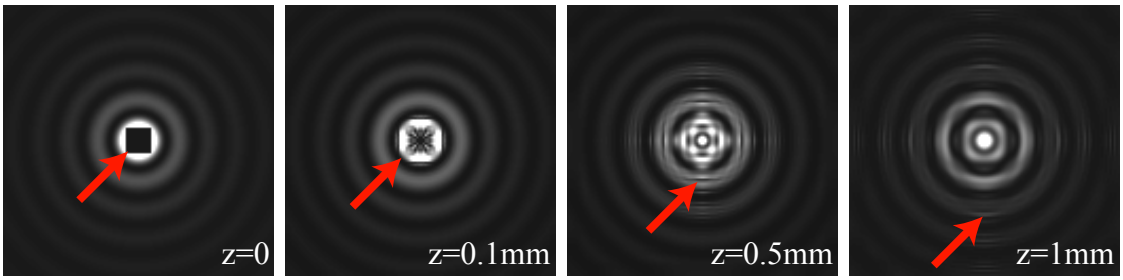


Figure 6.2: The xy cross-sections of the self-reconstruction of a Bessel beam behind a square obstruction. (a) A square obstruction has been placed in the centre of the beam. Frames (b), (c) and (d) show the beam after increasing propagation distances, z. It is evident that the disturbance, highlighted by the arrow, moves outwards after further propagation.

beams, an optical conveyor belt has been created [50] and by interfering more than one Bessel beam, arrays of elongated traps of limited control have been produced [51]) it would be almost impossible to generate any desired intensity distribution normally attributed to optical trapping in this way. However, it is possible to shape such an intensity distribution using appropriate modifications of various algorithms, such as those described in [8, 39, 15]. We chose to modify the Curtis-Koss-Grier algorithm [39] to shape the intensity distribution in a self-reconstructing beam.

6.2 Algorithms for shaping self-reconstructing light beams

6.2.1 Modification made to beam shaping algorithm

We modify the Curtis-Koss-Grier algorithm [39] to shape optical traps that are self-reconstructing. Our modification is not specific to the Curtis-Koss-Grier algorithm (described fully in section 6.2.2) and could be applied to many light shaping algorithms.

As discussed in section 6.1, light beams are self-reconstructing due to the fact that they are composed of plane waves traveling at an angle to the beam axis; there is an absence of light traveling at less than a minimum inclination angle α with respect to the beam axis. As shown in figure 6.1(b) this angle α defines the length of a shadow, ℓ, that will be cast by a trapped particle of a particular radius, r, hence defining the region after which the beam, and specifically any shaped trap, will be reconstructed. The shadow length, particle radius and minimum inclination angle are related through the equation

$$\tan\alpha = \frac{r}{\ell}. \tag{6.1}$$

To create traps that are self-reconstructing we must use only light propagating at an angle greater than a minimum inclination angle α with respect to the propagation direction, or indeed any direction in which the beam is the beam is self-reconstructing. It is the wave numbers of a Fourier space component, k_x and k_y, that determine the direction at which this component will propagate. By creating dark discs of radius

$k_{r,min}$ in Fourier space we are able to both dictate in what direction the beam will be self-reconstructing and define the minimum inclination angle α, which is related to $k_{r,min}$ through the equation

$$\tan\alpha = \frac{k_{r,min}}{\sqrt{k^2 - k_{r,min}^2}}, \tag{6.2}$$

where $k = 2\pi/\lambda$. As is shown in figure 6.3, we can define dark discs in Fourier space as dark discs of radius ρ in the hologram's illumination. The dark disc of radius ρ in the hologram plane is related to the angle α through the equation

$$\tan\alpha = \frac{\rho}{f}, \tag{6.3}$$

where f is the focal length of the Fourier lens (or the effective focal length of any combination of lenses used between the hologram plane and the Fourier plane). For the results shown in the next section we chose to simply centre this disc at the centre of the hologram plane ($(k_x, k_y) = (0, 0)$), meaning that the beams are self-reconstructing in the z direction. By changing the location of this disc we could create beams that would self-reconstruct in other directions. The only modification that is required to be made to a light shaping algorithm is the introduction of this dark disc in the hologram's illumination. As mentioned before, this could be introduced into any algorithm that allows the k-space intensity to be directly specified. We chose to modify the Curtis-Koss-Grier algorithm, an algorithm frequently used in the shaping of optical traps.

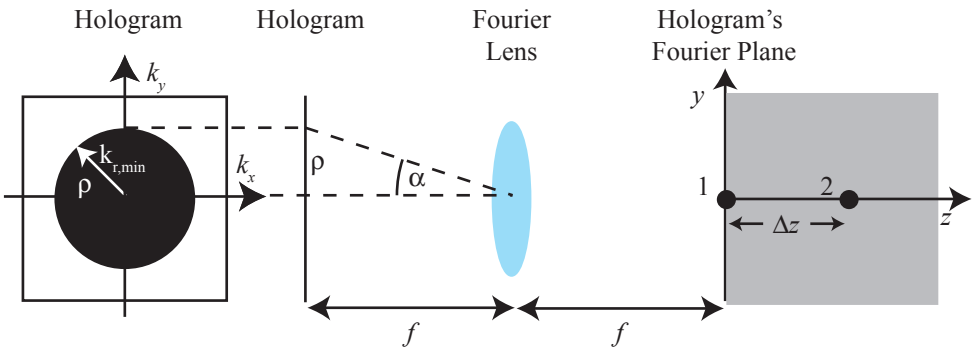


Figure 6.3: Set-up for self-reconstructing Fourier holography. The hologram illumination directly defines the Fourier space structure of the beam: a dark disc of radius ρ in the hologram illumination intensity defines a dark disc of radius $k_{r,min}$ in Fourier space. The radius of the dark disc directly defines the minimum inclination angle α of the beam used to shape two traps (1 and 2) where trap 1 is located in the hologram's Fourier plane given a Fourier lens of focal length f.

6.2.2 The Curtis-Koss-Grier algorithm

The Curtis-Koss-Grier algorithm [39] is a modified version of the Gerchberg-Saxton algorithm [8], described in appendix A, that is optimised to shape multiple optical traps in multiple planes. Unlike the original Gerchberg-Saxton algorithm, which can be used to shape the intensity of a light beam across an entire plane, the Curtis-Koss-Grier algorithm in its simplest form only calculates the optical field at the discrete locations in the trapping volume that correspond to the position of desired optical traps. At all other positions the beam is automatically made dark. This makes the Curtis-Koss-Grier algorithm fast in comparison and hence useful for the shaping of optical traps.

Before describing the algorithm as a whole it is useful to understand the Fourier relationship used to calculate the field at each discrete trap from the field in the

hologram plane, and the corresponding inverse Fourier relationship used to calculate the field in the hologram plane from the field at each discrete trap. First consider a number of traps positioned at the Fourier plane surrounded by complete darkness. We define the centre of the hologram's Fourier plane, which we define to be the origin of a coordinate system $\Delta\vec{r} = (x, y, z)$. The field at the trap position $u_j = a_j \exp(i\phi_j)$, where the trap number $j = 1....N$, is then related to the field $U_\varepsilon = A_\varepsilon \exp(i\Phi_\varepsilon)$ at pixel number ε ($\varepsilon = 1, ..., M$) in the hologram plane, by an inverse Fourier transform

$$U_\varepsilon = \frac{1}{M} \sum_{j=1}^{N} u_j \exp\left(i\overrightarrow{\Delta r_j} \cdot \overrightarrow{k_\varepsilon}\right). \tag{6.4}$$

Here $\overrightarrow{\Delta r_j}$ is the position of the jth trap, $1/M$ is a normalisation factor and $\overrightarrow{k_\varepsilon}$ is the position of the εth pixel in the hologram plane expressed in terms of the Fourier space coordinates, k_x, k_y and k_z. The Fourier space coordinates can be calculated from the real space coordinates in the hologram plane, X and Y. Figure 6.4 shows the relationship between Y and k_y. Both Y and k_y are related to the angle α through the equations $\tan\alpha = Y/f$ and $\sin\alpha = -k_y/k$ respectively, where f is the focal length of the Fourier lens, $k = 2\pi/\lambda$ is the wavenumber where λ is the wavelength of the light and the angle α is defined in figure 6.4. By considering the small angle approximation, Y is therefore related to k_y through the equation

$$k_y = -\frac{2\pi Y}{\lambda f}. \tag{6.5}$$

Similarly,

$$k_x = -\frac{2\pi X}{\lambda f}. \tag{6.6}$$

The z component of the wave vector, k_z, is related to the others through the equation

$$k_z = \sqrt{k^2 - k_x^2 - k_y^2}. \tag{6.7}$$

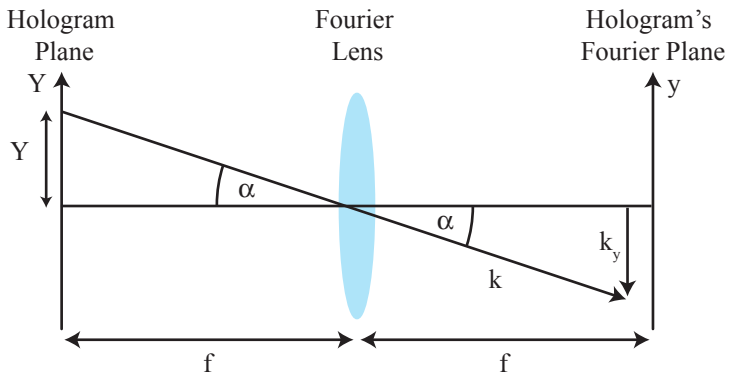


Figure 6.4: The definition of the parameters that can be used to calculate the Fourier space coordinate k_y from the real space coordinate Y_0 in the hologram plane.

If we restrict ourselves to the case of a single trap, $j = 1$, located at the centre of the hologram's Fourier plane, $\overrightarrow{\Delta r_j} = (0, 0, 0)$, then $\overrightarrow{\Delta r_j} \cdot \overrightarrow{k_\varepsilon} = 0$, hence equation (6.4) becomes

$$U_\varepsilon = \frac{1}{M} u_j. \tag{6.8}$$

Considering again our more general case of multiple traps, equation (6.4) can be turned round, to give the field at trap number j positioned at the centre of the hologram's Fourier plane calculated from the field at each element of the hologram plane, U_ε, forming the Fourier transform,

$$u_j = \sum_{\varepsilon=1}^{M} U_\varepsilon \exp\left(-i\overrightarrow{\Delta r} \cdot \overrightarrow{k_\varepsilon}\right). \tag{6.9}$$

Again equation (6.9) can be simplified when considering a single trap $j = 1$ positioned at the centre of the hologram's Fourier plane ($\overrightarrow{\Delta r_j} = (0, 0, 0)$). Then

$$u_j = \sum_{\varepsilon=1}^{M} U_\varepsilon. \tag{6.10}$$

Both equations (6.8) and (6.10) are simplified Fourier transforms that can be used to calculate U_ε and u_j respectively for a case where we aim to find a hologram that gives one trap positioned at the centre of the hologram's Fourier plane. However, it is usual within optical tweezers to have multiple traps in positions such that $\Delta\vec{r}_j \neq 0$ $(j = 1, ..., N)$. To accommodate the shift $\Delta\vec{r}_j$ from the centre of the Fourier plane to the desired trap location we include a kernel in the Fourier transform and inverse Fourier transform. The kernel $K_j(\overrightarrow{\Delta r_j}, \overrightarrow{k_\varepsilon})$ – essentially an additional hologram – shifts the trap field from the actual trap location back into the centre of the Fourier plane. Equation 6.10 then becomes $U_\varepsilon = \frac{1}{M} u_j k_j^{-1}$. For the case of multiple traps located at various locations, the field at each element in the hologram plane U_ε can be calculated from the field at each of the trap locations u_j using the following relationship:

$$U_\varepsilon = \sum_{j=1}^{N} u_j K_j^{-1}(\overrightarrow{\Delta r_j}, \overrightarrow{k_\varepsilon}), \tag{6.11}$$

where λ is the wavelength of the light, f is the focal length of the Fourier lens and $\vec{k_\varepsilon} = (k_x, k_y, k_z)$ is the position of the εth element in the hologram plane expressed in terms of the Fourier space coordinates. The field at trap number j can be calculated from the field at each element of the hologram plane, U_ε, using the following equation:

$$u_j = \sum_{\varepsilon=1}^{M} U_\varepsilon K_j(\overrightarrow{\Delta r_j}, \overrightarrow{k_\varepsilon}). \tag{6.12}$$

As mentioned above, the addition of a kernel to the Fourier transform is the equivalent of experimentally adding an additional hologram to the optical setup. The form of this additional kernel hologram depends on the shift required to return the optical trap to the centre of the Fourier plane. By introducing a lens (or the

phase hologram equivalent of a lens) into an optical system, any foci of light will be shifted in the lateral direction (z), alternatively by introducing a wedge (or the phase hologram of a wedge, a blazed phase grating) into an optical system, any foci of light will be shifted in the transverse direction (x, y) (see figure 6.5). Therefore if the desired optical trap needs to be shifted in the lateral direction to move the trap back to the centre of the Fourier plane, the kernel, $K_j(\overrightarrow{\Delta r_j}, \overrightarrow{k_\varepsilon})$, is calculated as the phase hologram of a lens ,

$$K_j(\overrightarrow{\Delta r_j}, \overrightarrow{k_\varepsilon}) = \exp(ik_z^2 \cdot \Delta z). \tag{6.13}$$

Alternatively we can add the equivalent of a blazed grating to the hologram to shift the desired optical trap in the transverse direction back to the centre of the Fourier plane, achieved by calculating the kernel, $K_j(\overrightarrow{\Delta r_j}, \overrightarrow{k_\varepsilon})$, using the grating equation ,

$$K_j(\overrightarrow{\Delta r_j}, \overrightarrow{k_\varepsilon}) = \exp\left(-i\lambda(k_x \sin\alpha + k_y \sin\beta)\right) \tag{6.14}$$

where $\alpha = -\arctan(\Delta x_j/f)$ and $\beta_j = -\arctan(\Delta y_j/f)$.

The relationships described in equations (6.11) and (6.12) are used in the Curtis-Koss-Grier algorithm in a slightly modified form [39]. As discussed in Appendix A, the constraint that means light shaping problems have to be solved as inverse problems is that the hologram illumination profile A is set. We therefore cannot simply use equation (6.12) to calculate U from our desired trap intensities a_j; this would result in an illumination intensity that was not the hologram illumination profile, A. The algorithm is iterative; the first iteration ($n = 1$) begins by setting a random phase distribution, Φ^1, in the hologram plane. We combine this with the known illumination intensity, A, to give the field $U^1 = A\exp(i\Phi^1)$. In the following

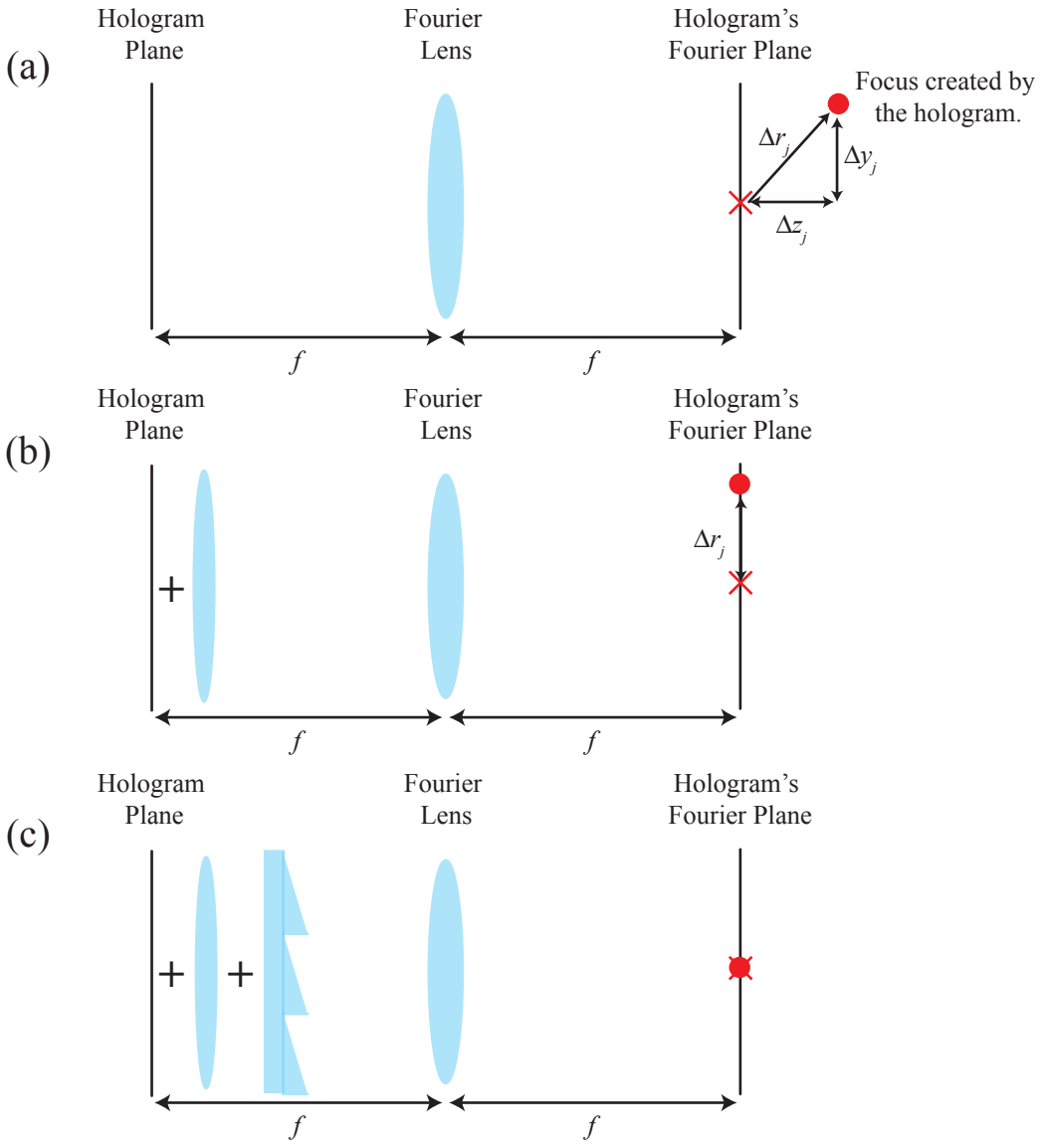


Figure 6.5: The introduction of kernels as lenses or wedges can shift the trap back to the centre of the Fourier plane. The focus created by the hologram is shown by the red circle and the red cross shows the centre of the hologram's Fourier plane. (a) shows the desired position of the trap shifted in x, y and z from the centre of the Fourier plane. The focus can be shifted back to the Fourier plane by the use of a lens (b), and shifted into the centre of the Fourier plane with the addition of a wedge (c). The kernel is exactly the phase hologram of the lens/wedge combination.

iterations (here the nth iteration) we apply U^n to equation (6.12) to give the resulting field at discrete trap number j,

$$\begin{aligned} u_j^n &= \sum_{\varepsilon=1}^{M} A_\varepsilon \exp(i\Phi_\varepsilon^n) K_j(\overrightarrow{\Delta r_j}, \overrightarrow{k_\varepsilon}) \\ &\equiv a_j^n \exp(i\phi_j^n). \end{aligned} \tag{6.15}$$

This is repeated for each of the N discrete traps. Like in the Gerchberg-Saxton algorithm we replace the resulting amplitude a_j^n at each trap with the desired amplitudes a_j and retain the resulting phase ϕ_j^n to give $u_j^n = a_j \exp(i\phi_j^n)$. From this we calculate the resulting field at the hologram plane U_{n+1} using equation 6.11,

$$\begin{aligned} U_{n+1} &= \sum_{j=1}^{N} a_j \exp(i\phi_j^n) K_j^{-1}(\overrightarrow{\Delta r_j}, \overrightarrow{k_\varepsilon}) \\ &\equiv A^{n+1} \exp(i\Phi^{n+1}). \end{aligned} \tag{6.16}$$

Again we retain the resulting phase Φ^{n+1}, which then forms the starting point of iteration $n+1$. The process of Fourier transform – replace intensity – inverse Fourier transform – replace intensity is repeated. Φ^n converges such that the u_j^{n+1}s are a good approximations to the desired values.

The Curtis-Koss-Grier algorithm of course shares many similarities with the Gerchberg-Saxton algorithm, namely the symmetric Fourier transform – replacement – inverse Fourier transform relationship used to bring both algorithms to convergence. The Curtis-Koss-Grier algorithm can be considered a three-dimensional version of the Gerchberg-Saxton algorithm described in Appendix A useful for shaping optical traps over a volume. Both the Curtis-Koss-Grier algorithm and the Gerchberg-Saxton algorithm converge much quicker than the direct search algorithm [15]. However, the direct search algorithm generally converges to a better hologram;

the Curtis-Koss-Grier and Gerchberg-Saxton algorithm with a suitably defined cost function [52] commonly converge to a local rather than a global minimum. Perhaps this could be avoided by some form of noise annealing (chapter 2) .

6.3 Examples of shaped self-reconstructing light beams

To demonstrate the self-reconstructing algorithm we investigate a particularly simple situation: shaping two traps positioned one behind the other with a separation of variable distance Δz as shown in figure 6.3. The first trap contains a completely absorbing disc-shaped particle (in the transverse plane) of variable radius r. We simulate wave-optically the shadow cast by this particle and investigate the effect the shadow has on the formation of the second trap.

Figure 6.6 shows the yz intensity cross sections for two traps separated by a distance $\Delta z = 8.25\mu$m. Frames 1,2 and 3 are calculated for varying radius r of the particle in trap 1. Frame 1 shows the beam with particle of radius $r = 0$ (i.e. no particle in trap 1); trap 2 in this case is obviously fully formed. Frame 2 shows the same beam, but now with a particle of radius $r = 3\mu$m positioned at trap 1. The second trap is fully reconstructed because it is not positioned in the shadow cast by the trapped particle, i.e. the length of the shadow ℓ cast by the particle is less than the trap separation Δz. Frame 3 shows the beam with a trapped particle of radius $r = 9\mu$m. Now the shadow cast by the trapped particle is longer than the trap separation Δz, so the second trap is not reconstructed.

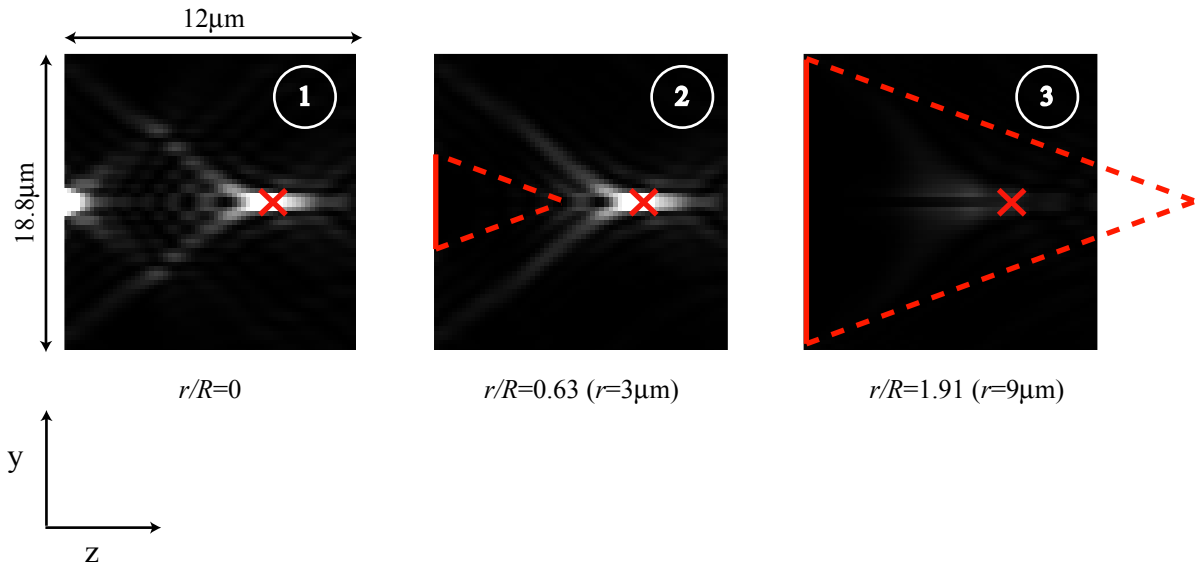


Figure 6.6: yz intensity cross-sections showing the self-reconstruction of beams behind obstructions of increasing size, all shown for the same trap separation Δz and different obstruction sizes. The solid red line indicates the size of the obstruction which casts a shadow shown by the dashed red line; the cross marks the position of trap 2 in the undisturbed beam. Cross-section 1 contains no obstruction in first trap. Cross-section 2 is produced by simulating the presence of an obstruction in the first trap of radius r less than the critical radius R. Cross-section 3 was produced by simulating an obstruction in the first trap of $r > R$. Like all the other results shown in this section, this figure was calculated for a hologram of size 20mm×20mm which was uniformly illuminated outside the dark disc of radius $\rho = 12.5$mm. We simulated this on a grid of size 256×256 for light of wavelength $\lambda = 633$nm; the effective focal length was $f = 11$mm. In trap 2 of the figure the trapping geometry is such that the second trap is reconstructed despite the presence of the particle in the first trap.

We call the particle radius for which the length of the particle's shadow equals the trap separation Δz the critical radius R. For a trapped particle of radius $r > R$, trap 2 is not reconstructed as a result of the shadow that is cast (frame 3 of figure 6.6); for a trapped particle of radius $r < R$, trap 2 is reconstructed as the shadow length ℓ is shorter than Δz (frame 2 of figure 6.6). By insertion into equation 6.1, the critical particle radius R for the given trap separation Δz comes out to be

$$\tan \alpha = \frac{R}{\Delta z}. \tag{6.17}$$

This is investigated in more detail in figure 6.7, which shows the intensity at the position of trap 2, I_2, plotted as a function of r/R for constant values of Δz and α. As the particle radius is increased from 0 to the critical radius ($0 < r/R < 1$), the intensity in trap 2 I_2 remains approximately constant; the shadow cast by the trapped particle is shorter than the trap separation Δz, hence the second trap is fully reconstructed. As r/R is increased further ($r/R > 1$), the intensity in trap 2 begins to decrease, ultimately to 0; the trapped particle casts an increasing shadow over the second trap, hence the second trap is not fully reconstructed.

Figure 6.8 shows yz intensity cross sections for a fixed particle of radius $r = 3\mu$m with varying trap separation Δz. The red cross shown marks the location of the second trap given that no particle is present in the first trap. Frame 4 in figure 6.8 shows a cross section in which the second trap is not reconstructed; the length of the shadow cast ℓ is greater than the trap separation Δz. Frame 6 shows a cross section where the shadow cast by the trapped particle is less than the trap separation Δz, hence the second trap is fully reconstructed.

Now there is a critical trap separation Z equal to the length of the shadow cast

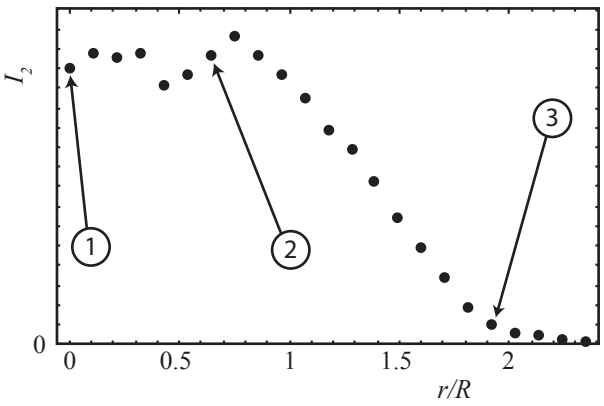


Figure 6.7: Intensity in the second trap, I_2, as the radius of the obstruction, r, is increased from 0 to show that for obstructions of r less than the critical radius R the second trap will reconstruct fully given the presence of the obstruction. This is shown by the almost constant value of I_2 until $r = R$. For values of $r > R$ the peak intensity decreases. The yz cross-sections corresponding to points 1,2 and 3 are shown in figure 6.6.

by the trapped particle, given by

$$\tan \alpha = \frac{r}{Z}. \tag{6.18}$$

If the trap separation Δz is less than the critical trap separation Z, trap 2 is not reconstructed (frame 4 of figure 6.8); if it is greater than the critical trap separation Z, trap 2 is outside the shadow and therefore reconstructed (frame 6 of figure 6.8). In the graph of the intensity in trap 2, I_2, as a function of $\Delta z/Z$ (figure 6.9), these two cases show up respectively as a darkened intensity ($\Delta z/Z < 1$) and as full intensity ($\Delta z/Z > 1$).

Note that for the graph in figure 6.9 a new hologram had to be calculated for each data point. As the calculation of the hologram involves random processes, this resulted in a less smooth curve than the one shown in figure 6.7, which was calculated for the same hologram.

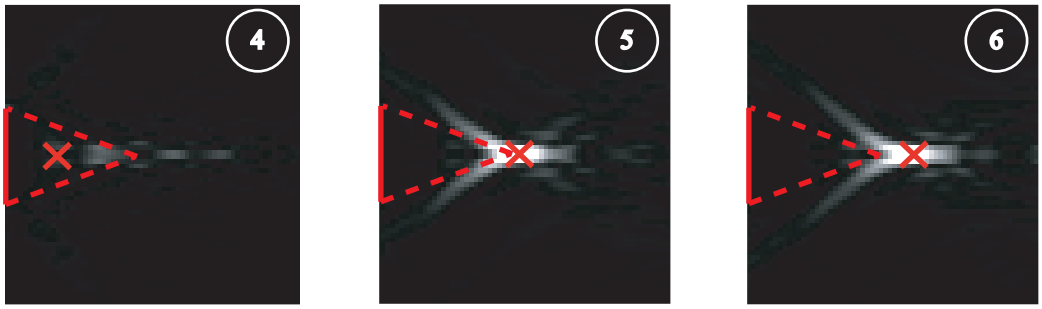


Figure 6.8: yz intensity cross-sections showing the self-reconstruction of beams behind an obstruction of radius $r = 3\mu m$. Each frame shows the cross section for a different trap separations Δz. The red line indicates the size of the obstruction which casts a shadow shown by the red dotted line. Cross-section 4 is shown for Δz less than the critical trap separation Z. Cross-section 5 is shown for $\Delta z \approx Z$ and cross section 6 is shown for $\Delta z > Z$. Parameters are the same as those used in figure 6.6.

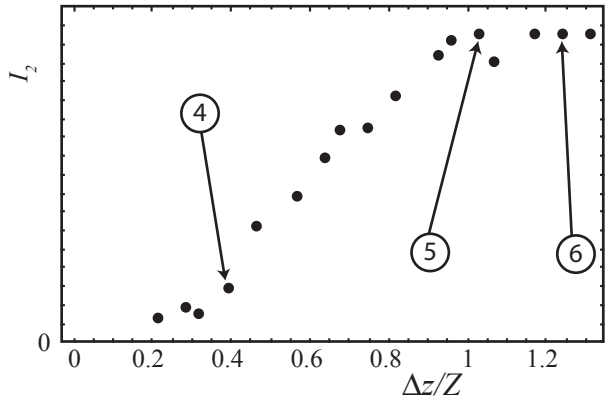


Figure 6.9: Intensity in the second trap, I_2 as the trap separation Δz is increased for a fixed obstruction radius $r = 3\mu m$. As Δz is increased from the critical separation Z the peak intensity in the second trap remains constant showing that for $\Delta z > Z$ the second trap reconstructs given the presence of the obstruction. Figure 6.8 shows the yz cross-sections corresponding to points 4,5 and 6. Parameters like in figure 6.6.

6.4 Light beams that self-reconstruct in directions other than the propagation direction

Here we describe the potential to extend the concept of the generalized self-reconstructing algorithm to shape light beams that may be useful for the trapping of photonic crystal structures [53].

It is possible to create light beams that are self-reconstructing in directions other than the direction of propagation and in multiple directions. This is achieved by specifying the correct dark disc in the hologram illumination. The beam will be self-reconstructing in the direction around which its k-space representation is dark. The length of the shadow cast by the trapped particle is defined by the radius of the dark disc. By creating a hologram illumination that contains multiple dark discs centered on various locations we can then create light beams that are self-reconstructing in different directions.

6.5 Conclusions

We have demonstrated that it is possible to make a simple modification to a light shaping algorithm so that it can then be used to shape a light beam that is self-reconstructing. We explain generalized self-reconstructing light beams using a simple ray-optical argument. By wave-optically simulating a completely absorbing trapped particle in our shaped self-reconstructing beam we are able to demonstrate its self-reconstructing properties. Any introduction of a reflecting or refracting particle,

which could lead to optical binding [54], would, of course make our simulations more complicated. This could form the basis of future work.

Chapter 7

Evanescent wave shaping

The size of bright structures in travelling-wave light fields is limited by diffraction. This in turn limits a number of technologies, for example optical trapping. One way to beat the diffraction limit is to use evanescent waves instead of travelling waves. In this chapter we discuss the application of light shaping algorithms to shape the three dimensional intensity distribution of evanescent waves. We simulate 3D intensity shaping of evanescent-wave fields using this approach, and we investigate some of its limitations. The work discussed in this chapter was published in reference [55].

7.1 Introduction

Optical tweezers trap microscopic particles in the intensity maxima or minima of a light beam. The smallest possible size of a maximum or minimum and the proximity of multiple maxima or minima to each other is limited by the diffraction limit. This therefore places a limit on the optical trapping resolution. We previously addressed beating the diffraction limit in chapter 5, where we applied light shaping algorithms

produce super-oscillations. Another way to beat the diffraction limit and achieve superresolution is to use superpositions of evanescent waves to form optical traps.

The transverse wavelength of evanescent waves can be significantly shorter than that of a traveling wave, with the same frequency. This results in the ability to beat the diffraction limit. Superpositions of small numbers of evanescent waves have previously been used for optical trapping [56] and sorting [57].

In this chapter we describe a method of using a light shaping algorithm, direct search algorithm [15], to shape the intensity distribution of an arbitrary number of evanescent wave superpositions. We demonstrate this numerically and show that this approach can be used to shape evanescent wave superpositions in three dimensions. We apply the algorithm to shape intensity maxima in evanescent waves at desired three dimensional locations. The shaping of such evanescent wave fields in this way is applicable to optical trapping.

In the following sections we discuss the creation of superpositions of evanescent waves, the application of the direct search algorithm to evanescent wave shaping and present our results.

7.2 Creation of superpositions of evanescent waves

We have previously discussed the use of the direct search algorithm in chapter 5. For our application of the direct-search algorithm to evanescent-wave shaping, the algorithm is written with a specific experimental design in mind (but it can be applied more generally). In this section we discuss two candidate designs for the creation of

superpositions of evanescent plane waves.

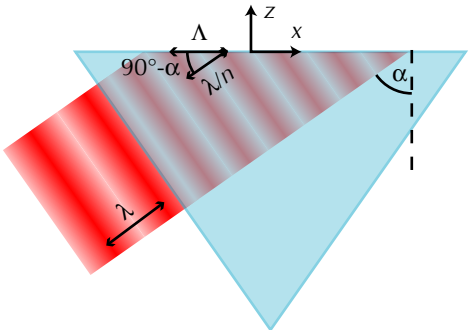


Figure 7.1: Geometry of the creation of evanescent waves by total internal reflection of a prism surface. A plane wave of vacuum wavelength λ is incident on a prism surface with angle of incidence α. The transverse wavelength along the prism surface, Λ, is given by the equation $\Lambda \sin(\alpha) = \lambda/n$.

Figure 7.1 shows the geometry of a plane wave entering a glass prism and hitting its top side, where total internal reflection (TIR) is taking place. During TIR evanescent waves are produced on the surface of the top side of the prism.

The wavelength outside the prism (in air) is λ, that inside the prism glass of refractive index n is λ/n. We choose our coordinate system such that the TIR surface is in the $z = 0$ plane and that the plane wave in the prism is travelling perpendicular to the y direction. Above the prism, the evanescent waves have the form of plane waves, namely

$$u(x, y, z) = \exp(i(k_x x + k_z z)), \tag{7.1}$$

where $k_x^2 + k_z^2 = (2\pi/\lambda)^2$ (in our coordinate system $k_y = 0$). However, in evanescent waves k_z is purely imaginary, and the wave therefore decays exponentially in the z direction.

The smallest transverse wavelength that can be achieved in evanescent waves using TIR as described above is λ/n, which corresponds to grazing incidence at the TIR surface. Small transverse structures can be achieved with small transverse wavelengths, which in turn can be realized using high-n glass. One example of a suitable glass is Schott SF11, which has a refractive index of $n > 1.7$ for a wide range of wavelengths, and which is relatively cheap as it is frequently used in femtosecond-laser correction.

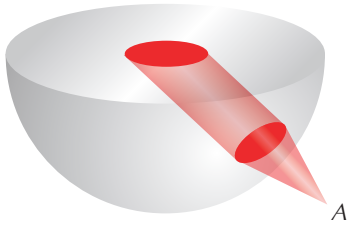


Figure 7.2: Glass hemisphere configuration creating an evanescent wave field in the centre of the flat surface. Illuminating the sphere with a point light source, A, which can be for example the end of a single-mode fiber, can create a collimated beam inside the glass, which in turn can create an evanescent plane wave above the centre of the hemisphere's flat side. Illuminating with two or more coherent point light sources at different positions can create a superposition of evanescent waves.

An alternative design uses a glass hemisphere in place of a prism (figure 7.2). This is a more versatile design as the hemisphere can be illuminated from any direction ($k_y \neq 0$) and with any value of ($k_x \neq 0$). Such versatility makes this design more suitable for use with the direct search algorithm. The hemisphere's flat side acts as the TIR surface. We propose to illuminate the hemisphere with a number of fiber ends, each effectively acting as a point light source. If a fiber end is positioned at the correct distance from the glass hemisphere, the curvature of the glass collimates the beam from the fiber inside the glass. The correct distance can be calculated from the

lens maker's formula, which states that a spherical glass surface of refractive index n and radius of curvature r has a focal length

$$f = \frac{r}{n-1}. \tag{7.2}$$

Therefore each fiber end has to be positioned this distance f from the spherical surface to create a plane wave inside the glass hemisphere.

We limit the setup such that all of the evanescent planes waves have the same global polarization, so they will interfere. The different plane waves inside the glass hemisphere are then incident on the TIR surface, the hemisphere's flat side, from different directions. This creates different evanescent waves on the other side of the TIR surface. By varying each fibres illumination we can create varying desired interference patterns.

The illumination fibres could be gathered together in a bundle and illuminated using a phase-only spatial light modulator (SLM). The SLM is imaged onto the end of the fibre bundle - each SLM pixel addressing a fibre end. The phase at both ends of the fibres can therefore be controlled directly. The intensity at the input fibre ends, and therefore also the intensity of the corresponding evanescent plane-wave components, is given by the intensity distribution of the illuminating laser beam.

7.3 Application of the direct search algorithm to evanescent wave shaping

With a view to potential applications in optical trapping, we aim to create superpositions of N evanescent plane-wave components that have bright spots at M trap

positions with coordinates (x_j, y_j, z_j) $(j = 1, ..., M)$. We restrict ourselves to changing the phases of the evanescent plane waves, all of which are of the same intensity (generalization to different intensities is straightforward; a traveling-wave example can be found in [15]). We characterize the ith plane-wave component by its transverse wave-number values, $k_{x,i}$ and $k_{y,i}$. The corresponding plane wave can then be written in the form

$$u_i(x, y, z) = \exp(i(k_{x,i}x + k_{y,i}y + k_{z,i}z)) \tag{7.3}$$

$$= \exp(i(k_{x,i}x + k_{y,i}y)) \exp(-\beta_i z), \tag{7.4}$$

where

$$k_{z,i} = \sqrt{(2\pi/\lambda)^2 - k_{x,i}^2 - k_{y,i}^2} \tag{7.5}$$

and $\beta_i = -ik_{z,i}$ describes the exponential decay in the z direction. We then want to optimize the phases ϕ_i of the plane-wave amplitudes in the superposition

$$v(x, y, z) = \sum_{i=1}^{N} \exp(i\phi_i) u_i(x, y, z) \tag{7.6}$$

such that the intensities at the trap positions (x_j, y_j, z_j),

$$I_j = |v(x_j, y_j, z_j)|^2, \tag{7.7}$$

are as bright and as equal as possible.

We again utilise the direct search algorithm [15] as a brute-force approach to solving such "inverse" problems [58]. Initially, we set the phases of all beam amplitudes to random values. We then calculate the quality of the resulting superposition in terms of the intensity at the trap positions according to the definition

$$Q = \sum_j \ln(I_j + 1). \tag{7.8}$$

This cost function is chosen as it has two properties important to solving this problem: 1) Increasing any one of the intensities I_j increases Q. 2) Transferring intensity from a brighter to a less bright trap position also increases Q. Other cost functions could be chosen for the same reason, however comparison of cost function was not the focus of this work but could be the basis of later work. Now we make a random change to the superposition: we pick a random component and change the phase of its amplitude a_i to a random value. We calculate the quality of the altered superposition, again according to the definition (7.8), but calling this new quality Q'. If the new superposition is better than the old one, that is if $Q' > Q$, we accept the random change; otherwise we discard it. We repeat the process of randomly picking one component, randomly altering its phase, and keeping only changes that lead to improvements until the quality does not change any longer for several hundred iterations.

7.4 Results

We have tested this approach to evanescent-wave shaping for various target distributions. The results we present in this section show that the approach to evanescent wave shaping we have taken works, but they also demonstrate its limits.

The simulations in this section were performed for specific choices of additional parameters. All our simulations were performed for $\lambda = 632.8$nm. We also made a choice on the specific arrangement of optical fibers around the glass sphere: our model represents an arrangement of optical fibers that are equally spaced in their

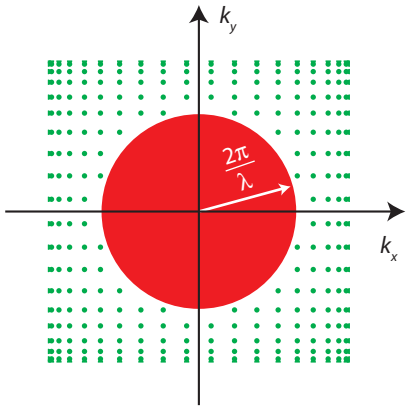


Figure 7.3: Example of the transverse wave numbers k-space distribution of evanescent plane-wave components used in our simulation. The dots represent evanescent-plane-wave components. They all lie outside the shaded area in the centre, which represents travelling waves.

projection angles into the xz and yz planes, whereby travelling-wave components are not allowed. An example of the corresponding k-space distribution is shown in figure 7.3.

Figure 7.4 demonstrates shaping of the transverse intensity distribution of the evanescent-wave field immediately above the hemisphere's surface ($z = 0$). It also demonstrates that the intensity features can be sub-wavelength in size.

The algorithm can also shape the transverse intensity in planes a finite distance above the hemisphere's surface. Figure 7.5 shows the result of optimization of the intensity at two points a finite distance above the prism surface. Figure 7.5(a) shows the transverse (x, y) intensity cross-section away from the prism surface. The corresponding (x, z) cross-section (figure 7.5(b)) demonstrates that, unlike in similar algorithms used for shaping travelling waves, simply optimising the intensity at a

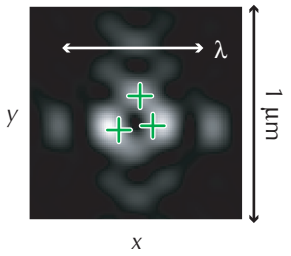


Figure 7.4: Shaping of the transverse intensity distribution of an evanescent-wave superposition, here directly above the prism surface ($z = 0$). "+" symbols indicate positions where the intensity was maximized. The k-space components are those shown in Fig. 7.3. The shaped intensity features are sub-wavelength in size.

number of points does not produce 3D intensity maxima.

Figure 7.6 shows the results of attempting to force the creation of 3D intensity maxima. In addition to trap positions where the intensity is maximised, we modified the algorithm to minimise the intensity on a set of points on spheres around the maximum-intensity trap positions, simply by extending the definition of Q to include a term such that an increase of the intensity at any one of L "no-trap" positions (x_j, y_j, z_j), where $j = M + 1...M + L$, leads to a *decrease* of the quality:

$$Q = \sum_{j=1}^{M} \ln(I_j + 1) - \sum_{j=M+1}^{M+L} \ln(I_j + 1). \tag{7.9}$$

Interestingly, we did not succeed in creating 3D intensity maxima. This might well be due to the fact that we restrict ourselves to superpositions of evanescent plane waves that all have the same intensity. In [55] it is suggested that it may also be – fully or in part – due to a more fundamental reason related to Earnshaw's theorem.

Finally, we address the commonly held belief that the intensity of evanescent-wave fields can only ever fall off in the longitudinal direction. Figure 7.7 demonstrates that,

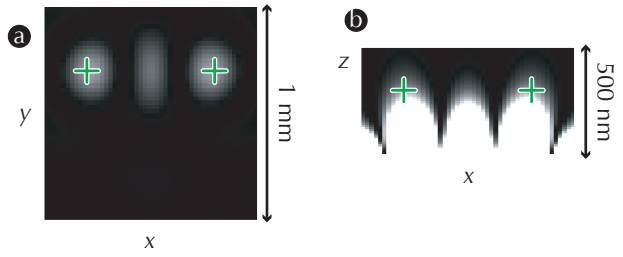


Figure 7.5: (Color online) Transverse (a) and longitudinal (b) intensity cross-section after optimization of the intensity at two trap positions ("+" signs). The trap positions are 300nm above the prism surface. The k-space distribution is that shown in Fig. 7.3. Unlike in similar algorithms used to shape travelling waves, (b) shows that simply optimising the intensity at a number of points does not produce 3D intensity maxima.

along a line in the transverse (z) direction, the intensity can not only rise in the z direction, but it can also go through one or more maxima. (Note, however, that in 3D all of the intensity maxima along the line in the z direction are saddle points.) figure 7.8 explains how an out-of-phase superposition of exponential functions (such as amplitudes of different evanescent waves) can have a maximum; superpositions of more exponentials can have more maxima.

7.5 Conclusions

In this work, we have investigated the potential use of evanescent-wave holography for optical micromanipulation. Our results are useful not only for showing that evanescent-wave intensities can be shaped in 3D, but also for investigating the limitations of holographic evanescent-wave shaping.

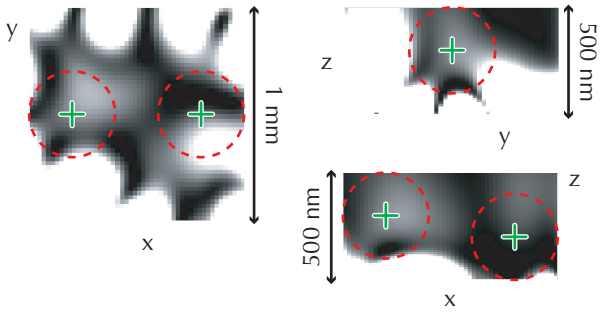


Figure 7.6: (Color online) Intensity resulting from an attempt to force the creation of two 3D intensity maxima. The algorithm tries to maximize the intensity at the trap positions, which are indicated by "+" signs (green in the online version); it tries to minimize the intensity at the points marked with "-" signs (red in the online version). k-space distribution as in Fig. 7.3. 3D intensity maxima are not successfully shaped.

We have shown in figure 7.4 that we can successfully shape evanescent-wave fields that have desired sub-wavelength features. We also demonstrate, figure 7.7 that in the transverse direction (as we move away from the surface of the prism) the evanescent-wave field can go through multiple intensity maxima. Showing that evanescent-wave intensity maxima can be shaped away from the surface of the prism.

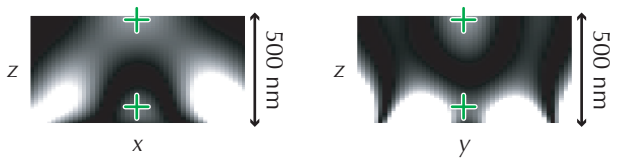


Figure 7.7: (Color online) Demonstration of intensity maxima away from the prism interface. The intensity was maximized in two points, each marked with a "+", on the line $x = y = 0$. The intensity is shown in two planes that both include the line: $y = 0$ (a), and $x = 0$ (b). What appears to be a 3D maximum in (a) turns out to be a saddle point in (b). The intensity was calculated for a k-space distribution different from that shown in Fig. 7.3.

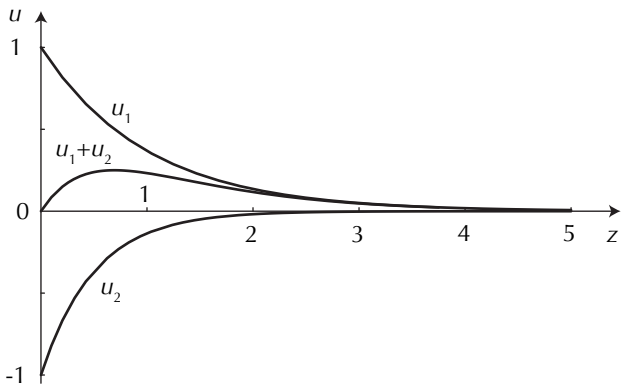


Figure 7.8: Local maximum in the field, u, sum of two evanescent waves, $u_1(z) = \exp(-z)$ and $u_2(z) = -\exp(-2z)$ where z is the propagation distance (dimensionless units).

Chapter 8

Conclusions

Rather than repeat the conclusions made at the end of each previous chapter this chapter simply makes some overall concluding comments.

In the first half of this thesis we discussed the use of inverse techniques to locate gas sources from downwind concentration measurements. This is done with an application in oil prospecting in mind. The software resulting from this work, Recon, is currently used by the LightTouch team at Shell Global Solutions.

We have demonstrated that when solving inverse problems in this way the choice of cost function has a large bearing on the result. The results in chapter 4 suggest that the choice of cost function has greater bearing on the results than choice of forward models we compare. This is due to the fact that changes to the gas source distribution, for example, are accepted or discarded based solely on the an improved value of the cost function. In a mathematical sense it is essentially the cost function that determines the underlying characteristic of the source distribution. For example, cost functions can be defined to favour sources that are smoothly distributed or highly localised. This choice is made on the basis of experience gained from knowledge of

the problem itself. I believe there is no universal cost function suited to solving all inverse problems. Rather each problem requires the cost function to be chosen with care and knowledge of the problem itself.

The fact that Shell Global Solutions continue to pursue this technology is proof that it is a viable solution to gas location problems. This approach to solving inverse problems can be computationally intense. With a desktop PC running Recon with one of our typical gas measurement data sets can take between many minutes and many hours to converge. The time to converge is dependent on the amount of annealing, the number of measurements made and the number of potential sources. Recent developments in computing technologies should make this type of approach viable for solving this problem and others. Programmable graphics cards are one such computing development which in cases have reportedly improved processing time by 100 to 1000 times.

The holographic light shaping problems, we discussed in the second half of this thesis, could also benefit from being implemented with graphics cards. Such an implementation would make more rigorous and varied algorithms available to real time holographic light shaping. We could therefore foresee the realisation of the suggested experimental setup discussed in chapter 7 to manipulate desired three dimensional evanescent fields in real time.

The work discussed in chapter 7 demonstrates that evanescent fields can be shaped in three dimensions and suggests an experimental setup to realise this. Further to this we show that multiple evanescent intensity maxima can be shaped in the transverse direction from the TIR surface.

Using evanescent fields to make optical traps is one suggested way to improve trap resolution. Another approach which we have described in this thesis utilises superresolution and is described in chapter 5. In this work, we show numerically, that by shaping only a small part of the beam we can create superresolution optical traps which have an improved relative trap stiffness and trap resolution. However, both improvements come at a cost to trap strength.

Another light shaping problem we have described in this thesis, in chapter 6, is shaping self-reconstructing beams. We have demonstrated that it is possible to modify a light shaping algorithm to shape self-reconstructing light beams. We shape a self-reconstructing beam which contains multiple optical traps along the direction of propagation. We demonstrate it's self-reconstructing properties by numerically simulating an absorbing particle in the first trap and show that on propagation the beam reconstructs with the remaining traps preserved.

Appendix A

Gerchberg-Saxton Algorithm

The Gerchberg-Saxton Algorithm [8] is used to calculate the phase hologram that can be used to produce a desired intensity of a light field for a given hologram illumination. The Gerchberg-Saxton Algorithm is based on the Fourier relationship between the hologram plane and the Fourier plane as shown in figure A.1.

We set the intensity in the hologram plane to be the known illumination intensity, $I_H(x_i, y_j)$, and choose an arbitrary distribution for the initial phase in the hologram plane, $\phi_H^n(x_i, y_j)$. The initial complex field in the hologram plane, $U_H^n(x_i, y_j) = I_H(x_i, y_j)\exp(i\phi_H^n(x_i, y_j))$, is Fourier transformed to give the field in the Fourier plane, $U_F^n(x_i, y_j) = I_F^n(x_i, y_j)\exp(i\phi_F^n(x_i, y_j))$. As we are interested only in shaping the intensity we leave the phase in the Fourier plane, $\phi_F^n(x_i, y_j)$, unchanged and replace the intensity with the desired intensity in the Fourier plane, $I_F(x_i, y_j)$. The replaced field in the Fourier plane, $U_F^n(x_i, y_j) = I_F\exp(i\phi_F^n(x_i, y_j))$, is inverse Fourier transformed to give the new field in the hologram plane, $U_H^{n+1}(x_i, y_j) = I_H^{n+1}(x_i, y_j)\exp(i\phi_H^{n+1}(x_i, y_j))$. The intensity at $U_H^{n+1}(x_i, y_j)$ is replaced by the illumination intensity and the phase again remains unchanged. The replaced $U_H^{n+1}(x_i, y_j) =$

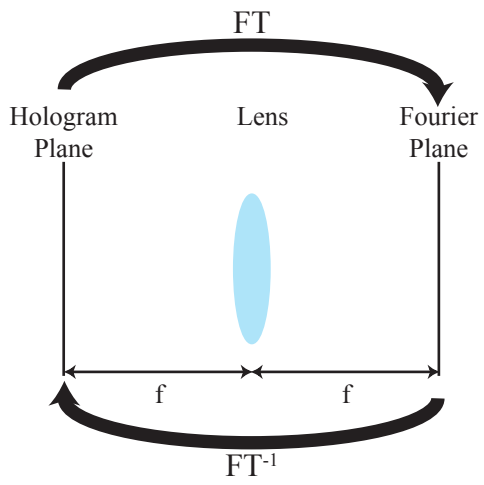


Figure A.1: The Fourier relationship between hologram and Fourier plane.

$I_H(x_i, y_j) \exp(i\phi_H^{n+1}(x_i, y_j))$ is again Fourier transformed to give the new field in the Fourier plane $U_F^{n+1}(x_i, y_j)$ and the process of replacement inverse Fourier transform replacement Fourier transform is repeated until the algorithm converges. The algorithm has converged when the Fourier transform of the replaced field in the hologram plane, $U_H^{n+j}(x_i, y_j) = I_H(x_i, y_j) \exp(i\phi_H^{n+j}(x_i, y_j))$, gives the best approximation to the desired intensity $I_F(x_i, y_j)$ and no further iterations of the algorithm make any improvement. Figure A.2 shows a schematic of the Gerchberg-Saxton algorithm.

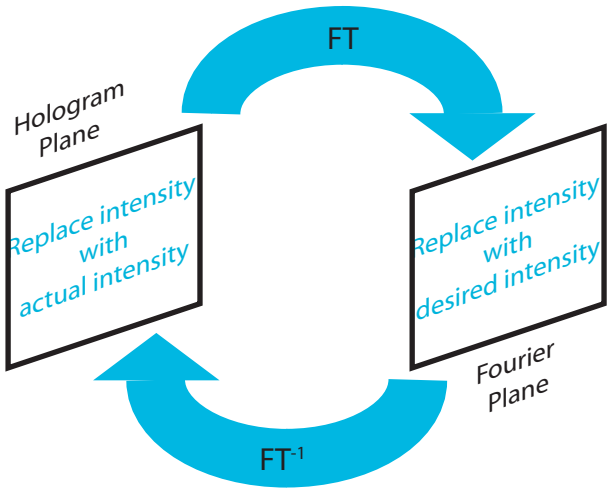


Figure A.2: The Gerchberg-Saxton algorithm.

Appendix B

Recon

Recon is the software, written in Labview, that we have written to locate gas sources. The software utilises the inverse techniques discussed in chapters 2, 3 and 4.

Figure B.1 shows a flow chart detailing the main inputs, outputs and processes completed by Recon. The initial stage in the process, to process the measured data, can be used to remove the background gas concentration and average the measurement data in the time domain. The next stage in the process calculates the forward model, which in this case is a gas dispersion model, resulting in a matrix, A, which is used during the iterations to calculate the downwind gas concentration measurements resulting from the trialed source distribution. The algorithm itself follows and is detailed in the flow chart shown in figure B.1.

Screen shots from Recon have been taken which show the user interface. Figure B.3 shows the section of the program which calculates the forward model. In the case shown it is the Gaussian plume model which is being used, however, one of others can be selected here by changing the tab.

Figure B.4 shows the interface of the section of the program where the iterations

of the algorithm are carried out. In this case it is a χ^2 cost function with no annealing that is being used, again this can be changed by changing the appropriate tab.

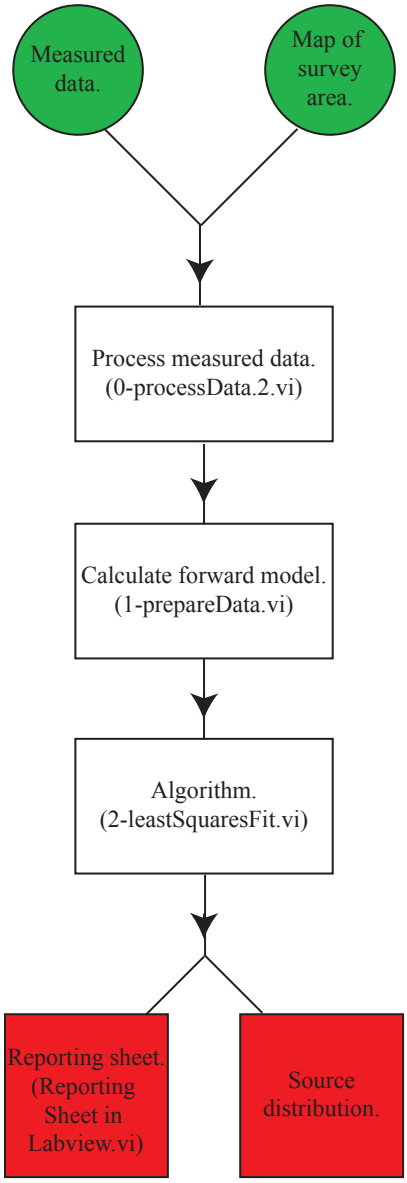


Figure B.1: A flow diagram highlighting the input, output and main processes completed by Recon.

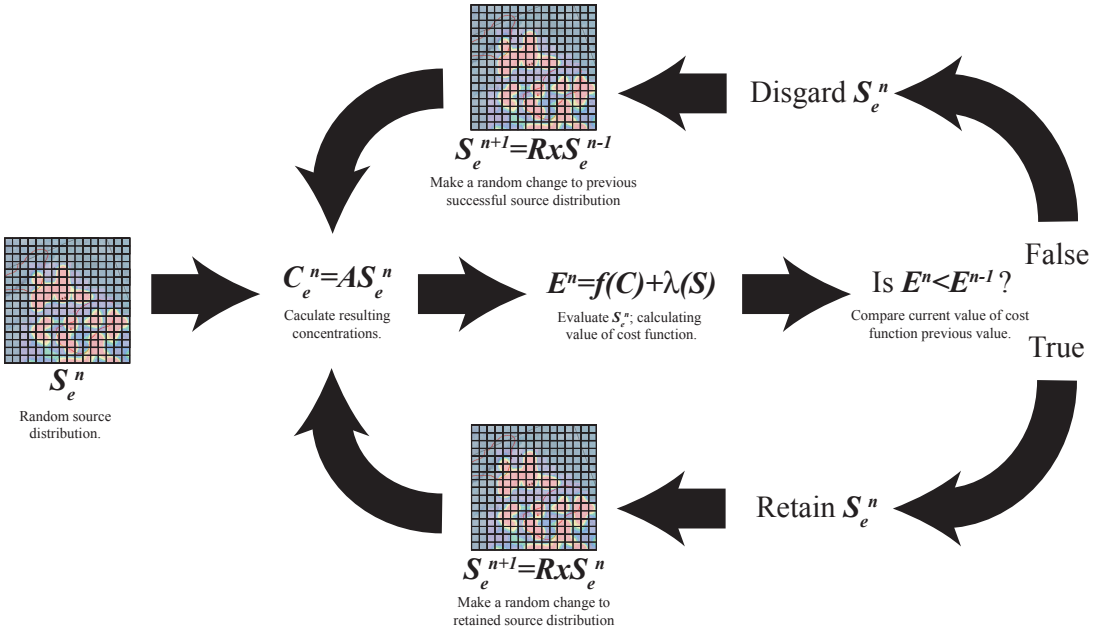


Figure B.2: The algorithm.

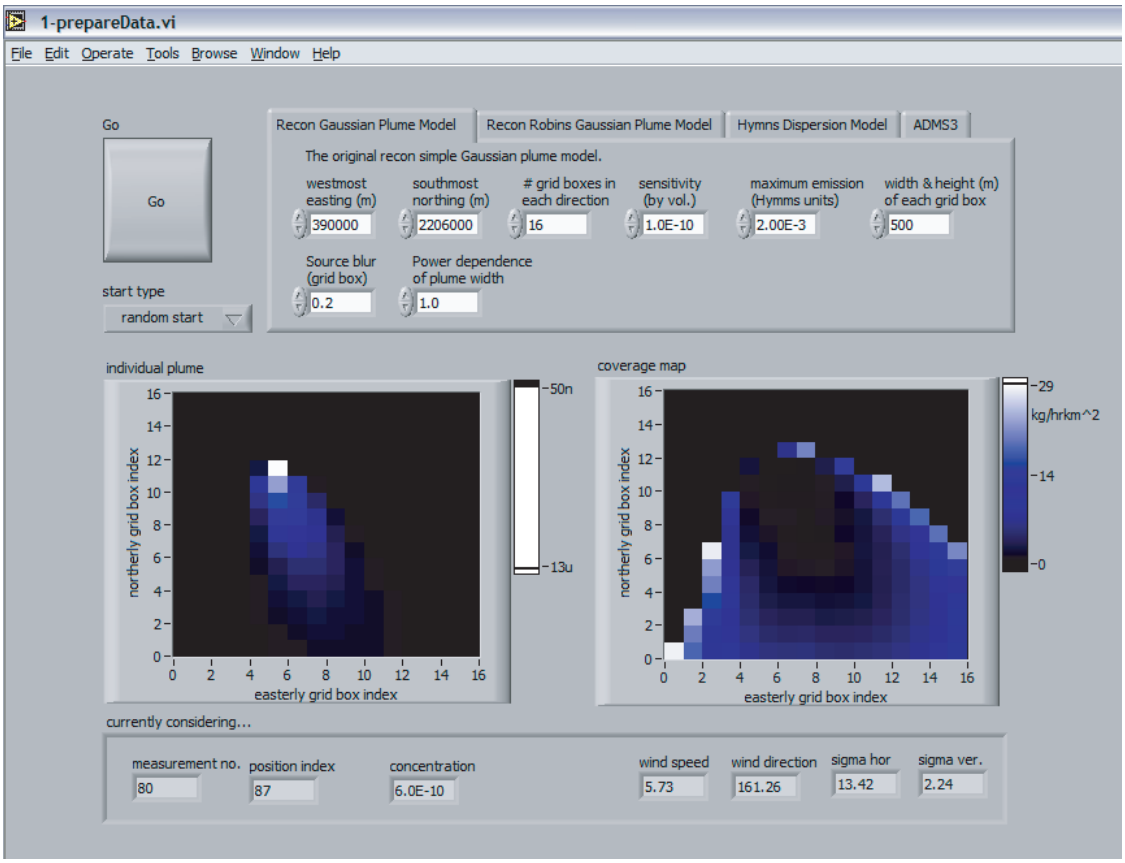


Figure B.3: A screen shot taken from Recon showing the forward model being calculated.

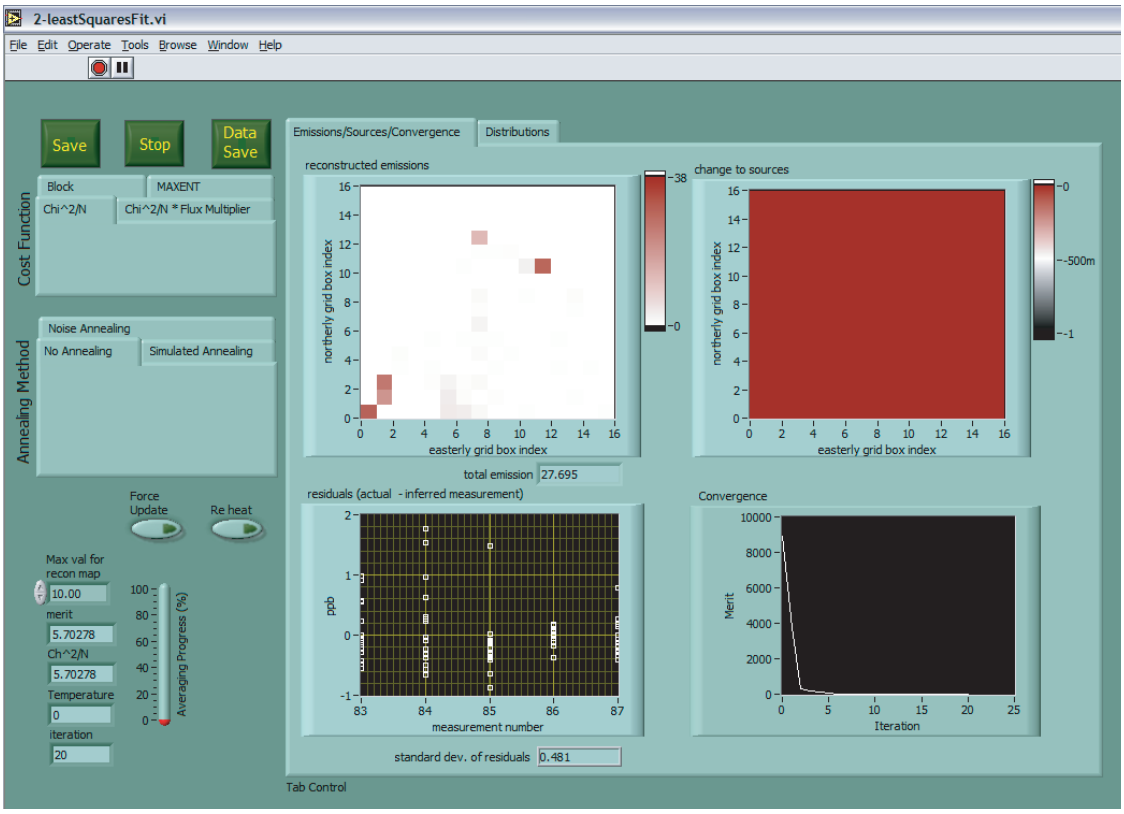


Figure B.4: A screen shot taken from Recon showing the progressing iterations.

Appendix C

Direct Binary Search Algorithm

The direct binary search algorithm [15] can be used to shape a desired light field; both phase and intensity if desired. The algorithm aims to find the phase hologram that will give the best approximation to the desired intensity $I_{F_D}(x_i, y_j)$, desired phase $\phi_{F_D}(x_i, y_j)$ or desired field $U_{F_D}(x_i, y_j)$.

Like in the Gerchberg-Saxton algorithm described in Appendix A we set the illumination intensity to $I_H(x_i, y_j)$. We know that we can calculate using a Fourier transform the field in the Fourier plane from the field in the hologram plane. However, if we inverse Fourier our desired field in the Fourier plane we will get both a random phase and intensity distribution in the hologram plane. As we set the intensity in the hologram plane to be the illumination intensity $I_H(x_i, y_j)$ it is this constraint on the problem that means we must solve this problem as an inverse problem. The Fourier transform in this problem is our forward model; used to calculate the unknown result, field in Fourier plane, from the known event, field in the hologram plane.

The initial phase is a random distribution $\phi_H^n(x_i, y_j)$. We Fourier transform the initial complex field in the hologram plane $U_H^n(x_i, y_j) = I_H(x_i, y_j) \exp(i\phi_H^n(x_i, y_j))$ to

give the corresponding field in the Fourier plane $U_F^n(x_i, y_j) = I_F^n(x_i, y_j) \exp(i\phi_F^n(x_i, y_j))$. We then must numerically evaluate the fit of the guessed intensity distribution $I_F^n(x_i, y_j)$ to the desired intensity distribution $I_{F_D}(x_i, y_j)$ which we do using a cost function . In this case where the algorithm is shaping a desired intensity distribution the most simple cost function would be a least squares comparison between $I_F^n(x_i, y_j)$ and $I_{F_D}(x_i, y_j)$,

$$E^n = \sum_i \sum_j \mid I_F^n(x_i, y_j) - I_{F_D}(x_i, y_j) \mid^2 . \tag{C.1}$$

This cost function can be modified to compare phase or field depending on what is desired. If the value of the E^n is less than the value of the cost function calculated at the previous iteration we retain $\phi_H^n(x_i, y_j)$. During this initial iteration we always retain $\phi_H^n(x_i, y_j)$ as $E^{n=0} = \infty$.

All subsequent iterations begin by selecting a random pixel in the hologram plane and replacing the phase at that pixel with a random value to give a new field $U_H^{n+1}(x_i, y_j) = I_H(x_i, y_j) \exp(i\phi_H^{n+1}(x_i, y_j))$. Again we Fourier transform U_H^{n+1} to give the field in the Fourier plane due to this random change $U_F^{n+1}(x_i, y_j) = I_F^{n+1}(x_i, y_j) \exp(i\phi_F^{n+1}(x_i, y_j))$. We again numerically evaluate the random change using a cost function like that described in equation C.1. If E_{n+1} is less than E_n at the last successful iteration we retain $\phi_H^{n+1}(x_i, y_j)$ and if not we discard $\phi_H^{n+1}(x_i, y_j)$ and retain the phase at the previous successful iteration $\phi_H^{n+1}(x_i, y_j)$. We then begin another iteration by making a random change to the retained phase distribution.

Bibliography

[1] M. Bertero and P. Boccacci, *Introduction to Inverse Problems in Imaging* (Taylor & Francis, 1998).

[2] T. Flescha, J. D. Wilsona, L. A. Harperb, and B. P. Crenna, "Estimating gas emissions from a farm with an inverse-dispersion technique," Atmospheric Enviroment **39**, 4863–4874 (2005).

[3] M. Lehning, D. P. Y. Chang, D. R. Shonnard, and R. L. Bell, "An inversion algorithm for determining area-source emissions from downwind concentration measurements," Journal of Air and Waste Management Association **44**, 1204–1213 (1994).

[4] B. Hirst, G. Gibson, S. Gillespie, I. Archibald, O. Podlaha, K. D. Skeldon, J. Courtial, S. Monk, and M. Padgett, "Oil and gas prospecting by ultrasensitive optical gas detection with inverse gas dispersion modelling," Geophysical Research Letters **31** (2004).

[5] L. C. Thomson, B. Hirst, G. Gibson, S. Gillespie, P. Jonathan, K. D. Skeldon, and M. J. Padgett, "An improved algorithm for locating a gas source using

inverse methods," Atmospheric Environment **41**, 1128–1134 (2007).

[6] L. Horvitz, "Geochemical-exploration for petroleum," Science **229**, 821–827 (1985).

[7] D. Schumacher, *Exploring for Oil and Gas Traps* (American Association of Petroleum Geologists, 1999).

[8] R. W. Gerchberg and W. O. Saxton, "A practical algorithm for the determination of the phase from image and diffraction plane pictures," Optik **35**, 237246 (1972).

[9] G. Gibson, S. D. Monk, and M. J. Padgett, "A field-portable, laser-diode spectrometer for the ultra-sensitive detection of hydrocarbon gases," Journal of Modern Optics **49**, 769 – 776 (2002).

[10] I. R. MacDonald, J. R. Kelly, Jr, S. E. Best, R. Venkataramaiah, N. L. Guinasso, Jr, and J. Amos, *Hydrocarbon Migration and Its Near-Surface Expression*, vol. 66 (AAPG Mem., 1996).

[11] B. A. Martin and S. J. Cawley, "Onshore and offshore petroleum seepage: Contrasting a conventional study in papua new guinea and airborne laser flourescing over the arafura sea," The APEA Journal **31**, 333–353 (1991).

[12] G. G. Walton, "Three-dimensional seismic method," Geophysics **37**, 417 (1972).

[13] R. R. Draxler, "Determination of atmospheric diffusion parameters," Atmospheric Environment **100**, 99–105 (1975).

[14] F. A. Gifford, *Lectures on Air Pollution and Environmental Impact Analyses* (American Meterological Society, 1976).

[15] M. A. Seldowitz, J. P. Allebach, and D. W. Sweeney, "Synthesis of digital holograms by direct binary search," Applied Optics **26**, 2788–2798 (1987).

[16] N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller, and E. Teller, "Equation of state calculations by fast computing machines," Journal of Chemical Physics **21**, 1087–1092 (1953).

[17] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery, *Numerical Recipes in C++,W. H. Press* (Cambridge University Press, 2002).

[18] S. G. Gopalakrishnan and M. Sharan, "A Lagrangian particle model for marginally heavy gas dispersion," in *Atmospheric Environment* [18], pp. 3369–3382.

[19] D. Carruthers, R. Holroyd, J.C.R.Hunt, W. Weng, A. Robins, D. Apsley, D. Thomson, and F. Smith, "Uk-adms: A new approach to modelling dispersion in the earth's atmospheric boundary layer," Journal of Wind Engineering and Industrial Aerodynamics **52**, 139–153 (1994).

[20] A. Riddlea, D. Carruthersb, A. Sharpea, C. McHughb, and J. Stockerb, "Comparisons between fluent and adms for atmospheric dispersion modelling," Atmospheric Environment **38**, 10291038 (2004).

[21] C. S. Patterson, L. C. McMillan, C. Longbottom, G. M. Gibson, M. J. Padgett, and K. D. Skeldon, "Portable optical spectroscopy for accurate analysis of ethane in exhaled breath," Measurement and Science Technology **18**, 14591464 (2007).

[22] A. Ashkin, J. M. Dziedzic, J. E. Bjorkholm, and S. Chu, "Observation of a single-beam gradient force optical trap for dielectric particles," Optics Letters **11**, 288–290 (198).

[23] G. Gibson, L. Barron, F. Beck, G. Whyte, and M. Padgett, "Optically controlled grippers for manipulating micron-sized particles," New Journal of Physics **9**, 14 (2007).

[24] L. C. Thomson, Y. Boissel, G. Whyte, E. Yao, and J. Courtial, "Simulation of superresolution holography for optical tweezers," in *New Journal of Physics* [24], p. 023015.

[25] M. Berry, "Faster than fourier," in "Fundamental Problems in Quantum Theory," , J. A. Anandan and J. Safko, eds. (World Scientific, Singapore, 1994).

[26] R. Zambrini, L. C. Thomson, S. M. Barnett, and M. Padgett, "Momentum paradox in a vortex core," Journal of Modern Optics **52**, 1135–1144 (2005).

[27] G. P. Karman, M. W. Beijersbergen, A. van Duijl, D. Bouwmeester, and J. P. Woerdman, "Airy pattern reorganization and subwavelength structure in a focus," J. Opt. Soc. Am. A **15**, 884–899 (1998).

[28] R. W. Gerchberg, "Super-resolution through error energy reduction," Optic Acta **21**, 709–720 (1974).

[29] E. Noel, R. R. Kahn, and H. S. Dhadwal, "Optical implementation of a regularized gerchberg iterative algorithm for super-resolution," Optical Engineering **32**, 2866–2871 (2003).

[30] C. Blanca and S. Hell, "Axial superresolution with ultrahigh aperture lenses," Opt. Express **10**, 893–898 (2002).

[31] S. Hell and E. H. Stelzer, "Fundamental improvement of resolution with a 4pi-confocal fluorescence microscope using two-photon excitation," Optics Communications **93**, 277–282 (1992).

[32] S. Hell and J. Wichmann, "Breaking the diffraction resolution limit by stimulated emission: stimulated-emission-depletion fluorescence microscopy," Optics Letters **19**, 780–782 (1994).

[33] S. W. Hell, "Far-field optical nanoscopy," Science (2010).

[34] P. Wei, O. F. Tan, Y. Zhu, and G. H. Duan, "Axial superresolution of two-photon microfabrication," Appl. Opt. **46**, 3694–3699 (2007).

[35] L. E. Helseth, "Smallest focal hole," Optics Communications **257**, 1–8 (2006).

[36] J. B. Pendry, "Negative refraction makes a perfect lens," Phys. Rev. Lett. **85**, 3966–3969 (2000).

[37] J. E. Molloy and M. Padgett, "Lights, action; optical tweezers," Contemporary Physics **43**, 241–258 (2002).

[38] A. Ashkin, "Forces of a single-beam gradient laser trap on a dielectric sphere in the ray optics regime," Biophys. J. **61**, 569–582 (1992).

[39] J. E. Curtis, B. A. Koss, and D. G. Grier*, "Dynamic holographic optical tweeezers," Optics Communicaitons **207**, 169–175 (2002).

[40] V. Garces-Chavez, D. McGloin, H. Melville, W. Sibbett, and K. Dholakia, "Simultaneous micromanipulation in multiple planes using a self-reconstructing light beam," Nature **419**, 145–147 (2002).

[41] J. Leach, G. Sinclair, P. Jordan, J. Courtial, M. J. Padgett, J. Cooper, and Z. J. Laczik, "3d manipulation of particles into crystal structures using holographic optical tweezers," Optics Express **12**, 220–226 (2004).

[42] J. Leach, K. Wulff, G. Sinclair, P. Jordan, J. Courtial, L. Thomson, G. Gibson, K. Karunwi, J. Cooper, Z. J. Laczik, and M. Padgett, "Interactive approach to optical tweezers control," Applied Optics **45**, 897–903 (2006).

[43] M. Born and E. Wolf, *Principles of Optics* (Pergamon Press, Oxford, 1993).

[44] A. Rohrbach, "Stiffness of optical traps: Quantitative agreement between experiment and electromagnetic theory," Physical Review Letters **95**, 168102 (2005).

[45] M. V. Berry and S. Popescu, Journal of Physics A **36**, 6965–70 (2006).

[46] K. D. Wulff, D. G.Cole, R. L. Clark, R. D. Leonardo, J. Leach, J. Cooper, G. Gibson, and M. J. Padgett, "Aberration correction in holographic optical tweezers," Optics Express **14**, 41705 (2006).

[47] J. Durnin, J. Miceli, J. J., and J. H. Eberly, "Diffraction free beams," Physical Review Letters **58**, 1499–1501 (1987).

[48] Z. Bouchal, J. Wagner, and M. Chlup, "Self-reconstruction of a distorted non-diffracting beam," Optics Communications **151**, 207–211 (1998).

[49] L. C. Thomson and J. Courtial, "Holographic shaping of generalized self-reconstructing light beams," in *Optics Communications* [49], pp. 1217–1221.

[50] T. Eizmr, V. Garcs-Chvez, K. Dholakia, and P. Zemneka, "Optical conveyor belt for delivery of submicron objects," Applied Physics Letters **86**, 174100–174101 (2005).

[51] M. Anguiano-Morales, M. M. Mndez-Otero, M. D. Iturbe-Castillo, and S. Chvez-Cerda, "Conical dynamics of bessel beams," Optical Engineering **46(7)**, 078001–1–078001–9 (2007).

[52] R. D. Leonardo, F. Ianni, and G. Ruocco, "Computer generation of optimal holograms for optical trap arrays," Optics Express **15**, 1933–1922 (2007).

[53] E. R. Dufresne and D. G. Grier, "Optical tweezer arrays and optical substrates created with diffractive optics," Review of Scientific Instruments **69**, 1947–1977 (1998).

[54] S. A. Tatarkova, A. E. Carruthers, and K. Dholakia, "One-dimensional optically bound arrays of microscopic particles," Physical Review Letters **89**, 283901–1–283901–4 (2002).

[55] L. C. Thomson, G. Whyte, M. Mazilu, and J. Courtial, "Simulated holographic three-dimensional intensity shaping of evanescent-wave fields," in *Journal of the Optical Society of America B* [55], pp. 849–853.

[56] P. J. Reece, V. Garcs-Chvez, and K. Dholakia, "Near-field optical micromanipulation with cavity enhanced evanescent waves," in *Applied Physics Letters* [56], p. 22116.

[57] T. Cizmar, M. Siler, M. Sery, P. Zemanek, V. Garces-Chavez, and K. Dholakia, "Optical sorting and detection of sub-micron objects in a motional standing wave," in *Physical Review B* [57], p. 035105.

[58] A. Tarantola, *Inverse Problem Theory* (Society for Industrial and Applied Mathematics, 2005).

Index