ABSTRACT

Title of Dissertation: REAL-TIME PATH PLANNING FOR AUTOMATING OPTICAL TWEEZERS BASED PARTICLE TRANSPORT OPERATIONS

Ashis Gopal Banerjee, Doctor of Philosophy, 2009

Directed by: Professor Satyandra K. Gupta
Department of Mechanical Engineering

Optical tweezers (OT) have been developed to successfully trap, orient, and transport micro and nano scale components of many different sizes and shapes in a fluid medium. They can be viewed as robots made out of light. Components can be simply released from optical traps by switching off laser beams. By utilizing the principle of time sharing or holograms, multiple optical traps can perform several operations in parallel. These characteristics make optical tweezers a very promising technology for creating directed micro and nano scale assemblies. In the infra-red regime, they are useful in a large number of biological applications as well. This dissertation explores the problem of real-time path planning for autonomous OT based transport operations. Such operations pose interesting challenges as the environment is uncertain and dynamic due to the random Brownian motion of the particles and noise in the imaging based measurements. Silica microspheres having diameters between (1-20) µm are selected as model components.

Offline simulations are performed to gather trapping probability data that serves as a measure of trap strength and reliability as a function of relative position of the particle under consideration with respect to the trap focus, and trap velocity. Simplified models are generated using Gaussian Radial Basis Functions to represent the data in a compact form. These metamodels can be queried at run-time to obtain estimated probability values accurately and efficiently. Simple trapping probability models are then utilized in a stochastic dynamic programming framework to compute optimum trap locations and velocities that minimizes the total, expected transport time by incorporating collision avoidance and recovery steps. A discrete version of an approximate partially observable Markov decision process algorithm, called the QMDP_NLTDV algorithm, is developed. Real-time performance is ensured by pruning the search space and enhancing convergence rates by introducing a non-linear value function. The algorithm is validated both using a simulator as well as a physical holographic tweezer set-up. Successful runs show that the automated planner is flexible, works well in reasonably crowded scenes, and is capable of transporting a specific particle to a given goal location by avoiding collisions either by circumventing or by trapping other freely diffusing particles. This technique for transporting individual particles is utilized within a decoupled and prioritized approach to move multiple particles simultaneously. An iterative version of a bipartite graph matching algorithm is also used to assign goal locations to target objects optimally. As in the case of single particle transport, simulation and some physical experiments are performed to validate the multi-particle planning approach.

REAL-TIME PATH PLANNING FOR AUTOMATING OPTICAL TWEEZERS BASED PARTICLE TRANSPORT OPERATIONS

by

Ashis Gopal Banerjee

Dissertation submitted to the Faculty of the Graduate School of the
University of Maryland, College Park in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
2009

Advisory Committee:

Professor Satyandra K. Gupta, Chair/Advisor
Associate Professor Hugh Bruck
Professor Abhijit Dasgupta
Professor Amitabh Varshney
Professor Michael Zachariah

DEDICATION

To my parents and all those who made it possible

ACKNOWLEDGEMENTS

First and foremost I would like to express heartfelt appreciation and sincere gratitude to Prof. Satyandra K. Gupta for providing me this wonderful opportunity to conduct research under his astute guidance. His boundless energy, wonderful analytical skills, cool and calm composure, and motivational power has made this experience a truly memorable one. I am sure this will stand me in good stead in my future research and professional career as well.

I would also like to thank Dr. Hugh Bruck, Prof. Abhijit Dasgupta, Prof. Amitabh Varshney, and Prof. Michael Zachariah for readily accepting to be in my dissertation committee. I am grateful to Dr. Wolfgang Losert for allowing and helping me to conduct experiments in the Cell Dynamics Lab. I would also take this opportunity to thank Dr. Thomas LeBrun and Arvind Balijepalli for their help and advice when I first started working on optical tweezers. I also want to thank the National Science Foundation and the National Institute of Standards and Technology for supporting my dissertation research work.

Then I am thankful to University of Maryland, College Park, Department of Mechanical Engineering and the Institute for Systems Research for their support and facilities. I also want to express my appreciation to all my CIM lab friends, namely, Atul, Petr, Peyman, Alex, and Martin for their help and constant words of encouragement.

Of course, all this has been made possible by my parents and other family members. I can never thank Mom and Dad enough; they are the most wonderful parents one can ever hope to have. Their unfailing emotional support, love and affection have carried me through trying circumstances. I would not have been able to enjoy this achievement fully without them.

Last but not the least I want to thank all my dear friends: Arvind, Basu, Sahoo, Anand, Wojciech, Souvik, Snehaunshu, Sachin, Subhojit, Suvajyoti, Tania, Prabha, Tapi, Vipul, Srini, Sourav, Missy, Dora and many others for advice, friendly rebukes and loads of affection.

TABLE OF CONTENTS

List of Figures

List of Tables

List of Commonly Used Acronyms

OT	Optical tweezers
SLM	Spatial light modulator
RBF	Radial basis function
GPU	Graphics processing unit
PRM	Probabilistic roadmap
MDP	Markov decision process
POMDP	Partially observable Markov decision process
QMDP_NLTDV	QMDP_WithNonLinearTimeDependentValueFunction

Chapter 1

INTRODUCTION

1.1 Background

Micro and nano components (components with micro or nano scale size and features) can be used to exploit new phenomena that take place at those small scales [Bhus04]. Potential applications of such components include bio-sensors, electronic components, photonic devices, solar cells, and batteries [Niem04, Wils02]. These component based devices are expected to revolutionize health care, energy, communication, and computing industry [Ratn02].

In order to construct useful devices, the components need to be assembled together. Assembling micro and nano scale components to make functional devices remains a challenge despite rapid advances in imaging, measurement, and fabrication at the small scales. Two types of assembly processes are possible. The first type of process is self-assembly [Requ06] which is very useful for large scale production. The second type of process is directed assembly. This process is useful for prototyping new design concepts, small scale production, device repair, and creating templates for certain types of self-assembly processes. A number of manipulation techniques for directed assembly of components have emerged (e.g., fluid flow based [Yi06, Ong08], magnetic manipulation [Bent04, Frie05], dielectrophoresis [Kim06, Li02], scanning probe microscope based manipulation [Meye04, Moka07], and optical tweezers [Ashk00, Bali05, Grie03]). A comparative survey of different manipulation techniques can be found in [Sitt01].

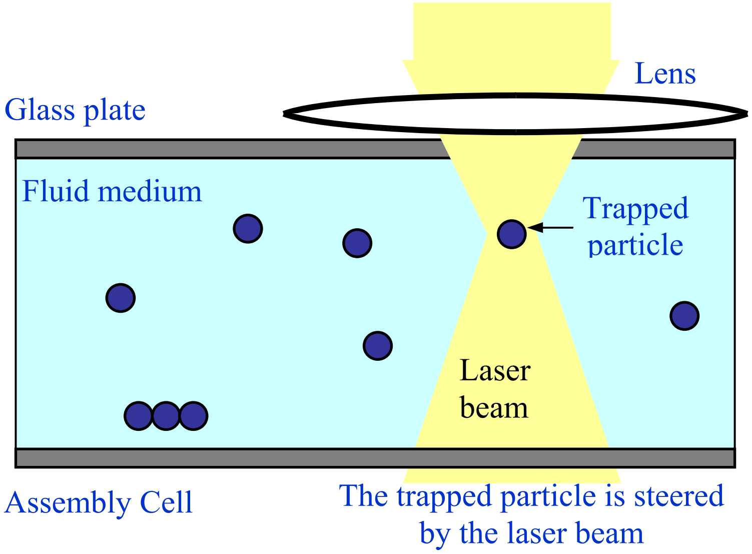


Figure 1.1: Schematic illustration of optical trapping

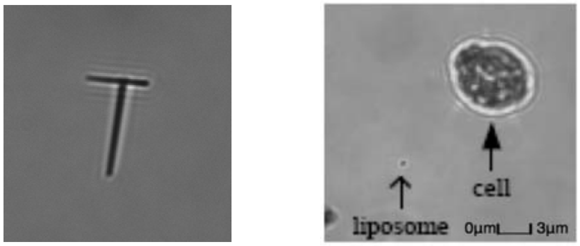
Broadly speaking, the directed manipulation techniques can be classified into three categories: fluid flow based, contact, and non-contact. The first category typically uses micron and nanometer sized channels to precisely control and manipulate fluids along with the components present in them. In the second category, a physical probe (microscopic cantilever beam) with a sharp tip at the end or a micro-electro-mechanical gripper is used to push and manipulate components. Atomic force microscope is the most common type of scanning probe based techniques. In the last category, an appropriate optical, electrostatic, or magnetic field (i.e., trap) needs to be created to trap the component. The field is then controlled to move the component in the fluidic workspace. A schematic illustration of optical trapping is shown in Figure 1.1. The interaction between the field and the component is stochastic in nature due to the Brownian motion of the component and the presence of uncertainty in the component location as a result of sensor noise. Unfortunately, the offline planning approach that works well at the macro scale for optimizing operation efficiency does not work at the nano scale. To cope with the stochastic nature of the problem, automated micro and nano scale manipulation requires a new real-time and automated planning paradigm that takes actions based on the estimated state of the workspace in a feedback loop.

1.2 Motivation

As discussed in the previous Section, optical tweezers (OT) are emerging as a promising tool for trapping and manipulating micro and nano scale components in a fluid medium. If these components can be assembled together to form complex 3D shapes, then useful devices can be created with improved electrical, thermal or optical properties. However, currently all such assemblies are formed either manually or semi-automatically.

Few examples of such assemblies are shown in Figures 1.2 and 1.3 respectively. Figure 1.2 (a) represents a T-junction formed using two ZnO nanowires. Figure 1.2 (b) shows a cell-liposome assembly that can be potentially used for drug delivery. Both these assemblies have been created by manually manipulating the components using rasterized scanner based optical tweezers at the Manufacturing Engineering Laboratory in the National Institute of Standards and Technology (NIST). Figure 1.3 shows the gradual morphing of a 15 μm diamond unit cell from eighteen 1 μm silica spheres that were originally arranged in the form of two linear columns. The time at which each video frame was taken is shown in the top left corner. Although programmable holographic optical tweezers were used to morph the linear columns into a 3D cubical structure automatically, initial positioning of the spheres was done manually.

Consequently, it takes a long time (few minutes to even hours) to complete the assembling process. Moreover, during manipulation, due to manual control, some unwanted components also get trapped. Thus, a lot of time is wasted in removing those components from the final assembly at a later point of time. Sometimes as a result on undesired trapping, collisions between multiple components may result in de-trapping as well. By de-trapping, it means that some of the originally trapped components get displaced from their stable trap locations, which, in turn, results in them being no longer trapped by the tweezers. In a nutshell, the overall process is very tedious and inefficient.



(a) ZnO wires (b) Cell and liposome

Figure 1.2: Examples of assemblies formed by optical tweezers (Image courtesy: Arvind Balijepalli)

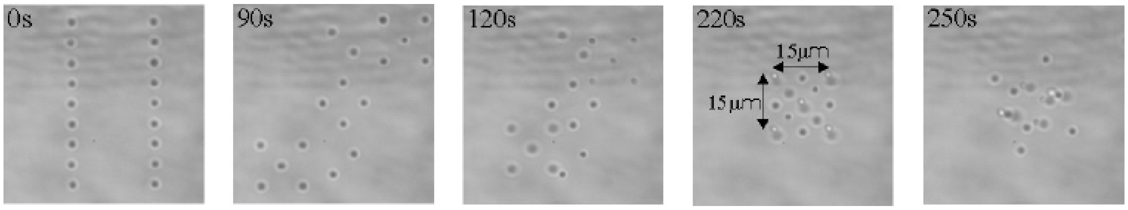
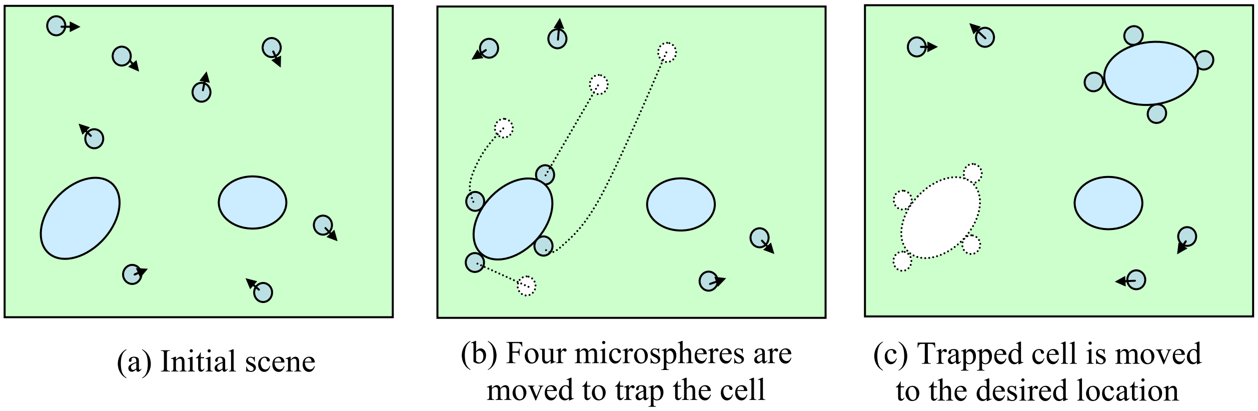


Figure 1.3: Formation of a 15 μm diamond unit cell from eighteen 1 μm silica spheres (Image source: [Sinc04]).

So far, optical tweezers have been primarily used to create directed assemblies which can be utilized for limited prototype production, and repair work. However, if this technology is coupled with chemical assembly or micro and nano fluidics, then the potential is enormous, provided the process can be sped up. As shown in [Cast05], complex 3D structures such as micro-wires can be produced if such integration is performed. Then the assemblies formed by optical tweezers will also serve as the templates for bulk-scale self-assembly process.

In the biological domain, recently researchers have shown that sorting of cells can be enhanced by attaching silica microspheres coated with antibodies to cell surfaces. The notion of attaching microspheres to cells for indirectly manipulating them can also be potentially utilized to place a large number of cells precisely on a substrate for quantitative studies of inter cell communication. The indirect method will limit both the peak and average light exposure of the cells as compared to direct optical trapping, where the trap focus is positioned within a cell causing damage due to local heating and formation of undesirable free radicals. This idea is shown schematically in Figure 1.4. *Saccharomyces cerevisiae* (type of yeast cells) have been successfully transported manually at a laser power of 0.4 W by attaching three 4.74 μm nominal diameter silica beads using the holographic BioRyx 200 system (see Figure 1.5). The cell-sphere complex was moved together as a single group by using in-built capabilities of the system. The individual trap focal positions are shown using '$\oplus$' symbols. Details about the BioRyx set-up are specified in Chapter 4. As expected, the manual operation turned out to be quite challenging and time-consuming.

Hence, there is a need for automating the process of assembling components or manipulating biological objects. Automation involves real-time path planning for the components that need to be transported from their current locations to some other desired locations. This refers to the software code that needs to be fed to the controller hardware to automatically set the locations and velocities of the individual optical traps (laser beams). Irrespective of whether multiple optical traps are created by time-sharing the laser beam across multiple trap locations, or by generating holograms, the planner should take the estimated locations of all the components from the imaging system as the input and generate optimum values. This process must be repeated every time a fresh set of image data becomes available. This dissertation describes development of such a planning system that will enable automated transport operations in the future.


(a) Initial scene

(b) Four microspheres are moved to trap the cell

(c) Trapped cell is moved to the desired location

Figure 1.4: Using microspheres to transport cells (small circles denote microspheres and larger ovals denote cells)

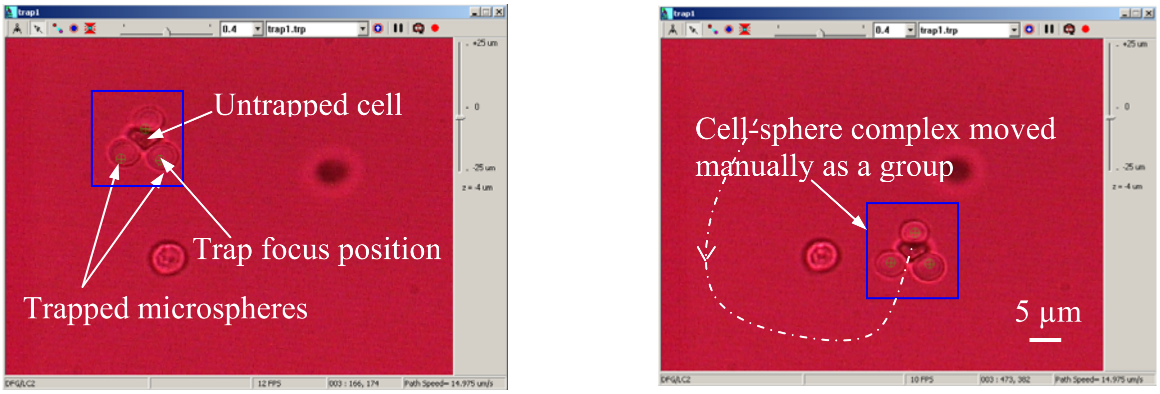


Figure 1.5: Manual indirect transport of yeast cell using three silica microspheres

1.3 Research Issues

This work identifies three primary research issues or challenges in order to perform real-time path planning for transporting particles using optical tweezers with minimum amount of manual intervention. They are described in details as follows.

1. *Understanding optical trap-particle interaction*: Particles first need to be trapped before they can be transported to the desired goal locations in the assembly cell. Thus, information that is very useful in this context is the probability with which a particle will be trapped in a spatial region close to a stationary or moving optical trap, i.e. laser beam. The probability arises as a result of stochastic, Brownian motion exhibited by any particle immersed in a fluid medium, due to which it either may diffuse towards or away from the trap focal region, where attractive gradient forces dominate. Henceforth, this will be referred to as the *trapping probability*. This probability acts as a surrogate measure of the trap robustness as well as reliability as a function of trap velocity and the relative position of the particle with respect to the trap focus. Any trapping event involves a complex interplay of various forces: optical, Brownian, gravity, and buoyancy. In the absence of any analytical expression or experimental results for the probability distribution function, this dissertation explores the use of physically-accurate simulation to estimate trapping probability values. However, simulations need to be performed at very small time intervals (of the order of μs); so it takes a very long time to gather relevant data. Hence, real-time simulations cannot be used for path planning. On the other hand, offline simulation can be used to generate data at discrete points in the parameter space (consisting of

spatial coordinates and velocity components) which can then be represented by computationally efficient simplified models.

2. *Handling large degree of uncertainty in planning*: Once particles have been reliably trapped by laser beams, they have to be transported in an optimal manner to the given goal locations. However, this problem is challenging due to the presence of two forms of uncertainties. Firstly, all the particles are constantly moving around in the workspace following the laws of Brownian motion. This means that the actual position, velocity and acceleration of any particle cannot be determined exactly ahead of time. In other words, during transport operations, random collisions may occur between the trapped and originally untrapped particles. This may result in situations where certain particles are no longer trapped or additional, undesirable particles are trapped by the laser beams, thereby increasing the overall transport time as beams have to be switched off and on or re-positioned before resuming the operation. Thus, any path planning algorithm needs to consider fast, random motion of untrapped particles in the workspace so that such unwanted collisions can be avoided. Secondly, imaging based sensors such as Optical Section Microscope used at NIST, provide noisy data. The positions of all the particles in the system are known only with a certain degree of certainty. Thus, the controller may never be able to position the trap focus exactly where it wants to, because the particle itself may be located in a slightly different position. Similarly, circumventing other particles to avoid potential collision events also becomes more challenging as their true positions are never known. So, this dissertation investigates different motion planning paradigms to select the one

whose underlying mathematical framework is best suited for tackling such a stochastic optimization problem.

3. *Planning to facilitate real-time operations*: Unlike many other robot navigation, planning and control problems commonly encountered in practice, the time available for planning in this case is limited by the imaging interval and/or controller update rate, which is typically of the order of few milliseconds. Ideally, the planning time should be very short as compared to both the time intervals. This leaves any planner with very little time to evaluate different options and select the best one. The problem is significantly magnified in the case of transporting multiple particles, where the aim is to complete transporting all the particles to their respective goal locations in the minimum, expected time. This involves avoiding unwanted collisions not only with the freely diffusing particles, but also among the particles that are being transported themselves. That is why, all feasible options or control actions that are guaranteed to yield sub-optimal solutions need to be pruned to reduce the search space. Modifications to the existing planning approaches are explored in this research work to compute the optimum control policy that satisfies the rigorous time and performance constraints.

1.4 Dissertation Scope and Outline

Currently, optical tweezers are primary used in manipulating spherical components (glass, silica, latex, polystyrene, metals like silver, gold, and biological cells). So, this dissertation will focus on transporting spherical particles using optical tweezers. Henceforth, spheres, objects, and particles will be used interchangeably in the text. Although many of the techniques and approaches developed as a part of the current research work are equally applicable for other types of components such as nanowires, flat particles etc. all the simulation and experimental validation will be carried out on spheres only. Amorphous silica is chosen as the sphere material since it can be easily procured, maintained, and a lot of experimental work has already been conducted on it. Due to the ease of conducting experiments and availability of well-established trapping force models, the current work is restricted to dielectric microspheres that are homogeneous and isotropic. However, it may be noted here that it is anticipated that all the proposed techniques can be generalized to nanometer scales without any significant need for altering the basic framework to maintain a similar level of performance. Moreover, all the transport operations are performed only in the same horizontal plane.

The rest of the dissertation is organized as follows. The next chapter surveys state-of-the-art literature in the related areas of micro and nano manipulation, model simplification or metamodeling techniques, single and multiple robot motion planning approaches under uncertainty, and indirect optical manipulation of cells. Chapter 3 presents a radial basis function based approach to generate simplified models for estimating trapping probability from the offline simulation data. Chapter 4 describes a stochastic dynamic programming framework to minimize the expected time taken to transport a single particle from its current location to the goal. Chapter 5 describes a decoupled and prioritized approach for transporting multiple particles simultaneously after assigning goal locations optimally. Finally, Chapter 6 summarizes the intellectual contributions of the current work, highlights anticipated benefits to the industry and the research community, and outlines scope for future work.

Chapter 2

LITERATURE SURVEY

The current dissertation work is inter-disciplinary. So, a rich body of literature existing in different fields of engineering and science needs to be reviewed in this Chapter. Since it is not possible to review all the papers available in the literature, representative works that are either recent or seminal and are directly relevant are discussed in the Chapter.

Section 2.1 deals with micro and nano manipulation. This is a vast area of research in itself. Keeping in mind the application domain, the survey is primarily restricted to optical tweezers. Since this work focuses on spheres, manipulation of spheres using optical tweezers has been reviewed in details. Some representative work on manipulation of other types of components using optical tweezers, and other manipulation techniques has been reviewed later in the same Section. The next Section presents a survey of metamodeling techniques as one such technique will be used to represent the trapping probability simulation data. After reviewing few survey articles covering a wide variety of such approximation or simplification techniques in the first part of the Section, the second part exclusively reviews use of radial basis functions for several types of applications.

Section 2.3 deals with single robot motion planning under uncertainty. This is an area of great interest in the robotics community. After briefly discussing some books and seminal papers that presented several classical motion approaches, this Section primarily reviews state-of-the-art literature on two specific techniques: probabilistic roadmaps and stochastic dynamic programming. These two classes of techniques are most commonly used nowadays to deal with planning problems involving stochastic uncertainty. Hence, a comparative understanding of both the classes of techniques is essential to select the one that is more suitable for the current application. The next Section surveys some of the recent work on decoupled motion planning of multiple robots. After briefly mentioning the coupled approach, this Section discusses different methods that either relies on a centralized or a decentralized architecture within the overall decoupled framework. The final Section establishes the need for indirect cell manipulation as opposed to direct manipulation and presents some of the recent work in this area. Each Section contains a summary where the rationale behind selection of a particular approach or model and the need to extend it for the application being investigated in the dissertation has been emphasized.

2.1 Micro and Nano Manipulation

2.1.1 Different Optical Tweezers Systems and Set-ups

Light beams or optical traps exert small optical forces of the order of picoNewtons (pN), and for components smaller than tens of micrometers and up to few nanometers, the forces can be designed to “grasp” and move them to desired positions. Using this property of light beams, optical tweezers have been developed to successfully trap and transport micro and nano scale components of many different sizes and shapes [Ashk00, Frie98]. Optical tweezers do not make physical contact with the trapped component and hence do not cause any damage due to deformation. Components can be simply released from the optical traps by switching off the laser beams. Moreover, since optical tweezers systems use inexpensive lasers, microscopes, telescopes and video cameras, the overall cost of the equipment is relatively modest.

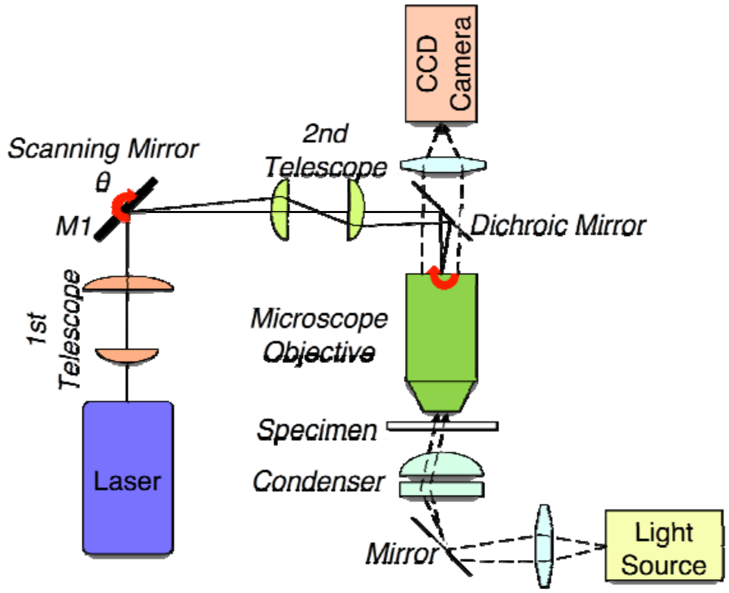


Figure 2.1: Rasterized scanner based optical tweezers set-up (Image source: [Bali06])

Although different types of optical tweezers systems and set-ups exist in various universities, industries, and research laboratories across the world, the two most popular ones are rasterized scanning mirror based and hologram based. A modular and flexible software architecture and system framework for performing assembly operations using the former type of set-up is described by Balijepalli *et al*. [Bali05, Bali06]. In this set-up shown in Figure 2.1, the laser source generates the laser beam. The scanning mirrors are used to create a variety of trap shapes by rapidly moving the laser beam over a certain area. In addition, they are used to create multiple beams by time sharing the beam coming out of the laser source among different traps. The telescope is used to steer the beam and magnify its size. This beam magnification is needed to fill the entrance aperture of the microscope objective, a key requirement to maximize trapping force for optical tweezers.

The microscope objective is used to focus the beam and form the trap. Communication among different modules such as interface, control/automation, and hardware is performed using a protocol known as neutral messaging language. A set of representative examples including an assembly of five 3 µm silica spheres, trapped gold dimers, out-of-plane rotation of alumina silicate nanowires and so on illustrates the capabilities of this system.

Although components can be observed in the optical microscope, resolution is insufficient to estimate 3D locations and parameters such as length, radius etc. with sufficient accuracy. Tao et al. [Peng07a, Peng07b] have developed algorithms to compute such information from a stack of images at various cross-sections (horizontal planes). In case of microspheres, the images are first segmented to obtain areas of interest. The image gradient information in the obtained areas is then used to locate the position of spheres in the XY plane. Finally, signature curves are computed to obtain the Z-location of the spheres. A similar methodology has been used to estimate the length, location, and orientation of nanowires by combining additional feature extraction techniques based on modified Hough transform.

A dynamically reconfigurable version of the second type of set-up is described in [Curt02, Grie03]. It is schematically shown in Figure 2.2. Holographic optical tweezers use a computer-generated diffractive optical element (DOE) to split a collimated laser beam into several separate beams, each of which acts as a conventional optical tweezer. Originally implemented with micro-fabricated DOEs, later on they have been implemented on computer-addressable liquid crystal spatial light modulators (SLMs). The authors have presented new algorithms for computing phase holograms so that real-time, dynamic reconfiguration of trap patterns becomes possible. This tweezers set-up can sculpt the wavefront of each optical trap separately, so that a mixed array of traps based on different light modes can be created, including optical vortices, axial line traps, optical bottles, and optical rotators. Representative examples on morphing of 3D patterns created by silica and polystyrene particles clearly show that this ability to manipulate individually structured optical traps in 3D opens up exciting new possibilities for engineering, diagnostics, and manufacturing at the small scales.

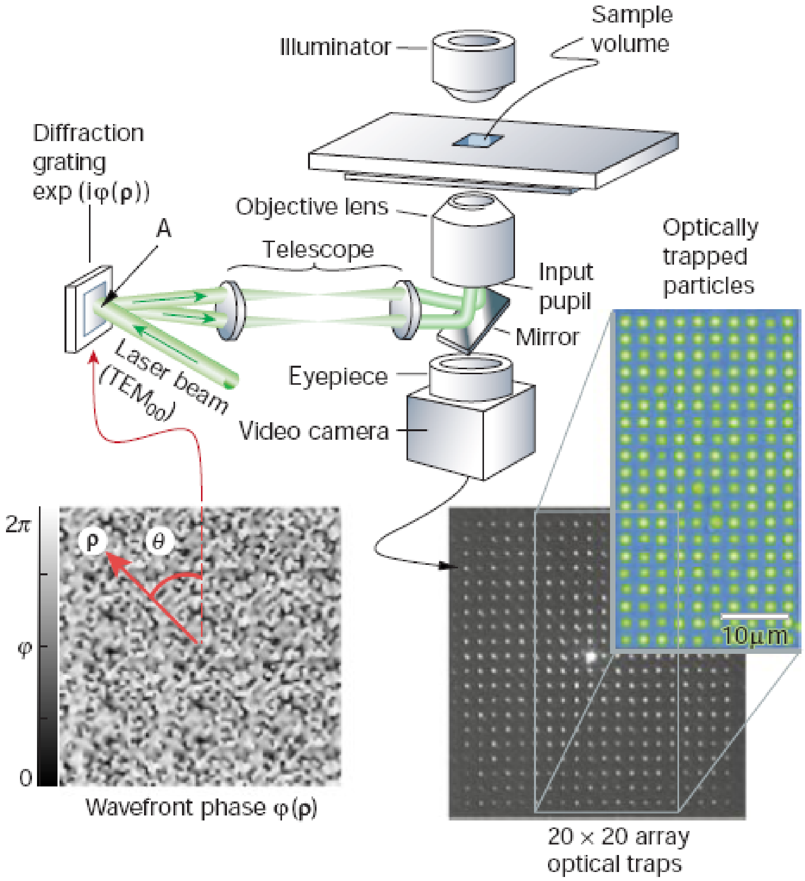


Figure 2.2: Schematic illustration of dynamic holographic optical tweezers (Image source: [Grie03])

It may be noted here that both the set-ups have their own relative advantages and limitations. The extent and complexity of multi-particle patterns that can be formed using scanning mirror based optical tweezers is limited by the time required to re-position each of the diffusing particles, since the beam dwells briefly on one particle before moving to the other. Even if this scanning is performed at high rates, operations are primarily restricted to the focal plane of the lens. Thus, mostly they are useful for organizing planar assemblies. Holographic tweezers suffer from no such limitations. However, dynamic operations are still much slower as compared to the scanner based tweezers, although improved algorithms such as modified Gerchberg-Saxton algorithm has been proposed by certain researchers [Sinc04].

2.1.2 Manipulation using Optical Tweezers

2.1.2.1 Spheres

Ever since Ashkin *et al*. [Ashk86] demonstrated that optical gradient forces are sufficient to trap small dielectric particles, several researchers have worked on manipulation of spheres. Experimental, analytical, and simulation techniques have been developed to understand the physics of optical trapping and compute or measure the trapping forces. Two models are commonly used to compute the optical trapping forces. These models are complementary to each other and are applicable in two different size regimes [Wrig93]. The Gaussian beam electromagnetic field (EM) model, originally developed by Barton *et al*. [Bart88, Bart89], gives reasonably accurate results when the laser wavelength (λ) is larger than the sphere diameter. In this model, expansion coefficients that describe the incident and scattered laser fields are derived from an infinite series representation of the electric and magnetic fields. Forces are then computed using Maxwell’s stress tensor.

Subsequently, Rohrbach and Stelzer [Rohr01] have generalized this model for arbitrary time-invariant laser EM fields. They have obtained the two major components of the optical force, namely gradient (acting perpendicular to the beam propagation direction) and scattering (acting along the propagation direction) separately. Moreover, they have utilized these values to compute experimentally verifiable parameters such as force constants (trap stiffness) and trapping potential depths [Rohr02]. In a recent work, Rohrbach [Rohr05] has found good quantitative agreement between experimental and electromagnetic theory based analytical results in this sub-wavelength regime. He has shown that for all the three dimensions, the measured extent of harmonic optical trapping potentials comes close to the theoretically predicted extent, provided all the instrument parameters are taken into consideration. Moreover, the strong asymmetry of the trapping potential is verified in all the three dimensions.

The second model is the well-known geometric ray optics one, originally developed by Ashkin [Ashk92], which is applicable when λ is much smaller than the sphere diameter. This model has been later used by other researchers [Im02, Ukit06] to compute the axial and transverse trapping efficiencies with minor modifications. The underlying principle behind this model is to decompose the overall light beam (assumed to be Gaussian with input beam waist equal to objective lens aperture radius) into individual rays, each with appropriate intensity, direction and state of polarization, that propagate along straight lines in a homogeneous medium. Although diffractive effects are neglected here, each ray can change direction when it reflects, refracts and changes polarization at dielectric interfaces following usual Fresnel formulae. Each incident ray is partially reflected out of the sphere and partially transmitted inside the sphere. Each transmitted or refracted ray successively undergoes infinite number of internal reflections before it eventually emerges from the sphere. As net momentum must be conserved in this process, the change in momentum of the ray exerts optical forces (both gradient and scattering) on the sphere. The overall force experienced by the sphere can hence be obtained by integrating (summing up) the contributions of all such rays that are incident on the lens aperture. Actual equations are shown in Appendix A.

Svoboda and Block [Svob94a] have shown that in the sub-wavelength (Rayleigh) regime, metals and dielectrics behave similarly from the light scattering point-of-view. In fact, larger polarizability of the metals implies that the optical trapping forces are greater. This makes nanometer sized metallic particles very useful for biological applications.

An analytical solution is available for homogeneous isotropic spheres that lie in the intermediate size regime between small and large particle (with respect to laser wavelength) approximations. This is known as the Lorenz-Mie theory. The original theory was developed for plane waves only, and its extension to non-plane case is known as the generalized Lorenz-Mie theory [Gous82]. It has been used quite extensively for modeling the optical trapping forces for isotropic spheres [Neve06]. One of the major difficulties associated with this theory is the usual paraxial representation of laser beams as solutions of the scalar wave equation instead of the vector Helmholtz equation. Nieminen *et al*. [Niem03] have chosen a least squares fit to generate a Helmholtz beam with a far field matching that is expected from the tightly focused beam. This implies that the incident and the scattering field can be represented in terms of a discrete basis set of functions which can be truncated at some finite values. Assuming a linear relationship between the incident and scattered fields, the coefficients of the two basis functions are related by a simple matrix (*T*) equation. Recently, Nieminen *et al*. [Niem07] have developed a MATLAB toolbox to compute the trapping forces and torques using this T-matrix method, where the basic functions consist of vector spherical wavefunctions.

2.1.2.2 Other Components

A detailed history of optical trapping and manipulation of small, neutral components, atoms and molecules till the dawn of this millennium can be found in [Ashk00]. Special emphasis has been given to biological objects. This list includes colloidal tobacco mosaic virus, *E. coli* bacteria, pigmented red blood cells, green algae, amoebas, other protozoans, vesicles, DNA etc. It should be noted here that trapping with infra-red Nd:YAG laser at 1.064 μm wavelength reduced the damage due to reduction in absorption of water, and molecules such as chlorophyll and hemoglobin. This paper highlights the immense potential of optical tweezers to investigate various mechanical properties of biological objects, such as compliance and viscoelasticity. It also reports several experiments that have been conducted to use "laser scalpels" for cutting and moving cells and organelles and to study the behavior of molecular motors, namely microtubules and actin filaments.

Galajda and Ormos [Gala03] have successfully demonstrated orienting flat microscopic particles using linearly polarized light. They have found out that the orienting torque originates from the anisotropic scattering of the polarized light. Simple viscous, drag force equation for a cylindrical object and ray optics model are used to characterize the orientation torque. Thus, the torque exerted can be measured and controlled and the position of the particle inside can be completely regulated. This angular trapping effect provides a useful extension of current optical tweezers applications. For example, this method provides a way to investigate the torsional properties of macromolecules attached to anisotropic objects.

Plewa *et al*. [Plew04] have demonstrated trapping and manipulation of carbon nanotubes for the first time using holographic optical tweezers. Carbon nanotubes are highly valued for their mechanical, electrical, chemical, and optical properties; hence, they form the basic building blocks of most nanotechnological applications. This is a rather surprising result because individual nanotubes are substantially smaller than the wavelength of light; hence, they should not be amenable to optical trapping. However, they have shown that nanotubes bundles, and possibly even individual tubes can be transported at high speeds, deposited on substrates, untangled, and even selectively ablated. One such operation is shown in Figure 2.3, where a dispersion of single-walled nano tubes is gathered into an optical trap and translated in water at up to 100 μm/s. The arrows in (c) indicate the direction of motion. This opens up numerous possibilities for highly parallel nanotube processing with light that can be used to create many new devices.

Yu *et al*. [Yu04] have presented a technique for manipulating and assembling one-dimensional CuO nanorods using line optical tweezers. In the presence of line optical tweezers, it is found that the nanorods get aligned with their long axis coinciding with the direction of the optical field. Moreover, they are found to travel along with the line tweezers towards the end with higher intensity. Thus, they can be manipulated to form some basic shapes, such as ‘T’, ‘L’, cross and triangle. Another interesting application shown in the paper is bridging a nanorod across two gold electrodes. This will enable the study and measurement of electrical properties of the nanorod under consideration.

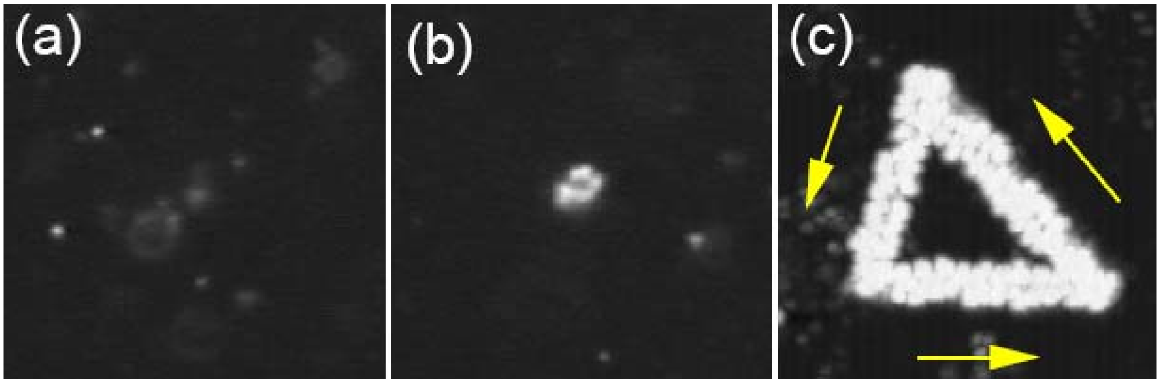


Figure 2.3: Carbon nanotubes processing using holographic optical tweezers (Image source [Plew04])

2.1.2 Other Manipulation Techniques

As mentioned earlier in this Chapter, hundreds of published articles can be found on various manipulation techniques. Only few recent, representative work spanning both micro and nano manipulation of biological as well as inorganic components will be reviewed in this sub-Section.

Dielectrophoresis refers to the movement of particles in non-uniform electric field. It is of two types – negative and positive. During negative dielectrophoresis, the dielectric constant of the particle is less than that of the fluid medium and the particle moves towards the location having the smallest electric field gradient. The reverse phenomenon takes place during positive dielectrophoresis.

Li and Bashir [Li02] separated live and heat-treated (dead) cells of *Listeria innocua* bacteria on micro-fabricated devices with interdigitated electrodes using ac dielectrophoresis. Difference in dielectric properties between alive and dead cells as a function of ac signal frequency is utilized to perform the separation operation. Both the types of cells collect at the center of the electrodes in negative dielectrophoresis, whereas they accumulate at the electrode edges in positive dielectrophoresis due to a combination of dielectrophoretic and electro-hydrodynamic forces. The authors have shown that the behavior of the two types is different in the frequency range from (30-100) kHz, although both experience negative dielectrophoresis at lower frequencies and positive dielectrophoresis at higher values. More specifically, a 90% separation efficiency is achieved at 50 kHz and 1V peak-to-peak potential difference as the live cells experience positive dielectrophoresis, while the dead cells still experience negative dielectrophoresis. This can be potentially useful in micro scale sample preparation and diagnostic applications in biochips.

Kim *et al*. [Kim06] reported an ac and dc dielectrophoresis method to align and manipulate semiconductor gallium nitride (GaN) nanowires using variations in the type of electric field as well as signal frequency. They observed that the ability to align and form an assembly of nanowires (single or a bundle configuration) depends on the magnitude of both ac and dc fields. In terms of alignment, ac dielectrophoresis achieves a higher yield rate of approximately 80% over the entire array in the chip as compared to dc dielectrophoresis. The authors have successfully demonstrated formation of hybrid p-n junction structures by assembling n-type GaN nanowires with a p-type Si substrate. These hybrid structures show well-defined current biasing behavior with a low reverse, leakage current of about 3 X 10^{-4} A at -25 V. So they can prove to be useful in electronic and optoelectronic devices.

Bentley *et al*. [Bent04] have demonstrated manipulation of non-magnetic, CuSn metal alloy nanowires by capping them with magnetic (Ni) ends. These nanowires were prepared by sequential electro deposition of nickel and copper/tin bronze alloys into alumina membrane with pores 200 nm in diameter. The separated nanowires were suspended in solvents and were oriented and spun in circles as “nano stirbars” using external magnetic fields. These segmented nanowires were also trapped between magnetized Ni stripes. About 66% of the nanowires were aligned within 10° of the horizontal with the assistance of a horizontal magnetic field. Most of them touched the outer edge of a stripe and extended beyond it to form cantilevers. Few (13%) formed a bridge between the two stripes. This method can be used in the future to position a wide variety of nano wires (and not merely magnetic ones) for mechanical testing and incorporation into nano scale devices.

Friedman and Yellen [Frie05] have performed a review of the physical fundamentals on the topic of manipulation of colloidal particles based on their magnetic and other kinds of properties using magnetic separation techniques. They have summarized the relevant models and emphasized the recent work on the use of micro systems in separation, assembly, and manipulation of non-magnetic particles in magnetic fluids. The focus is on issues pertaining to the scaling of magnetic forces with particle size and gradient generating structures. The authors have concluded that inclusion of hydrodynamic interactions based on particle concentration and self-consistent calculations of fluid velocity profiles in trajectory based models are needed to develop more accurate models of handling magnetic beads in micro systems and lab-on-a-chip applications.

A detailed description of different scanning probe microscopy techniques including various force models and modes of operation, with particular emphasis on scanning tunneling microscopy and atomic force microscopy, can be found in [Meye04]. Mokaberi *et al*. [Moka07] have described a fully automated system for building arbitrary planar patterns of nano particles using atomic force microscope (AFM). For a given, random distribution of particles on a substrate surface and a desired pattern, the planner determines the paths required to perform the manipulation operations. The planner outputs a sequence of primitive commands for positioning and pushing operations involving linear motions. The commands are executed by a software that accounts for thermal drift, creep and hysteresis by applying Kalman filters and Prandtl-Ishlinskii operators suitably. Experimental results on manipulation of 15 nm Au particles on mica show that patterns can be built in minutes that would have normally taken a day for an experienced user to construct interactively.

Mechanical grippers can also be used for handling micron-sized components. A review of the existing systems for micro-part handling is available in [Sanc05]. Primarily, three types of micro grippers are popularly used: friction, pneumatic, and magnetic. One of the key challenges is the effect of the grip principle on the accuracy of the part gripper relation. That is why, often sensor control is required to compensate for the inaccuracies resulting from the adhesive forces. A recent survey of compliant micro grippers in the context of micro-motion device technology for biotechnology applications is available in [Ouya08]. The authors have pointed out that various microfabrication techniques are being increasingly used to design electrostatically, electrothernally, or electromagnetically actuated grippers to grasp objects of different shapes steadily with high positioning accuracy and a large number of degrees of freedom.

Microfluidics has been shown to be a useful technology for manipulating a wide variety of micro and nano scale objects. A review of the latest advances in manipulating magnetic particles for their use in magnetic separation, immuno-assays, magnetic resonance imaging, drug delivery, and hyperthermia can be found in [Gijs04]. Yi *et al.* have reviewed the use of microfluidics for the manipulation and subsequent analysis of biological cells [Yi06]. Such systems possess lot of advantages in terms of controlled transportation and immobilization of cells, as well as separation, mixing, and dilution of chemical reagents for analysis of intracellular parameters and detection of cell metabolites. They can also be used for cell lysis, cell culture, and cell electroporation. Some of the popular microfluidic devices include cytometer, biochemical sensing chip, and whole cell sensing chip. A more recent survey of the fundamental principles and applications of microfluidics for cell manipulation and DNA amplification is available in [Ong08].

Several researchers have also worked on combining two or more manipulation techniques. For example, Ozkan *et al.* [Ozka03] have described an electro-optical system that uses the physical properties of mammalian cells (charge, dielectric permittivity etc.) to rapidly pattern and manipulate cells in a microarray format. It employs electrophoretic arraying of cells in a dc field due to intrinsic negative surface charge, and remote optical manipulation of individual cells by vertical-cavity surface emitting laser driven infra-red optical tweezers. The platform is optically transparent, thereby enabling fluorescent reporters of cell events.

Sitti has conducted a survey of different nano manipulation systems [Sitt01]. He has grouped the various approaches according to starting point (top-down versus bottom-up), utilized process (self-assembly versus physical such as electrical, optical etc.), interaction type (contact versus non-contact), and operation type (teleoperated, semi-autonomous, and automatic). The main components of such systems, namely manipulators (AFM, optical tweezers etc.), physical models, sensors, actuators, and controllers have been discussed in details. He has summarized the applications of nano manipulation systems in biology, material science, and computer technology and concluded that some of the open areas of research include integration of self-assembly with precise physical manipulation, automation, design of nano grippers, and modeling of sensing of nano scale forces.

2.1.3 Summary

It is clear from the preceding discussion that there is no clear winner among the different manipulation techniques. Lot of work has been done on each of them to enhance their capabilities and currently all of them can successfully manipulate components of varying shapes, sizes, and properties. It is due to the versatility in terms of manipulating neutral, non-magnetic components and an interest to apply the results of this research work to biological systems that optical tweezers have been selected. Although other techniques have been used to manipulate and study different biological systems, optical tweezers are most popular since the damage on cells, tissues, and macromolecules (DNA, proteins etc.) due to light exposure is less than that as a result of external electric or magnetic field, or physical contact forces. Although microfluidics is a very useful technology for gross motion, fine motion control for transporting individual components is difficult to achieve.

Spheres form a natural choice for components that will be manipulated using optical tweezers. They have been shown to be particularly amenable for trapping purposes, as evident from the vast body of literature that exists on different trapping models. Although many researchers are actively working on trapping and orienting nanowires and nanotubes, the trapping physics is not yet well understood. Hence, it is difficult to manipulate them with a high degree of reliability and precision. The reasons for selecting silica as the sphere material during simulation are already mentioned in Chapter 1. As discussed earlier in the same Chapter, lack of automation hinders fast creation of micro and nano scale assemblies using optical tweezers. This dissertation will focus on automated transport of spherical particles so that the framework can be utilized in automating the entire assembly operation in the future.

2.2 Metamodeling Techniques

2.2.1 Different Types

Surface fitting and interpolation are the two major classes of metamodels. Detailed comparison of different surface fitting techniques can be found in [Chiv95]. Broadly, surfaces can be represented in two ways – algebraic and parametric. Algebraic surfaces again can be either represented implicitly or explicitly. Some of the common fitting techniques include least squares, least absolute deviation, least median of squares fit, and patch method. The last technique is most popular in case of parametric surfaces. The authors have proposed several criteria namely, data collection method, intersection algorithms, constraint management, surface extension, local control, and data storage requirement to compare the two representation schemes. They have concluded that while algebraic surfaces offer greater flexibility in terms of fitting techniques and data collection, parametric surfaces are better suited for post-processing or downstream operations such as intersections, local control etc. Moreover, since algebraic surfaces are infinite and are not bounded by the bounding box of the measured data points unlike their parametric counterparts, they are good for extension type of operations. While algebraic surfaces provide an attractive option if the data is collected in a random fashion, parametric surfaces are better for gridded data.

Similarly, a comparative study of various scattered data interpolation methods has been carried out in [Amid02]. The methods surveyed in this paper can be classified into four categories – triangulation or tetrahedrization based methods, inverse distance weighted methods or the Shepard family of methods, radial basis function methods, and natural neighbor methods. Some of the criteria used for comparing the different methods include smoothness in terms of derivative continuity, possibility of extrapolation, local versus global control, efficiency, speed, suitability for large data sets, extensibility to higher dimensions, need for pre-processing steps, ease of use and so on.

The authors have concluded that none of the existing methods is universally satisfactory and it is the task of the user to select the one that is best suited for his/her application. Triangulation methods are local, capable of handling large data sets efficiently, computationally simple, and reasonably accurate. However, triangulation or tetrahedrization has to be performed before the actual interpolation step. Inverse distance weighting methods are usually less accurate and less efficient than the other techniques. However, they are commonly used for higher dimensional problems due to their simplicity. Radial basis functions (RBFs) are emerging as one of the most promising methods in terms of fitting power and visual smoothness. Natural neighbor interpolation methods are quite robust and perform equally well in cluttered as well as sparse areas of the given data sets. However, they are computationally more intensive and often require elaborate pre-processing operations in the form of construction of Voronoi tessellation. Although certain other methods such as Kriging are popularly used in certain applications like geology and mining, in general RBFs have been shown to have better performance.

In the panel discussion summarized in [Simp02], each of the five panelists gave one or two examples of use of approximation methods using surrogate models or metamodels, described the current state-of-the-art, and identified challenges and future research directions. Some of the common challenges include using Kriging techniques as opposed to typical response surface based methods for design and analysis of computer experiments, visualizing data from such models, capturing uncertainty, and handling problems with large number of variables. The merits of using sequential and adaptive approximation methods by combining response surface and Kriging techniques were also discussed by the panelists. It was pointed out that sequential sampling strategies such as Latin hypercube sampling can enhance the accuracy of the model in the regions of interest.

A review of metamodeling techniques from engineering design optimization perspective was conducted by Wang and Shan [Wang07]. The authors have discussed the pros and cons of several sampling methods, metamodel choices, and fitting techniques. They have pointed out that while Kriging and other Gaussian process models are accurate for nonlinear problems, they are difficult to obtain and use since a global optimization technique is applied to identify the maximum likelihood estimators. On the other hand, polynomial models are easy to construct and easy to use, but they are less accurate. RBFs provide a nice tradeoff between Kriging and polynomials. Recent advances in this field include use of Multivariate Adaptive Regression Splines, Support Vector Machines, Artificial Neural Networks, and hybrid models. As concluded in [Chiv95], each metamodel type performs better with a particular set of fitting techniques.

The authors are of the opinion that in spite of the recent surge in development of improved metamodels, approaches are still more or less infeasible for large scale problems. Effective decomposition strategies need to be devised in order to tackle the so-called "curse of dimensionality". They have also concluded that uncertainties in the metamodels themselves pose new challenges in optimization problems, although the models themselves can be used to filter out noise from computer simulation data. In addition, further work needs to be done to generate minimum number of sample points intelligently and develop a generic metamodeling framework so that models of variable fidelities can be used in different regions of the design space.

A study of metamodeling techniques and ensembles for evolutionary computing has been done by Lim et al. [Lim07]. They have studied four different metamodels, namely, Gaussian Process (GP), RBF, Polynomial Regression (PR), and Extreme Learning Machine (ELM). Empirical results reveal that while all the metamodels show improvement as compared to standard Genetic Algorithm (GA), some of the metamodels, namely, PR and GP exhibit great robustness. Interestingly enough, an ensemble of metamodels and multi-surrogates yields an even better solution quality on the same benchmark problems for an identical computational budget.

2.2.2 Radial Basis Functions

As discussed in the previous sub-Section, many of the limitations associated with the other techniques have been overcome using RBFs. They are circularly symmetric functions centered about a single point known as the mean or the center. They are quite popularly used by the computer graphics community over the past two decades or so. Other researchers have started applying them in the fields of engineering design, reverse engineering, and shape optimization over the past few years.

Some of the advantages include their compact representation, sensitivity to local features, ability to suppress noise in data, and ability to incorporate varying level of smoothness [Juba07]. Some of the common choices for RBFs are thin-plate splines, multiquadrics, inverse multiquadrics, polyharmonics, and Gaussians. One of the earliest uses of RBFs dates back to 1990, when Hardy [Hard90] applied multiquadrics to represent topological surfaces for given sets of sparse and scattered measurements.

Goshtasby [Gosh00] has used inverse multiquadric and Gaussian RBFs to partition a large set of non-uniformly spaced points in the plane into subsets and fit a parametric curve to each subset. This can be used to find patterns among the points and describe the physical phenomenon analytically if the points are obtained as measurements from a physical or simulation experiment. Similarly, if the points are measured from a geometric model, it can be used for model reconstruction purposes. This method is particularly well-suited for dense and noisy point sets. Test results show that the number and types of ridges and contours (parametric curves) depend on the choice of the basis function.

Carr *et al*. [Carr01] have used polyharmonic RBFs to reconstruct smooth, manifold surfaces from point clouds and to repair incomplete meshes. A greedy algorithm is used during the fitting process to reduce the number of RBF centers that are required to represent a surface. This makes it computationally feasible to represent complicated objects of arbitrary topology using RBFs. The scale-independent, "smoothest interpolator" characterization of polyharmonic splines makes them particularly suitable for fitting to non-uniformly sampled point clouds and partial meshes containing large, irregular holes. A non-interpolating approximation is used when the data is noisy. Test results on several examples, such as a laughing Buddha, a dragon, an asteroid, a human skeleton hand, and a turbine blade containing intricate internal structure, highlight the effectiveness of this fitting technique. One such example is shown in Figure 2.4, where 544,000 point cloud is represented by 80,000 centers with a relative accuracy of 5 X 10^{-4} in the final frame.

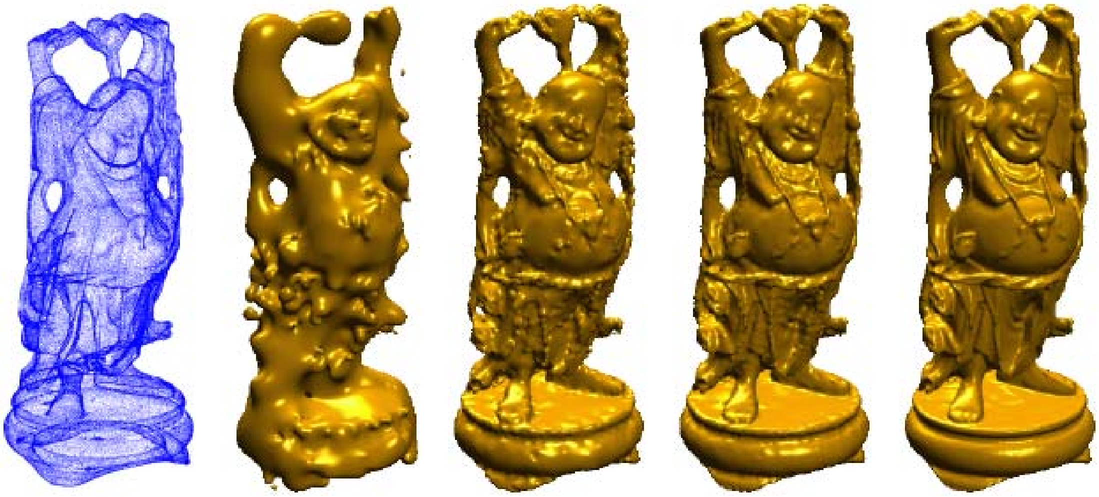


Figure 2.4: Iterative RBF fitting to a dense point cloud (Image source: [Carr01])

Turk and O' Brien [Turk02] have used RBFs to create interpolating implicit surfaces. A 3D implicit function is created by specifying 3D locations through this surface must pass (data points), and also identifying locations that are interior or exterior to the surface. The iso-surface of this function describes the desired surface. These surfaces can be easily used for constructive solid geometry operations, interference detections, interactive manipulation or sculpting, polygonal tiling, ray tracing and so on. These manipulations are difficult to achieve on other popular types of implicit surfaces, such as the sums of spherical or ellipsoidal Gaussian functions, known as blobby balls. This particular formulation also enables conversion of a polygonal model to a smooth, implicit model and formation of blends between two objects.

Weiler *et al*. [Weil05] have pointed out that although both splines and multiquadrics are considered to be accurate, Gaussian RBFs offer certain advantages over the other two types because they are computationally less challenging, concise, robust, and possess some desirable basis function properties. One such property is that they decay to zero rather than tending towards infinity; thus, if a particular region in space is poorly fitted by an RBF, then the overall modeling error is restricted. They are also less expensive to compute using modern graphics processing units (GPUs).

The authors have presented a kd-tree based method to fit Gaussian RBFs to an unstructured volumetric vector field. They have used principal component analysis to cluster and determine the centers of the Gaussian RBFs. They have also selected some of the centers to lie at the crests and the troughs of the low frequency components present in the data. An approximate, iterative method is used to solve the system of equation for the RBF weights quickly. This fitting framework has been coupled with GPU based volume rendering algorithms that proceed by slicing. For each fragment in a slice, a fragment program iterates over the RBF parameters, computes the scalar value at that location, and performs a lookup operation of the corresponding color from a 1D texture map.

Wu *et al*. [Wu05] have developed a new scheme for 3D reconstruction of implicit surfaces from large scattered point sets using RBFs. A partition of unity (POU) method and binary tree is used to organize the point sets into overlapping local sub-domains. A local surface is then reconstructed for each of the sub-domains from non-disjunct subsets of the points. In contrary to the conventional RBF fitting methods, where a full set of off-surface points are used in each sub-domain, this scheme uses a single off-surface point that avoids a trivial solution to the system of linear RBF equations. Careful choice of this offset point is shown to yield an efficient and robust local fitting technique that offers a high level of scalability. Global solution is obtained by combining the local solutions with the POU equations. The level set propagation methodology for dynamic surfaces is adapted to smooth the reconstructed surfaces using curvature diffusion. Test results on various standard graphics examples like Stanford bunny, Venus face etc. show satisfactory results.

Mullur and Messac [Mull05] have combined the effectiveness of multiquadric RBFs with the flexibility of non-radial basis functions (N-RBFs) to come up with extended radial basis functions (E-RBFs) to explore new frontiers in engineering design. The authors have pointed out that although RBFs offer lots of advantages over traditional response surface methods (RSMs) in terms of handling multi-dimensional interpolative approximations, they are not very useful in uncertainty handling and multi-objective optimization. A linear programming based approach has been developed to impose convexity constraints on the E-RBFs. This is another significant modification to the characteristics of traditional RBFs.

Comparative study of E-RBFs with RBFs and RSMs over seven surface functions shows that E-RBFs perform consistently well under widely varying conditions from simple polynomial functions to highly non-linear large scale problems. On the other hand, RSMs perform well for problems with low non-linearity, and RBFs perform satisfactorily for highly non-linear but not very large scale problems. However, further work needs to be done to investigate the performance of E-RBFs for problems having more than 100 variables. Similarly, the possibility of incorporating constraints other than convexity can also be explored.

Wu *et al*. [Wu06] have presented two approaches for 3D surface reconstruction from laser scanned data sets using RBFs. In the first method, a point set is organized using a balanced binary tree. Similar to the technique proposed in [Wu05], only one off-surface point in the quasi-normal direction (obtained by Eigen analysis) is used for RBF interpolation in every sub-domain. These sub-domains are controlled such that they overlap mildly and contain an adequate number of points for efficiency and stability. Another least squares RBF fitting technique is proposed that overcomes the problem of numerical ill-conditioning and over-fitting of traditional techniques. Although it is time-consuming, it can represent surfaces with lesser number of RBF centers.

Wei and Wu [Wei06] have used RBFs in conjunction with level sets to solve structural shape and topology optimization problems. They have come up with a parametric representation of level sets, which is shown to have certain advantages over the conventional discrete representation. The propagation of the front or the boundary of the implicit surfaces (modeled by RBFs) towards an optimal shape and topology by solving the Hamilton-Jacobi partial differential equation is reduced to the evolution of the ordinary differential equation of the RBF coefficients or parameters. Steepest descent method has been used here to perform gradient based search in the parametric space.

Two integration schemes have been applied in this paper for different applications. The boundary integration scheme is useful for shape optimization and fluid simulation problems, where the velocity extension procedure is time-consuming. On the other hand, the volume integration scheme applies velocity extension. Hence, there is no need to explicitly trace the front curves of the implicit surface, thereby enhancing the efficiency of the numerical computation. Both these schemes avoid the re-initialization process that is unavoidable with the discrete level set method. Numerical experiments on minimum compliance design of a 2D structure show promising results, although the stability and convergence of this method need to be formally investigated in future.

2.2.3 Summary

From the discussion in the previous two sub-Sections, it is clear that RBFs form an ideal choice for the current application as they can be regarded as some sort of a hybrid between pure surface fitting (like response surfaces) and interpolation techniques (like Kriging). Not only do they preserve local features that may be present in the high-dimensional trapping probability surface, they are also robust with respect to outliers that typically occur in any stochastic experiment. Among RBFs, Gaussians seem to be the most appropriate choice. Although E-RBFs may be very accurate, they are not required here as the parametric space is not going to be of very high dimensions (expected to be less than 10). Similarly, an ensemble of metamodeling techniques is not required as evolutionary computing will not be carried out here. For implicit surface representation of large, noisy data sets, Gaussian RBFs are effective on their own.

Arrangement of RBFs in the form of a data structure that allows fast querying operations is also important in the current work. Binary trees are well-suited for this purpose. That is why, a specialized form of binary tree, namely the kd-tree is utilized in the current work. The effectiveness of combining RBFs with a kd-tree based spatial partitioning scheme has been already demonstrated by Weiler *et al*. [Weil05]. Some other reasons for selecting kd-trees over octrees (or their higher dimensional versions) have been explained later in Chapter 3. However, so far no work has been done to characterize the extent of usefulness of any RBF fitting procedure to micro or nano scale assembly simulation data. This will be investigated in the current dissertation by repeating the same set of simulation experiments for particles of different sizes having varying levels of stochastic motions.

2.3 Single Robot Motion Planning under Uncertainty

2.3.1 Classical Techniques

Single robot motion planning has received a lot of attention over the past few decades. Broadly, the methods can be classified into two types: planning with complete information or with incomplete information. Inspired by the Piano Mover's problem, a large number of problems have been studied in the former domain [Schw86]. Many of the approaches utilize the notion of *configuration space* (*C*-space) originally introduced by Lozano-Perez [Pere83]. In *C*-space, the robot is shrunk to a point, and the surrounding objects, called obstacles, are expanded accordingly. For polygonal or polyhedral robot and obstacles, this amounts to computing the Minkowski sum of the robot polygon (or polyhedron) with the image of the obstacle polygon (or polyhedron) under consideration. Free subspace of *C*-space is the complement of the union of the expanded obstacles. Any path that lies in a continuous subset of the free space is a physically realizable path for the point robot.

Latombe has provided a detailed description of the popular methods in this complete information or offline planning category, including road map, cell decomposition, potential field etc. [Lato91]. In the roadmap approach, the free *C*-space is retracted, reduced to or mapped onto one-dimensional lines. Hence, this approach is also known as Retraction, Skeleton, or Highway approach. The search space is limited to this network, and hence, it reduces to a graph search problem. Some of the well-known roadmaps include Visibility graph, Voronoi diagram, Silhouette, and the Subgoal network. In the cell decomposition algorithms, the free *C*-space is decomposed into a set of simple cells and the adjacency relationships among the cells are computed. A collision-free path between the start and the goal configurations of the robot is obtained by identifying those two points and then connecting them with a sequence of connected cells. In the potential field method, a robot is represented in *C*-space as a point particle moving under the influence of an artificial potential field whose local variations represent the nature of the free space. This function can be defined as the sum of an attractive potential pulling the robot towards the goal configuration, and repulsive potentials pushing it away from the obstacles.

The other class of incomplete information based planning problems has been studied only in the recent past. Lumelsky has presented a detailed survey of sensing, intelligence and motion based approaches to solve these problems [Lume06]. He has developed a class of Bug algorithms for a planning in 2D using tactile sensing only in unstructured environments consisting of arbitrary-shaped, but finite-sized obstacles. These algorithms have been extended to deal with visual sensing, based on the sensor's circular field of vision. Significant modifications to the basic algorithms have made to apply them in case of two or three-dimensional robot arm manipulators. Thus, these algorithms are potentially very useful in industrial settings where most of the actual job is performed by arm manipulators.

Masehian and Sedighizadeh [Mase07] have performed a chronological review of the existing classical and heuristic approaches in robot motion planning. As expected, the heuristic techniques such as artificial neural network, simulated annealing, genetic algorithm, fuzzy logic, particle swarm optimization, ant colony optimization, tabu search, wavelets etc. have increased in popularity over the past decade or so. Even so, *C*-space and Voronoi diagram based approaches are still being used widely in different applications.

2.3.2 Probabilistic Roadmaps

Kavraki *et al*. [Kavr96] first presented the technique of probabilistic roadmaps (PRMs) for path planning in higher dimensional, static *C*-spaces. The technique proceeds in two phases: a learning phase and a query phase. In the learning phase, a probabilistic roadmap is constructed and stored as a graph. The nodes or vertices of the graph correspond to collision-free configurations and the edges represent feasible paths between those configurations. The nodes are selected using uniform random distribution based sampling of the free *C*-space and the paths are computed using a fast, local planner. In the query phase, any given start and goal configurations of the robot are connected to two nodes of the roadmap and it is then searched for a path joining these two nodes. The technique is easy to implement and can be applied to almost any holonomic robot. Although efficiency depends on the selection of certain scene and robot dependent parameters during the learning phase, this is a relatively easy operation. Experimental results on planar, articulated robots with many degrees of freedom show that planning can be done in fraction of a second after learning for a few dozen seconds. Theoretical results bounding the number of roadmap nodes required for solving planning problems in spaces with certain geometric properties have been presented by Kavraki and Latombe [Kavr98].

An obstacle based PRM for obtaining high quality roadmaps for crowded scenes has been developed by Wu [Wu96]. The salient feature of this approach is that roadmap candidate points are chosen on the constraint surfaces corresponding to obstacles present in the workspace. Hence, it is likely to contain difficult paths that traverse through long, narrow passages. This is useful for contact planning apart from collision-free path planning.

Boor *et al*. [Boor99] have presented a new Gaussian sampling strategy for PRM based planners. This gives better coverage in the difficult portions of the free *C*-space. Test results on a rectangular robot with three degrees of freedom in scenes involving small corridors, and large number of obstacles show significant reduction in the number of nodes as well as the time taken to complete sampling in comparison to random sampling based technique. This reduction in time is expected to be even more important in case of three-dimensional robots having six degrees of freedom as the time required for adding a node is much more than the time required for testing a sample.

Branicky *et al*. [Bran01] have proposed the use of quasi-random sampling techniques in the context of PRM framework. Two quasi-random variants of PRM based path planner have been implemented in this work: (a) a classical PRM with quasi-random sampling, and (b) a quasi-random Lazy-PRM. The main feature of Lazy-PRM is that the roadmap is initially constructed without considering collisions. Once a query is given, A^* search is performed to obtain a solution. If any of the solution edges are in collision, they are removed from the roadmap and A^* search is repeated. In the worst case, all edges may have to be checked for collision; however, the problem is usually solved well before this happens. Both the techniques show significant computational benefits in terms of reduction in the number of nodes (and corresponding query operations) as compared to their randomized counterparts.

Apaydin *et al*. [Apay02a] have extended this technique to develop a new algorithm, known as stochastic roadmap simulation (SRS) to explore the kinetics of molecular motion by examining multiple pathways simultaneously. This avoids the common problem encountered by many other molecular motion simulation techniques, such as Monte Carlo, which generates one pathway at a time and, thereby, spends most of the time in overcoming local minima of the energy landscape defined over a molecular conformation space. SRS encodes all the paths compactly in the form of a graph that is constructed by randomly sampling the molecular conformation space. Each edge represents a potential transition of the molecule from one conformation to the other and is associated with a probability denoting the likelihood of this transition. This graph can be viewed as a Markov chain, leading to efficient computation of ensemble properties. Application to two biological problems, namely, probability of protein folding, and estimated, expected time to escape from a ligand-protein binding site shows that SRS yields slightly more accurate results than Monte Carlo simulation. Moreover, computation time is reduced by several orders of magnitude.

In another related work, Apaydin et al. [Apay06b] have applied SRS in studying ligand-protein interactions. They have considered a parameter called escape time, which is the expected number of simulation steps required for the ligand to escape from the 'funnel of attraction' of the binding site. This parameter acts as a metric for analyzing the interactions. The results for six mutant complexes agree well with biological interpretations, while those for seven ligand-protein interactions show the effectiveness of the metric to distinguish the catalytic site in five cases.

Hsu *et al*. [Hsu02] have also extended the PRM framework to develop a randomized motion planner for robots that must achieve a specified goal under kinematic and/or dynamic motion constraints without colliding with moving obstacles having known trajectories. Motion constraints are encoded with a control system that samples the robot's state X time space by selecting control inputs randomly, and then integrating the equations of motion. The roadmap is not pre-computed here; instead, for every query, a new roadmap is generated to connect the initial and the goal state X time point. It is shown that if the state X time space satisfies the expansiveness property, then a slightly idealized version of the implemented is guaranteed to locate a trajectory, if one exists, with the probability quickly approaching 1, as the number of points increases. This planner has been tested both in simulated environments as well as on a real robot. If a change in the expected obstacle motion takes place while the robot is executing the planned trajectory (based upon visual information regarding obstacle motion before planning starts), then the robot recomputes the trajectory on the fly. Extensions to the planner have been carried out to deal with time delays and uncertainties that are inherently present in any robotic system interacting with the physical world.

Leven and Hutchinson [Leve02] have presented a technique similar in nature to the PRM approach for generating collision-free paths for robots operating in changing environments. Unlike the traditional PRM approach, the graph that represents a roadmap in the *C*-space is not constructed for a specific workspace. Instead, it is constructed for an obstacle-free workspace that encodes a mapping from workspace cells to nodes and arcs (or edges). When the environment changes, this mapping is used to make appropriate modifications in the graph, and plans are generated by searching the modified graph. Random samples of the *C*-space, generated using an importance sampling approach, are connected together to form the roadmap. Compression schemes are used to exploit redundancy in encoding mapping information. Implementation on serial-link manipulators shows that few seconds are required to create the roadmap for a 20 joint planar robot and a 6 joint 3D robot. Enhancement of the robustness of the roadmap with respect to changes in the environment increases this processing time to minutes and even hours depending upon the number of nodes.

Lot of work has also been carried out on replanning in unknown and dynamic environments. Bruce and Veloso [Bruc02] have introduced extensions to previous work on Rapidly-Exploring Random Trees (RRTs), which themselves are variants of the PRM technique. Their planning algorithm, known as execution extended RRT, uses a waypoint cache and adaptive cost penalty search to improve the replanning efficiency and the quality of the generated paths. Simulation and physical experiments on a RoboCup F180 robot controller show that the robot can travel significantly faster, especially when kd-tree is used for nearest neighbor searching while reducing oscillations and local minima problems encountered by a reactive scheme. However, post-processing steps and better heuristics may be required to correct the inaccuracies in the kinematics model and reduce the planning time further respectively.

Zucker *et al*. [Zuck07] have presented a Multipartite RRT (MP-RRT) algorithm that bias the sampling distribution and re-use branches from previous planning iterations for rapid replanning in dynamic environments. Simulation experiments show that this algorithm outperforms some of the existing RRT techniques in high-dimensional configuration spaces with unknown moving obstacles, especially when a greedy smoothing heuristic is used to incrementally smooth the path in the configuration space. However, it needs to be verified whether the MP-RRT algorithm is capable of generating more robust plans as compared to the existing approaches.

A greedy but probabilistically complete (and hence safe) state space exploration strategy for replanning under kinodynamic constraints in non-holonomic systems using incrementally updated tree data structure has been presented in [Bekr07]. The authors have built upon the previous work using RRTs to reduce the cost of achieving safety by controlling the planning cycle duration and employing lazy evaluation. Searching is biased using information extracted from sensor data as opposed to "Voronoi-bias" selection strategies employed by previous techniques. Simulated experiments indicate favorable computational performance against existing alternatives. However, this approach does not account for uncertainty in the form of sensor noise and positioning error.

Rodriguez *et al*. [Rodr07] have extended the roadmap approach to come up with a heuristic framework for motion planning in environments with moving obstacles. This approach assumes that the robot has no knowledge about the trajectories of the moving obstacles. It also distinguishes the obstacles into two types: hard with which collisions must be avoided at all costs, and soft referring to other agents or robots with which some collisions are permitted. A global roadmap is first constructed taking the hard obstacles into account only. This global map is then integrated with kinodynamic local planning to consider the soft obstacles, and neighboring agents to compute safe plans. The local paths are updated as the environment changes. Numerical experiment results on various test cases including corridors, rotors and attacking agents environments show that successful plans can be generated about (70-80) % of the time if about 10-15 soft obstacles are present. Typically, few hundred steps are needed to generate a successful plan within few tens of milliseconds. However, further work needs to be done to apply this framework in real-world settings.

A similar notion of global roadmap and local collision avoidance is used by Sud *et al*. [Sud07a] to present an approach for real-time path planning for multiple virtual agents in complex, dynamic scenes. They have introduced a new data structure, called multi-agent navigation graph (MaNG). This is constructed using first and second order Voronoi diagrams. Voronoi diagrams encode the connectivity of the space and provide a path of maximal clearance for a robot from other obstacles. While first order Voronoi diagrams contain all the points that are closest to a site, second order diagrams contain all the points that are closest to two given sites. Combination of the two avoids the computational burden associated with considering every agent separately, and treating all the other agents and environment objects as obstacles.

MaNG is computed using graphics hardware and culling operations are performed to achieve speed-up. Applications of this algorithm in pursuit-evasion (fruit stealing) and crowd simulation scenarios consisting of hundreds of moving agents, each with a distinct goal, show that 5 to 20 frames per second (fps) performance can be achieved in case of 15 static obstacles and few (2 to 5) moving obstacles. Running time analysis shows that the construction of discrete Voronoi diagram is the bottleneck step and has a cost of $O(rm^2 + m \log n)$ associated with it, where n is the number of sites, m X m is the grid size, and r depends on the tightness of the computed Voronoi region bounds. Although this approach computes an optimal path for every frame, it does not guarantee coherence of paths across frames, or convergence over a period of time.

Further work on real-time navigation of independent agents in dynamic environments using lazy roadmap based technique has been done by Sud *et al*. [Sud07b]. Another approach for motion planning of multiple robots in dynamic environments has been proposed by Gayle *et al*. [Gayl06]. They have developed a new data structure, called the Reactive Deformable Roadmap, such that each link in this roadmap can be deformed in response to the motions of other robots or obstacles obeying Newtonian physics and Hooke’s law to compute collision-free paths. Yan *et al*. [Yan06] have integrated environment information and constraints for unmanned air vehicle into the procedure for building flight roadmaps for mission-adaptable route planning in uncertain and adversial environments. Missiuro and Roy [Miss06] have proposed an extension of the PRM algorithm that computes motion plans that are robust to uncertain maps. This generates less collision-prone trajectories with fewer samples than the traditional approach.

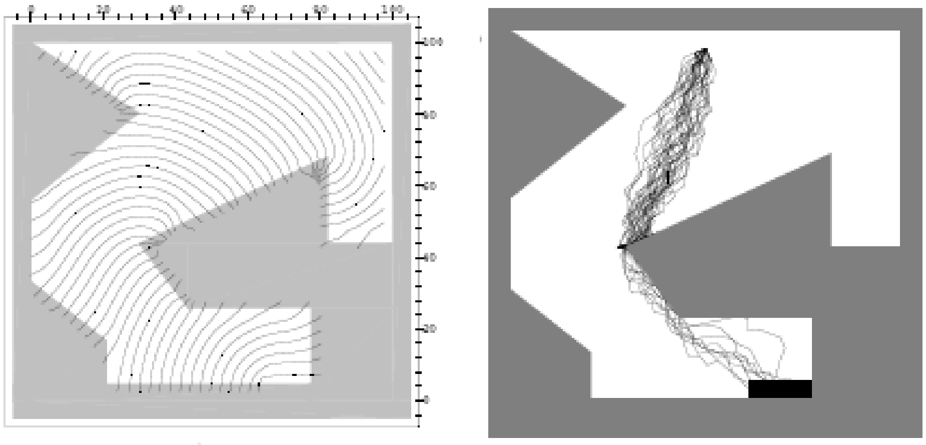
2.3.3 Stochastic Dynamic Programming

LaValle and Sharma [LaVa97] have presented a framework for robot motion planning in environments that is changing with time, and is not completely predictable. They have classified the sources of uncertainty into four types: uncertainty in configuration sensing, configuration prediction, environment sensing, and environment prediction. Although the paper primarily deals with the last type of uncertainty, several extensions have been presented that can handle other types of uncertainties as well.

The changing environment is modeled in a flexible manner by combining *C*-space approaches with a Markov process. The notion of a motion strategy has been used to provide a motion command to the robot for every contingency plan that it is confronted with. A desired performance criterion, such as time or distance, is specified and an optimal motion strategy with respect to that criterion is determined using dynamic programming based algorithm. Although the paper focuses on a finite environment space of reasonably small size, applications on various problems involving changing *C*-space, hazardous regions and shelters, and processing of random service requests clearly highlight the benefits of the approach.

LaValle and Hutchinson [LaVa98] have presented an objective based framework for motion planning under both and control and sensing uncertainties. They have generalized classical preimages [Lato91] to performance preimages and preimage planning to motion strategies with information feedback by combining concepts of stochastic optimal control and dynamic game theory with preimage planning approaches. They have considered the executed robot trajectory as the performance criterion. They have presented dynamic programming based algorithms to numerically compute performance preimages and motion strategies that will optimize the this performance criterion either under nondeterministic uncertainty (worst case analysis) or under probabilistic uncertainty (expected or average case analysis). A large number of computed examples of performance preimages, and optimal motion strategies show the fundamental differences between nondeterministic and probabilistic analysis, and the effect of the two types of uncertainties either separately or in conjunction to each other. While the isoperformance curves come closer to each other in case of worst case analysis as the computed costs are higher, the curves become crowded and do not align perfectly with the obstacles if sensor imperfections are included. The sample paths under implementation of the optimal strategies also show greater variation from one simulation to the other in the latter case. Isoperformance curves and sample trajectories using probabilistic analysis for a particular test case involving both forms of uncertainties is shown in Figure 2.5.

A unified, game-theoretic framework for design and analysis of algorithms to solve a wide class of robotics problems has been proposed by LaValle [LaVa00]. This is motivated by the similar benefits that were achieved by using a unified *C*-space approach for basic path planning problems. The proposed approach can handle many kinds of complications such as sensing and modeling uncertainties, nonholonomy, robot body dynamics, multiple robots and goals, optimality criterion, unpredictability, and nonstationarity in addition to geometric workspace constraints. Other than robotics, this has potential applications in virtual prototyping, computer graphics, and computational biology. By adopting this approach, a general dynamic programming based algorithm has been obtained for computing approximate solutions to a broad class of motion planning problems. This provides useful flexibility in substituting uncertainty models, understanding the relationship between nondeterministic and probabilistic uncertainty models, and replacing approximate methods by precise formulations for mathematical analysis.



(a) Isoperformance curves

(b) Sample trajectories

Figure 2.5: Optimal motion strategies under probabilistic uncertainty and imperfect state information using dynamic programming based approach (Image source: [LaVa98])

Alterovitz *et al*. [Alte05] have used a discrete version of the infinite horizon dynamic programming to compute an optimal control sequence for steering flexible needles with bevel tips inside soft tissues. These needles can be steered around obstacles to reach targets inaccessible to rigid needles by controlling two degrees of freedom at the needle base (bevel direction and insertion distance). This is analogous to nonholonomic motion planning for a Dubins car without any reversal.

By formulating the planning problem as a Markov decision process (MDP), this approach handles control action uncertainty in needle motion due to uncertainties in tissue properties, needle mechanics, and interaction forces. The expected cost to reach a target in the imaging plane due to insertion distance, direction changes, and obstacle collisions is minimized. The discretization errors as a result of sampling of the state space of needle tip positions and orientations are bounded mathematically. Comparison with the deterministic motion case clearly shows that significantly different motion plans are obtained in the two cases. However, further work needs to be done to investigate the nature of the state transition probability distribution and improve computational performance for interactive applications or larger state spaces with smaller discretization errors.

An attempt to combine probabilistic roadmaps with partially observable Markov decision process (POMDP) has been made by Prentice and Roy [Pren07]. They have developed a belief-space variant of the probabilistic roadmap algorithm, called the Belief Roadmap (BRM), by using a factored form of the covariance matrix associated with linear Gaussian systems. This factored form enables combining several prediction and measurement steps into a single linear, one-step transfer function, thereby leading to more efficient posterior belief prediction during planning. Theories from linear filtering and optimal control have been applied to prove the covariance factorization theorem that can then be utilized for updating the Extended Kalman filter instead of solving the non-linear Riccati equation. Experimental results indicate that BRM provides more accurate localization than shortest path planning algorithms under conditions of varying noise and small maximum sensor range. Computation time is also reduced by two orders of magnitude and scales well with respect to the path length and depth of the search tree.

Doshi and Roy [Dosh08a] have utilized the structure of many preference elicitation problems to obtain POMDP solutions with exponentially fewer belief points while retaining the quality of the solution. In such problems, the agent robots must discover some hidden preferences from another (often human) agent. The approach is based upon the assumption that the agent's optimal control action type depends only on its uncertainty over states and not on the particular states that it is confused about. Empirical simulation results show that computation can be sped up by several orders of magnitude; however, more compact encoding schemes for models will be useful for large scale problems.

Doshi *et al*. [Dosh08b] have presented an approximate Bayesian reinforcement learning technique for obtaining the POMDP model parameters by treating them as additional hidden states in a "model-uncertainty" POMDP. This framework of simultaneous learning and planning avoids the problem of long training periods of the robots or agents to acquire the POMDP parameters from the domain of operation while guaranteeing correctness and convergence. The notion of meta-queries has been introduced to accelerate learning and infer the effects of potential disasters without actually experiencing them. Results on some well-known POMDP examples, involving both discrete and continuous-valued parameters, show that this approach performs better than the non-learning as well as the passively learning versions. However, more efficient sampling strategies as well as heuristics for allocating more computation to better solutions are required for real-time applications.

Thrun *et al*. [Thru05] have presented a comprehensive survey of techniques in the emerging field of probabilistic robotics. It is a subfield of robotics pertaining to perception and control. By relying on statistical techniques for representing information and decision-making, it accommodates the uncertainty that arises in different applications. All the algorithms are based on a single, underlying mathematical foundation: Bayes rule and its temporal extension, known as Bayes filter. This book presents both discrete and continuous versions of the value iteration equation based upon Bellman optimality equation for a Markov decision process. A finite world POMDP algorithm is also described to capture sensor uncertainty in addition to control action uncertainty, and perform value iteration over the belief state instead of the actual state. Finally, approximate versions of the previous algorithm are presented to achieve superior computational performance. Complete mathematical derivations of all the assumptions from first principle, test cases to illustrate behavior, and detailed discussion of the strengths and weaknesses of each algorithm make this book particularly useful for understanding and comparing different planning approaches capable of dealing with uncertainties.

An up-to-date, text book style presentation of both geometric and probabilistic motion planning approaches, along with techniques for position estimation and mechanical control systems can be found in [Chos06]. LaValle [LaVa06] has presented a unified treatment of different kinds of planning algorithms. All the dynamic programming based approaches have been grouped together under the category of decision-theoretic planning. A detailed study of such algorithms has also been done by Kane *et al*. [Kane06]. Since LaValle's book covers a wide variety of approaches ranging from graph search techniques (like A*, Dijkstra's algorithm etc.) for discrete planning, to probabilistic roadmaps and randomized potential fields for sampling based motion planning, and navigation functions for feedback motion planning, it provides a very good reference for comparing fundamentally different approaches.

2.3.4 Summary

Discussion of different motion planning approaches shows that dynamic programming has been used most widely to deal with uncertainty. Although recently researchers have started working on extending the probabilistic roadmap framework, algorithms that can generate theoretically guaranteed optimum strategies in real-time for a large number of unpredictably moving obstacles have not yet been developed. Moreover, unlike in other approaches, both forms of uncertainty, namely control action and sensing can be easily incorporated within the dynamic programming framework. Since the current dissertation focuses on transporting particles in fluid medium where all other particles (obstacles) are exhibiting Brownian motions, probabilistic models are better suited to represent uncertainties than nondeterministic bounded intervals. This is because particle positions obey normal distributions, with mean displacements equal to zero, and variances related to the diffusion lengths. In other words, the current planning problem can be mathematically represented as a POMDP. One of the approximate POMDP algorithms, known as the QMDP algorithm described in [Thru05], has been adapted in this work to achieve computational efficiency.

However, in its basic form, this algorithm cannot compute the optimum solution within a few milliseconds for environments consisting of as many as 50 or more constantly moving obstacles. So pruning conditions have to be developed to reduce the search space drastically. Similarly, means of attainting faster convergence of the value iteration equation have to be explored. Finally, no POMDP algorithm currently accounts for the change taking place in the environment within the time it takes for the planner to compute the optimum strategy. This effect needs to be considered here due to the rapidity of the changes occurring in the environment.

2.4 Decoupled Multiple Robot Motion Planning

2.4.1 Methods

Most multi-robot planning algorithms can be either categorized as *coupled* or *decoupled*. Coupled algorithms combine the states of all the individual robots into a single state space representation, and plan the paths concurrently. The method developed by Svestka and Overmars [Sves98] uses probabilistic roadmaps for static environments. It first constructs a roadmap for every robot and then combines these simple roadmaps into a composite one. Simulation results on five car-like robots in complex scenes show that the problem can be solved in the order of seconds once the pre-processing step is completed in few minutes. The primary limitation is that the size of the configuration space grows exponentially with the number of robots in the system.

On the other hand, decoupled methods plan for the motion of individual robots separately. One approach is to decouple path planning with mutual collision avoidance. This implies that obstacle-free paths are obtained first. Then, the velocities of the individual robots are adjusted to avoid collisions [Peng05]. Alternatively, a geometry-based, coordination-diagram approach can be utilized to independently combine the generated paths of many robots such that collisions are avoided [Sime02]. A bounding box representation is used for every obstacle so that the *n*-dimensional coordination space can be represented implicitly by exploiting the cylindrical structure of the coordination diagram obstacles. This method is resolution-complete and is able to efficiently handle up to 10 robots in the worst-case and more than 100 robots in practical situations. However, extensions are necessary to determine the groups of few robots that may cause deadlocks resulting in the approach returning no solution. In that case, a centralized planner may be needed to resolve the local conflicts.

Decoupled methods may also be classified as using a *centralized* or a *decentralized* architecture. The latter type uses independent planning techniques such as potential fields [Barr91]. Lumelsky and Hariharan [Lume97] have used a cocktail party model, where they have assumed that the robots have no prior knowledge about the scene (consisting of stationary obstacles only) or about the paths and objectives of the other robots. The robots do not communicate with each other and no restrictions are imposed on the paths or shapes of the robots and the obstacles. Clearly, no provable strategy can be employed in this situation; the maze searching based heuristic method demonstrates good performance and robustness characteristics in simulation experiments. However, like any other decentralized technique, this method cannot guarantee convergence.

The centralized planners typically compute the individual paths sequentially in a single processor and then combine the plans to avoid collisions. Combination of plans inherently involves assigning priorities to robots, which affects the quality of the resulting solution. Such priority assignment is carried out either by considering all possible combinations of priorities as shown for three robots in [Azar97]. Bennewitz *et al*. [Benn01] have used an optimization process to overcome the limitations of incompleteness and lack of optimality. Unlike many other approaches that apply a single priority scheme which often fails in cases when valid solutions exist, the authors have used a randomized search with hill climbing to generate solutions and minimize the path lengths. Constraints obtained from the task specifications are used to focus the search. Both physical (on CS-Frieburg RoboCup team as shown in Figure 2.6) and simulation experiments (in cyclic corridors and unstructured environments) show the superior performance of this method both in terms of the plan efficiency as well as the ability to find valid plans.

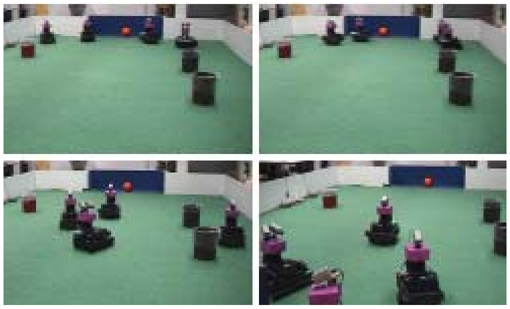


Figure 2.6: Coordinated path planning for four robots in the CS-Freiburg RoboCup team (Image source: [Benn01])

An alternative multiphase planning approach that is scalable to a large number of robots linearly has been presented recently in [Peas08]. The authors have used a graph and spanning tree representation to create and maintain obstacle-free paths for each robot. Simulation in an underground mine environment consisting of tunnels and corridors takes less than 1.5 s for 100 robots in a 1.5 GHz processor. However, the process of generating a suitable roadmap or graph representation for arbitrary obstacles and nonholonomic motion constraints can be quite challenging. Also, it may be more advantageous to consider a hybrid approach that can take advantage of the useful features of both sequential and multiphase planners. Further work is also required to implement this method on a distributed architecture instead of the current centralized one.

2.4.2 Summary

It is clear from the preceding discussion that in general decoupled methods offer significant computational benefits over any coupled approach. However, some of the decoupled techniques suffer from lack of completeness and optimality. That is why, the priority optimization method presented in [Benn01] has been selected as the candidate for developing a planning approach to transport multiple particles. Although the multiphase method described in [Peas08] has better scalability, in its current form, its usefulness for rapidly changing environments is not clear.

It is useful to note here that none of the current techniques consider the problem of selecting goals for robots. All of them assume that the assignment of goals to individual robots is already specified in the problem statement or definition. However, that is not necessarily the case for transporting particles using optical tweezers. In certain situations, the planner has to generate the assignment automatically based on some criterion that will minimize the expected time to complete the full operation. So an optimum way of doing that needs to be explored in the current work. Moreover, ways to reduce the search space so that optimal priorities can be determined within few hundred milliseconds for a reasonable number of particles (10-20) also have to be considered.

2.5 Indirect Optical Manipulation of Cells

2.5.1 Techniques

As mentioned earlier in this Chapter, optical tweezers have been used extensively to trap and manipulate cells by focusing laser beams directly on them. However, due to the tight focusing of the laser, considerable photodamage can be inflicted on trapped cells, including the death of cells as noted by Ashkin [Ashk89]. A range of assays have shown that focused laser light can also affect cell function without destroying the cell. The underlying mechanism for photodamage has been proposed to be due to the creation of reactive chemical species [Svob94b, Liu96], local heating [Liu96], two-photon absorption [Koni95, Koni96] and singlet oxygen through the excitation of a photosensitizer [Neum99]. These studies caution that direct cell trapping may not be desirable.

That is why researchers have recently started investigating the possibility of manipulating cells by indirect means. Sun *et al*. [Sun01] have used irregularly shaped diamond microparticles as handles for manipulating cells. Both the direction and speed of rotation of the diamond particles can be easily controlled in a linearly polarized beam with fundamental Gaussian mode due to the irregular optical shape of the diamond particles. The authors have been successful in imparting independent linear translation and rotation to the mesophyll protoplast cells. Ferrari *et al*. [Ferr05] have mimicked the mechanical environment of living cells in tissues by surrounding them with optically trapped micron-sized beads as shown in Figure 2.7. They have used two different tweezer set-ups, namely one 2D set-up generated by acousto-optical deflectors and another 3D set-up using diffractive optical elements. By dynamically varying the configuration geometry and the trapping forces, the authors have been successful in creating a controlled environment where only mechanical stimuli are present and the biological response is monitored.

Paterson *et al.* and Dholakia *et al.* [Pate05, Dhol07] have demonstrated passive optical sorting of lymphocytes from a mixed population containing both lymphocytes and erythrocytes in the absence of any externally driven fluid flow using Bessel light beam. This separation is enhanced by attaching silica microspheres coated with antibodies to a subpopulation of lymphocytes by creating a larger difference in effective refractive index between the fluid medium and the component being transported. This principle of tagging silica microspheres proves to be significantly more efficient than simple passive optical separation in case of bone marrow and progenitor/stem cells [Pate07]. The cells collected after separation are also viable, as evident from the facts that they are able to clone *in vitro* and show a lack of expression of Caspase 3, a marker of apoptosis.

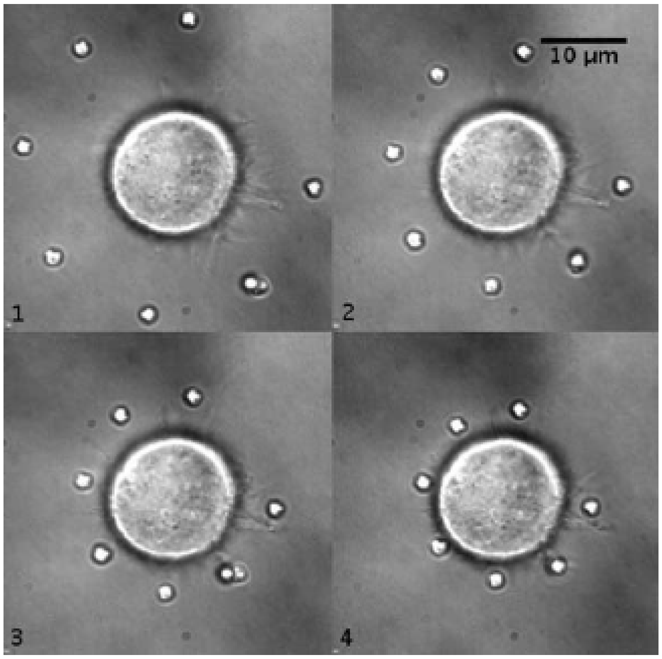


Figure 2.7: Indirect ND cell manipulation using 2 μm diameter latex beads in an acousto-optical deflector based optical tweezer set-up (Image source: [Ferr05])

2.5.2 Summary

From the preceding discussion it is clear that indirect cell trapping and manipulation offers considerable advantages over direct trapping in terms of the damage incurred to the cells. Successful demonstration of using optically trapped microspheres as handles to move, sort, or stimulate cells means that they can also be used to transport cells over large distances in the workspace and then form regular arrangements by accurate positioning. As mentioned in Chapter 1, such arrangements can be used for quantitative studies on cell-signaling. However, this will first require transporting the microspheres to the cell boundary before the cell-sphere group can be moved together. This dissertation provides a step towards image-guided, autonomous manipulation of cells by developing the framework for transporting multiple microspheres simultaneously.

Chapter 3

GENERATING SIMPLIFIED TRAPPING PROBABILITY MODELS FROM TRAP-PARTICLE SIMULATION

Simulations have emerged as a powerful tool in the traditional manufacturing and assembly areas to facilitate a wide variety of decision making [Vin06, Liu06]. We expect simulations to play a major role in micro and nano scale assembly as well. However, simulations at the small scales pose new challenges due to the presence of a higher level of uncertainty. Since path planning will involve trapping the desired particles and avoiding collisions with other particles present in the workspace, knowledge of trapping probability is very useful. Many decisions need to be taken within a few milliseconds during real-time planning; so the trapping probability estimation method must be extremely fast. As already discussed in Chapter 1, trapping physics based simulations are computationally intensive. Currently, it takes few seconds to complete a single simulation run at any particular point in the space. Moreover, multiple simulation runs are required at every point to account for the stochastic nature of the motion. The raw data generated by simulation is also expected to be very large in size due to the need for varying several different parameters in a combinatorial manner. This Chapter describes a computational framework to perform offline micro scale particle motion simulations and proposes a model simplification technique to represent the raw data in a compact form to estimate probabilities at any arbitrary point in the parameter space at run-time. The approach presented in this Chapter is reported in [Bane08a, Bane09a].

3.1 Framework to Simulate Sphere Behavior in Single Beam OT System

A computational framework has been developed using the C++ programming language to simulate the physical behavior of a sphere in a single beam optical tweezers system. The laser beam focus is initially placed at the origin of the global coordinate system (0, 0, 0). It may be kept either stationary or moved with a constant speed in the horizontal plane or along the vertical axis. As shown in Figure 3.1, most of the experiments are reported with the coordinate system oriented such that the positive Z-axis points vertically downwards and coincides with the beam propagation direction.

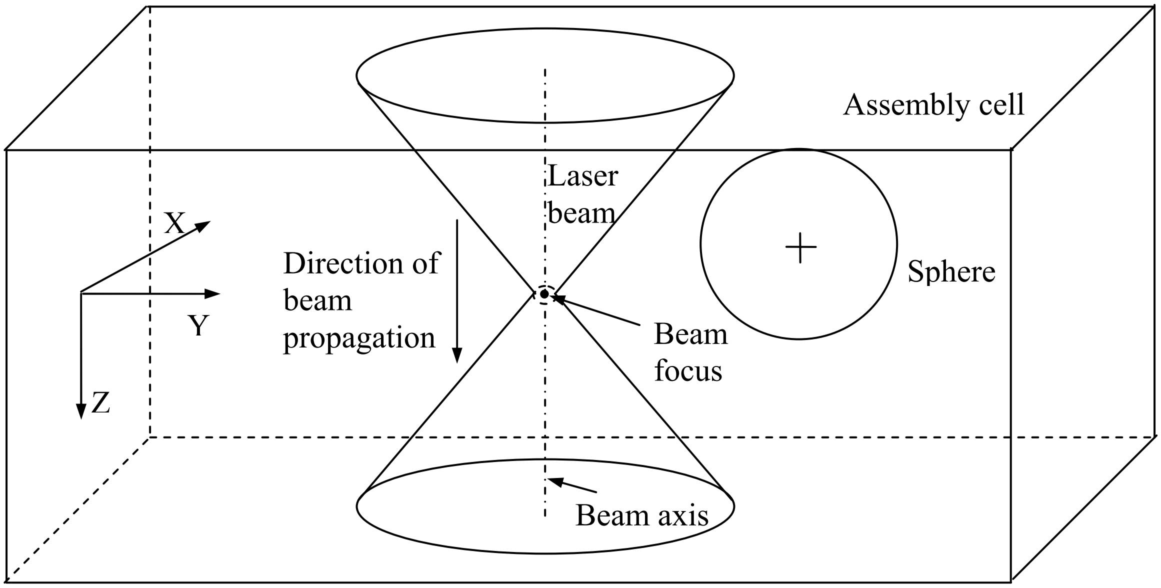


Figure 3.1: Schematic illustration of simulation set-up

3.1.1 Simulation of Sphere Motion

A particle moving with a velocity V in a fluid medium experiences a rapidly fluctuating force (due to a large number of collisions with the surrounding liquid molecules), as well as a hydrodynamic drag force. These forces are related with each other and are modeled using Langevin’s equation given by Equation (3.1) [Bali09].

$$\frac{dV(t)}{dt} = -\frac{\gamma}{m}V(t) + \frac{\xi}{m}\Gamma(t) \tag{3.1}$$

This equation assumes a fluid with viscosity η as a function of temperature T (η = 1.002 x 10^{-3} Pa-s at T = 293 K for water). For micron-sized particles, the drag coefficient γ is given by Stokes' law as $6\pi\eta R_a$, where R_a is the radius of the spherical particle. Slip-correction factor is ignored as the Knudsen number is much greater than 1. The scaling constant $\xi = \sqrt{2\gamma K_B T}$, where K_B is Boltzmann's constant, is obtained by imposing the requirements of the fluctuation-dissipation theorem [Weis89]. The presence of the stochastic force term $\Gamma(t)$ prevents direct analytical solution of this equation. Therefore, the Langevin's equation needs to be expressed in finite difference form before proceeding to integrate it numerically. The difference form of this equation is given by Equation (3.2) [Bali08], where the stochastic term $\Gamma(t)$ is replaced with a standard normal distribution and the scaling constant includes the time-step δt. The external force term, F_{ext}, allows one to include gravitational or optical trapping forces. Any fluid flow resulting due to thermal currents generated because of laser heating of the medium is not taken into account here.

$$\frac{V(t+\delta t)-V(t)}{\delta t} = A(t+\delta t) = -\frac{\gamma}{m}V(t) + \frac{1}{m}\sqrt{\frac{2\gamma K_B T}{\delta t}}N(0,1) + \frac{F_{Ext}}{m} \tag{3.2}$$

In the simulation, the sphere experiences three external forces – the optical trapping force, gravitational force, F_g, and buoyancy, F_b. The latter two forces are constant for a sphere of a given size and are given by:

$$F_g = \frac{4}{3}\pi\rho_s R_a^3 g \text{ and } F_b = \frac{4}{3}\pi\rho_f R_a^3 g \tag{3.3}$$

In Equation (3.3), ρ_s and ρ_f are the densities of sphere (silica with ρ_s = 2000 kg/m^3) and fluid (water with ρ_f = 1000 kg/m^3), respectively, and g is the acceleration due to gravity.

The optical trapping forces for relatively large-sized spheres (5.0 µm or more diameter) are calculated by numerically integrating the ray optics model based force equations A.1 and A.2 given in the Appendix A using Romberg's formula (extension of Simpson's 3-point rule). The MATLAB toolbar [Optt07] described in [Niem07] is utilized to compute the forces for the smaller sized spheres. It used a vector spherical wave function formulation of the generalized Lorentz-Mie theory known as the *T*-matrix method. The trapping laser beam is Gaussian and circularly polarized, its power *P* is chosen to be 100 mW, with wavelength of the incident light, λ = 532 nm. The speed of light in free space is, c = 3x10^8 m/s, the refractive index of water n_1 = 1.33, the refractive index of silica n_2 = 1.46, the refractive index of oil n_{oil} = 1.51 and the numerical aperture (NA) of the oil-immersion objective is 1.4.

Figures 3.2 and 3.3 show the trapping forces along the axial and transverse directions for a 5.0 µm diameter sphere that is trapped offset from the trap focus. Figure 3.3 shows that the transverse forces are negligible and the axial forces have opposite signs above and below the trap focus. Moreover, the axial forces acting on a sphere, whose center is located above the trap focus, are much stronger than those acting on a sphere, whose center is located below the focus. Figure 3.3 reveals that the transverse forces are much stronger than the axial forces and also that the transverse forces are anti-symmetric about the trap focus while the axial forces are symmetric. Figures 3.2 and 3.3 agree qualitatively and quantitatively with the data published in the literature [Ashk92, Wrig94]. In order to minimize the computation time, a lookup table of the trapping forces was generated with uniform grid spacing of 0.125 µm for spheres having diameters less than or equal to 2.0 µm, a spacing of 0.25 µm for spheres having diameters less than or equal to 2.0 µm, and 0.5 µm for all other sphere sizes. Bilinear interpolation is then used to recover the optical trapping force at run time.

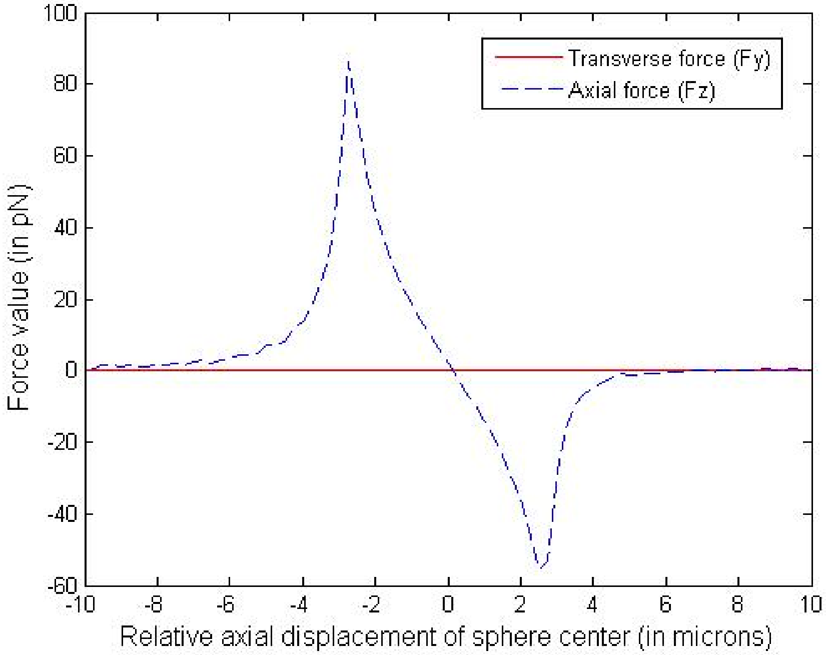


Figure 3.2: Axial and transverse force components when 5.0 µm diameter sphere center is displaced along laser beam axis from the focus

The gravitational force, buoyancy and the optical trapping forces are combined into the external force term, F_{ext} of Equation (3.2) to calculate the acceleration, $A(t + \delta t)$ at the end of the time-step, δt. The velocity form of the second order Verlet integrator [Swop82, Alle87] is used to generate a list of the particle's position, velocity, and acceleration at the end of each time step. The time-step is chosen so that it is much smaller than the characteristic time-scale of the physical process. From Equation (3.1), it is seen that the characteristic time-scale for this model is given by the relaxation time, $\frac{m}{\gamma}$ [Bali09]. Therefore, the time-step is chosen such that $\delta t << \frac{m}{\gamma}$. In the simulations, δt is selected as the closest multiple of 100 ns that is smaller than the relaxation time.

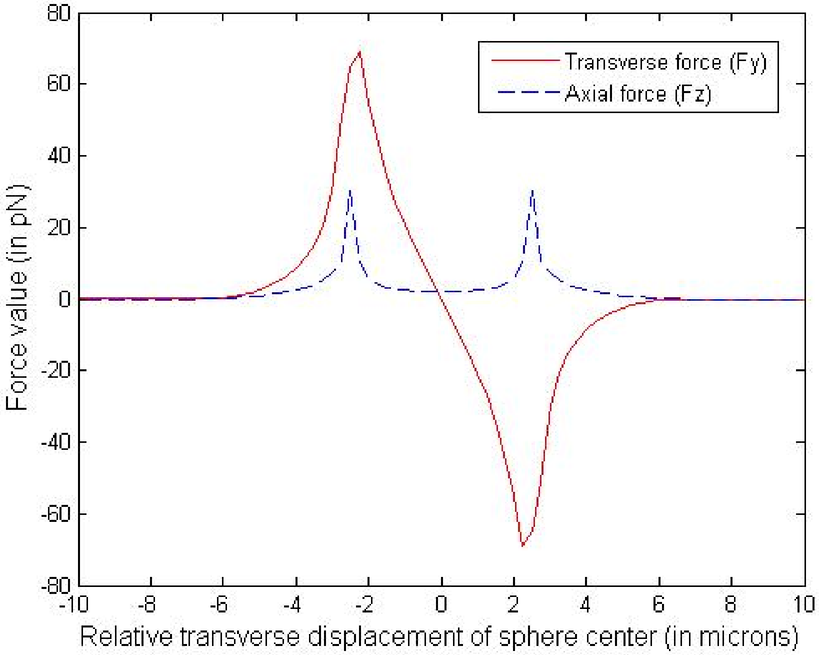


Figure 3.3: Axial and transverse force components when 5.0 µm diameter sphere center is displaced along transverse axis from laser beam focus

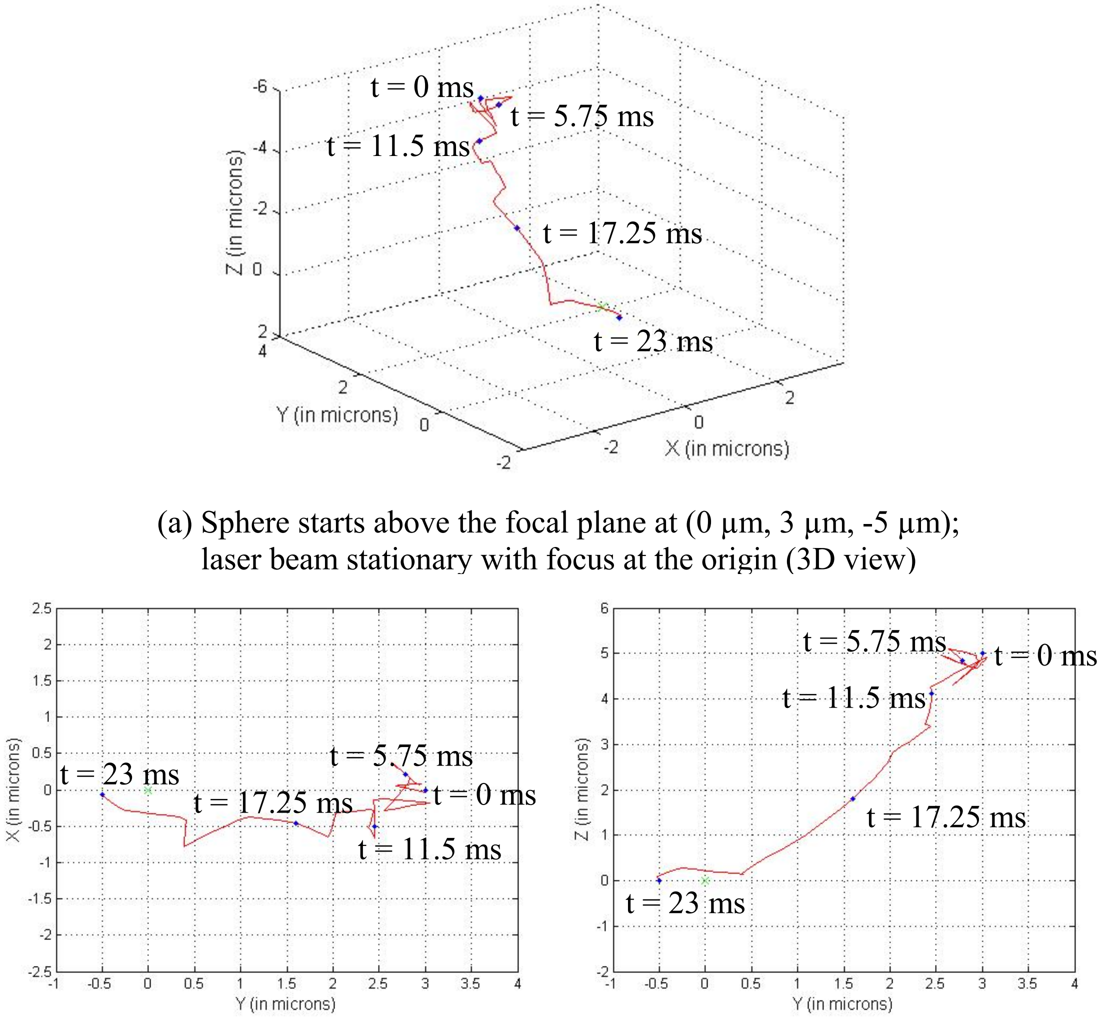
A particular run is terminated when one of the following conditions are satisfied: (a) The sphere is trapped, (b) The sphere goes completely outside the geometric boundary of the laser cone, (c) The sphere reaches the bottom of the workspace under the influence of gravity, or (d) A fixed number of iterations is completed.

Termination criterion (a) depends on various factors. A trapped sphere is held at a certain distance below the beam focus, where the axial component of optical trapping force exactly balances the combined effect of gravity and buoyancy. This depth varies based upon the sphere size. Moreover, in case of a stationary laser beam, the displacements along X and Y-axes are comparatively small (of the order of the diffusion length), whereas for a moving beam, these displacements can be larger and depend both on the direction and magnitude of the beam velocity. To address this issue, first some pilot tests were conducted to ascertain the approximate relative position of entrapment of a sphere of a given size with respect to a laser beam moving along a particular direction with a constant speed. This information was later utilized to quickly determine whether a sphere has been trapped. The sphere is considered to be trapped if it moves along with the beam by keeping the same relative displacement from the beam focus.

Termination criterion (b) is suppressed in two cases to incorporate certain important physical effects. Firstly, it is not used when a sphere goes outside the laser cone boundary sufficiently close to the beam focal region. This needs to be done because such a sphere still experiences the effect of strong gradient force due to the proximity of the laser and may be trapped eventually. Secondly, it is not applied when a beam is moving in the horizontal plane. This is required by the fact that a sphere, which is outside the cone boundary at any point of time, will diffuse and fall down under the combined action of Brownian motion, gravity, viscous drag, and buoyancy such that it can come under the influence of strong optical trapping forces at a later stage which may be sufficient to trap it.

During the simulation runs, the sphere center is initially positioned close to the laser beam in the work space. These close locations ensure that the beam motion will result in the sphere boundary intersecting the laser beam boundary within one path planning time span. Henceforth, this subset of the workspace is referred to as the *local workspace*. The planning time span depends upon the system setup and has been chosen to be equal to 50 ms in the simulations (assuming that the beam position will be updated at a frequency of 20 Hz). This also governs the choice of the fixed number of iterations in the termination criterion (d). The trapping probability of any sphere that lies outside the local workspace or which will not be trapped within 50 ms can be estimated and utilized for motion planning purposes at subsequent time steps.

Figures 3.4, 3.5, and 3.6 show few representative trajectories to depict the applicability of the above mentioned termination criteria. Each figure graphs the spatial trajectory of a 5.0 μm diameter sphere that is initially placed at different locations inside the local workspace. Although simulation is performed at intervals of 2.5 μs (computed value of δt for 5.0 μm sphere), data generated after every 0.25 ms is used for plotting purposes. The laser beam is kept stationary in all the cases, with its focus (shown by a 'x' mark) located at the origin. It can be seen from Figure 3.4(b) and 3.4(c) that the sphere settles within 0.5 μm of the focus and its total displacement along X-axis is significantly smaller than the displacements along Y and Z axes. When the sphere center is initially not displaced along the X-axis, F_{Ext} does not have any X-component and displacement occurs purely due to random Brownian motion coupled with viscous drag. However, as soon as the sphere drifts away from the origin of X-axis, strong optical trapping forces pull it back. It may also be noted that in both in Figures 3.4 and 3.5, the sphere initially diffuses randomly around its original position, before diffusion brings it to a stronger trapping force region after which it moves directly towards the trap focus. Figure 3.6 depicts the condition when a sphere that is located quite far away from the focal plane does not get trapped. Instead, it starts settling towards the bottom following a random trajectory.

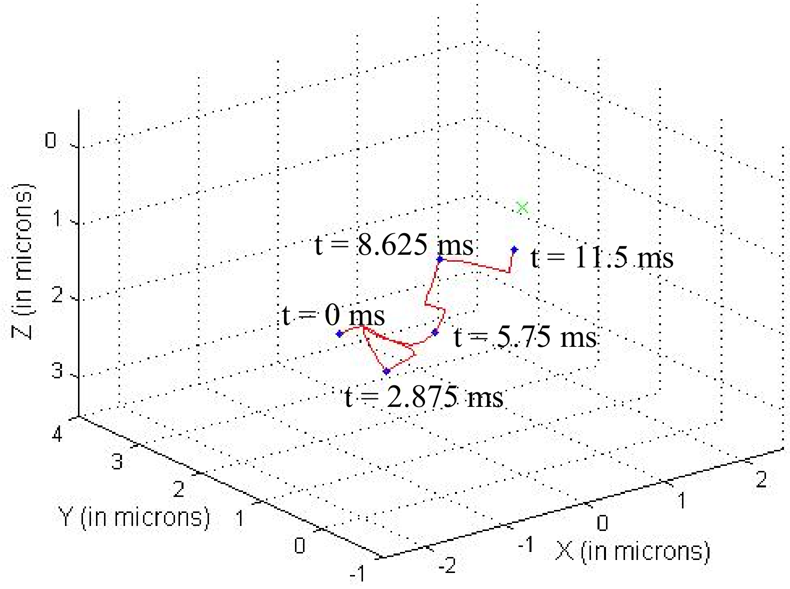


(a) Sphere starts above the focal plane at (0 µm, 3 µm, -5 µm);
laser beam stationary with focus at the origin (3D view)

(b) Above trajectory in XY plane

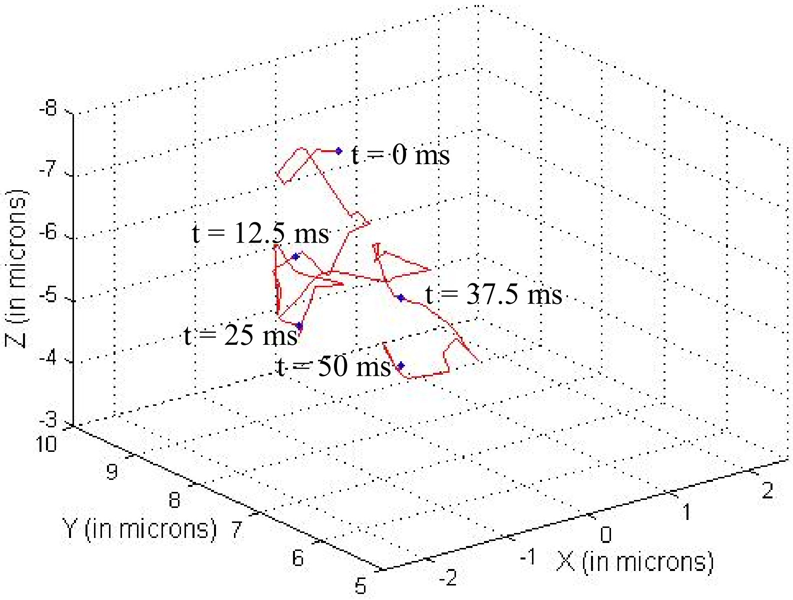
(c) Above trajectory in YZ plane

Fig. 3.4: Trajectory of 5.0 µm diameter sphere positioned above focal plane that is trapped by a stationary laser beam



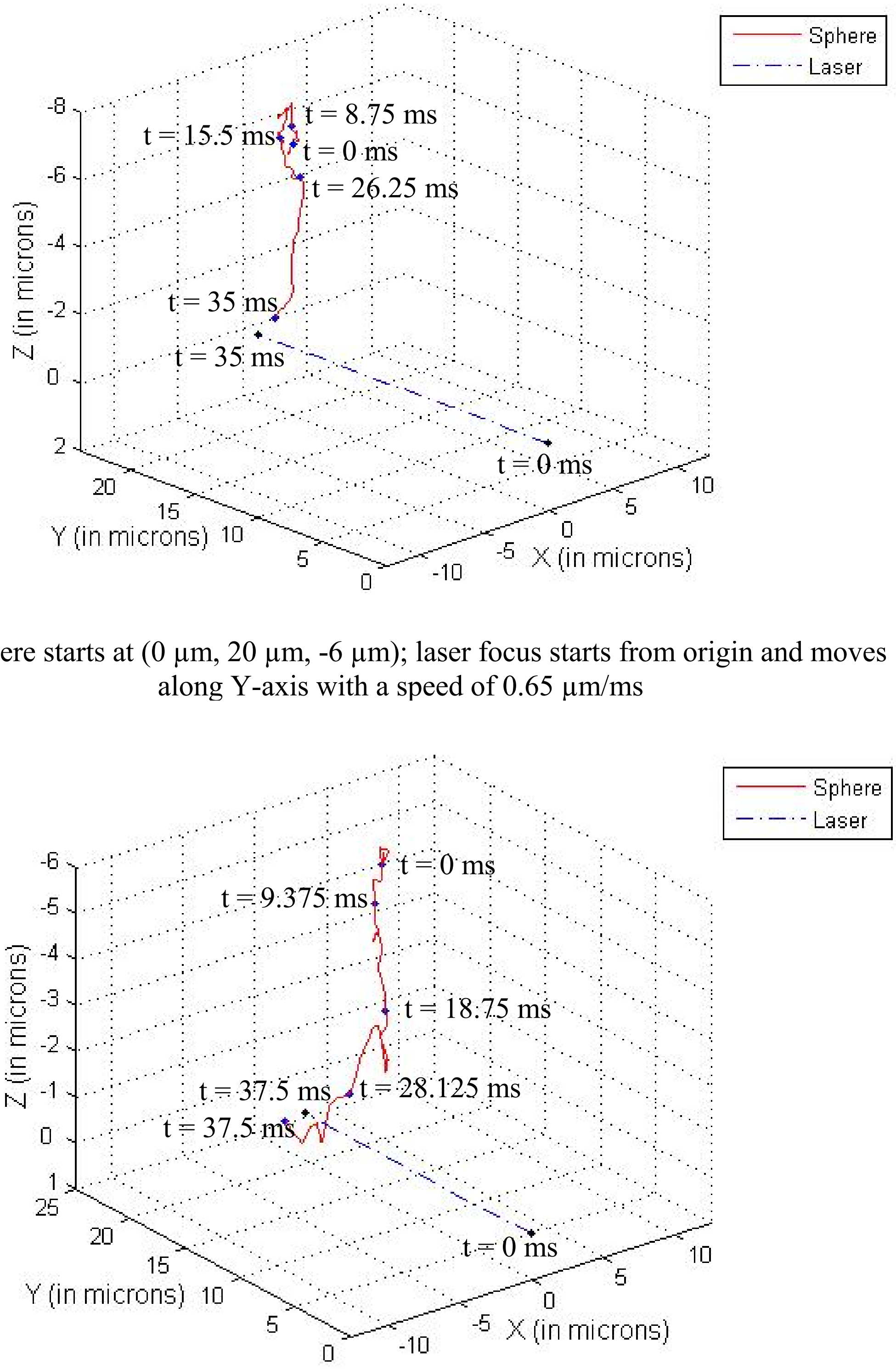
Sphere starts below the focal plane at (0 µm, 3 µm, 2.75 µm); laser beam stationary with focus at the origin

Figure 3.5: Trajectory of 5.0 µm diameter sphere positioned below focal plane that is trapped by a stationary laser beam



Sphere starts at (0 µm, 9 µm, -7 µm); laser beam stationary with focus at the origin

Figure 3.6: Trajectory of 5.0 µm diameter sphere that is not trapped by a laser beam



(a) Sphere starts at (0 µm, 20 µm, -6 µm); laser focus starts from origin and moves along Y-axis with a speed of 0.65 µm/ms

(b) Sphere starts at (5 µm, 20 µm, -5 µm); laser focus starts from origin and moves along Y-axis with a speed of 0.65 µm/ms

Fig. 3.7: Trajectories of 5.0 µm diameter spheres that are trapped by moving laser beams

Figure 3.7 represents the trajectories of spheres of the same size, originally located quite far above the focal plane, under the influence of moving laser beams. In Figure 3.7(a), the sphere center is not originally displaced along the X-axis. It initially exhibits a net vertically downward motion. Subsequently, it experiences stronger trapping forces as the beam moves closer to it, and is eventually pulled towards the beam focus before finally settling at a certain distance away from the beam focus along the direction of motion (Y-axis). However, when the sphere center is displaced along all the 3 axes (see Figure 3.7(b)), after moving vertically downwards for a short distance, it starts moving along a circular path in the horizontal plane before being trapped.

3.1.2 Generating Trapping Probability Estimates using Dynamics Simulation

Simulation is performed $N = 100$ times at every discrete point in the 4-dimensional space (3 spatial coordinates and 1 velocity component) to estimate the trapping probability based on the number of times the sphere gets trapped by the laser beam. The reason for using only 1 velocity component has been explained later in this section. To restrict the error bound in probability estimate within ±0.1 with 95% confidence level, N was chosen to be equal to 100. In any particular trial, the outcome can be either 0 (non-trapped) or 1 (trapped). Thus, it follows a binomial distribution. However, for a sample size greater than 30, it can be approximated using a normal distribution with mean (μ) equal to estimated probability p_e and variance (σ^2) equal to $\frac{p_e\left(1-p_e\right)}{N}$ [Mont94]. As 95%

confidence interval roughly corresponds to $\mu \pm 2\sigma$, the maximum error of ±0.1 occurs when $p_e = 0.5$.

Overall four separate sets of experiments were performed. Two sphere sizes, having diameters of 2.0 μm and 5.0 μm respectively, were used. For each sphere size, two experiments were conducted. In the first experiment, the in-plane velocity component (v_{xy}) was kept zero. In the second experiment, the out-of-plane component (v_z) was kept zero. It is assumed that since the beam will either transport the trapped sphere along the transverse plane or along the axial direction, both the velocity components need not have non-zero values in the same simulation run. In order to reduce the computational burden, the coordinate system is rotated so that the new Y-axis always coincides with the direction of the in-plane velocity vector. Thus, in any particular simulation, four parameters are present, namely the relative *x*, *y,* and *z* coordinate of the sphere with respect to the beam focus and one velocity component. For the sake of convenience in setting up experimental conditions, the spatial 3D volume is represented in polar coordinates. Discretization is carried out at an interval of every 0.5 μm for both the sphere sizes along the radial (r_{xy}) and axial directions (r_z) and 5° along the azimuthal direction (θ_{xy}).

The expressions for maximum permissible laser speeds such that the optical trapping force is sufficient to overcome viscous drag are given by:

$$v_{xy,\max} = \frac{Q_{tr,\max} n_1 P}{6c\pi\eta R_a} \text{ and } v_{z,\max} = \frac{Q_{ax,\max} n_1 P}{6c\pi\eta R_a} \tag{3.4}$$

where $Q_{tr,\max}$ and $Q_{ax,\max}$ are the maximum transverse and axial trapping efficiencies [Ashk92]. These efficiencies have been chosen to be 0.16 and 0.07 respectively based upon the peak values of the transverse and axial force components in Figure 3.3. Although axial forces have higher peaks when the sphere center is offset axially from the trap focus (Figure 3.2), it is reasonable to select the smaller value so that Equation (3.4) is applicable for all scenarios. Simulation experiments reveal that the sphere often escapes from the trap near the maximum speed limit. This, by itself, is a random phenomenon and has a probability associated with it based upon the sphere size and the trap speed. The effect is far more pronounced in case of the smaller sphere as it has higher diffusivity. Hence, speeds have been restricted to 86% of the maximum value for the 5.0 μm sphere and to less than 1% for the 2.0 μm sphere. This restriction is imposed for both the velocity components and the domain has been sampled uniformly by 5 points (including zero) for estimating trapping probabilities. For an out-of-plane component, both upward and downward laser motions have been considered.

Combinatorial experiments have then been carried out by varying one parameter at a time and keeping all the others constant. However, this means that a large number of grid points are constructed for estimating trapping probabilities inside the local workspace. In order to obtain results in a reasonably short period of time, computationally redundant simulations are avoided based on the insight regarding the physical behavior of the system. They are listed below:

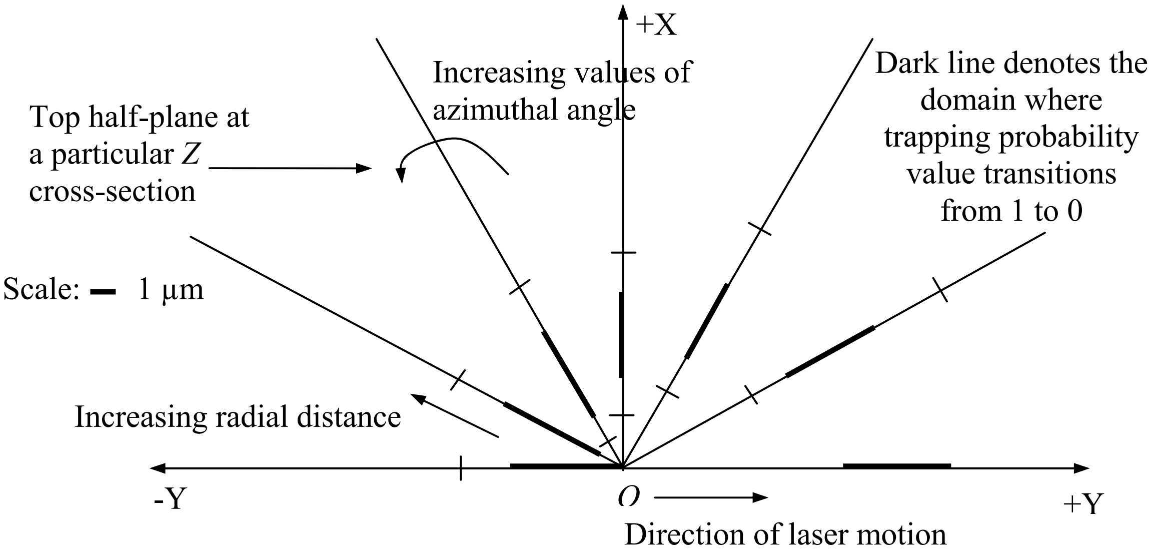
- Azimuthal variation is not considered in the experiments where the laser beam is stationary or moves along the axial direction since the *x* and *y*-components of the optical trapping force are symmetric about the vertical axis.
- Data points are placed only at the top half-plane at every horizontal cross-section of the 3D volume where azimuthal variations need to be taken into account. This is

because all points in the bottom half-plane will exhibit symmetry about the transverse (Y) axis since the laser velocity has no component along the X-axis.

- Simulation is not performed at all the data points lying in the top half-plane at any particular cross-section even where azimuthal variations have to be taken into account. Experiments are carried out at all the points lying along the direction of motion (Y-axis) to identify the domain of interest, which refers to the region along Y-axis within which the estimated trapping probability changes from 1 to 0 or vice versa. At the next azimuthal direction, this domain of interest is elongated by 1.5 μm at the boundary where trapping probability transitions from a fractional value to 0 or vice versa and by 3 μm where it transitions from 1 to a fractional value or vice versa. This process is repeated by obtaining the domain of interest iteratively at any θ_{xy}. This works well in practice because trapping probability values do not change drastically for the same value of r_{xy} at neighboring values of θ_{xy}.

Application of the last strategy has been illustrated in Figure 3.8. About 10 weeks were required to gather all the data. Two PCs were used; they consist of Intel Core 2 Quad processors with 8 GB of RAM and 2.83 GHz clock speed. All the PCs run on Microsoft Windows Vista Business operating system. Microsoft Visual Studio 2005 was used as the compiler in all the cases.

Simulation data is used to generate a list of figures that help in visualizing the trapping probability estimates meaningfully. These figures show probability contour plots for a 5.0 μm diameter sphere in different planes for a stationary or moving laser beam that has an in-plane velocity component. The simulation data includes significant noise. This noise arises as a result of stochastic (Brownian) forces on the spheres, which introduces variability in the trapping behavior. Thus, the trapping probability at any two neighboring grid points may show a local trend that is different from the overall trend. For example, in case of a stationary laser beam, instead of exhibiting a monotonically decreasing trend at the focal plane, a grid point may actually report higher probability value as compared to an adjacent one to the left. Of course, this discrepancy lies within the error limits identified at the beginning of Section 3.1.2. Hence, all the contour plots have been smoothened out using in-built MATLAB function to filter the noise and display the broad trends.


+X
Increasing values of azimuthal angle
Top half-plane at a particular *Z* cross-section
Dark line denotes the domain where trapping probability value transitions from 1 to 0
Scale: ▬ 1 μm
Increasing radial distance
-Y
O
+Y
Direction of laser motion

Gridding is shown at intervals of 30° instead of 5° for ease of representation

Figure 3.8 Spatial discretization for conducting offline simulation experiments involving laser motion in horizontal plane

The first contour plot (Figure 3.9) is a baseline, which represents the trapping behavior of a sphere under the influence of a stationary laser beam. The plot is shown only in half-YZ plane, as the contours in the other half are symmetric about the Z-axis. Similar plot can be observed in the XZ plane. Contours follow circular patterns in XY-plane. In Figure 3.9, contours meet the beam axis below the focal plane. However, this phenomenon cannot be seen above the focal plane. Instead, higher probability contours tend to come towards beam axis, whereas, lower probability contours remain more or less vertical. Moreover, they tend to diverge as we move away from the focal plane. This occurs due to a combination of Brownian motion and gravitational pull. Any sphere that is originally positioned way above the focal plane has a greater chance of moving away or towards the beam axis in addition to settling down at the bottom. During its random motion, it may either come under the influence of stronger optical trapping forces or may go outside the beam boundary. So, the contours span a wider region and move away from each other in opposing directions.

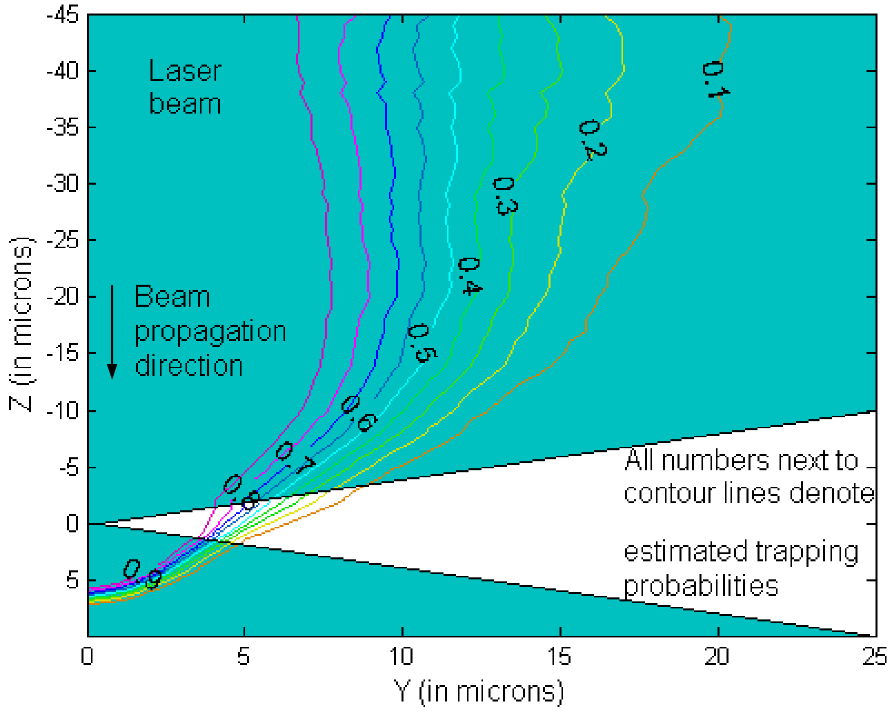


Figure 3.9: Estimated trapping probability contours for 5.0 µm diameter sphere under influence of a stationary laser beam

Figures 3.10 and 3.11 have been drawn for the case when the laser moves along +Y-axis with a speed of 0.65 μm/ms. Significantly different behavior from the stationary case can be observed. Figure 3.10 reveals that the contours converge in the negative Y-plane. Higher probability contours occur closer to the origin where optical trapping forces are stronger. The contours also tend to shift upwards with increasing positive values in Y due to the combined effect of laser motion and gravity. As the laser moves along positive Y-axis, spheres that are originally positioned above the focal plane and quite far away from origin, may get trapped as they come under the influence of strong trapping forces. But the spheres that are positioned below the focal plane may not experience strong trapping forces as they may have already moved downwards by the time laser has come close enough to them.

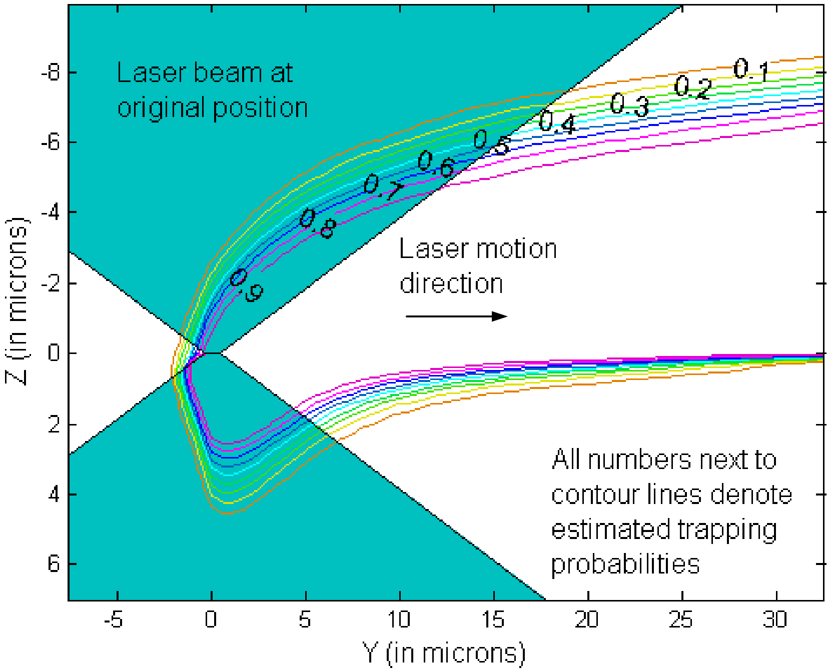
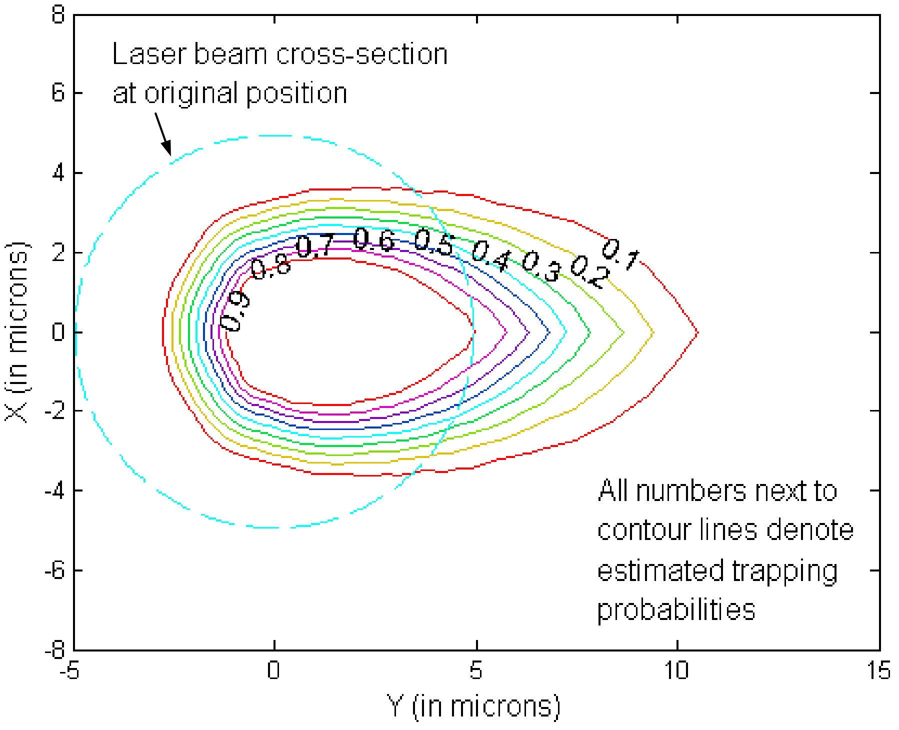


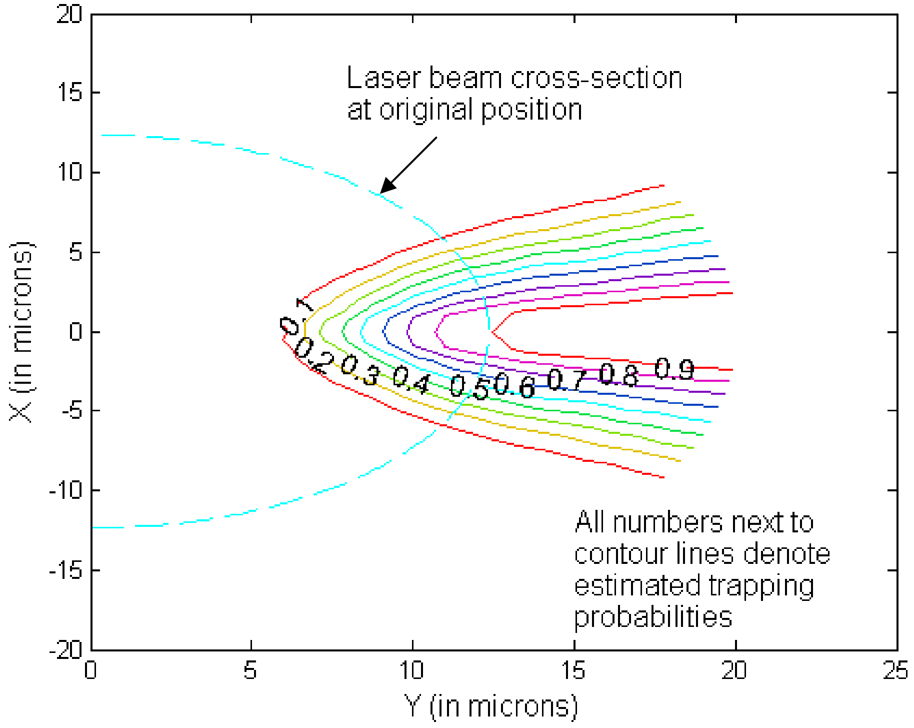
Figure 3.10: Estimated trapping probability contours for 5.0 μm diameter sphere in YZ plane under influence of a laser beam moving along +Y-axis with a speed of 0.65 μm/ms

This differential behavior at two horizontal cross-sections has also been illustrated in Figure 3.11. At both the cross-sections, the contours do not follow circular patterns unlike the stationary beam case; rather they take elliptical shapes and are distorted along the direction of laser motion. Significantly less trapping takes place in the negative Y-plane as the laser quickly moves away from the origin. It may also be noted that the contours below the focal plane move close to each other in the positive Y-plane instead of following parallel bands due to the fact that spheres located even slightly below the focal plane and quite far away from the origin do not get trapped within the planning time span.

Figure 3.12 represents the contour plot when the laser moves along negative Z-axis with a constant speed of 0.325 μm/ms. The basic nature is similar to the plot shown in Figure 3.9 for the stationary beam case. The only significant difference lies in the span or coverage of the contours. In case of the stationary beam, the contours spread out to about 20 μm along the transverse axis. They only cover a distance of about 11 μm when the laser is moving vertically upwards. The diffusion length of a 5.0 μm sphere is less than one micron during the planning time span. Hence, this difference cannot be attributed to any stochastic noise. Instead, this phenomenon can be explained by considering the fact that due to the upward motion of the laser beam, even spheres that are originally located far above the focal plane effectively behave like spheres that are placed close to the focus of a stationary beam. Hence, if the centers of such spheres are displaced transversally by large amounts, then they do not experience very strong optical trapping forces and thus do not get trapped.



(a) Plot at z = 2 μm



(b) Plot at z = -5 μm

Figure 3.11: Estimated trapping probability contours for 5.0 μm diameter sphere in XY plane under influence of a laser beam moving along +Y-axis with a speed of 0.65 μm/ms

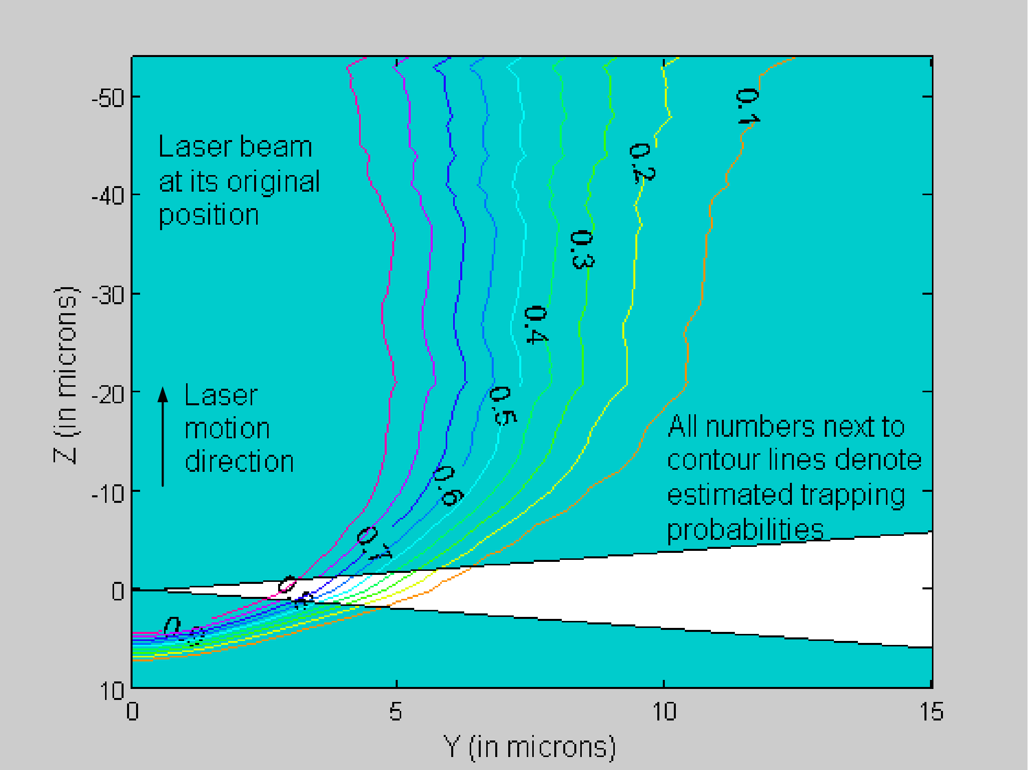


Figure 3.12: Estimated trapping probability contours for 5.0 μm diameter sphere in YZ plane under influence of a laser beam moving along -Z-axis with a speed of 0.325 μm/ms

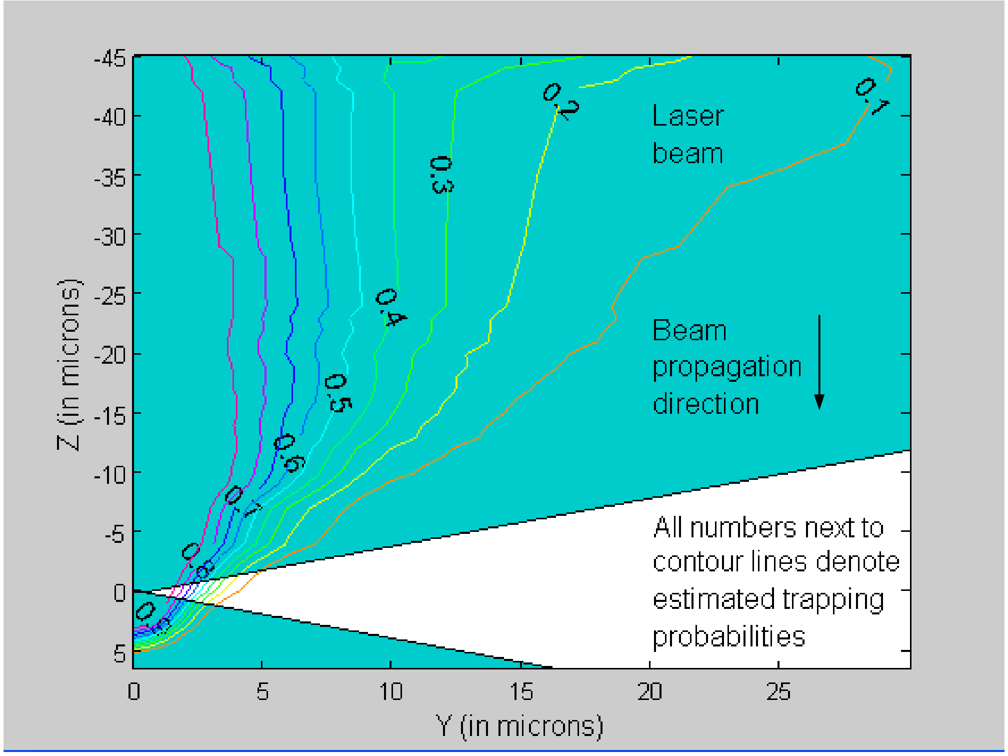


Figure 3.13: Estimated trapping probability contours for 2.0 μm diameter sphere under influence of a stationary laser beam

Figure 3.13 presents the contour plot for a 2.0 μm diameter sphere under the influence of a stationary laser beam. Although the overall trend is similar to that of the 5.0 μm diameter sphere, certain important distinctions can be noted. The major difference is that the contours exhibit greater divergence as one moves away from the beam focus in the negative Z-plane. This behavior primarily arise due to two reasons – enhanced effect of Brownian motion and reduced effect of gravity on smaller sized sphere. A sphere located significantly above the focal plane and away from the beam axis can drift close enough to the strong trapping force region. Similarly, a sphere located not that far away from the axis above the focal plane, can drift sufficiently away so that it never gets trapped. This divergence behavior starts from approximately z = -4 μm, instead of z = -10 μm in the 5.0 μm case. Corresponding span along the Y-axis from where transition originates is also somewhat less in this case. Since strong optical trapping forces exist approximately up to sphere diameter along both the axes (as can be verified from Figures 3.2 and 3.3), divergence starts earlier in this case.

3.2 Generating Simplified Models from Simulation Data

3.2.1 Selecting Model Simplification Technique

Although offline simulation has presented a lot of data, it cannot be used directly for run-time estimation of trapping probability. As already discussed earlier, some form of simplified model is necessary to represent this data compactly and then use it efficiently to compute the estimated trapping probability at any point in 4-dimensional parametric space. Advantages of using Gaussian RBFs over other metamodeling techniques have been explained in Chapter 2. Hence, Gaussian RBFs were selected for developing simplified models. Each of the four data sets obtained from simulation is separately represented and modeled using Gaussian RBFs. Symbolically, it takes the following form:

$$p\left(\vec{x}\right) = w_0 + \sum_{i=1}^{m^{/}} w_i \phi_i \left(\vec{x}\right) \tag{3.5}$$

where $\vec{x}$ is the position vector of a point in 4-dimensional parameter space,

$p\left(\vec{x}\right)$ is the computed trapping probability value at that point,
$m^{/}$ is the number of RBFs,
ϕ_i is the *i*th RBF,
w_0 is a constant term and
w_i is the weight of the *i*th RBF

Since Gaussian RBFs are being used, $\phi_i\left(\vec{x}\right) = \exp\left(-\frac{r^2}{2}\right)$, where RBF radius $r = \left\| \frac{\vec{x} - \overset{\rightarrow}{\mu_i}}{\overset{\rightarrow}{\sigma_i}} \right\|$. Here, $\overset{\rightarrow}{\mu_i}$ is the position vector of the *i*th RBF center and $\overset{\rightarrow}{\sigma_i}$ represents the width of the *i*th RBF. The components of this width vector may be selected anisotropically so that it assumes different values along the different axes. However, the current work is restricted to axis-aligned RBFs only, thereby sacrificing some generality for the sake of a more compact representation. By axis-aligned, it is meant that the axes of the hyper-ellipsoidal Gaussian RBF surface are aligned along the axes of the 4-dimensional parameter space.

3.2.2 Fitting Gaussian RBFs to Simulation Data Sets

The method reported in [Juba07] has been adapted and extended for this application. Specifically, Juba and Varshney have developed a method for fitting 3D volumetric data using RBFs. Here, the parametric space is four dimensional and involves both distance and velocity parameters. Hence, a different spatial partitioning scheme had to be developed.

A single RBF is first fitted to a 2^4 resolution down sampled version of the given data set. This down sampling is done by obtaining all possible combinations of the minimum and maximum values of each of the four parameters (i.e., 2^4 points). This is taken as the root of the RBF hierarchy, which is represented in the form of a k-dimensional tree (kd-tree). This fitted RBF is then evaluated at all the data points in the entire parameter space. If the root mean square (RMS) error is above 2% and the maximum error is above 4%, then the parameter space is partitioned into two halves by a hyperplane that is perpendicular to the *X*-axis such that the two half-spaces have approximately same number of data points. These threshold errors are just representative values that have been chosen to evaluate the performance of this fitting procedure. Henceforth, each sub-space will be termed as a *region*. It actually represents a node in the kd-tree. Thus, it can be seen that kd-tree is a binary tree which will be progressively constructed during the fitting operation.

Two other 2^4 resolution down sampled versions of the parameter space are then obtained in the newly constructed regions by selecting points using uniform, random distribution. A new child RBF is fitted to the residual error in each of the two regions using the down sampled versions and the error is evaluated using both the child as well as the parent RBF at all the data points. If both error measures lie below their respective thresholds in any of the two regions, then it is not sub-divided any further. Otherwise, that region is partitioned by a hyperplane perpendicular to the Y-axis, the kd-tree is updated and this process continues. If region partitioning is required at the 3rd and 4th levels of the kd-tree, then hyperplanes perpendicular to Z and V-axis are used respectively. This cycle of choice of partitioning hyperplane is then repeated for subsequent levels in the tree. In actual implementation, this process is carried out in a depth-wise manner. Moreover, if in the region under consideration some or all of the four parameters assume boundary values that are different from the minimum or maximum values, then one additional data point is considered from the corresponding neighboring region. This avoids noticeable discontinuities at the region boundaries.

Although down sampling could have been done again by considering minimum and maximum parameter values, it is found that random sampling leads to a slightly lower number of RBFs. However, it is better to follow the former approach while obtaining the root RBF as it enables us to fit a function that can broadly span the entire parameter space. The process terminates when all the regions have been adequately fitted using varying number of RBFs such that RMS and maximum errors lie below their respective thresholds in each and every one of them. It may be argued that the overall process can be sped up by using a hex tree (analogous to octree in 3D) decomposition method as this directly splits up the space into 16 sub-spaces at one run. However, a kd-tree is a better choice in this case because of the differences in dimensions of the parameters along the 4 axes (3 are distances and 4th is velocity). It also provides a convenient way of discretizing the overall space into sub-spaces of roughly equal cardinalities by forcing the partitioning plane to pass through the median value of the corresponding parameter. Moreover, any dimensionality of the parametric space can be easily handled by the kd-tree. This only entails increasing the number of partitioning planes (equal to *k*) to a higher value. Thus, kd-tree based partitioning offers lot of flexibility in incorporating additional parameters in future applications.

Therefore, obtaining this compact model reduces to a careful selection of the constant term w_0, RBF weights, RBF centers, and RBF widths. w_0 has been selected as zero for all results reported in this work. Thus, this term can be neglected henceforth. The initial guess value of the center of the newly created RBF is taken as the position vector of the data point that has the maximum residual approximation error. The RBF's weight is set as the RMS error value of the region under consideration. This can be rationalized by taking note of the fact that this causes the center data point to have zero error since Gaussian RBFs have a value of 1 at their centers.

The anisotropic widths are selected using maximum likelihood estimation scheme which assumes that the data to be fitted to a particular RBF is a histogram of samples from a Gaussian (normal) probability density function with the previously selected mean or center. Then the width or spread correlates to its standard deviation and is computed such that this set of samples will have the highest probability of being generated. Since the data values (trapping probabilities) are estimated based upon a set of 100 simulation runs, it is meaningful to adopt this approach in the current work. The relevant equation for computing the *x*-component of the *i*th RBF's width vector (σ_{xi}^2) is given below. Other components are obtained similarly.

$$\sigma_{xi}^2 = \frac{\sum_{j=1}^{n} p_j}{2\sum_{j=1}^{n}\left(x_j - \frac{\sum_{j=1}^{n} x_j p_j}{\sum_{j=1}^{n} p_j} \right) p_j} \tag{3.6}$$

where n is the number of data points that are being approximated by the ith RBF and p_j is the estimated trapping probability of the jth data point.

Once the RBF center and all the widths have been obtained, improved values of the weights are computed by minimizing the sum of squared errors

$$\psi = \sum_{j=1}^{n}\left[\sum_{i=1}^{m^{/}} w_i\phi_i - p_j\right]^2 \tag{3.7}$$

This is equivalent to solving the following system of linear equations in the least-squares senses, as usually $m^{/} < n$.

$$\begin{bmatrix} \phi_1\left(\vec{x}_1\right) & \cdots & \phi_{m^{/}}\left(\vec{x}_1\right) \\ \vdots & \ddots & \vdots \\ \phi_1\left(\vec{x}_n\right) & \cdots & \phi_{m^{/}}\left(\vec{x}_n\right) \end{bmatrix} \begin{bmatrix} w_1 \\ \vdots \\ w_{m^{/}} \end{bmatrix} = \begin{bmatrix} p_1 \\ \vdots \\ p_n \end{bmatrix} \tag{3.8}$$

The entire method has been implemented in MATLAB. All the steps are formally described in the algorithm FIT-GAUSSIAN-RBFs. Performance of this algorithm on all the four data sets has been shown in Table 3.1. The first two data sets correspond to 5.0 μm diameter sphere, whereas the remaining two correspond to 2.0 μm diameter sphere. Odd numbered data sets deal with the cases where horizontal component of laser velocity is varied; correspondingly even numbered data sets capture the effect of out-of-plane laser velocity component. Even if simulation data is gathered only in the top-half cross-sectional plane or at few locations using adaptive sampling scheme, probability estimates at other data points are stored by copying values suitably using a pre-processor program. In actual implementation, the first dimension (x-coordinate) is neglected for the even numbered data sets and the second dimension (y-coordinate) is treated as a radial coordinate that can only assume positive values. This reduces the fitting time and storage space of RBF representation and follows from the fact that trapping behavior of a sphere only depends on its relative axial and radial coordinate for a beam that has no in-plane velocity component.

Algorithm: FIT-GAUSSIAN-RBFs

Input:

- A set of M data points in 4-dimensional parameter space $D = \{d_1, \ldots, d_M\}$, with an estimated trapping probability p_i associated with every point d_i, $i = 1, \ldots, M$. The parameters associated every data point include x, y, z-coordinates of the initial location of sphere center under consideration and either in-plane or out of plane component of laser velocity. This can be represented by a 4-tuple (x_i, y_i, z_i, v_i).

Output:

- An incrementally constructed partial kd-tree K that represents the input data concisely using $M^{/}$ number of Gaussian RBFs (same as the total number of tree nodes), such that every node corresponding to a region in 4D parametric space is fitted by a single RBF.

Steps:

1. Obtain a 2^4 resolution down sampled version from D by obtaining all the points whose 4-tuple is formed by some combination of minimum and/or maximum values of the parameters. These points represent the boundary points of the local workspace.
2. Fit a Gaussian RBF to this down sampled data set using the function FIT-SINGLE-RBF and store it as the root node of previously empty K.

3. Estimate the RMS error of approximating the entire parameter space consisting of *M* data points by the single Gaussian RBF $e_{RMS} = \sqrt{\frac{\sum_{i=1}^{M}\left(p_i - p_i^{/}\right)^2}{M}}$, where $p_i^{/}$ is the reconstructed trapping probability value using the fitted RBF at the *i*th data point.
4. Compute the maximum error in approximating the parameter space by the same Gaussian RBF $e_{Max} = \max\left(\left|p_i - p_i^{/}\right|\right), \forall i = 1, \ldots, M$.
5. **If** e_{RMS}> 2% **or** e_{Max}> 4%, then proceed to step 6; **else** output *K* and terminate.
6. Initialize current level (*l*) to be equal to 1, current parent node (n_c) to be equal to *NULL* and current region (r_c) as the entire parameter space.
7. **While** e_{RMS} > 2% **or** e_{Max} > 4% in r_c **and** r_c contains at least 2 data points, **do**
    i. Partition r_c into two halves by a plane perpendicular to one of the axes such that each half-plane roughly contains the same number of points. Choice of this axis depends on the level *l*. If *l* belongs to the sequence {1, 5, …} select *X*-axis; if it is a part of the sequence {2, 6, …} select *Y*-axis; if it belongs to {3, 7, …} choose *Z*-axis and if it is a multiple of 4, then choose *V*-axis. Store the two half-planes as children nodes of the root of *K*. By pre-sorting the original data set *D* along each of the 4 dimensions, median coordinate along the corresponding dimension is used to split the current data set.
    ii. Increment *l* by 1 and update n_c as the node corresponding to the current region and r_c as the left child region.
    iii. Obtain a 2^4 resolution down sampled version from the points stored in r_c by random selection.

iv. Search whether any of the 4 parameters have a boundary value that is different from its minimum and maximum values. **If** yes, then add another randomly selected point from the neighboring region that shares a common partitioning hyperplane perpendicular to that parameter axis to the down sampled version.

v. Fit a new Gaussian RBF to r_c by calling the function FIT-SINGLE-RBF.

vi. Estimate e_{RMS} in approximating p_i values in *all* the data points present in the left child region using the newly fitted RBF as well as all of its parent or ancestral RBFs spanning up to the root node from current level l. Similarly estimate e_{Max} as well.

8. **If** $e_{RMS} < 2\%$ **and** $e_{Max} < 4\%$ in all the K regions, then output K and terminate. **Else** go to next step.

9. **If** node corresponding to r_c is the left child of n_c, then update r_c as the right child region and go to step 7 iii and continue recursively inside the while loop. However, **if** the corresponding node is the right child of n_c, then traverse backwards in K until a node is reached, such that the previously visited node is its left child. Store this node value as n_c, update r_c as the right child region of n_c and set l as the tree level value of right child. Then go to step 7 iii.

Function: FIT-SINGLE-RBF

Input: A down sampled set of data points $D^/$ present in a particular K node region, such that $D^/ \subseteq D$, and corresponding $p_i^/$ values.

Output: A newly constructed Gaussian RBF that approximates p_i values for all the points present in $D^/$ and updated $p_i^/$ values.

Steps:

1. Perform a linear search to obtain the data point belonging to set $D^{/}$ that has maximum *residual* error using the expression $e_{Max} = \max\left(\left|p_i - p_i^{/}\right|\right), \forall i = 1,...,n^{/}$, where $n^{/}$ is the cardinality of $D^{/}$. Select this data point as the RBF center. **If** the root RBF is being constructed, then the geometric center corresponding to the mean value of each of the 4 parameters is selected.
2. Choose the anisotropic widths using Equation (6).
3. Represent the RBF using Equation (5), with the constant term always set to zero and weight being neglected temporarily and obtain updated e_{RMS} only for this down sampled data set $D^{/}$. Updated e_{RMS} is computed by modifying $p_i^{/}$ to incorporate the effect of the new RBF. Set the weight to be equal to e_{RMS}.
4. Obtain final value of weight by performing a lease squares error fitting over the data set $D^{/}$ using Equations (7) and (8).

Table 3.1: Results of fitting Gaussian RBFs to 4 different simulation data sets

Data set number	Number of data points (in million)	Fitting time (in s)	Number of fitted RBFs (in thousand)	RMS error (in %)	Maximum error (in %)
1	6.354	1456.9	107.53	1.75	3.97
2	0.066	8.3	0.44	1.97	3.23
3	0.301	159.7	12.18	1.76	3.98
4	0.002	0.6	0.03	1.98	3.25

It is found that maximum errors typically occur repetitively in two local zones – a) while fitting an RBF to a kd-tree region that has been formed by partitioning parent region based on velocity parameter and b) while fitting to a region created by partitioning based on *z*-coordinate value above the focal plane only when in-plane velocity is taken into account. The former scenario arises in all the 4 data sets due to the presence of only 5 discretization levels in the velocity parameters. However, the latter scenario occurs exclusively in odd numbered data sets due to the differential trapping behavior at various cross-sections explained using Figure 3.11. The number of fitted RBFs (and hence the fitting time) depends on the size of data set, the size of the sphere as well as which velocity component is considered. In general, it is lower when the data set size is smaller and out-of-plane velocity is considered (lesser features observable in the contour plots as compared to the in-plane velocity cases). However, proportionately it is much higher for the smaller sized sphere as more complex features are present due to greater diffusivity.

3.2.3 Querying kd-Tree to Compute Estimated Trapping Probability Values

Once all the data sets have been modeled, it is pretty straightforward to compute the estimated trapping probability value online at any point in the parameter space. The point of interest need not coincide with any of the grid or data points because one may be interested in computing the probability value at any arbitrary location in the parametric space. This estimation is done by summing up the values of all the RBFs that overlap that point. The steps are outlined below in the form of the algorithm OBTAIN-PROBABILITY-VALUE.

Algorithm: OBTAIN-PROBABILITY-VALUE

Input:

- A point in local workspace with fully specified 4-tuple ($x_{\text{int},} y_{\text{int}}, z_{\text{int}}, v_{\text{int}}$) at which one is interested in computing the trapping probability.
- kd-tree *K* obtained as output of the algorithm FIT-GAUSSIAN-RBFs

Output:

- Estimated trapping probability value p_{est}.

Steps:

1. Initialize node (n_e) and region (r_e) that are being currently explored as the root of *K* and the entire parameter space respectively.
2. Initialize a list of explored kd-tree nodes *L* as an empty set.
3. **While** p_{est} is not computed **or** both children of n_e have not been explored, **do**
    i. Insert node n_e in list *L*.
    ii. Perform a linear search of all the data points stored in r_e to ascertain whether the given point lies inside r_e based on all the values specified in the 4-tuple.
    iii. **If** yes, then go to left child of n_e (if one exists) and update n_e and r_e accordingly. However, **if** n_e is a leaf node, then carry out a backward traversal of *K* from n_e to root node to identify all the parent RBFs and compute p_{est} using Equation (5).
    iv. **Else** backtrack to nearest unexplored node in *K* (present in the set of nodes of *K* but not in *L*) as is done in any *depth-first* tree traversal, update n_e and r_e and continue inside the loop recursively.
4. Output p_{est}.

One thousand points are randomly chosen from each of the four parametric spaces such that none of the points coincides with any of the data points that were used to fit the Gaussian RBFs. Simulation experiment is then carried out 100 times at each such point and the trapping probability is estimated. The query algorithm is then used to compute trapping probabilities at all those selected points. The performance of the algorithm (and hence that of the overall fitting technique) in terms of both computational speed as well as accuracy is shown in Table 3.2. Timing data clearly reveals the significance of developing a simplified model and the impracticality of using offline simulation for real-time motion planning. As expected, the error measures are slightly higher than the corresponding numbers in Table 3.1 because those values are recorded based on grid points, whereas these points never coincide with any point that has been used to fit RBFs. The increase in error is more for the maximum error measure as diffusion (particularly for the 2.0 μm sphere) results in noisy data at certain points in the parametric space. However, none of the error values is markedly high which shows the efficacy of the fitting technique.

Table 3.2: Performance of query function to estimate trapping probability at 1000 randomly selected points in 4 parametric spaces

Data set number	Overall simulation time (in hours)	Overall query time (in ms)	RMS error (in %)	Maximum error (in %)
1	26.37	105.9	1.84	4.34
2	24.34	6.8	2.06	3.49
3	153.38	16.4	1.86	4.57
4	144.12	1.2	2.07	3.62

Now, in order to transport one or more spheres from a given initial location to target location(s), one needs to develop intelligent motion planning strategies that can perform this operation without any manual intervention in the least possible expected time. This requires avoiding imminent collisions with other spheres present in the workspace because the laser beam may often lose the trapped sphere as a result of such impacts. If that happens, then the laser needs to be moved to a new location so that the stable trap location is close enough to the displaced sphere center. Due to latency effects and limitations in the accuracy of the imaging system, a lot of time may elapse before the laser can be positioned satisfactorily. This makes the process inherently inefficient.

Hence, run-time knowledge of trapping probability at any spatial location close to beam focus at any permissible laser speed will help in predicting possible sources of collisions. Once potentially colliding spheres are identified, multiple laser beams can be switched on so that they can trap those spheres and either hold them stationary or move them away from the spheres that are being transported. Alternatively, the controller can command currently operational laser beams to go along different paths at a faster or slower rate to circumvent those potentially colliding spheres. Circumvention can occur by a suitable combination of rotation about Z-axis and horizontal or vertical translation. Moreover, trapping probability information gives us greater range in positioning the beam focus for trapping purposes. If one can easily estimate the region within which trapping probability is greater than 90%, then the beam can be placed anywhere in that region knowing that one can almost certainly trap the sphere of interest. This will help in developing more robust and efficient algorithms.

3.3 Summary

In order to automatically manipulate microspheres using optical tweezers, it is essential to efficiently compute an estimate of the probability with which they will be trapped by a moving laser beam. Since real-time simulation cannot be used in automated planning applications, a systematic approach has been developed for generating simplified trapping probability models based upon offline simulation data.

This Chapter investigates the computational challenges that arise in transport operations performed using optical tweezers. It describes the trapping behavior of a dielectric sphere in a spatial region close to the focus of a stationary laser beam. This behavior is quantified in terms of trapping probability which is estimated by performing repeated simulation runs by positioning the sphere center initially at the point of interest. It also describes the trapping behavior of a microsphere in a spatial region close the focus of a moving laser beam. The Chapter also presents a model simplification technique for fast and accurate online computation of trapping probability estimate at any arbitrary point in 3D space for any value of laser speed using Gaussian radial basis functions. This should be a very useful tool in automated path planning as it can enable us to trap or avoid trapping other components during manipulation of the desired components in the workspace. Although simulation has been performed only for dielectric microspheres, the approach is general in nature and can work with other trapping force models as well.

Chapter 4

DEVELOPING A STOCHASTIC DYNAMIC PROGRAMMING FRAMEWORK FOR SINGLE-PARTICLE TRANSPORT

Optical tweezers (OT) based single-particle transport operation involves trapping the desired particle (*target object*), transporting it to the user-specified goal location, and avoiding collisions with other particles (*obstacles*) present in the workspace if possible. This is similar to the single robot motion planning problems that have been discussed in Chapter 2. However, as explained in Chapter 1, the current problem has some unique characteristics due to the small size scales involved. The environment is constantly changing due to the random, Brownian motion of the objects. In addition, OT attracts objects into the laser trap centers even as far away as few microns; so care must be taken to maintain sufficient separation from all the obstacles. By utilizing synchronized time sharing or dynamically created holograms, multiple optical traps can manipulate several objects in parallel. Hence, additional traps can be used to keep nearby obstacles stationary by trapping them. Due to the uncertainty in identifying object positions using an imaging-based sensor, and the time delay between identification and trap placement, the event of trapping an object has a non-zero probability of failure associated with it. This suggests that the approaches used in the domain of probabilistic robotics need to be utilized here. This Chapter develops a stochastic dynamic programming framework to perform the above-mentioned planning operation. More specifically, it uses the well-known Markov assumption to cast a 2D version of the problem as a *partially observable Markov decision process* (POMDP). The automated planner is then validated by performing physical experiments in a holographic tweezer set-up and its performance is characterized using simulation. A lot of the material covered in this Chapter is reported in [Bane08b, Bane09b].

4.1 Problem Formulation

A 3D rectangular parallelepiped shaped assembly cell set-up is considered as the *workspace W*. It is assumed that the laser beam propagates upward along the +*z*-axis. Spherical objects will only be transported in the same horizontal plane. The physics of the problem is modeled as follows:

- Any object in the vicinity of a laser beam in the workspace experiences five forces, namely, optical trapping, Langevin/thermal (causing Brownian motion), viscous drag, gravity, and buoyancy.
- A normal distribution based position sampler is used to generate random, Brownian positions of any object at discrete time instants. Mean of the distribution is time-invariant for every individual object, whereas the standard deviation is equal to the diffusion length.
- Random position sampling is combined with the ray optics model for trapping forces (given in Appendix A), drag term given by Stokes' law, and well-known constant expressions for gravity and buoyancy to generate a list of positions, and velocities of the objects. It should be pointed out here that the trapping force values obtained from the ray optics model are suitably reduced based on the experimental data available for smaller-sized beads that are close to 1 μm in diameter ([Wrig93], [Wrig94]).

- Diffusion length of an object in 3D as a function of time *t* is expressed as $r_t = \sqrt{6Dt}$, where the diffusion coefficient (*D*) is given by Einstein-Smoluchowski relation to be $D = \frac{k_B T}{\gamma}$. Here k_B is the Boltzmann constant and γ is the viscous drag coefficient. From Stokes' law, $\gamma = 6\pi\eta R_a$, where R_a is the radius of the object and η is the viscosity of the fluid medium. It may be noted here that the diffusion length along each dimension is independent of one another and is equal to $\sqrt{2Dt}$.
- Sensor uncertainty or *measurement probability* is modeled using two independent normal distributions (having identical standard deviations σ) along the two coordinate axes. Means of both the distributions coincide with the respective coordinate values of the object under consideration.
- Collisions are modeled using "billiard ball-like" rigid-body, elastic approach. This implies that the total kinetic energy also remains conserved in addition to linear momentum. At a particular simulation step, only pair-wise collisions are handled and chain collisions are ignored. This works well due to the small time interval at which object positions are calculated.
- If multiple objects are trapped by a single laser beam, then they collide with each other and consequently either both of them may get trapped (smaller-sized objects) or none of them may remain trapped (larger objects).
- For relatively large objects (diameter more than 10 μm), the maximum permissible trap speed ($v_{\max}$) is obtained by equating the viscous drag force with the maximum optical trapping force. However, the Brownian motion term dominates for smaller-

sized objects. In that case, $v_{\max}$ is obtained by performing simulation or physical experiments.

- If multiple traps are switched on, then $v_{\max}$ is reduced proportionately. However, this effect can be compensated by increasing the power up to the maximum, attainable value (2.0 W in the holographic tweezer set-up).

Based on the trapping probability contour plots (Figures 3.9 and 3.13) shown in Chapter 3, traps are modeled by a set of two concentric, horizontal circles such that their inner and outer boundaries correspond to *trapping probability* contours equal to 1 and 0 respectively. The probability has a value of 1 inside the circle centered at the laser beam focus with radius equal to trap radius (l_{tr}), is equal to 0 outside the circle with radius equal to the sum of trap radius and trap width ($l_{tr} + l_{tw}$), and varies linearly as a function of distance of the object center from the trap focus (*d*) within the annular region, determined by l_{tr} and ($l_{tr} + l_{tw}$). This is shown in Figure 4.2. Both the radii of the inner circles (l_{tr}) and the widths of the transition zones (l_{tw}) are functions of laser power, object radius, and material properties. Actual values are obtained by performing a large number of simulations as described in Chapter 3. In order to account for trap motion, the lengths of the major axes of the half-ellipses along the direction of motion (as shown in Figure 3.11) at the focal plane for the maximum trap speed are selected as l_{tr} and l_{tw}.

It may be noted here that, unlike many other problems, the time available for planning in this case is limited by the imaging interval (Δt_i), image processing time (t_d), controller update interval (Δt_c), and the continuous random motion of all the non-trapped objects. Fig. 4.2 shows the event diagram for the planning and imaging modules. Ideally, the average planning time (t_p) should be very short as compared to both Δt_i and Δt_c such that the random motion does not change the obstacle positions so much that the plan is rendered ineffective. It is also useful to point out here that the imaging and controller update rates are not necessarily equal or synchronized with each other. In that case, the last available processed image data is utilized immediately for computing the plan. However, the plan is executed only at the next time instant in the AP series representing the controller update times.

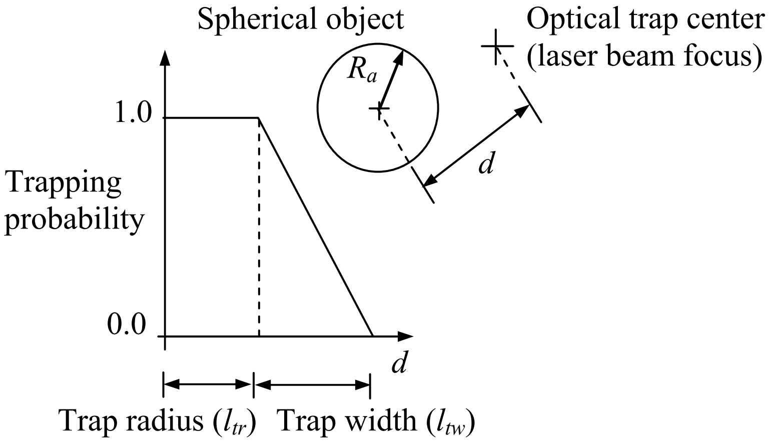


Figure 4.1: Trapping probability model

The radius of effective obstacle region (centered at the current, estimated object position) arising due to presence of optical trap, finite size, and imaging uncertainty is then given by: $r_{Sys} = \max\left\{\left(\left(l_{tr} + l_{tw}\right) + 3\sigma\right), \left(2R_a + 3\sigma\right)\right\}$. The rationale behind the definition of the effective obstacle radius is shown in Figure 4.3. Two objects may collide either due to geometric reasons or as a result of one object coming inside the trapping probability region of the other one. The greater amongst the two regions must be taken for conservative path planning purposes. However, each region must be expanded to account for the uncertainty in sensor readings. This is shown by the grey annular region in Figure 4.3.

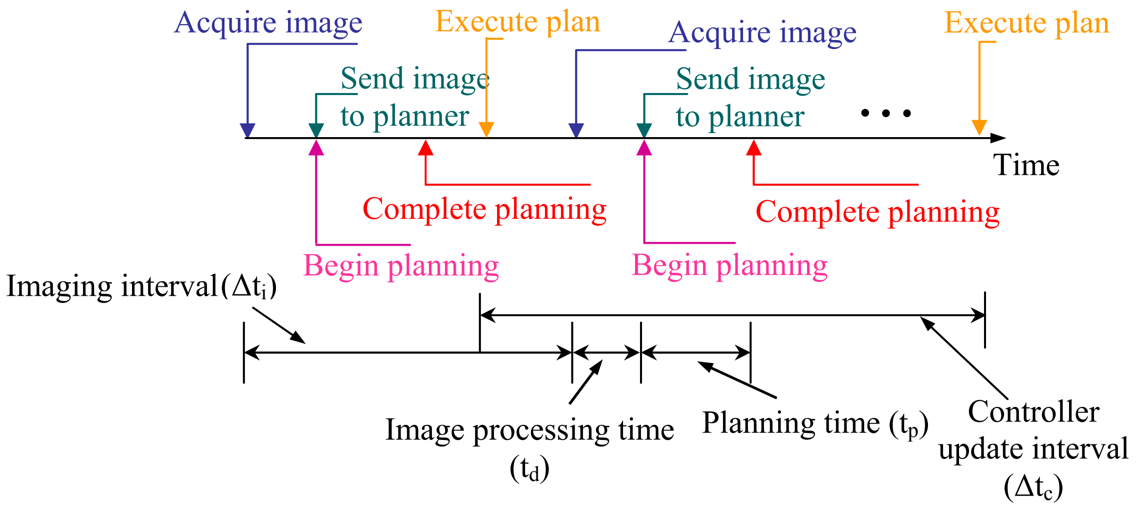


Figure 4.2: Event diagram for planning and imaging

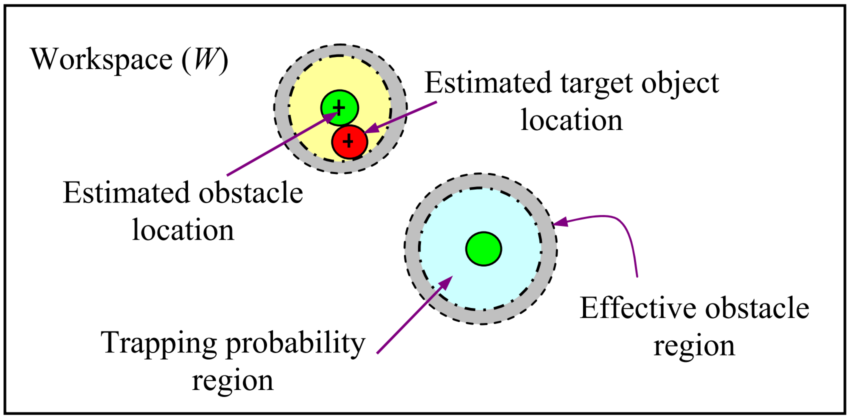

Workspace (*W*)
Estimated target object location
Estimated obstacle location
Effective obstacle region
Trapping probability region

Figure 4.3: Selection of effective obstacle radius

. Actual (physical or simulated) locations of all the objects in the workspace form the true workspace set, whereas the estimated or sensed locations form the estimated workspace set. The primary trap is solely used to transport the target object, whereas the secondary traps are utilized to keep the obstacles stationary. Mathematical definitions of these concepts are given below:

Definitions

True workspace state is represented as the set $X_t = \left\{X_1^t, \ldots, X_n^t\right\}$ where X_i^t is the actual coordinate $\left(x_i^t, y_i^t, z_i^t\right)$ of the *i^th^* object in *W*. Without any loss of generality, the first object can always be considered as the *target object* and the remaining (*n*-1) can be treated as *obstacles*.

Estimated workspace state is represented as the set $X_e = \left\{X_1^e, \ldots, X_n^e\right\}$ where X_i^e is the estimated coordinate $\left(x_i^e, y_i^e\right)$ of the *i^th^* object in *W*. The estimated z-locations of the object centers were only used in Section 4.5. The planner makes all decisions based on X_e and not X_t. It is useful to point out here that all the objects may not be detected at all the times due to limitations in the imaging hardware and feature recognition modules.

Control action set

- **Primary** – A 3-tuple $\left\{s, \left(f_x, f_y\right), \left(v_x, v_y\right)\right\}$ such that the 1st element is a binary integer and last two elements are pairs of floating point numbers. Here *s* denotes whether the *primary trap* that is transporting the target object is switched on (1 if on and 0 if off). $\left(f_x, f_y\right)$ represents the coordinate of the trap center (focus) and $\left(v_x, v_y\right)$ is the uniform velocity vector assigned to the trap. If $s = 0, \left(f_x, f_y\right) = \phi$ and $\left(v_x, v_y\right) = \phi$; else $f_x \in \left[0, x_{\max}\right], f_y \in \left[0, y_{\max}\right], v_x \in V_x, v_y \in V_y$. Here V_x and V_y represent the discrete set of permissible speeds along the X and Y-axes respectively (including 0).
- **Secondary** - A set of 9 doubles $\left\{\left\{s_1, \left(f_x^1, f_y^1\right)\right\}, \ldots, \left\{s_9, \left(f_x^9, f_y^9\right)\right\}\right\}$ such that the 1st element of every double is a binary integer that denotes whether the corresponding

secondary trap is switched on and $\left(f_x^i, f_y^i\right)$ represents the coordinate of the i^{th} trap center. If $s_i = 0, \left(f_x^i, f_y^i\right) = \phi, \forall i = 1,\dots,9$; else $f_x^i \in \left[0, x_{\max}\right], f_y^i \in \left[0, y_{\max}\right], \forall i = 1,\dots,9$. Figure 4.4 shows a typical workspace consisting of a target object and four obstacles.

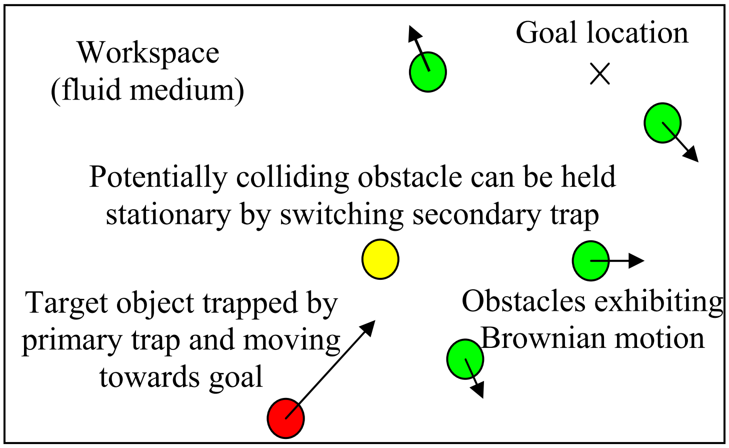

Workspace
(fluid medium)
Goal location
Potentially colliding obstacle can be held stationary by switching secondary trap
Target object trapped by primary trap and moving towards goal
Obstacles exhibiting Brownian motion

Figure 4.4: An example of workspace in OT based particle transport problem

Objective function

A motion planning approach can handle collisions in three possible ways: either by ignoring them and re-trapping the target object, or by circumventing obstacles, or by keeping them stationary at their current locations using a secondary trap. This is illustrated in Figure 4.6 and forms the basis of our objective function. *Estimated, expected transport time* T_e for an estimated workspace state X_e, a control action *A*, and a circular circumvention strategy is defined as follows:

$$T_e\left(X_e, A\right) = \min\left\{ \begin{array}{l} \left(\Delta t_c + \dfrac{\overline{F^{/}G}}{v_{\max}} + \displaystyle\sum_{i=1}^{n_c} p_i t_{avg}\right), \left(\Delta t_c + \dfrac{\overline{F^{/}G}}{v_{\max}} + \dfrac{\pi n_c r_{sys}}{v_{\max}}\right), \\ \left(\Delta t_c + \left(\dfrac{\overline{F^{/}G}}{v_{\max}}\right)\left(1 + \dfrac{w n_c \Delta t_c}{t_e}\right) + \left(1 - w\right)\displaystyle\sum_{i=1}^{n_c} p_i t_{avg}\right) \end{array} \right\}$$

The three terms in the above expression represent the estimated transport time using the three different collision handling strategies shown in Figure 4.5. The first term allows collisions to occur and then imposes a penalty by introducing the additional time taken to recover the target object. The second term explores the option of circumventing all the freely diffusing obstacles and the third term considers the possibility of avoiding collisions by trapping obstacles to prevent them from moving randomly, which decreases the maximum transport speed. $F^{/} = \left(f_x \hat{i} + f_y \hat{j} \right) + \left(v_x \hat{i} + v_y \hat{j} \right) \Delta t_c$ is the new location of the primary trap center after a time interval equal to Δt_c. n_c is the total number of expected collisions with obstacles based on the current X_e as the primary trap first moves with velocity $\vec{v} = v_x \hat{i} + v_y \hat{j}$ and then along $F^{/}G$. Such objects are termed as *potentially colliding obstacles*. t_{avg} is the average time to re-trap an object by positioning it at the displaced location or switching the laser off, allowing the objects to drift and then turning it back after collision. p_i is the trapping probability of the i^{th} potentially colliding obstacle based on the minimum distance of the object from the expected path followed by the primary trap. w is the *expected* probability of avoiding collision by keeping any obstacle stationary. This is not equal to 1 as sometimes collision is inevitable because the trap many pass through the effective obstacle region and keeping it fixed does not help in any way. t_e is the estimated time to reach G from the current location if the primary trap moves with the maximum speed.

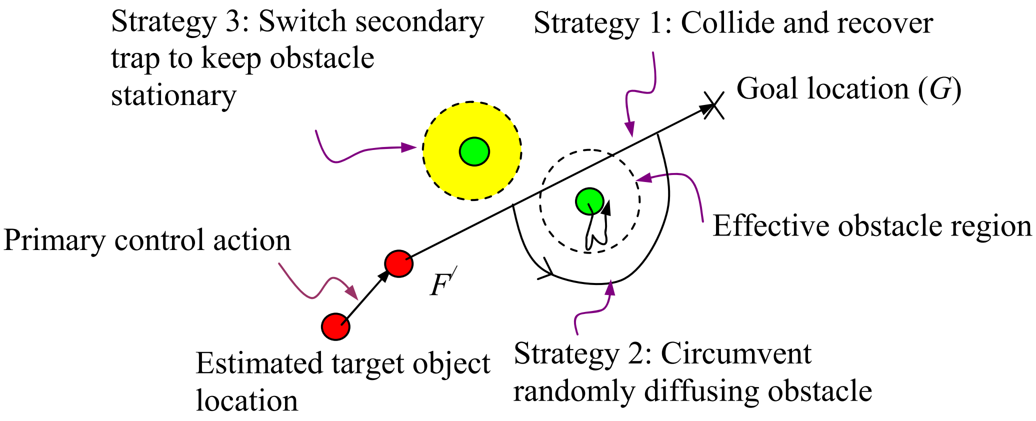

Strategy 3: Switch secondary trap to keep obstacle stationary
Strategy 1: Collide and recover
Goal location (G)
Effective obstacle region
Primary control action
F′
Estimated target object location
Strategy 2: Circumvent randomly diffusing obstacle

Figure 4.5: Different collision avoidance strategies

The above expression is valid for circular circumvention. However, the actual circumvention path may vary from one approach to the other. This will modify the constant multiplicative factor in the 3rd term of the 2nd element. Similarly, the value of *w* may differ as well. However, the basic nature of the definition remains unaltered.

Optimal solution

A sequence of control actions $\left\{A^1,\ldots,A^k,\ldots\right\}$ such that the total, estimated transport time $T=\sum_k T_e^k\left(X_e^k,A^k\right)$ is minimized, where the superscript *k* represents the *k*th instant when the planner is invoked.

Dominance of one control action over another

If $T_e^k\left(X_e^k,A_1^k\right)>T_e^k\left(X_e^k,A_2^k\right),\forall k$ then control action A_2 *strictly dominates* control action A_1.

If $T_e^k\left(X_e^k,A_1^k\right)\geq T_e^k\left(X_e^k,A_2^k\right),\forall k$ then control action A_2 *dominates* control action A_1.

If $T_e^k\left(X_e^k, A_1^k\right) \geq T_e^k\left(X_e^k, A_2^k\right) - \varepsilon^k, \forall k$ where ε^k is a random variable that depends on X_e^k with mean $\bar{\varepsilon}^k = 0$, then control action A_2 *weakly dominates* control action A_1.

If any one of the above conditions is valid, then A_1 can be pruned as it is expected to result in an inferior or sub-optimal solution. If none of the above conditions holds true, then A_1 and A_2 are *non-dominated* with respect to each other. Both of them need to be retained in this case as either one can result in a superior or optimal solution based upon the input parameters and X_e. If *each one* of the elements in a set of control actions dominates (in any one of the above mentioned three ways) *all* the elements of another set of control actions, then the former set is said to dominate the latter one. A *non-dominated control action set* is formed by including those control actions such that all of them are non-dominated with respect to at least one other control action present in this set.

4.2 Path Planning Approach

4.2.1 Algorithm Description

Since two types of uncertainties are present here, namely control action uncertainty and measurement uncertainty, this problem can be conveniently modeled as a POMDP. The control action uncertainty arises due to a combination of measurement uncertainty, trapping probability, and plan execution latency. If the primary trap focus is not placed within a distance equal to l_{tr} from the target object center, then there is a probability that the target object will not move exactly to $F^{/}$. In that case, it may drift slightly away from the inner circle of the trap, and then get pulled back inside it. As a result, it may not always move with the trap velocity and may end up in a slightly displaced location. As l_{tr} is quite large (of the order of R_a), this probability is reasonably small. Nevertheless, it needs to be taken into account. A POMDP model is also known as hidden Markov model or dynamic Bayes network. It utilizes the well-known Markov assumption that the past and future states are independent, provided the current state is known or estimated. Although inaccuracies in modeling the state transition and measurement probability may violate this assumption, Bayes filters used to derive the *belief state* is quite robust to such violations [Thru05]. Belief state refers to the posterior probability distribution of the current true state based on all the past estimated states and control actions.

POMDP algorithms require a *planning horizon* and an *expected cumulative payoff function* or *value function*. This function sums values over a number of time steps (*T*). If *T* is infinite, then one gets the *infinite-horizon* algorithm. The payoff function is analogous to the above described objective function, although the exact form is somewhat different. It uses the *value iteration* formulation of the *Bellman equation* (Step 3.a.i.3 in the algorithm pseudo-code) to ensure optimality of the selected control action. It initializes the value function and then recursively propagates information in reverse temporal order until the Bellman equation converges. The discrete version of an approximate POMDP technique, known as QMDP, has been used due to its ease of implementation and speed-up obtained over a full POMDP solution. In fact, QMDP is a hybrid between Markov Decision Process (MDP) and POMDP and is of the same complexity as MDP [Thru05] based on the assumption that the state is fully observable after one step of control. It generalizes the MDP-optimal value function defined over the estimated state into a POMDP-type value function defined in the belief state by computing the so-called Q function (Step 4.a in the pseudo-code) using the *converged* value function. This Q function is then utilized to select the control action that yields the minimum expected value (Step 6).

The state transition matrix is computed once for all possible locations of the target object center and control actions, and stored in the local memory so that it can be used every time the planner is invoked. Since only a single objective function is present in this case, the problem remains tractable. A rectangular grid with a uniform grid spacing along both the orthogonal directions is overlaid on W. The estimated coordinates of the target object and the obstacle centers are approximated by the nearest grid point. Based upon pilot trial runs, a constant probability value of 0.9 is assigned to the expected grid location of the target object corresponding to the given control action ($x_{n,k}$). Values of 0.025 are assigned to each of the four other neighboring grid points (x_m). This is shown in Figure 4.6. Belief function values are randomly sampled from the normal distribution representing the measurement uncertainty (Step 4.b) and are normalized so that they all add up to 1 (Step 5).

Certain important modifications have been made to the original QMDP algorithm to adapt it for the current application. Firstly, unlike what is commonly done in practice, the payoff function does not merely represent the one-step gain associated with that particular control action. Instead, it takes into account the overall gain corresponding to the given control action. Thus, it is similar in nature to the objective function and, consequently to the value function. In other words, the value function no longer iterates over the value obtained at the previous iteration added to a one-step gain amount. Rather, it iterates over the previous value added to a payoff amount that is of the same order as the value itself. This makes the iteration equation non-linear with respect to the distance (estimated, expected transport time in this case) of the goal from the current state, provided the given control action is implemented.

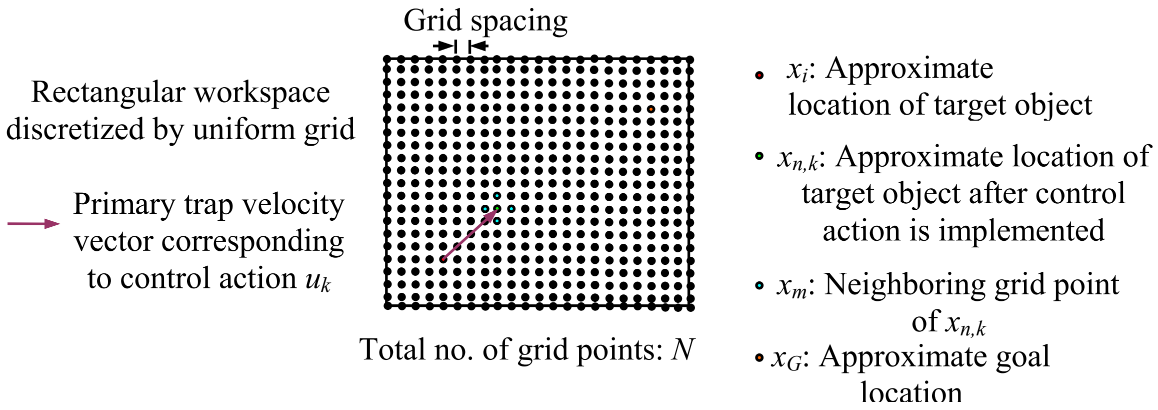

Grid spacing
Rectangular workspace discretized by uniform grid
Primary trap velocity vector corresponding to control action u_k
Total no. of grid points: N
x_i: Approximate location of target object
$x_{n,k}$: Approximate location of target object after control action is implemented
x_m: Neighboring grid point of $x_{n,k}$
x_G: Approximate goal location

Figure 4.6: Approximate representations in QMDP algorithm

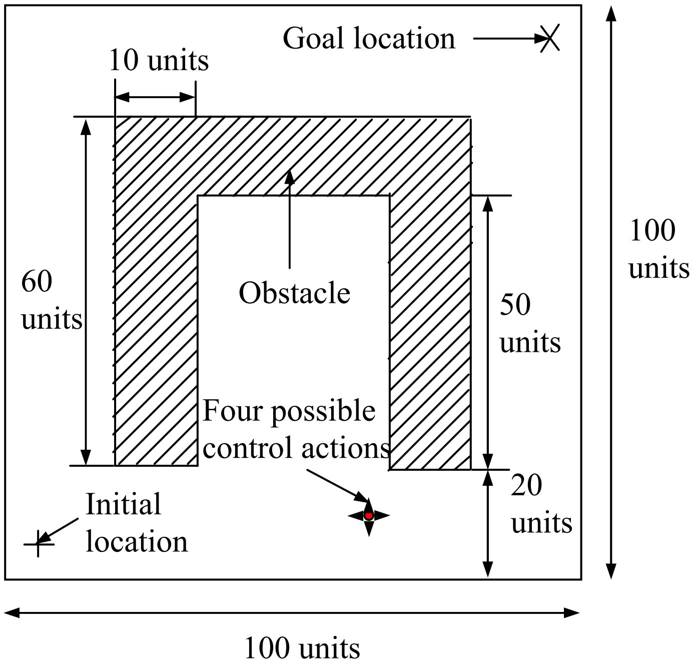


Figure 4.7: Test set-up for comparing one-step gain payoff function with overall gain payoff function

A comparison between the vanilla one-step payoff function and the overall gain-based payoff function is made using a simple 2D test set-up shown in Figure 4.7. The workspace is a square enclosure, with the length of each side being equal to 100 units. A concave obstacle is placed in the workspace as shown in the same Figure. The task is to find the optimum path from the given initial location (5,5) to the given goal location (95, 95) without passing through the obstacle. Only four possible control actions are considered, one each along the four directions that changes the current state by 1 unit along either ±X or ±Y-axes.

Two variants of the discrete, infinite-horizon MDP algorithm are used. The first one replicates the standard version that uses a fixed payoff function such that if any control action results in a new state that lies within the obstacle, then a penalty of 50 units is imposed; a nominal penalty of 1 unit is imposed if the resulting state lies in the free region, and a reward of 20 units is assigned if the goal location is reached. The second one uses the formulation described in this work. No sensor uncertainty is modeled and the effect of state transition probability is also ignored to isolate the effect of the payoff function on the convergence time and path quality. It is observed that there is no appreciable change in the quality of paths generated by the two algorithms. However, the overall gain-based formulation does, indeed, reduce the convergence time by approximately 46%. While the number of iterations required for convergence decreases by almost 57%, the net computational saving is slightly reduced due to the greater amount of time spent in computing the payoff function at every iteration. These values are based on the average taken over the entire path from the initial to the goal location. Although the exact reduction values will depend upon the shape, size, and number of obstacles, this simple experiment highlights the utility of modifying the payoff function. Further reduction in computational time will take place in OT set-ups as the state transition probability matrix need not be computed at run-time unlike in the case of the vanilla algorithm.

The second major modification in the QMDP algorithm is that time is introduced as an additional parameter within the convergence loop of the algorithm. As discussed earlier in this Chapter, this is necessitated by the fact that certain planning options that were optimal at the beginning of the planning time, may lead to inferior solutions with the passage of time and vice versa. Since all the non-stationary obstacles are exhibiting Brownian motion, it is known that the mean position of all the objects will remain unchanged with time. Hence, stochastically speaking, no better solution can be obtained by changing the locations of the obstacles from their current, estimated values during the course of the computation. However, the same is not always true for the target object. It may be still moving with the previous velocity even after it has reached $x_{n,k}$ unless it is commanded to act otherwise. This usually happens in the case of scanning mirror based set-up (but not for the holographic one). If by the time the plan is executed, it has already moved quite close to an obstacle, then any further motion along a similar direction will definitely result in collision. However, such a motion might have been the best option earlier when the planner started the computation. So, if the effect of the motion of the target object with time is accounted for, then such moves can be avoided and an appropriate circumvention plan can be obtained.

All the steps are formally described in the algorithm QMDP_WithNonLinearTimeDependentValueFunction (QMDP_NLTDV). Discussions on the optimality of the MDP portion of the algorithm and convergence of the Bellman equation are presented in Section 4.2.3. An important issue with the approach is the size of the control action set. It has to be *pruned* in order to obtain results within few milliseconds. The pruning conditions, based upon the concept of non-dominated control action set defined earlier, are explained in the following sub-Section.

Algorithm: QMDP_NLTDV

Symbols

- Set of rectangular grid coordinates $X = \{x_1, \ldots, x_N\}$ representing the state space
- Value function V
- Belief function b
- Expected payoff function r
- Time parameter t
- Iteration counter q
- Span of a vector $sp(w) = \max_i w_i - \min_i w_i$

Input

- Approximate goal location $x_G \in X$
- Approximate, estimated location of target object center $x_T \in X$
- Set of approximate estimated locations of all the obstacle centers $E = \{y_1, \ldots, y_{n-1}\}$, where $y_i \in X, \forall i$
- Set of pruned control action sets $U = \{u_1, \ldots, u_m\}$

- Set of state transition probability matrices $P = \{p_1, \ldots, p_m\}$, where $p_k \equiv p_k\left(x_j \mid u_k, x_i\right) \forall i, j$ represents the matrix corresponding to the *k*th control action set
- R_a, Δt_c, t_d, l_{tr}, l_{tw}, σ
- Maximum trap transport velocity vector $v_m = \begin{bmatrix} v_{xm} & v_{ym} \end{bmatrix}^T$; $v_{\max} = \left|v_m\right|$
- Previous primary trap velocity vector $v_p = \begin{bmatrix} v_{xp} & v_{yp} \end{bmatrix}^T$

Output

- Optimum control action set u^*

Steps

```
1. Initialize $t = t_d$ and $q = 0$.
2. For $i = 1$ to $N$, do $V^q\left(x_i, E, t\right) = \Delta t_c$.
3. Do while $q \le 1$ or $sp\left(V^q - V^{q-1}\right) \ge 1$
    a. For $i = 1$ to $N$, do
        i. For $k = 1$ to $m$, do
            1. If ($x_i == x_T$), compute the new grid location $x_{n,k} = \left(x_T + v_p t\right) + v_k \Delta t_c$, where $v_k$ corresponds to the primary trap velocity vector of control action set $u_k$; else compute $x_{n,k} = x_i + v_k \Delta t_c$
            2. If $\left\|x_i - x_G\right\| \le R_a$, $r\left(x_i, E, u_k, t\right) = 0$; else if $\left\|x_{n,k} - x_G\right\| \le R_a$, $r\left(x_i, E, u_k, t\right) = \Delta t_c$; else
            $r(x_i,E,u_k,t)=\min\left\{\begin{matrix}\sum_{l=1}^{n_c} p_l t_{avg}, \\ \left(\frac{\overline{x_{n,k}x_G}}{v_{\max}}\right)\left(\frac{n_s \Delta t_c}{t_e}\right)+\sum_{l=n_s}^{n_c} p_l t_{avg}\end{matrix}\right\}$ (see note at the end of the algorithm for explanation)

        3. Compute
            $\hat{V}(x_i,E,u_k,t)=r(x_i,E,u_k,t)+\sum_{j=1}^{N} V^q(x_j,E,t)\,p(x_j \mid u_k,x_i)$

    ii. Obtain $V^q(x_i,E,t)=\min_{u_k}\left[\hat{V}(x_i,E,u_k,t)\right]$

  b. Increment $q$ by 1 and $t$ by the actual time taken to complete Step a.

4. For $i$ = 1 to $N$, do

  a. For $k$ = 1 to $m$, do $Q(x_i,E,u_k,t)=r(x_i,E,u_k,t)+\sum_{j=1}^{N} V^q(x_j,E,t)\,p(x_j \mid u_k,x_i)$

  b. Sample $b_i$ randomly from a normal distribution with mean and standard deviation
     equal to $x_i$ and $\sigma$ respectively and multiply it with the predicted value.

5. Normalize belief function values such that $\sum_{i=1}^{N} b_i = 1$.

6. Return $u^* = \arg\min_{u_k} \sum_{i=1}^{N} b_i Q(x_i,E,u_k,t)$.
```

It should be noted here that all the objects whose effective obstacle regions intersect the trajectory of the primary trap as it moves from x_i to $x_{n,k}$ and then from $x_{n,k}$ to x_G along a straight line path are considered to be potentially colliding. n_s is the number of activated secondary traps. Trapping probability p_l for the l^{th} potentially colliding object is obtained by considering the minimum distance between its estimated location and the trajectory of the primary trap and utilizing l_{tr} and l_{tw}. The payoff function is derived from the objective function defined in Section 4.1, by ignoring the common Δt_c from all the three terms, not considering the 2nd term, and by substituting $w = 0$ for the potentially colliding obstacles that are trapped using secondary traps and $w = 1$ for the remaining ones. The selection of obstacles to be trapped is governed by pruning condition 4 explained in the following sub-Section. The 2nd (circumvention) term is not required in this algorithm as different ways of maneuvering around obstacles are automatically considered in the form of primary trap velocity vectors spanning the full horizontal plane at every state.

4.2.2 Control Action Set Pruning Conditions

Condition 1

Let $A^{/}$ be a set of control actions such that the centers of the primary traps corresponding to the elements in this set are located either at $X_1^e + \left(t_d + t_p\right) v_p$ or at X_1^e. For every element $A_j^{/} \in A^{/}$, let $A_{jd}^{/} \in A_d^{/}$ be the set of control actions that have identical secondary traps but distinctly different primary traps. If $\sigma, D << l_{tr}$ (as is usually the case), then $A_j^{/}$ weakly dominates $A_{jd}^{/} \forall j$. This is illustrated in Figure 4.8. Hence, we retain $A^{/}$ and discard $A_d^{/}$ from the set of control actions to be used for computing the optimum plan.

Rationale

From the physics of optical tweezers, it is clear that if X_1^t lies inside the circle centered at the focus of the primary trap in $A^{/}$ with radius equal to the trap radius (referred to as the *unit trapping probability circle*), then the target object is stably trapped and is expected to move along with the trap with zero relative velocity. In reality, it may still exhibit some diffusion, which can be ignored. If it lies in the annular region with inner radius equal to l_{tr} and outer radius equal to $\left(l_{tr} + l_{tw}\right)$, then it is not stably trapped. It may either drift towards the stronger trapping force region (with a probability given by the trapping probability value at the corresponding *d* value) or away from it. If it goes outside the annular region, then it experiences negligible optical trapping forces and undergoes free diffusion.

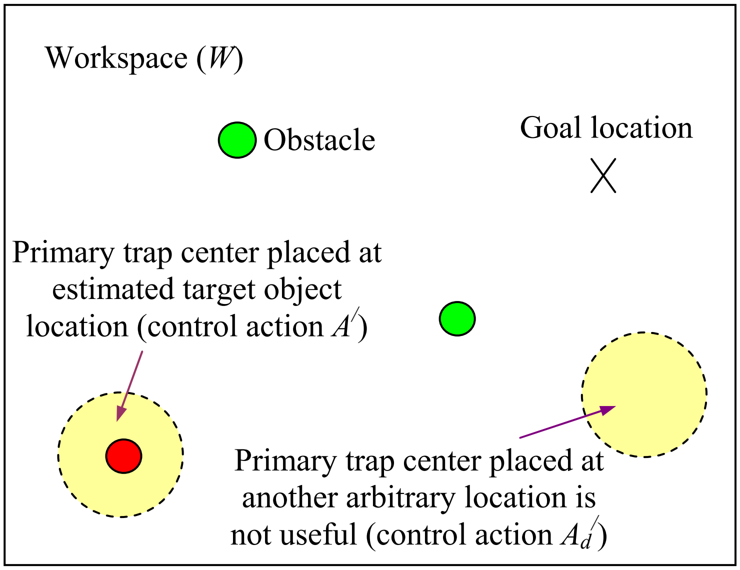

Workspace (*W*)
Obstacle
Goal location
Primary trap center placed at estimated target object location (control action $A^{/}$)
Primary trap center placed at another arbitrary location is not useful (control action $A_d^{/}$)

Figure 4.8: Pruning condition 1 illustration

Since the sensor provides an estimate of X_1^t with a certain degree of uncertainty (characterized by σ) to the planner, it can never position the primary trap exactly at X_1^t

with a probability of 1. However, if the primary trap center is placed at X_1^e, then it has a very high probability of lying inside the unit trapping probability circle as $\sigma << l_{tr}$ and X_1^e lies within a circle of radius 3σ, centered at X_1^t with a probability of approximately 0.9996. If we place the trap center at any other location in *W*, it is going to increase T_e value as the target is either not being moved closer to the goal *G* at all or it is being moved after spending additional time in bringing the object to a stable trap position ($\overline{F^{/}G}$ goes up in both cases).

Now, due to the image processing time (t_d) and the finite amount of time taken by the planner to come up with the optimal control action (t_p), the target object may get displaced in *W* from X_1^t. If the object is being constantly transported by the primary trap (as in the case of scanning mirror based set-up), then its estimated position after time $t = t_d + t_p$ will be equal to $X_1^e + \left(t_d + t_p\right)v_p$ (current iteration is at time $t = 0$). This additional $\left(t_d + t_p\right)v_p$ term is equal to zero for the holographic set-up. If instead of being trapped, the target object is freely diffusing in the fluid medium (due to collision event or inaccurate positioning or primary trap), then the diffusion length ($r_{t_d+t_p}$) within this time interval will be equal to $\sqrt{6D\left(t_d + t_p\right)}$. As $D << l_{tr}$, $r_{t_d+t_p} < l_{tr}$ for relatively small values of t_d and t_p of the order of few milliseconds, and hence the target object is expected to lie inside the unit trapping probability circle, centered at X_1^e, with a very high probability. Of course, due to an ensuing collision within this time interval $t_d + t_p$, the object may be knocked off from the primary trap and may start diffusing freely. However, this is not expected to occur in practice (again due to small values of t_d and t_p). Thus, this condition follows from definition of weak dominance of one control action set over another.

Condition 2

Let $A^{//}$ be a set of control actions such that the centers of the secondary traps corresponding to the elements present in this set are either null sets or distinctly different and located at the obstacle centers $X_i^e, i = 2, \ldots, n$. For every element $A_j^{//} \in A^{//}$, let $A_{jd}^{//} \in A_d^{//}$ be the set of control actions that have identical primary traps to that in $A_j^{//}$, but secondary traps where the centers are not positioned at $X_i^e, \forall i$. Then $A_j^{//}$ dominates $A_{jd}^{//} \forall j$. This is shown in Figure 4.9. Hence, we retain $A^{//}$ and prune $A_d^{//}$ from the set of control actions that will be used to compute the optimum plan.

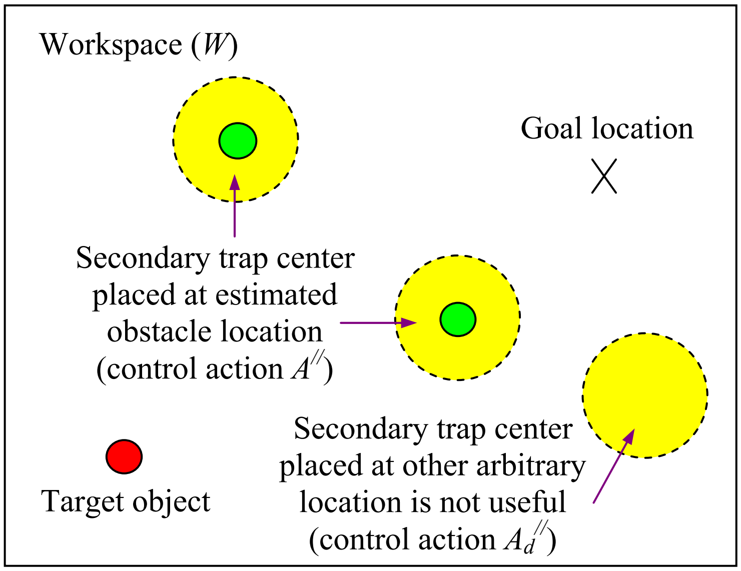

Workspace (W)
Goal location
Secondary trap center placed at estimated obstacle location (control action $A^{//}$)
Target object
Secondary trap center placed at other arbitrary location is not useful (control action $A_d^{//}$)

Figure 4.9: Pruning condition 2 illustration

Rationale

If the center $\left(f_x^{jl}, f_y^{jl}\right)$ of the secondary trap corresponding to control action $A_j^{//}$ is not a null set, s_l must be equal to 1. In other words, the l^{th} secondary trap is switched on. From the principle of working of optical tweezers, this results in a proportionate reduction of $v_{\max}$. Hence, the objective function value increases. Now, if $\left(f_x^{jl}, f_y^{jl}\right)$ is placed outside the trapping probability region centered at X_i^e, it will not be able to avoid any potential collision. Thus, the value of n_c remains same and correspondingly T_e is unaffected. If $\left(f_x^{jl}, f_y^{jl}\right)$ is placed anywhere inside the unit trapping probability circle (including at X_i^e), the value of n_c goes down by 1 and the value of T_e decreases. If $\left(f_x^{jl}, f_y^{jl}\right)$ is placed inside the annular region, the corresponding trap may or may not be able to trap the obstacle and keep it stationary at that position (henceforth referred to as *grasping*). So the value of the objective function may either remain unchanged or at best equal that in the last case. Hence, if $\left(f_x^{jl}, f_y^{jl}\right)$ is placed at X_i^e, then $A_j^{//}$ dominates other control actions which have l^{th} secondary trap centered anywhere else.

If we place multiple secondary traps at same X_i^e, it is not going to reduce n_c any further. Instead, it is going to decrease $v_{\max}$ even more and thereby increase the value of T_e. Hence, $\left(f_x^{jl}, f_y^{jl}\right)$ and $\left(f_x^{jm}, f_y^{jm}\right)$ must correspond to the estimated locations of the centers of two different obstacles $\forall l \neq m$ in order to introduce the possibility of decreasing T_e by eliminating additional collision. In other words, if the secondary traps in

$A_j^{//}$, centered at various estimated obstacle centers are distinctly different, then it dominates the set $A_{jd}^{//}$.

Condition 3

Let A_m be a control action such that the speed of the corresponding primary trap velocity vector is equal to the maximum, possible value. Further let $A^{///}$ be a set of control actions such that the primary traps corresponding to the elements present in this set have the same velocity vector direction as that of A_m, have lower speeds than v_m, and the secondary traps are all identical to that of A_m. If $d_{\min} > r_{Sys} + r_{\Delta t_c}$, where $d_{\min}$ is the distance of the nearest obstacle from the trajectory of the primary trap of A_m, then A_m weakly dominates $A^{///}$. This is illustrated in Figure 4.10. Hence, we retain A_m and discard $A^{///}$ from the set of control actions to be used for computing the optimum plan.

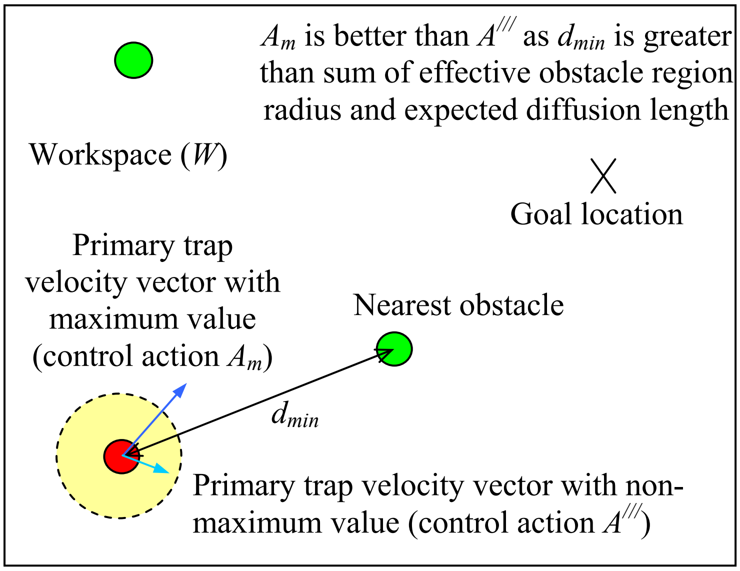

A_m is better than $A^{///}$ as d_{min} is greater than sum of effective obstacle region radius and expected diffusion length
Workspace (W)
Goal location
Primary trap velocity vector with maximum value (control action A_m)
Nearest obstacle
d_{min}
Primary trap velocity vector with non-maximum value (control action $A^{///}$)

Figure 4.10: Pruning condition 3 illustration

Rationale

The nearest obstacle is expected to diffuse by an amount equal to $r_{\Delta t_c}$ within the time interval between the current iteration and when the planner control action will be implemented next (equal to Δt_c). Hence, the distance may be different from $d_{\min}$ at time $t = \Delta t_c$ (current iteration is at time $t = 0$). However, in the worst case, it is expected to be equal to $d_{\min} - r_{\Delta t_c}$, when the obstacle moves by $r_{\Delta t_c}$ directly towards the trajectory of the primary trap and the minimum distance condition occurs exactly at time $t = \Delta t_c$. Even in that case, the target object trapped by the primary trap in A_m is expected to lie just outside the effective obstacle region given by r_{Sys}. Hence, A_m is expected to result in the least possible objective function value as compared to all the other control actions present in $A^{///}$, because $\overline{F^{/}G}$ will be shortest in this case since maximum distance towards the goal is covered within the time interval Δt_c. This is true since no collision is expected to occur when the primary trap is moving at the maximum speed.

Condition 4

Let $A_d^{////}$ be a set of control actions such that the first elements of at least p of the secondary traps are equal to 1, where $1 \le p \le 9$. For every element $A_{jd}^{////} \in A_d^{////}$, let $A_j^{////} \in A^{////}$ be the set of control actions that have identical primary traps to that in $A_{jd}^{////}$ but different secondary traps such that the first elements of all of them are equal to 0. Also let $d_{\min}^i$ be the minimum distance between the trajectory of the primary trap of $A_j^{////}$ and X_i^e. If $d_{\min}^i < r_{Sys} - 3\sigma$ or $d_{\min}^i > r_{Sys} + r_{\Delta t_c}$, for m obstacles ($m \in [2, n]$), then $A_j^{////}$ weakly dominates $A_{jd}^{''''} \forall j$, when $p = m, m \leq 9$ and $p = 9, m > 9$. This is shown in Figure 4.11. Hence, we prune $A_{d}^{''''}$ from the set of control actions that will be used to compute the optimum plan.

Rationale

From condition 2, it is clear that in order to obtain an optimal solution, the l^{th} secondary trap corresponding to any control action will be switched on (i.e. s_l will be equal to 1 and $\left(f_x^l, f_y^l\right)$ will not be a null set) only when we want to grasp a particular obstacle in *W*. Moreover, each secondary trap will be used to grasp a different obstacle. Even then, under the given conditions, if any one of the secondary trap centers present in a control action belonging to the set $A_{jd}^{''''}$ is placed at X_i^e, T_e is expected to increase. If the first inequality in the condition statement is satisfied, it is expected that the target object (trapped by the common primary trap in $A^{''''}$ and $A_{jd}^{''''}$) lies within the effective obstacle region of the obstacle that is kept stationary by the secondary trap with a probability of approximately 0.9996. And if the second inequality in the condition statement is satisfied, the target object is always expected to lie outside the effective obstacle region (as explained in condition 3). In both the cases, the corresponding control action is not expected to generate any better solution by reducing the value of n_c ($v_{\max}$ always decreases).

If $m = p \leq 9$ obstacles are present in *W* that satisfy one of the two inequalities, then this argument is valid for any *p* secondary traps present in the corresponding control action. If $m > 9$ obstacles satisfy one of the two inequalities, then this argument holds true for all the 9 secondary traps. In other words, $A_j^{''''}$ weakly dominates $A_{jd}^{''''} \forall j$ as $A_j^{''''}$ is always expected to yield smaller objective function value. This condition ignores the effect due to the interaction between multiple optical traps. If the intersection of multiple traps modifies the effective obstacle region, then the inequalities have to be changed.

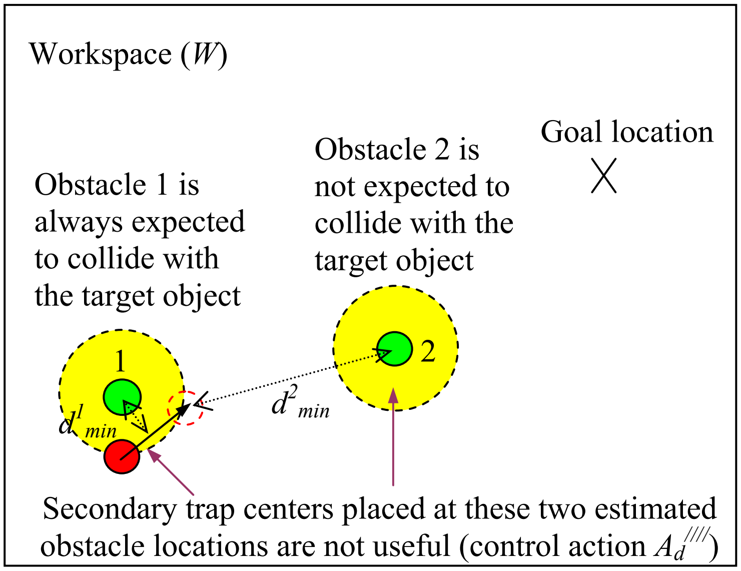

Workspace (*W*)
Goal location
Obstacle 1 is always expected to collide with the target object
Obstacle 2 is not expected to collide with the target object
1
2
d^{1}_{min}
d^{2}_{min}
Secondary trap centers placed at these two estimated obstacle locations are not useful (control action $A_d^{////}$)

Figure 4.11: Pruning condition 4 illustration

Pruning Strategy

Non-dominated control action set is discrete and consists of a subset of the control action set. Conditions 1, 2, 3, and 4 discretize the control action set by imposing restrictions on the choices of control actions from the optimality point of view. This subset of the control action set (identified by the four conditions) forms our non-dominated control action set from basic definition.

It should be noted here that since the path planning approach models control action uncertainty in the form of state transition probabilities and/or weighing factors in the objective function. This makes it robust to violations of the assumptions in condition 1 and high values of t_d and t_p .

4.2.3 Optimality and Convergence Discussion

Discussion on Optimality

The optimality of the MDP portion of the QMDP_NLTDV algorithm is believed to follow directly from the Proposition 13.13 given in [Gosa03] for the value iteration case. Although the payoff function may vary in the current case with time for the state corresponding to the estimated target object location, it is always bounded and finite since trapping probability, distance to goal, estimated time to reach the goal from current location, number of potentially colliding obstacles, and number of secondary traps all lie within well-defined ranges. The remaining parameters do not vary with time. Hence, the optimality proof remains essentially unchanged as there is no change to the limit as the number of iterations tends towards infinity.

Discussion on Convergence

Unlike in most existing algorithms where the payoff function takes on constant values, it depends on several parameters in the current algorithm. Some of the parameters, namely, goal location, controller update interval, time to re-trap the target object after collision, and maximum trap speed remain constant at all iterations inside the convergence loop and thereby do not have any effect on convergence. As mentioned in the optimality discussion, the other parameters may change with time if the state corresponds to the estimated target object location.

Now, typically the planning time t_p is constrained to lie within few tens of milliseconds based on t_d and Δt_c. Based on the fact that $v_{\max}$ is usually few µm/s for (1-20) µm diameter particles considered here, the target object is expected to move less than a micron within the time taken for convergence, which is going to be smaller than t_p . As a result, it is expected that at most one non-potentially colliding obstacle may become potentially colliding and vice versa. Trapping probability values of the other potentially colliding obstacles are not expected to change much either since the minimum distances of the obstacles from the expected trajectory of the target object do not change appreciably. Moreover, the increase in values for certain obstacles is expected to counterbalance the effect of decrease in the others in a typical optical tweezer workspace where obstacles are likely to be positioned randomly all around the target object. The effect of the change in the distance from $x_{n,k}$ to x_G in the numerator of the 1st element in the 2nd term is also counterbalanced by the corresponding change in t_e in the denominator.

Hence, the overall change in the payoff function with time is expected to be monotonic and quite small (dominated by whether the number of potentially colliding obstacle changes). Even if a switch occurs inside the payoff function from the 1st term to the 2nd or vice versa, this switch is only expected to happen once and then it always takes the value of the same term throughout the rest of the convergence process. It should also be noted that this change in the payoff function for one particular (target object location) state may not affect the span based convergence criterion at all since that depends only on the maximum and minimum value function. All the above-mentioned hypotheses are validated by performing the following set of simulation experiments. More details about the simulation framework are mentioned in the following sub-Section.

50 obstacles, each 2 μm in diameter, are placed in a 90 μm X 80 μm X 50 μm workspace. The planner is run when the target object is positioned at (45 μm, 40 μm, 5

μm) and the goal is located at (85 μm, 80 μm, 5 μm). Laser power is selected as 0.2 W. Δt_c, t_d, and t_d are set to be 100 ms, 100 ms, and 80 ms respectively. $v_{\max}$ is chosen as 7 μm/s. So, the target object can only move by 0.56 μm in the worst case. The planner is run 100 times, every time with an entirely random placement of obstacles in the XY plane and fixed Z location, equal to 5 μm. Convergence is always observed; 97% of the experiments resulted in convergence within 12 iterations and the remaining 3 converged after 15, 16, and 20 iterations respectively. Clearly, this shows that the changes in payoff function of the particular state only affected the convergence in 3 cases. Further analysis shows that the number of potentially colliding obstacles changed in 13 cases (always by one), the payoff function switched in 6 cases, and never switched back and forth in any case. The change in payoff function from the original value always lies within 5% and the change is monotonic in every experiment. It is also seen that change of more than 0.5% only occurs in the 13 cases where a new obstacle becomes potentially colliding or vice versa.

Based on the above discussion, the convergence of the Bellman equation then follows from the Proposition 13.20 given in [Gosa03] for the average reward case. This is true since all the states in every Markov chain associated with the MDP algorithm are recurrent as any one of them can be visited infinite number of times over an infinitely long period of time (i.e. there exists no transient or absorbing state) and every Markov chain is regular (discussed below). It may be noted here that since QMDP_NLTDV algorithm simply extends the MDP algorithm to the belief space by defining Q function and then finding the control action that minimizes the expected Q function value (Steps 4-6 in the pseudo-code), these additional steps do not have any other convergence requirements. In other words, QMDP_NLTDV algorithm converges when the Bellman equation converges.

Any arbitrary column *v* of the *N* X *N* state transition probability matrix p_k for transitioning from any state x_i to any other state x_j for a given control action set u_k is given by $p_k^v = \begin{bmatrix} 0 & \dots & 0 & 1/40 & 0 & \dots & 1/40 & 9/10 & 1/40 & \dots & 0 & 1/40 & 0 & \dots & 0 \end{bmatrix}^T$ where 9/10 is the probability value corresponding to the state $x_{n,k}$ (as defined in Step 3.a.i.1 of the pseudo-code) and 1/40 is the probability value corresponding to one of the four states x_m. 9/10 is always preceded and followed by non-zero entries due to the way probabilities are assigned to the neighboring grid points (as shown in Figure 4.6). Clearly, $\sum_{u=1}^{N} p_k^{uv} = 1$, although $\sum_{v=1}^{N} p_k^{uv}$ may not be necessarily equal to 1. In some cases, if $x_{n,k}$ corresponds to a boundary state (state lying on the workspace boundaries), then the column vector may look like $p_k^v = \begin{bmatrix} 9/10 & 1/30 & 0 & \dots & 0 & 1/30 & 1/30 & 0 & \dots & 0 \end{bmatrix}^T$ or some equivalent form.

A probability matrix is defined to be regular if all its entries are strictly positive [Sadu08]. Now, in order for every Markov chain to be regular, it needs to be verified whether the corresponding p_k or some power of p_k is regular [Gosa03, Sadu08]. In other words, if every element of the product matrix $\left(p_k\right)^n > 0$ for some positive integer *n*, then it can be concluded that the corresponding Markov chain is regular. This regularity property is shown numerically for the following representative 7 X 7 state transition probability matrix.

$$p_k = \begin{bmatrix} 1/40 & 1/40 & 0 & 9/10 & 9/10 & 9/10 & 9/10 \\ 1/40 & 0 & 0 & 1/30 & 1/30 & 1/30 & 1/30 \\ 9/10 & 1/40 & 1/40 & 1/30 & 0 & 0 & 0 \\ 1/40 & 9/10 & 1/40 & 1/30 & 1/30 & 0 & 0 \\ 1/40 & 1/40 & 9/10 & 0 & 1/30 & 1/30 & 0 \\ 0 & 0 & 1/40 & 0 & 0 & 1/30 & 1/30 \\ 0 & 1/40 & 1/40 & 0 & 0 & 0 & 1/30 \end{bmatrix}$$

Consequently, $$\left(p_k\right)^3 = \begin{bmatrix} 0.8157 & 0.0752 & 0.1024 & 0.1011 & 0.0747 & 0.0757 & 0.0757 \\ 0.0313 & 0.0233 & 0.0249 & 0.0050 & 0.0048 & 0.0048 & 0.0048 \\ 0.0444 & 0.7732 & 0.7916 & 0.0707 & 0.0977 & 0.0977 & 0.0976 \\ 0.0330 & 0.0536 & 0.0541 & 0.0465 & 0.0473 & 0.0472 & 0.0472 \\ 0.0708 & 0.0708 & 0.0259 & 0.7352 & 0.7343 & 0.7333 & 0.7333 \\ 0.0026 & 0.0014 & 0.0002 & 0.0204 & 0.0203 & 0.0203 & 0.0204 \\ 0.0019 & 0.0022 & 0.0009 & 0.0209 & 0.0209 & 0.0209 & 0.0209 \end{bmatrix},$$

where all the entries are strictly positive and the entries in every column add up to 1 within the limits of numerical errors.

To conclude the discussion on convergence, it can be summarized that the QMDP_NLTDV algorithm is always expected to converge for typical optical tweezers environments (randomly positioned objects). Indeed, non-convergence has never been encountered in any of the experiments reported or conducted as a part of this dissertation. If, however, convergence does not occur in a pathological case (expected only in a carefully construed scene) due to positioning of target object, goal, and obstacles in some particular way resulting in a large change in payoff function values with time, then the algorithm is designed to transition to a greedy heuristic described in [Bane08b] when 5 milliseconds of available planning time is remaining.

4.2.4 Simulated Planning Trajectories

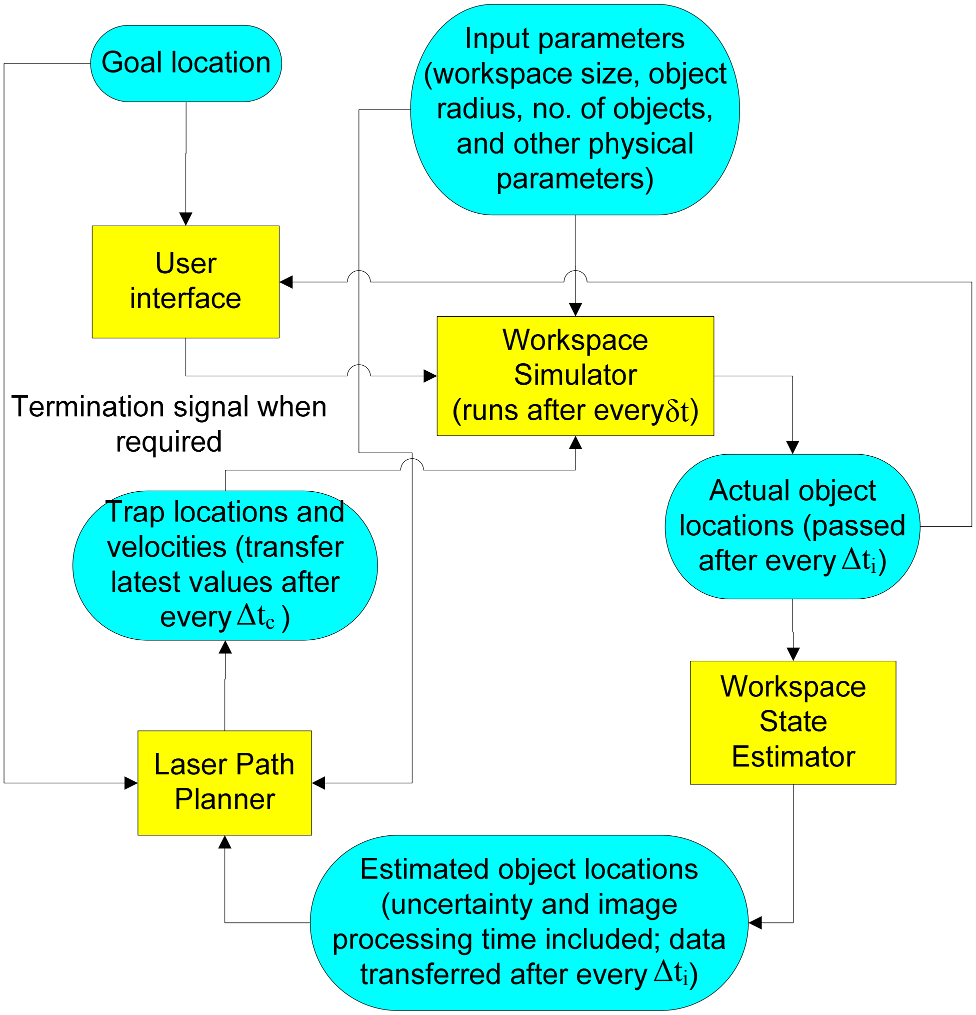

Goal location
Input parameters (workspace size, object radius, no. of objects, and other physical parameters)
User interface
Workspace Simulator (runs after every δt)
Termination signal when required
Actual object locations (passed after every Δt$_i$)
Trap locations and velocities (transfer latest values after every Δt$_c$)
Workspace State Estimator
Laser Path Planner
Estimated object locations (uncertainty and image processing time included; data transferred after every Δt$_i$)

Figure 4.12: Simulation framework: Connection between simulator, estimator, planner, and user interface

The overall physical system has been replicated in the form of a simulation framework shown in Figure 4.12. The workspace simulator takes physical parameters as the input and generates the simulated locations of the object centers after every time interval $\delta t < \frac{m}{\gamma}$ (as mentioned in Chapter 3), which is the relaxation time of the object. Here, *m* represents the mass of the object. The collision function is replicated from the free source code available online at [Coll3D]. The user interface receives data after every Δt_i (50-500 ms) time interval from the workspace simulator and decides whether the target object has reached the goal location. The workspace state estimator also obtains data at the same rate from the simulator and adds noise to the true locations to simulate imaging system uncertainty. It passes estimated positions of the object centers to the laser path planner with an appropriate delay (t_d always selected as 50 ms) to simulate imaging system latency. The planner then comes up with new locations and velocities of the traps. It transfers the latest values every Δt_c (50-1000 ms) to the workspace simulator.

Four test cases have been presented to visualize sample 2D trajectories generated by the QMDP_NLTDV algorithm. It is implemented in C++, using Microsoft Visual Studio.Net 2003 as the compiler. OpenGL is used for rendering purposes. Here, simulations have been performed on 2.0 μm diameter spheres, immersed in water at 20° C. Correspondingly, l_{tr} and l_{tw} are chosen to be equal to 1.0 μm and 1.5 μm respectively. The dimension of the workspace is 50 μm X 50 μm. All the other parameters have the same values as specified earlier in this Section. Figure 4.13 shows the trajectory followed by the target object in a moderately crowded scene consisting of 30 obstacles. The estimated locations of the object centers are shown by '+' signs, whereas, they have been drawn at their actual locations. Brownian motion trajectories of all the untrapped objects are also shown in the Figure. This Figure clearly illustrates the capability of the algorithm to detect the presence of obstacles and follow a circumvented path in order to avoid unnecessary collisions.

Figure 4.14 shows the usefulness of this dynamic programming framework as opposed to a simple, conservative scheme, wherein a search strategy can be employed to just follow the safest route based upon increasing collision regions (due to increase in diffusion length with time) in the free *C*-space. Such a conservative scheme will not be able to guide the target object through passages that may have opened up with time. Thus, it will not always yield the minimum, possible expected transport time. On the other hand, the infinite-horizon algorithm presented here is an adaptive and dynamic approach that constantly generates an optimum solution based upon the new state of the workspace, every time it is invoked. That is why, although the passage was too narrow in the initial workspace state (see Figure 4.14(a)), later on it became wider, and the primary trap could successfully transport the target object through this widened passage without causing any unwanted collision(s) (see Figure 4.14(b)). This results in an overall decrease of transport time in the range of (25 – 30) % as compared to the conservative scheme.

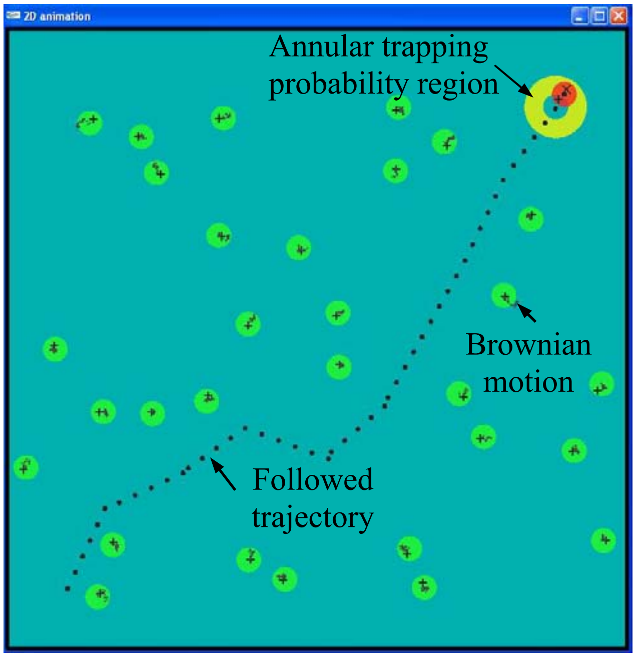
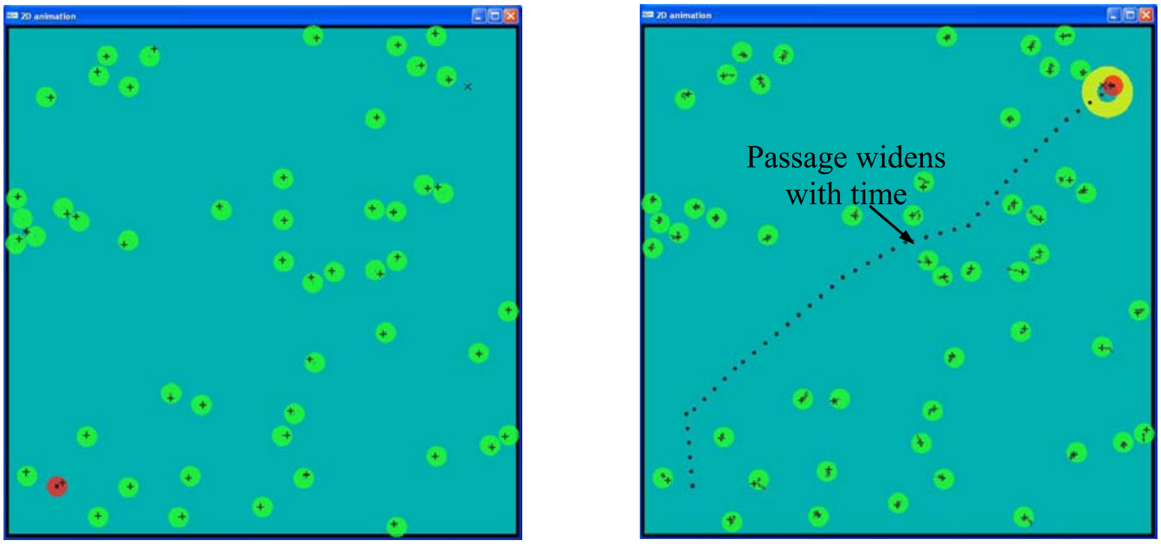


Figure 4.13: Circumventing obstacles in a moderately crowded scene



(a) Initial state

(b) Final state

Figure 4.14: Transporting the target object through an originally narrow passage that became wider with time

Figure 4.15 shows another interesting phenomenon in a similarly crowded scene consisting of 50 obstacles. Initial positions of all the objects are identical to that in the last scene. The maximum speed of the primary trap is intentionally reduced by 3 times as compared to the value used in the previous two test cases. Naturally, much more time is required to transport the target object to the goal location. It can be clearly seen that the effect of Brownian motion is also more pronounced in this case, as the diffusion length is proportional to the square root of the elapsed time. Now, during the transport operation, the planner intentionally allows collision with an obstacle. Due to small size, the momentum of the colliding objects is relatively small; hence, they do not move away from each other after impact. Instead, they get displaced by such a little amount that both of them remain within the inner circle of the trapping probability region. Thus, both of them get dragged along by the primary trap towards the goal. Since no secondary trap is switched on, there is no reduction in the maximum trap speed. Based upon the current set of parameters, this turns to be the better option as compared to a circumvented route that involves no collision. Another collision event takes place with an obstacle that was located very close to the goal. We have not implemented switching the trap off, and then activating it after the objects have drifted away in the current setting to demonstrate the effect of collisions.

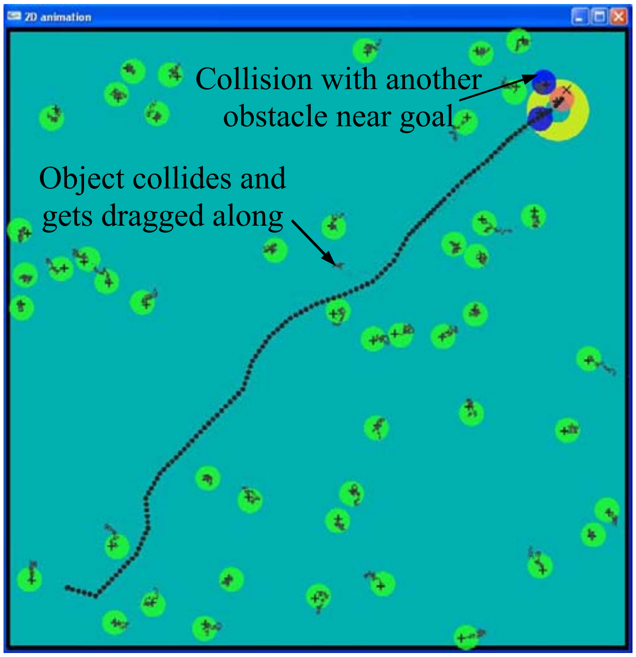

2D animation
Collision with another obstacle near goal
Object collides and gets dragged along

Figure 4.15: Collision and dragging of obstacle in a crowded scene where the relative rate of change in free *C*-space is quite high

The developed algorithm provides the option of changing the payoff function parameters so that an alternative non-collision path can also be computed if one wants to avoid the post-processing steps. This is shown in Figure 4.16 by increasing the value of t_{avg} to 150 ms. The initial state is same as that in the previous two test cases. A much longer, circuitous path is followed by the primary trap that not only avoids the first collision, but also goes around and above the goal before turning back to avoid trapping the obstacle lying close to the goal. This kind of flexibility will be very useful in practical applications.

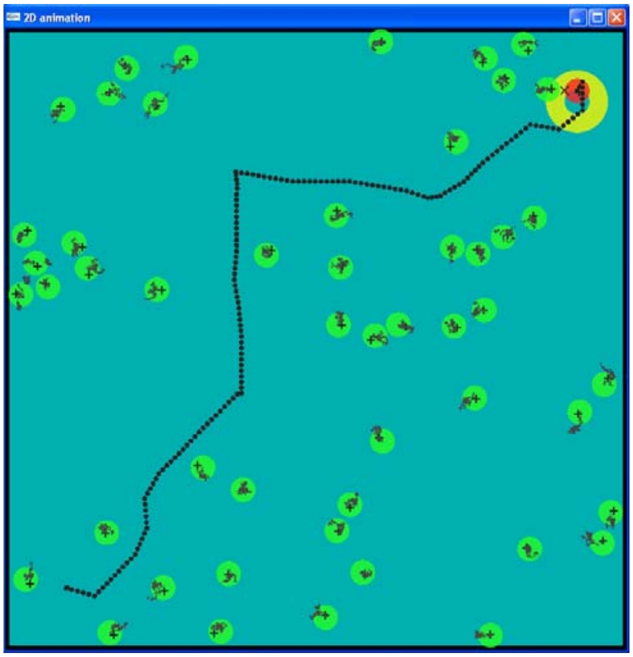


Figure 4.16: Avoiding collision and following a longer route in a crowded scene where the relative rate of change in free *C*-space is quite high

4.3 Experimental Validation

4.3.1 Optical Tweezer Instrument and Feedback Control

The optical tweezer used in these experiments was a BioRyx 200 (Arryx, Inc., Chicago, IL) holographic laser tweezer. The BioRyx 200 consists of a Nikon Eclipse TE 200 inverted microscope, a Spectra-Physics Nd-YAG laser (emitting green light of wavelength 532 nm), a spatial light modulator (SLM), and proprietary phase mask generation software running on a desktop PC. Nikon Plan Apo 60x/1.4 NA, DIC H oil-immersion objective is used. In order to impose a trajectory on the beads with a SLM, the optical trap is moved in a series of closely spaced steps aligned along the intended trajectory; the bead follows the steps of the trap quickly and then remains stationary until the next step. The stepping rate and step size determine the speed. The maximum rate at which traps can be set is the update rate of the SLM, 15 Hz, and the minimum step size is 150 nm.

The feedback control was achieved with a second PC equipped with a uEye camera (IDS, Inc., Cambridge, MA) for imaging the beads and software for executing the planning algorithm. A similar optical feedback approach in a system without stochastic fluctuations was implemented previously to control unstable crystal growth patterns [Pons07]. Beads were identified and located by thresholding the image and then calculating the center of mass of all the remaining blobs. For 2.01 μm diameter beads used in the physical planner experiments (see Section 4.3.4), this simple algorithm provided enough accuracy while maintaining a reasonable frame rate (1-2 frames/sec). Commands from the imaging/planning computer to the laser tweezer computer were sent via TCP/IP over an Ethernet connection between them.

4.3.2 Sphere Behavior without Optical Traps

The behavior of both 1.0 and 2.01 μm nominal diameter amorphous silica spheres has been observed in water at room temperature. The spheres have been procured from Bangs Laboratories, Inc., Chicago, IL. Refractive index of the spheres is 1.46 and density is 2000 kg/m^3. In the absence of any optical trap, they freely diffuse in water. Sequence of images has been taken for 1.0 μm diameter beads at five different cross-sectional planes which clearly show the effect of Brownian motion and the settling of objects at the bottom (cover slip glass plate) due to the effect of gravity. The camera frame rate is 3.68; dimension of each image is 512 x 640 pixels, where the length of each pixel is 100 nm.

Near the bottom plate, the well-focused beads (seen as bright white spots) are stuck to the plate and are more or less stationary. At planes well above the bottom plate, many previously bright spots become dark and vice versa as the beads drift in and out of the imaging planes. Estimation of the average in-plane diffusion length of ten beads each at the four imaging planes other than the bottom one between 10 successive frames corresponds well to the theoretically predicted value of 2.16 μm. The same experiment is repeated inside the simulator for the same number of silica spheres (with same material properties) at identical horizontal planes. The values lie within 5% of the experimental and theoretical numbers. This shows that Brownian motion has been accurately modeled in the simulator. Application of the image processing and feature recognition modules on the images also indicates that $0.25R_a$ is a good choice for *σ*.

The number of beads lying at the different imaging planes has been counted at the instants when the images have been taken. It is assumed that the first observation at the bottom plate starts at time $t = 0$ s. It must be noted here that the observations (or actual planning experiments) cannot be started at the very instant the dilute sample is created. Some time will be spent in placing it on the microscope objective, adjusting the z-position of the objective and so on. However, once we begin our observations, time at which all further readings have been taken can be easily recorded. Reading is continuously taken for about 16 s at every imaging plane; then the reading is taken at the next level and the same cycle is repeated once more for all the five planes.

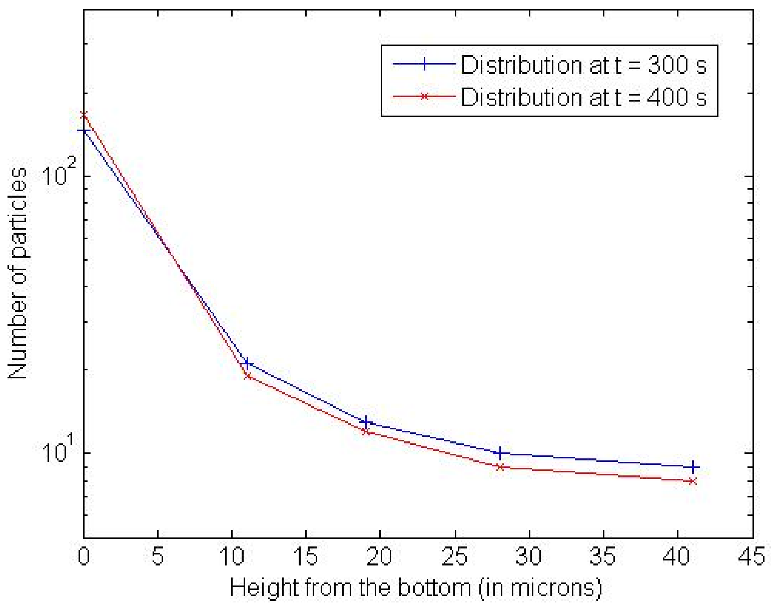


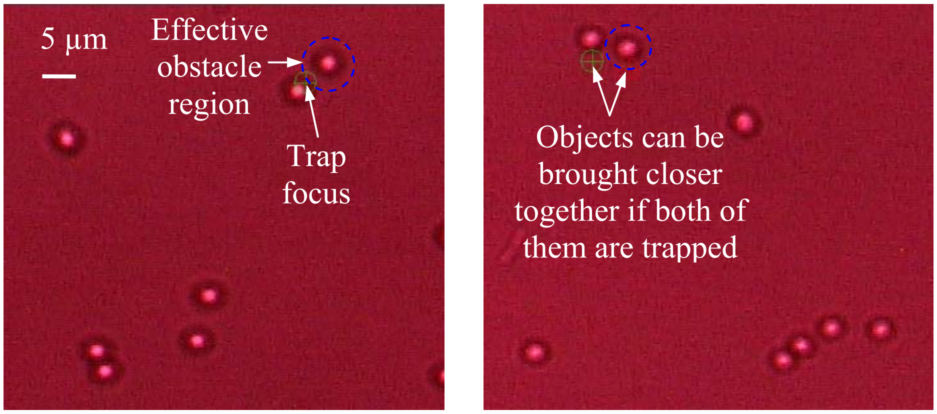
Figure 4.17: 1.0 μm diameter silica sphere number distribution as a function of depth from the bottom plate

Four different number values (two each at the beginning and ending of the first and second observations) obtained for every imaging plane are used to fit a linear curve. Values are extrapolated from the fitted curves to obtain the expected number of beads at that particular height (z-value) at a particular time instant. The expected logarithmic distribution of spheres at *t* = 300 s and 400 s are shown in Figure 4.17. Theoretically, the distribution should be linear. A progressive settling of beads can be seen at the bottom plate accompanied by a corresponding decrease in the number of beads at all the other planes. A similar experiment has been repeated inside the simulator by placing exactly the same number of spheres at different heights as observed for *t* = 300 s. XY positions of the spheres are assigned randomly. Simulation is allowed to run for 100 s and then the number of spheres is counted at different z-levels. Values again correspond within ±5% of those occurring in the physical set-up (as shown for *t* = 400 s). This shows that all the non-trapping forces, namely thermal, viscous drag, gravity, and buoyancy have been correctly accounted for in the simulation as the number distribution depends on all these forces simultaneously. Physically-obtained number distribution also provides a good starting point for all the simulation experiments. Though fluid flow can be observed experimentally under some conditions, it does not seem to play a role in the experiments shown here.

4.3.3 Demonstration of Effective Obstacle Region

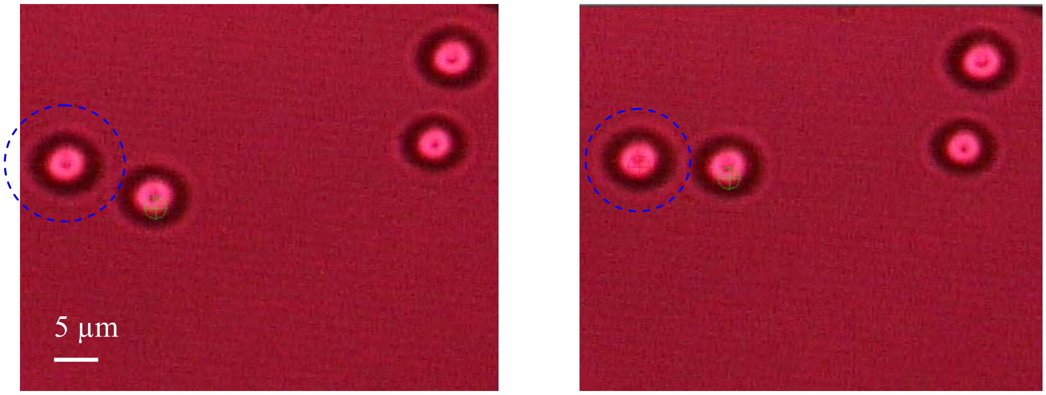
As mentioned earlier in Section 4.2, the trap radius (l_{tr}) and the trap width (l_{tw}) are functions of the object size, material properties, and laser power. Consequently, r_{sys} also depends on these physical parameters. Keeping the object material (silica) and overall laser power (0.4 W) fixed, dependence of r_{sys} on the object size and available laser power is shown in the following two figures.

Figure 4.18 shows the change in r_{sys} for 2.01 µm diameter spheres based on the number of traps that are activated. The size of the effective obstacle region is greater in Fig. 4.18(a), when only the target object is trapped, as compared to that in Fig. 4.18(b), when both the target object and the nearby obstacle are trapped. In the former case, it is approximately equal to 4.0 µm, whereas in the latter case, it is equal to 3.25 µm. These values are used to benchmark those obtained from the dynamics simulation so that accurate values are used in the simulation results presented in the following Section. This variation can be explained based on the fact that multiple trapping reduces the laser power available in each of the traps individually. As a result, due to the presence of the secondary trap, the obstacle can no longer get attracted into the primary trap, as long as the primary trap is not placed sufficiently close to it. Thus, it can be seen that switching secondary traps not only enables one to get rid of the uncertainty arising due to random Brownian motion during planning, but also increases the free *C*-space available for target object motion. Figure 4.19 shows a similar trend for the larger 4.74 µm nominal diameter silica spheres. It is useful to note here that the value of r_{sys} is much greater in this case. This happens because the optical trapping forces are much stronger for larger-sized spheres, till the forces start diminishing beyond a certain radius.


5 µm
Effective obstacle region
Trap focus
Objects can be brought closer together if both of them are trapped

(a) Only target object trapped (b) Both target object and obstacle trapped

Figure 4.18: Change in the effective obstacle region for 2.01 µm diameter silica sphere based on the number of set traps

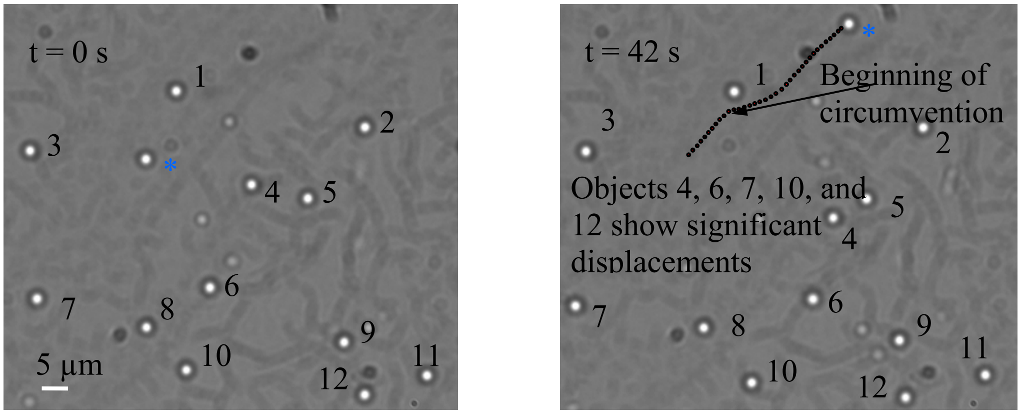


(a) Only target object trapped (b) Both target object and obstacle trapped

Figure 4.19: Change in the effective obstacle region for 4.74 µm nominal diameter silica sphere based on the number of set traps

4.3.4 Sphere Transportation using Automated Planner

2.01 μm diameter silica beads have been successfully transported using the current planner described in Section 4.2 automatically without any manual intervention. The dimension of the camera image is increased to 896 x 800 pixels. An initial and a goal location are specified in the controller user interface by manually clicking at two spots in the imaged workspace. The bead that is located nearest to the initial location is automatically selected as the target bead. It is then transported to the desired goal location by the POMDP planner; the planner is terminated when the target bead reaches within 1.0 μm radius of the second selected spot. Two such representative experimental runs near the bottom plate are shown in this paper to illustrate the capability of the system. The trap charge is always set to 0.0 to create a point trap.

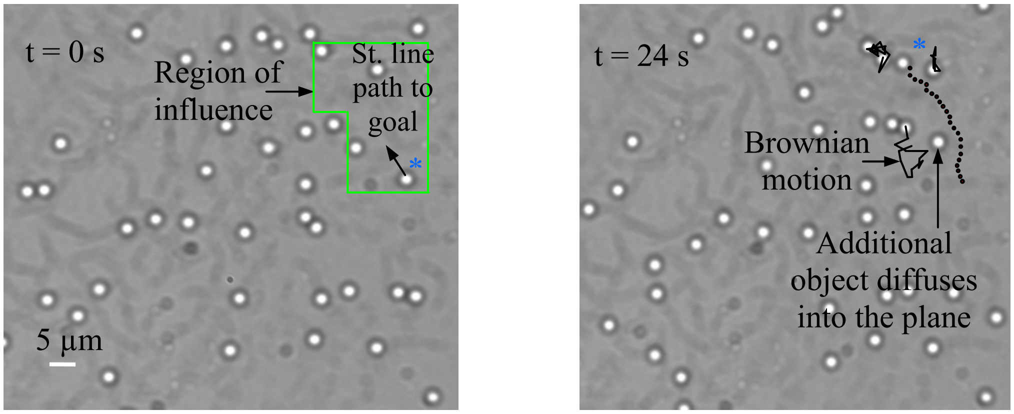

t = 0 s
1
2
3
*
4
5
6
7
8
9
10
11
12
5 μm
t = 42 s
1
*
Beginning of circumvention
3
2
Objects 4, 6, 7, 10, and 12 show significant displacements
5
4
6
7
8
9
11
10
12

(a) Initial scene

(b) Final scene

Figure 4.20: Obstacle circumvention by automated planner in holographic optical tweezer set-up

Figure 4.20 depicts the effective circumvention of an obstacle by the target object (represented by '*' sign by its side) moved by an optical trap at a speed of 2.23 μm/s. Laser power is set at 0.2 W. The time at which each frame is recorded is shown on the top left-hand corner. The overall trajectory of the target object is displayed using dots in Figure 4.20(b). A number is assigned to every object so that it can be easily tracked in the two figures. Figure 4.21 shows that even if a straight line path exists between the initial position of the target object and the goal, Brownian motion of nearby obstacles and diffusion of out-of-plane objects into the plane under consideration will cause the trap to follow a circuitous route. The random trajectory of the objects that influence the trap motion is also shown in Figure 4.21(b). In this experiment, the trap is moved at a faster speed of 4.45 µm/s.



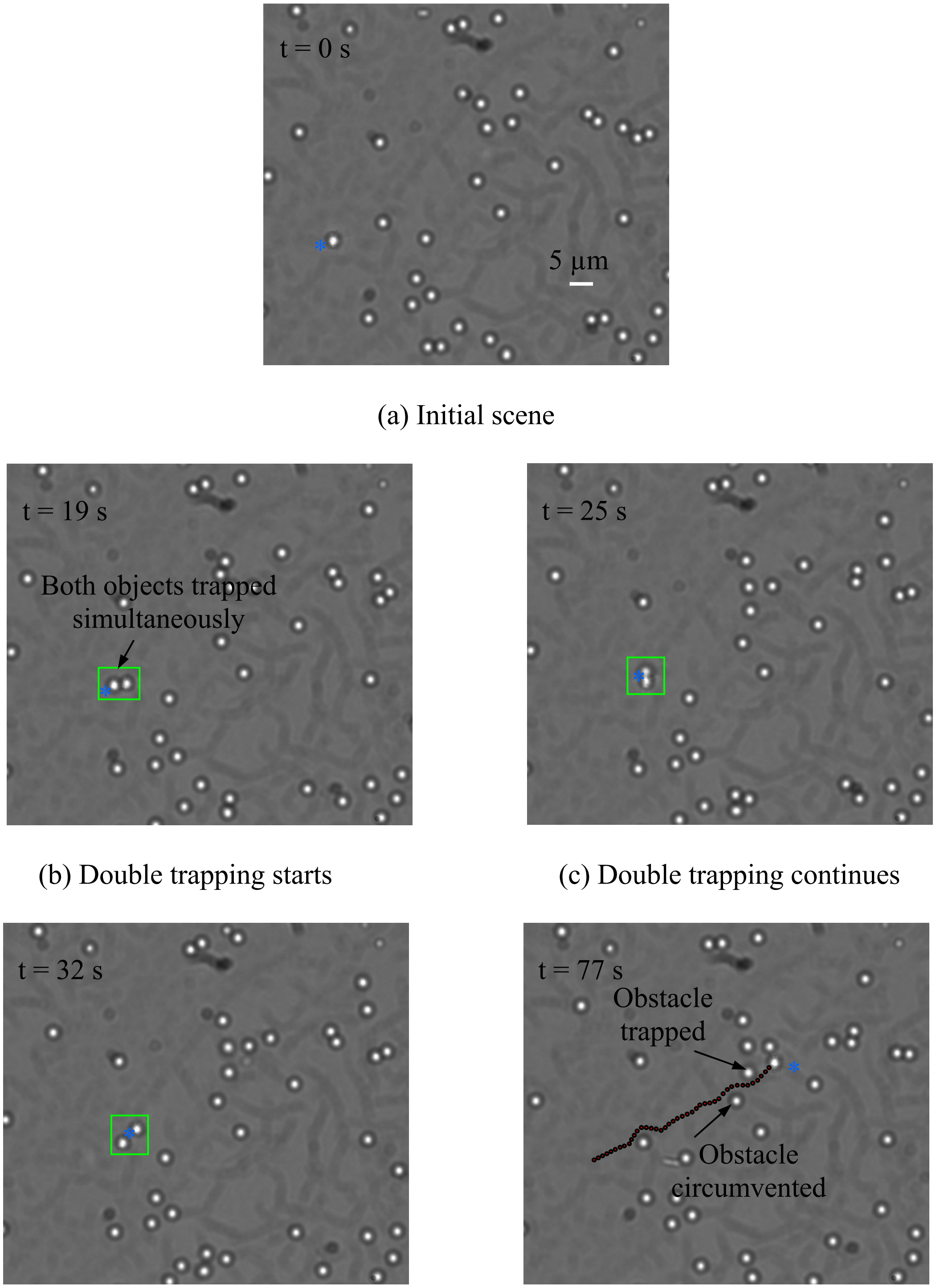
(a) Initial scene

(b) Final scene

Figure 4.21: Forced deviation of automated planner from linear trajectory due to obstacle diffusion

Figure 4.22 shows a combination of obstacle trapping and circumvention by the automated planner transporting the target object at a speed of 2.23 µm/s. Laser power is set at 0.2 W. The time at which each frame is recorded is shown on the top left-hand corner. The overall trajectory of the target object is displayed using dots in Figure 4.22(e).The planner traps two obstacles and circumvents another. Various stages of the first multiple (double) trapping event are shown in Figure 4.22(b)-4.22(d). A rectangular box has been drawn to highlight the two objects that are being trapped simultaneously. It can be seen that multiple trapping allows the target object to move much closer to the obstacle as compared to circumvention. Thus, by adjusting the payoff function parameters, if one type of collision avoidance strategy is made more likely to occur than the other, then the planner performance can be improved. For example, multiple trapping will be more useful in a crowded scene where the object positions are known with a reasonably low degree of uncertainty.

Collision and multiple trapping of beads are found to occur during transportation in crowded scenes after more beads have settled down. Intentionally, the trap is first placed at a slightly different position from the estimated bead center (see Figure 4.23(a)). As expected, the bead gets pulled in and then the trap starts moving at a speed of 2.23 μm/s. When obstacles are present in all possible routes to the goal, the planner decides to move along a relatively shorter path, allowing the possibility of collision with a nearby object. This results in simultaneous transport of multiple objects to the goal (see Figure 4.23(b)). As in the 3rd simulation test case (Figure 4.15), switching off the laser has not been incorporated due to the time it takes for the holographic tweezer to heat up and reach full intensity.



(a) Initial scene

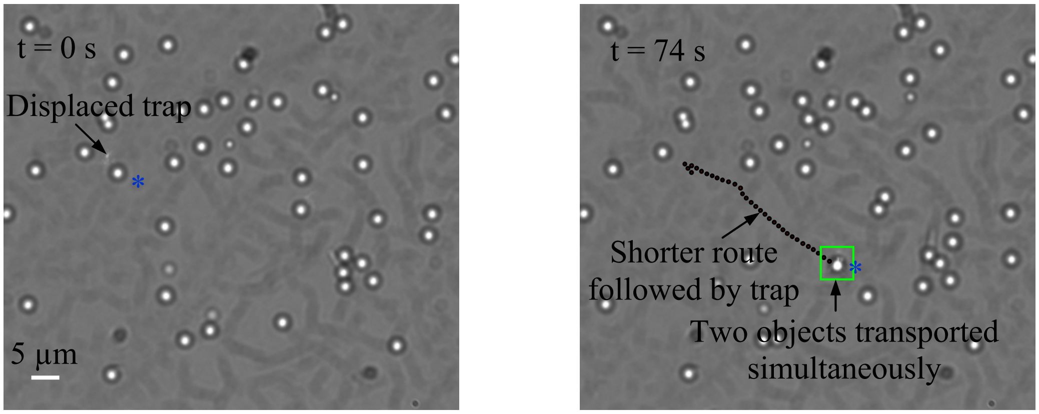
(b) Double trapping starts

(c) Double trapping continues

(d) Double trapping ends

(e) Final scene

Figure 4.22: Obstacle trapping as well as circumvention by automated planner


t = 0 s
Displaced trap
5 μm
t = 74 s
Shorter route followed by trap
Two objects transported simultaneously

(a) Initial scene

(b) Final scene

Figure 4.23: Automated transport involving collision to follow shorter route in a workspace having large number of obstacles

4.4 Simulation Results

Additional experiments have been carried out in the validated simulator to investigate the performance of the POMDP algorithm under a wide variety of operating conditions since it is not possible to perform them in the physical set-up due to the time needed to conduct a large number of runs to achieve statistically meaningful results. All the experiments are run on a Pentium 4 processor PC with 3.6 GHz clock speed and 1 GB of RAM. Every experiment is repeated 100 times and the average transport time is used as the performance measure. The workspace size is taken to be 89.6 μm x 80 μm x 50 μm (same as that used for planning experiments). $\Delta t_i, \Delta t_c$ and t_{avg} are always selected as 0.25 s, 0.5 s and 0.5 s respectively so that they are consistent with the values observed in the holographic tweezer set-up. A silica microsphere, having the same properties as used in the physical experiments, is initially positioned at (5 μm, 5 μm) in the XY plane and transported to (85 μm, 75 μm) in the same horizontal plane at room temperature in all the experiments. However, the z-location of the target object is varied in certain cases. Water is always used as the fluid medium. Laser wavelength and numerical aperture of the objective lens are also assigned constant values as specified in Section 3.1. Unlike in Section 4.2.3, time is not included as a parameter inside the convergence loop of the planning algorithm. This maintains consistency with the fact that the trap in the holographic tweezer stops once it has reached the intended location for a given control action command. However, the algorithm is general enough to deal with other kinds of set-ups where the traps continue moving as long as they are not commanded to act otherwise.

In the first two sets of experiments, the laser power and the imaging uncertainty standard deviation have been varied respectively, keeping all the other factors constant. Laser power and imaging uncertainty are chosen as the independent parameters because they are expected to change significantly from one set-up to the other. So simulation will enable one to identify the extent of effect on the transport time and correspondingly on the quality or effectiveness of the planning algorithm. In both the cases, initially 1.0 μm diameter objects are distributed in the workspace following the number distribution observed for t = 300 s in Figure 4.17. The first time any experiment is carried out, XY locations of the object centers are assigned following a uniform, random distribution. However, these values are stored and re-used in the other 99 trials. The target object is always placed at the z = 6 μm plane. This implies that the planner initially deals with 50 obstacles due to 2D imaging restrictions. Under such a situation, for 7200 grid points, an initial (non-pruned) primary control action set consisting of 7200 elements at every grid point, and 9 secondary traps that can be activated, average planner computation time is about 60 ms. However, with time the number of obstacles will change as objects diffuse in and out of the horizontal plane. Other parameters remaining constant (as expected in a typical physical set-up), the computation time varies almost linearly between (20-115) ms if the number of obstacles changes from 10 to 100. $v_{\max}$ is chosen for every data point by carrying out pilot simulation runs before starting the experiments.

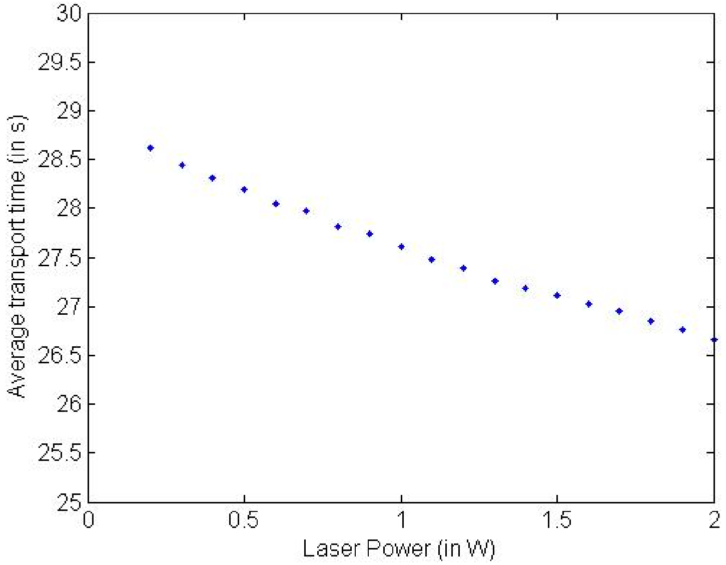


Figure 4.24: Transport time variation with respect to laser power

Figure 4.24 shows the variation in average transport time with respect to laser power. σ is chosen to be 0.125 µm. It can be seen that the time decreases only slightly as the power increases. This happens because $v_{\max}$ increases almost linearly with power; however, l_{tr} and l_{tw} also increase correspondingly meaning that less free *C*-space is available. Thus, although the primary trap can transport the target object faster, it has to circumvent more or activate more number of secondary traps in order to avoid collisions. Figure 4.25 shows the influence of imaging uncertainty on the average transport time. Laser power is kept constant at 0.2 W. As expected, the transport time increases as the imaging uncertainty becomes higher. For higher values of σ, since the sphere positions will be known with a much lesser degree of certainty, more collisions will take place, thereby increasing the transport time.

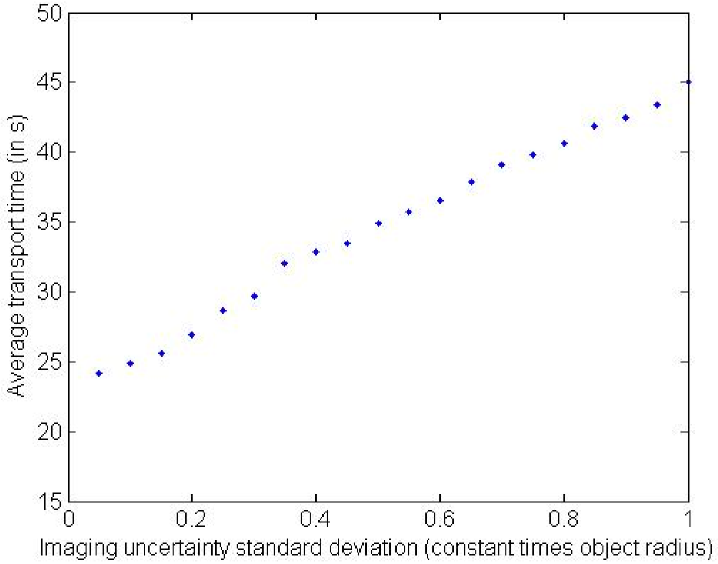


Figure 4.25: Transport time variation with respect to imaging uncertainty

The last experiment enumerates the extent of performance degradation as a result of ignoring the presence of spheres at other cross-sectional planes. Since they (particularly those lying relatively close by) may also get trapped by any activated laser beam, additional collisions will take place that have not been accounted by the planner. Consequently, the overall transport time will increase. As shown in Figure 4.26, this effect can only be seen near the bottom of the plate. This occurs due to the fact that many more objects are present near the bottom plate; hence, a larger number of objects will be undetected and consequently, greater number of unpredicted collisions will happen. The transport time more or less follows an exponential trend in accordance with the distribution of objects in the workspace.

The experiment is performed both for 1.0 and 2.01 μm diameter silica spheres, using laser power of 0.2 W and σ equal to 0.25 times the object radius. Initial distribution of the 2.01 μm diameter particles is taken to be the same as shown for t = 300 s in Figure 4.17. $v_{\max}$ is more for the 2.01 μm diameter spheres (7 μm/s) as compared to the 1.0 μm diameter spheres (4.5 μm/s). On the other hand, diffusivity is half for the larger sphere as compared to the smaller one. Together, these two factors counteract the effect of larger horizontal trapping regions for the bigger spheres, thereby resulting in smaller transport times. However, the overall variation trend is similar in both the cases.

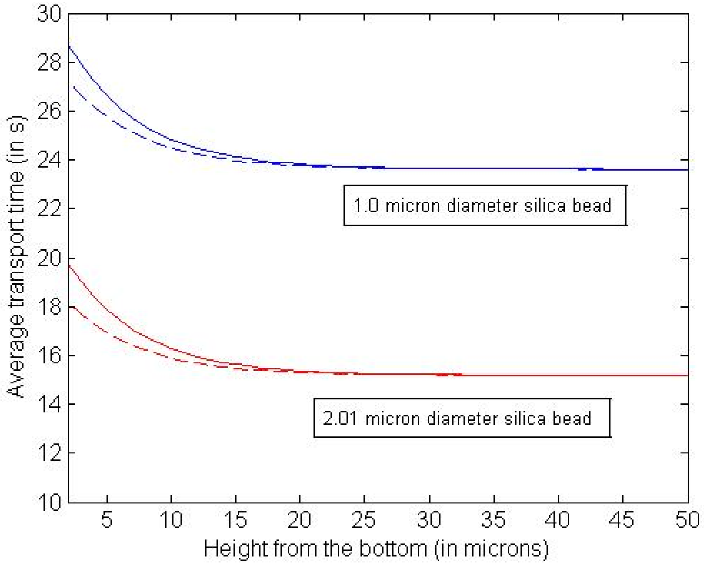


Figure 4.26: Planner performance comparison with (dashed lines) and without (solid lines) knowledge about the z-coordinates of the object centers

It should be noted here that the experiments have not been performed at z = 0 and 1 μm planes because of the fact that additional physical phenomena need to be modeled near the cover slip glass surface. Correction to the viscous drag term [Wrig94] (and correspondingly to the diffusivity terms which become anisotropic as well) needs to be made. Moreover, sliding friction and adhesion forces also have to be taken into account in order to transport objects that are stuck to the bottom plate. Another point that should be mentioned here is that in the simulation we have not increased the image processing time when z-locations of the object centers are also determined. But as shown in [Peng07a], when data from a stack of images taken at various z-planes are combined together to estimate the z-coordinate, computation time is significant. Thus, the planner may not be that effective as the scene for which the optimal plan is generated will have changed quite a lot by the time the plan is actually implemented. So the planner with 2D imaging may not perform significantly inferior as compared to the one with full 3D imaging.

4.5 Summary

Optical tweezers systems provide environments where obstacles exhibit random, Brownian motion and both forms of uncertainty (control action and measurement) are inherently present. Real-time operations also mandate that planning should be done repeatedly at short intervals of the order of milliseconds based on the latest information about the workspace.

This Chapter presents a stochastic dynamic programming framework to transport a single particle using OT from its current location to a given goal location in the minimum expected time. In order to utilize the stochastic dynamic programming framework, workspace state, control action set, value function, payoff function, and belief state are formally defined based upon the underlying physics of the problem. An existing, approximate infinite-horizon POMDP algorithm is modified to enhance computational speed and accuracy. Several conditions are presented to establish the rationale behind pruning the control action space. Discussion on convergence of the recursive Bellman equation and optimality of the MDP portion of the modified infinite-horizon algorithm show that the planning approach is always guaranteed to yield near-optimal solutions. Application of the infinite-horizon algorithm on several test cases clearly reveals its flexibility and effectiveness. Experiments conducted with silica beads provide data to validate the physical accuracy of the simulator. Moreover, successful runs show that the automated planner is capable of transporting a specific object by either circumventing or trapping freely diffusing obstacles. Finally, simulation experiments are carried out to characterize the performance of the planning algorithm under varying conditions of laser power, imaging uncertainty, and object number density.

Chapter 5

DEVELOPING A DECOUPLED AND PRIORITIZED APPROACH FOR MULTIPLE-PARTICLE TRANSPORT

A prioritized and decoupled stochastic dynamic programming framework is developed for transporting multiple particles simultaneously based upon the modified infinite-horizon algorithm presented in the preceding Chapter. Any symbol that is not defined in this Chapter has the same meaning as in the previous one. The notion of decoupled path planning has been taken from the multiple robot motion planning literature. As mentioned in Section 2.4, decoupled planning offers significant computational speed-up over a coupled architecture. Unlike in the latter case where planning is carried out in the composite configuration space of the movable objects, here the single particle transport algorithm is used sequentially to plan the paths of all the target objects individually. This is, however, preceded by a goal assignment step, where an iterative version of a maximum flow bipartite graph matching algorithm is used to assign goal locations to the target objects optimally so that the entire process of transporting all the target objects can be completed in the minimum, expected time. Once individual plans are computed, a three-step process is followed to assign an optimum set of priorities to the target objects for coordinating their motions to come up with collision-free paths. Just like in Chapter 4, both simulation and physical experimental results are presented to highlight the effectiveness of the developed planner. Most of the material covered in this Chapter is described in [Bane09c].

5.1 Problem Formulation

As mentioned in Chapter 4, a 3D rectangular parallelepiped workspace W is considered for transporting spherical objects. We have classified all the objects into two basic types- *trappable* and *non-trappable*. The trappable objects can be further classified into *transportable* and *non-transportable*.

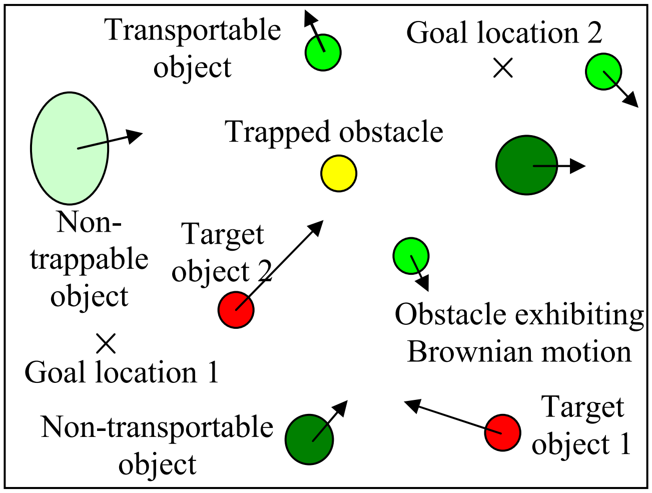


Figure 5.1: A typical workspace cross-section with two target objects and different types of obstacles

All the target objects belong to the transportable category, which consists of dielectric, colloidal, rigid particles of identical size and material properties. The non-transportable objects may not be rigid and can have different sizes and material properties, (e.g. density, refractive index). The non-trappable objects are just like the transportable objects, excepting the fact that they should not be trapped directly. Usually, they consist of biological objects such as cells, liposomes, and vesicles, which require indirect manipulation. If any of the objects is not perfectly spherical (cells are often oval shaped), then it is approximated using the bounding sphere. The diameter of all the objects is assumed to vary between (1-20) µm.

All the target objects are assumed to lie in the same horizontal plane and they will be transported to different goal locations in the identical horizontal plane. Figure 5.1 shows the horizontal cross-section of a typical workspace having multiple target objects (moving towards particular goal locations) and different types of obstacles. One such transportable obstacle is trapped to avoid collisions, whereas all the others are diffusing freely. The physics of the problem is modeled as in the previous Chapter (Section 4.1). It may be mentioned here that the optical trapping force values for various axially and transversally displaced locations of sphere centers with respect to the trap focus for certain spheres of fixed sizes are stored in the form of tables. For any sphere of intermediate size, quadratic interpolation is used. If multiple traps are switched on, then the overall laser intensity (power) is shared among all the traps and consequently the maximum trap speed ($v_{\max}$) is reduced proportionately. However, this effect can be compensated by increasing the power up to the maximum attainable value (2.0 W in the holographic tweezer set-up).

The radius of *effective obstacle region* (centered at the current, estimated obstacle position) arising due to the presence of optical trap, finite size, and imaging uncertainty is then given by: $r_{Sys} = \max\left\{\left(\left(l_{tr} + l_{tw}\right) + 3\sigma\right), \left(R_a + R_a^t + 3\sigma\right)\right\}$, where R_a^t is the radius of the target objects. Although true and estimated workspace state, goal set, control action set, and objective function have been mathematically defined in Chapter 4, the definitions need to be slightly modified here to account for multiple target objects, goal locations, and presence of different types of objects. The revised definitions are listed as follows:

Definitions

True workspace state is the set $X_t = \left\{ \left\{ X_1^{t_T}, \ldots, X_{n_T}^{t_T} \right\}, \left\{ X_1^{t_{NT}}, \ldots, X_{n_{NT}}^{t_{NT}} \right\}, \left\{ X_1^{t_{NP}}, \ldots, X_{n_{NP}}^{t_{NP}} \right\} \right\}$ where $X_i^{t_T}$ is the actual coordinate $\left(x_i^{t_T}, y_i^{t_T}, z_i^{t_T} \right)$ of the center of the i^{th} transportable object, $X_i^{t_{NT}}$ is the actual coordinate of the center of the i^{th} non-transportable object, and $X_i^{t_{NP}}$ is the actual coordinate of the center of the i^{th} non-trappable object in *W*. Without any loss of generality, the first *m* objects in the transportable set can always be considered as the target objects. Let $n = n_T + n_{NT} + n_{NP}$ be the total number of objects in *W*.

Estimated workspace state is the set $X_e = \left\{ \left\{ X_1^{e_T}, \ldots, X_{n_T}^{e_T} \right\}, \left\{ X_1^{e_{NT}}, \ldots, X_{n_{NT}}^{e_{NT}} \right\}, \left\{ X_1^{e_{NP}}, \ldots, X_{n_{NP}}^{e_{NP}} \right\} \right\}$ where $X_i^{e_T}$ is the estimated 2D coordinate $\left(x_i^{e_T}, y_i^{e_T} \right)$ of the center of the i^{th} transportable object, $X_i^{e_{NT}}$ is the estimated 2D coordinate of the center of the i^{th} non-transportable object, and $X_i^{e_{NP}}$ is the estimated 2D coordinate of the center of the i^{th} non-trappable object in *W*. Currently, we are not estimating the z-locations of the object centers. The planner makes all decisions based on X_e and not X_t. It is useful to point out here that all the objects may not be detected at all the times due to limitations in the imaging hardware and feature recognition modules.

Goal state is the set $G = \left\{ G_1, \ldots G_m \right\}$ where G_i represents the coordinate $\left(x_i^g, y_i^g, z_i^g \right)$ of the i^{th} goal location. This set is defined by the user and is known to the planner exactly. Every target object is assigned a unique goal location to which it needs to be transported. As mentioned earlier, $z_i^g = z_j^g, \forall i, j$ and is also equal to the common z-coordinate of the target object centers.

Control action set

- **Primary** – A set of m 3-tuples $\left\{\left\{s_1,\left(f_x^1,f_y^1\right),\left(v_x^1,v_y^1\right)\right\},\ldots,\left\{s_m,\left(f_x^m,f_y^m\right),\left(v_x^m,v_y^m\right)\right\}\right\}$, where s_i is a binary integer corresponding to the state of the ith *primary trap* (1 if on and 0 if off), $\left(f_x^i,f_y^i\right)$ represents the coordinate of the trap center (beam focus) and $\left(v_x^i,v_y^i\right)$ is the uniform velocity vector assigned to the trap. The z-coordinate of the trap center is always set to the common z-coordinate of the goal locations and the target object centers.
- **Secondary** - A set of (20-*m*) doubles $\left\{\left\{s_1,\left(f_x^1,f_y^1\right)\right\},\ldots,\left\{s_{20-m},\left(f_x^{20-m},f_y^{20-m}\right)\right\}\right\}$ such that the 1st element of every double is a binary integer that denotes whether the corresponding *secondary trap* is switched on and $\left(f_x^i,f_y^i\right)$ represents the coordinate of the *i*th trap center. As in the case of the primary trap, the z-coordinate of the trap center is always set to the common z-coordinate of the centers of the target objects. The choice of the maximum number of secondary traps is governed by the available laser power in the holographic tweezer set-up used for validating the automated planner.

Objective function

Estimated, expected transport time T_e^i for the ith target object moving to the jth goal location ($1 \le i, j \le m$), a given estimated workspace state X_e, a control action *A,* and a circular circumvention strategy is defined as follows:

$$T_e^i\left(X_e, A, G_j\right) = \min\left\{\begin{array}{l}\left(\Delta t_c + \dfrac{\overline{F'G_j}}{v_{\max}} + \sum_{k=1}^{n_c} p_k t_{avg}\right), \left(\Delta t_c + \dfrac{\overline{F'G_j}}{v_{\max}} + \dfrac{\pi n_c r_{sys}}{v_{\max}}\right), \\ \left(\Delta t_c + \left(\dfrac{\overline{F'G_j}}{v_{\max}}\right)\left(1 + \dfrac{w n_c \Delta t_c}{t_e}\right) + (1-w)\sum_{k=1}^{n_c} p_k t_{avg}\right)\end{array}\right\}$$

Estimated, expected total transport time for a given estimated workspace state X_e, goal state *G*, and a circular circumvention strategy is defined as $T_e = \max\left\{T_e^1, \ldots, T_e^m\right\}$. This follows from the fact that a transport operation is completed only when *all* the target objects have reached their respective goal locations. For practical purposes, it is assumed that the *i*th target object *reaches* the *j*th goal location when $\left\|X_i^{e_T} - G_j\right\| \leq \delta$, where δ is usually selected as R_a^t.

5.2 Path Planning Approach

5.2.1 Goal Assignment

Unlike in the case of single particle transport where a given target object is transported to the only specified goal location, selection of goals poses an interesting challenge in the case of multi-particle transport. Assignment of goals to target objects can be done either by the user or by the planner automatically. The former scenario is applicable in case of formation of directed assemblies, repair of devices, delivery of drugs via liposomes etc. In all such cases, specific target objects need to be transported to particular goal locations and positioning any other target object at that location may not be useful from the point of view of functional requirements. However, in the current work this possibility has not been considered as all target objects are assumed to be dielectric and colloidal having identical size and material properties. It may be mentioned here that although such applications will require extensions to our problem formulation, implementing user-selected assignment is a simple task. Instead, the focus is on developing an iterative optimal strategy for automated goal assignment. This is very useful in the kinds of biological applications mentioned in the introductory Chapter.

For every goal location, compute the estimated, expected transport time for every target object present in the workspace *W* based on the *initial* estimated workspace state using the expression given in Section 5.1. Let T_e^{ij} be the value for the i^{th} target object and j^{th} goal location. All other target objects are treated as obstacles that should not be trapped and kept stationary by activating any secondary trap. It is assumed that the target object moves along a straight line path from its current location to the goal. Furthermore, it is assumed that all the other goals are occupied by some target object. This ensures that in the worst-case scenario, if all other target objects have reached their goal locations while the one under consideration is still moving towards its corresponding goal, then collision can be avoided if necessary. This condition is modified in the subsequent iterations. All the target objects are then sorted in order of non-decreasing estimated, expected transport time. The overall process has a time complexity of $O\left(m^2 \log m\right)$, where *m* is the number of target objects.

Once all the T_e^{ij} have been computed, every element in the matrix table is traversed. If the corresponding value is greater than the current optimum (initially set to a very high positive number), then the algorithm moves over to the next element. Otherwise, a *feasible* assignment A_f of target objects to goal locations is determined if one exists; the current optimum is also set to the corresponding value and the current best assignment is set to A_f. Computation of A_f is explained in details later in this sub-section. The rationale behind this technique is that the optimum value of the objective function must be equal to one of the computed T_e^{ij} values (from basic definition). So if all the m^2 entries are individually explored and the feasible arrangement can be computed in polynomial time, then an optimum solution can be obtained in polynomial time as well. This observation makes the optimization problem computationally tractable.

After all the elements have been traversed, the T_e^{ij} values are updated based on the obtained feasible assignments. This means while computing the transport time value for a given pair of target object and goal location, all other goal locations may not necessarily be occupied by the corresponding target objects specified by the feasible assignment for the given pair. The goals will be occupied only when the estimated, expected transport times are greater than the time taken by the target object to move past those goals during transportation. In most cases it is observed that this will not change the transport times as majority of the goal locations may not lie anywhere close to the path of the target object and, hence, did not cause any effect in the first place. However, few values may be reduced. The optimum assignment generation procedure (by first sorting, then traversing and obtaining individual feasible assignments) is repeated until the same assignment is found in two successive runs. Convergence is ensured by the fact that the optimum transport time can never increase from one iteration to the other.

Computing feasible assignment for a particular pair of target object and goal location is the most important step in the entire goal assignment algorithm. The following approach has been developed for this purpose. First of all, any other assignment of that particular target object and goal location is discarded. The 1st entry among all the remaining ones is searched for every goal location. If the minimum value is greater than the corresponding expected, estimated transport time, then we conclude that no feasible arrangement exists and terminate the step. The last remaining entry is then searched for every goal location. If the maximum value is greater than or equal to the corresponding transport time, then it is concluded that the current assignment cannot lead us to the optimum solution. So the step is terminated and the algorithm moves on to the next element in the matrix table. Otherwise, all the *k* entries that are less than or equal to the corresponding transport time are retained. Clearly, it can be said that $m-1 \le k \le \left(m-1\right)^2$. If at least one entry is not present for every goal location, then also the step is terminated. Otherwise, a *bipartite graph* is created where the left-side nodes are the goal locations and the right-side ones are the target objects. Edges are created between nodes if the corresponding matrix table entry is retained. *Ford-Fulkerson maximum flow algorithm* [Corm97] is applied to compute the maximum matching *M* for the bipartite graph. If the cardinality of *M* is equal to (m – 1), then *M* yields the desired feasible arrangement. Run-time complexity of this maximum flow algorithm is $O\left(mk\right)$ [Corm97]. Although the Hopcroft and Karp algorithm runs in $O\left(\sqrt{m}k\right)$time [Corm97], Ford-Fulkerson method is adopted due to the relative ease of implementation.

5.2.2 Priority Assignment

Once goals have been assigned to all the target objects, the modified infinite-horizon algorithm is used sequentially to compute the optimum plan for every target object individually. Just as in the case of goal assignment, other target objects are treated as obstacles that should not be trapped by secondary traps. The remaining task is to resolve conflicts in the target object motions so that the total transport time is minimized in the expected sense. A three-step process, consisting of clustering, classification, and branch and bound optimization is adopted to assign priorities to the target objects. This priority list determines the sequence in which the final plans are computed for the target objects. Thus, if a target object is present lower down in the priority list, then its corresponding motion plan may have to be modified to avoid collisions with all the other target objects that occur higher up in the list. Each of the three steps is now presented in details.

Step 1: Clustering

This step involves clustering all the target objects into multiple clusters such that the priorities can be assigned *separately* in every cluster. This decomposes the overall priority assignment problem into several simpler sub-problems so that the assignment technique can be applied completely independently to each of the clusters. Before formally defining a cluster set, it is useful to introduce the notion of collision circle and effective collision circle corresponding to the primary control actions of the target objects.

Based upon the discussion in the previous sections, first the *set of feasible primary control actions* can be defined for the i^{th} target object $A_p^i = \left\{a_1^i, \ldots, a_l^i\right\}$, where $l = 2\pi / \Delta\theta$ is the cardinality of the set. l remains same for all the target objects. Here. $\Delta\theta$ is the angular resolution of the planning approach (selected as 5° as in Section 4.2). Again, $a_j^i = \left\{\left(f_x^i, f_y^i\right), \left(v_x^{ij}, v_y^{ij}\right)\right\}$, where $\sqrt{v_x^{ij2} + v_y^{ij2}} \le v_{\max}$, $\theta_j^i = \tan^{-1}\left(v_y^{ij} / v_x^{ij}\right)$, and

$\theta_{j+1}^{i} - \theta_{j}^{i} = \Delta\theta$ for $j = 1,\ldots,l-1$. $\left(v_x^{ij}, v_y^{ij}\right)$ represents the velocity vector that can be assigned to the primary trap transporting the i^{th} target object that makes an angle of $j\Delta\theta$ with the X-axis. As stated in Section 4.2, $\left(f_x^i, f_y^i\right)$coincides with $\left(x_i^{e_T}, y_i^{e_T}\right)$.

Collision circle corresponding to feasible primary control action a_j^i for the i^{th} target object is defined as a circle centered at the tip of the corresponding velocity vector $\left(f_x^i \hat{i} + f_y^i \hat{j}\right) + \left(v_x^{ij} \hat{i} + v_y^{ij} \hat{j}\right)\Delta t_c$ with radius $r_c^i = \max\left\{\left(R_a^t + 3\sigma\right), \left(l_{tr}^i + l_{tw}^i + 3\sigma\right)\right\}$, where l_{tr}^i and l_{tw}^i are the trap radius and trap width for that particular target object. Thus, the collision circle radius varies for different feasible primary control actions.

Effective collision circle for the i^{th} target object is defined as a circle whose center lies at $\left(x_i^{e_T}, y_i^{e_T}\right)$ with radius $r_{eff}^i = \max\left\{\left(R_a^t + v_{\max}\Delta t_c + 3\sigma\right), \left(l_{tr}^i + l_{tw}^i + v_{\max}\Delta t_c + 3\sigma\right)\right\}$. In other words, the effective collision circle occupies the maximum area among all the possible collisions circles for that particular target object.

Cluster set $C = \left\{c_1,\ldots,c_o\right\}$ is defined such that every individual cluster consists of a set of target objects where the following condition is satisfied. For every target object belonging to the cluster, there exist at least one target object in the same cluster such that the collision circles of one or more feasible primary control actions corresponding to the 1st target object intersect with the collision circles of one or more feasible primary control actions corresponding to the 2nd target object. Furthermore, $c_i \cap c_j = \phi$ for $1 \le i, j \le o, i \ne j$ and $\bigcup_{i=1}^{o} c_i$ spans all the target objects present in the workspace *W*.

Now a naïve way of forming clusters is to check for intersection of every collision circle of each target object with every collision circle of all the other target objects present in the system. This is computationally expensive and has a time-complexity of $O\left(m^2 l^2\right)$. That is why, the notion of effective collision circle has been developed to make this computation more efficient. For every target object, it is checked whether its effective collision circle intersects with the effective collision circle of all the other target objects one by one. If any pair-wise collision occurs, then both the target objects are inserted in the same cluster (if one or both of them are not already present in that cluster). A list of all the feasible primary control action velocity vectors that lie within the intersection region is also maintained in the so-called *forbidden list*. It is formally defined as the set $F = \left\{\left(f_1, f_2\right), \ldots, \left(f_1, f_m\right), \left(f_2, f_3\right), \ldots, \left(f_{m-1}, f_m\right)\right\}$, where $\left(f_i, f_j\right)$ contains the list of all the velocity vectors that lie within the intersection region for the i^{th} and the j^{th} target objects. $\left(f_i, f_j\right)$ will be an empty list if these two target objects do not belong to the same cluster. Thus, this conservative scheme has a time-complexity of $O\left(m^2\right)$ and works reasonably well because every target object will try to move the maximum possible speed in all the directions, unless it can specifically avoid colliding with the nearest obstacle by slowing down. An alternative method of computing polygonal intersection formed by the velocity vectors is not considered due to the relative difficulty of robust implementation and greater time-complexity of $O\left(m^2 \left(l \log l + k \log l\right)\right)$, where k is the output polygon complexity [Berg00]. Formation of two clusters, comprising of three and two target objects respectively, is shown in Figure 5.2. For the sake of representation clarity, the feasible primary control action velocity vectors that need to be inserted in the forbidden list are depicted for only one of the target objects. Exactly the same scheme that is used in Figure 5.1 is also applied here to represent the goal locations, target objects, and obstacles present in the scene.

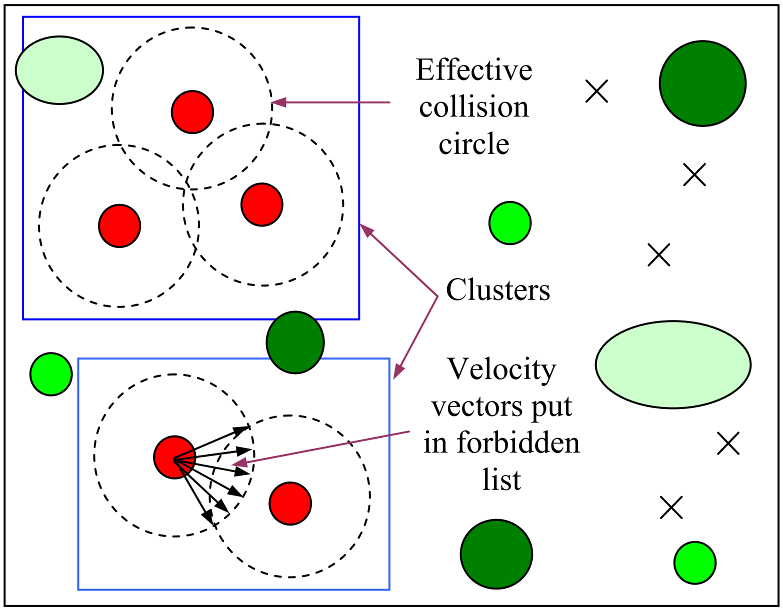

Effective collision circle
Clusters
Velocity vectors put in forbidden list

Figure 5.2: Clusters formed in a typical multi-particle transport operation

Step 2: Classification

The objective of this step is to classify every obtained cluster into two exhaustive and mutually exclusive sets such that any random priority can be assigned to the target objects present in one of them and an optimization technique can be employed on all the target objects remaining in the other set to compute collision-free paths. Let $c_k^{/}$ and $c_k^{//}$ represent the above-mentioned sets corresponding to the k^{th} cluster obtained in the previous step. An easily provable result has been developed to perform this classification process. This result can be mathematically stated as follows.

Let Q_i^j be the j^{th} expected Q function value (obtained as an output of the infinite-horizon algorithm presented in Chapter 4) of the i^{th} target object. Q_i^j is a function of a_j^i

and the estimated workspace state X_e and is computed by the algorithm by taking the expected value of the product of the converged values of the value function and the state transition probability added to the payoff function over the belief state. The payoff function is again analogous to the objective function defined in section 4.1. This implies that the individual expected, estimated transport time T_e^i for a given a_j^i directly correlates to Q_i^j.

Clearly, $Q_i^{\min} \leq Q_i^j \leq Q_i^{\max}$, and in the absence of any form of coordination among multiple target objects, optimum expected Q function value $Q_i^* = Q_i^{\min}$. The optimum solution for the overall problem is based on the *bottleneck Q* function value for all the target objects present in the classified cluster set $c_k^{//}$. It is given by $Q_{k,\max}^* = \max\left\{Q_p^*\right\}$ where the p^{th} target object is present in $c_k^{//}$. Now, if $Q_{k,\max}^* \geq Q_q^{\max}$ for every target object q present in the set $c_k^{/}$, then any random priority can be assigned in $c_k^{/}$ without degrading the estimated, expected total transport time T_e. This follows from the fact that if due to collision avoidance constraints, any feasible primary control action velocity vector corresponding to an expected Q function value other than the optimum one is chosen for any of the target objects present in $c_k^{/}$, then the corresponding T_e^q value increases. However, since the maximum T_e^q value is still less than the bottleneck value (i.e., value that constrains the optimum solution for the overall problem) after assigning priorities to the target objects present in $c_k^{//}$, T_e remains unchanged.

This provides a simple and effective way of reducing the overall computational burden by sub-dividing every cluster so that the optimization technique only needs to be carried out for a potentially fewer number of target objects. The random assignment of priorities in the first set of every cluster is done based on the original numerical indexing of the constituent target objects. In other words, if say for example, 2nd, 3rd and 7th target objects are present in this set of the 1st cluster, then the 2nd target object has the maximum priority (plan computed first), followed by the 3rd and the 7th target object respectively. The best (optimum) feasible primary control action is then going to be selected for the 2nd target object. All the feasible primary control actions whose velocity vectors lie in the set $\left(f_2, f_3\right)$ of the forbidden list will be ignored and the remaining best one will be selected for the 3rd target object. Similarly, all actions whose velocity vectors lie in either of the two sets $\left(f_2, f_7\right)$ or $\left(f_3, f_7\right)$ will be ignored for the 7th target object and the best, remaining one will be chosen. Assignment of priorities to the second classified set of every cluster is explained in the next step.

Step 3: Branch and bound optimization

Depth-first branch and bound algorithm is used to determine the optimal set of path planning priorities for the target objects present in the second set of every cluster. An empty root node is first created and all the target objects (elements) in that particular set are inserted as children nodes in order of non-increasing expected Q function values. Highest priority is assigned to the leftmost element of the 2nd level. The feasible primary control action velocity vectors of all the remaining target objects that intersect with the effective collision circle of the already selected target object are temporarily removed from consideration and the set of expected Q function values for the remaining target objects are updated accordingly. This is readily done by searching the forbidden list. All the remaining target objects are then inserted as children nodes of the already selected element in the same sorted order of non-increasing *Q* function values. The leftmost element is next added to the priority list and this process is repeated until all the target objects have been accounted for.

Attempt is made to refine the solution (i.e. obtain a smaller value than the current best) by incrementally constructing and traversing the tree in a depth-wise manner. The algorithm is terminated either when an user-specified maximum search time limit is exceeded or when all possible nodes have been created and explored. The algorithm may also terminate if the global optimum (best possible expected *Q* function available initially) is reached along a particular path from the root to a leaf node. Any path where the minimum expected *Q* function value for the non-added target objects exceeds the current optimum is pruned. This technique works well in practice due to the relatively small number of target objects that are usually present in the second set of any cluster.

The overall path planning approach is summarized in the form of a flowchart shown in Figure 5.3.

5.3 Experimental Validation

The holographic optical tweezer set-up described in Section 4.3 is used for demonstrating the feasibility of multi-particle transport. Due to memory problems while setting a reasonably large number of traps during continuous operation in the currently installed version of the Arryx system software, the planner cannot be run in an automatic mode. So all the experimental data are generated by directly positioning and controlling the traps in the BioRyx user interface instead of invoking the low-level system controller functions using the automated planner. The control actions are based on those generated by the planner in the simulated environment described in the next Section.

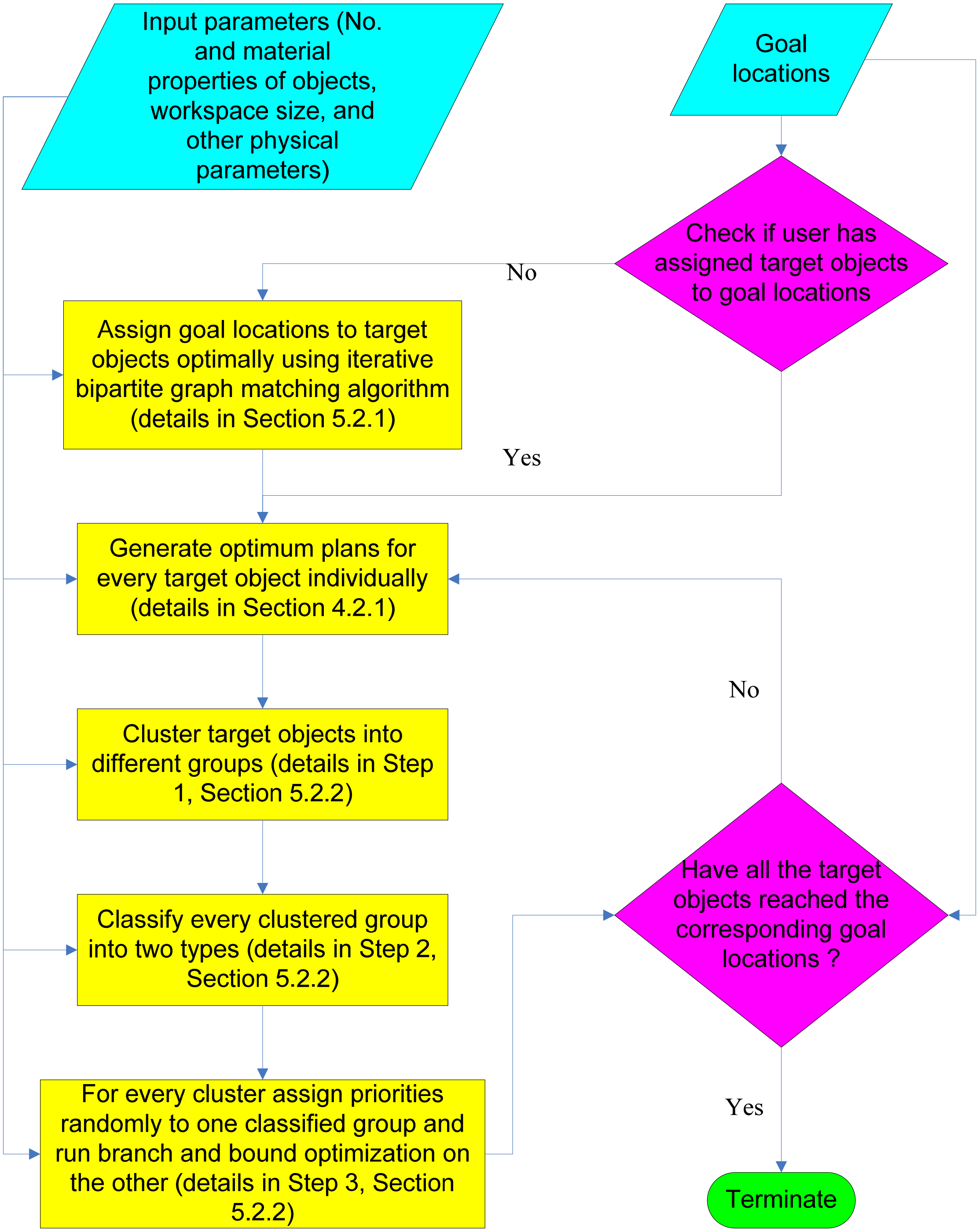
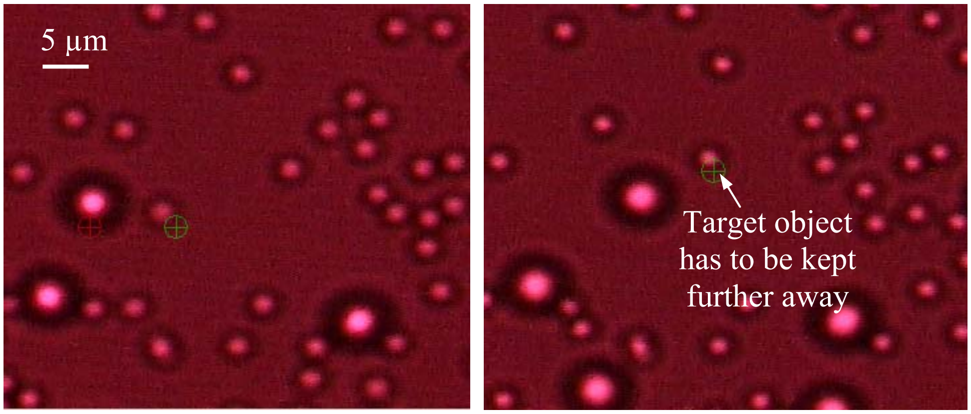

Input parameters (No. and material properties of objects, workspace size, and other physical parameters)
Goal locations
Check if user has assigned target objects to goal locations
No
Assign goal locations to target objects optimally using iterative bipartite graph matching algorithm (details in Section 5.2.1)
Yes
Generate optimum plans for every target object individually (details in Section 4.2.1)
No
Cluster target objects into different groups (details in Step 1, Section 5.2.2)
Have all the target objects reached the corresponding goal locations ?
Classify every clustered group into two types (details in Step 2, Section 5.2.2)
Yes
For every cluster assign priorities randomly to one classified group and run branch and bound optimization on the other (details in Step 3, Section 5.2.2)
Terminate

Figure 5.3: Multiple particle path planning approach flowchart

Figure 5.4 is analogous to the situations shown in Figures 4.18 and 4.19 which depict the change in the effective obstacle region radius due to the difference in the number of activated traps. However, in this particular case, the workspace contains both 2.01 and 4.74 μm diameter silica spheres. As in Figures 4.18 and 4.19, activating an additional secondary trap for the 4.74 μm diameter obstacle enables the smaller target object to come closer to the obstacle without resulting in any collision or multiple trapping. When both objects are trapped, the target object can be safely brought within 10 μm of the obstacle, whereas it needs to maintain a separation of 12 μm when the obstacle is diffusing freely. Laser power is set at 0.4 W in this experiment.

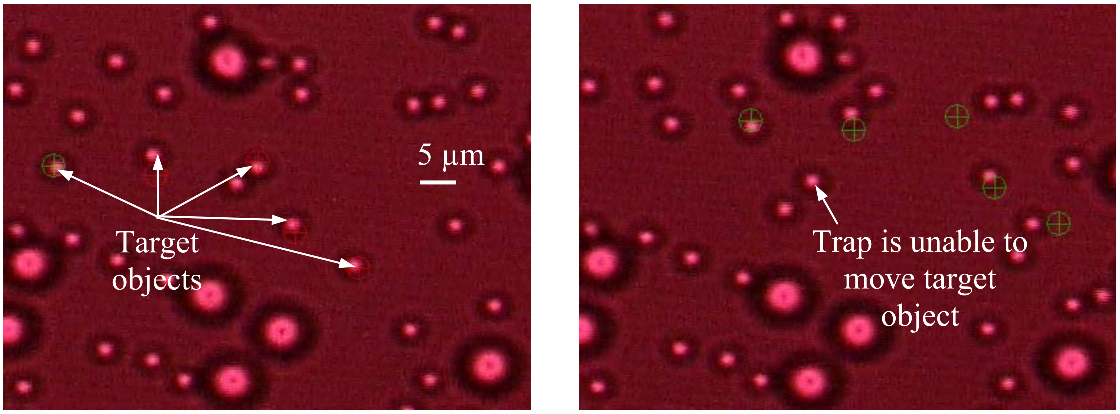


(a) Both target object and obstacle trapped (b) Only target object trapped

Figure 5.4: Change in safe distance to avoid trapping depending upon the number of set traps in workspace containing both 2.01 μm and 4.74 μm diameter silica spheres

Figures 5.5 and 5.6 show transport of multiple 2.01 μm diameter silica spheres simultaneously at 0.8 W and 0.6 W laser power respectively by grouping them together and moving the group as a whole with one single velocity. In Figure 5.5, all the five target objects are grouped together, whereas, the leftmost target object is not included in the group in Figure 5.6. The fact that certain target objects need not move in a particular planning step is often useful in preventing collisions among the target objects themselves.

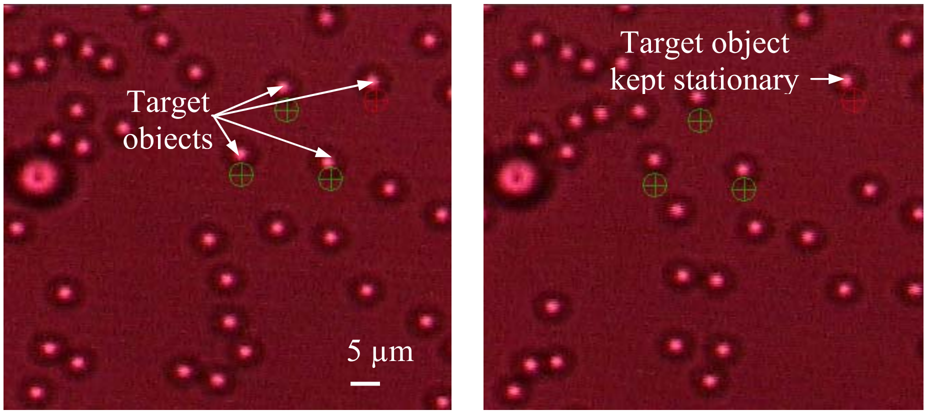
However, individual control of the traps is not possible in the existing user interface provided by the manufacturers. This results in the inability to move the middlemost target object by the trap in Figure 5.5. This demonstrates the need for automated planning that can adjust the individual trap speeds, directions, as well as carry out adaptive replanning to try and re-trap displaced objects.



(a) Initial scene

(b) Final scene

Figure 5.5: Transport of multiple 2.01 μm diameter spheres simultaneously



(a) Initial scene

(b) Final scene

Figure 5.6: Transport of multiple 2.01 μm diameter spheres with one kept stationary

5.4 Simulation Results

5.4.1 Simulated Trajectories

Two test cases have been presented to visualize sample trajectories generated by the multi-particle transport algorithm. It is implemented in C++, using Microsoft Visual Studio.Net 2005 as the compiler. OpenGL is used for rendering purposes. The dimensions of the workspace are 89.6 μm X 80 μm X 50 μm. Δt_i , Δt_c and t_{avg} are always selected as 0.25 s, 0.5 s and 0.5 s respectively so that they are consistent with the values observed in the holographic tweezer set-up. σ is taken to be 0.25 times the object radius. Water at room temperature (20° C) is always used as the fluid medium. Laser wavelength and numerical aperture of the objective lens are also assigned constant values as specified in Section 4.3. Laser power is chosen based upon the number of target objects such that the effective power in every primary trap is equal to 0.1 W. This implies that if three target objects are present, then the overall power is 0.3 W, whereas, if ten target objects are present, then the overall power is 1.0 W. Although target objects can be transported at greater speeds for higher values of laser power, it has been shown in Figure 4.24 that this does not correspond to a significant decrease in transport time due to an increase in trap radius and trap width values. Once a power value is selected at the start of any transport operation, it is not altered during the course of the operation. This means that if any other secondary trap is switched on, then the effective power decreases from the baseline value of 0.1 W for every trap.

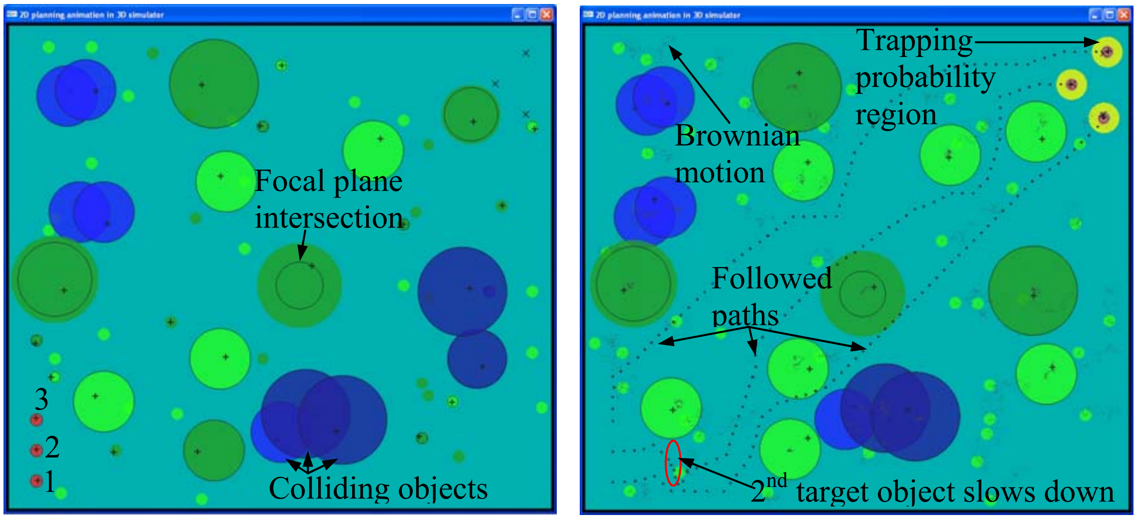
All the objects present in the workspace are considered to be trappable. The transportable objects are 2.01 μm diameter silica microspheres, whereas the non-transportable objects consist of a mixture of 10 μm and 15 μm diameter silica particles.

l_{tr} and l_{tw} are selected as 1.0 µm and 1.5 µm respectively at the baseline laser power of 0.1 W for the transportable objects. The corresponding values for the 10 µm diameter non-transportable objects are 5.0 µm and 7.0 µm respectively, whereas they are equal to 7.0 µm and 8.0 µm respectively for the 15 µm diameter objects.

Figure 5.7 shows the case where three target objects, initially positioned at (5 µm, 5 µm), (5 µm, 10 µm), and (5 µm, 15 µm) respectively, need to be transported to three goal locations whose coordinates are (85 µm, 65 µm), (80 µm, 70 µm), and (85 µm, 75 µm) respectively. The z-coordinate of all the target objects and the goal locations is equal to 5.55 µm. Forty-eight other transportable, twelve 10 µm diameter, and six 15 µm diameter non-transportable objects are also present. Initial X and Y coordinates of all the object centers are assigned following a uniform, random distribution. However, the z-coordinates of the object centers are given by an exponential distribution based upon the experimental data reported in Figure 4.17. This means that a greater number of objects are present closer to the bottom cover slip glass plate.

The estimated locations of the sensed object centers are shown by '+' signs, whereas, they have been drawn at their actual simulated locations. Based on experimental observations, an object is currently being sensed if a part of it lies within a 3 µm height zone in the workspace, centered about the focal or imaging plane. It can be seen that many objects which are sensed in the initial scene (Figure 5.7(a)), are no longer sensed in the final scene (Figure 5.7(b)). This occurs due to the settling of the objects at the bottom glass plate because of gravity. The size of every drawn object is proportional to its distance from the focal plane. If it lies below the focal plane, then it is drawn in a lighter shade. Brownian motion trajectories of all the untrapped objects are also shown in Figure

5.7(b). This clearly reveals the greater diffusivity of the smaller objects as compared to the larger ones.


2D planning animation in 3D simulator
Focal plane intersection
3
2
1
Colliding objects
2D planning animation in 3D simulator
Trapping probability region
Brownian motion
Followed paths
2nd target object slows down

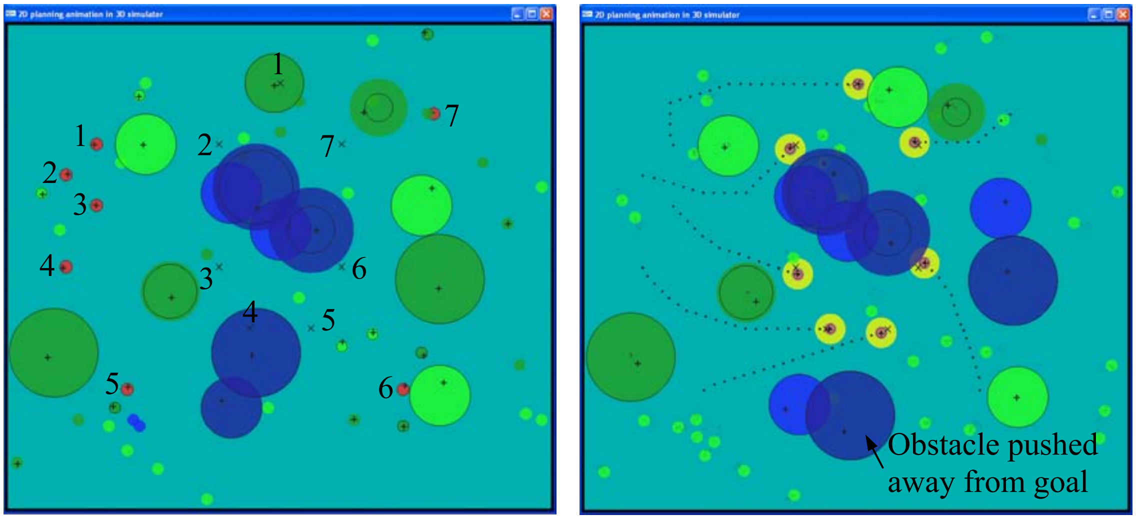
(a) Initial scene

(b) Final scene

Figure 5.7: Simultaneous transport of three 2.01 μm diameter target objects

The goal assignment algorithm assigns the target objects to the identically numbered goal location based on the minimum expected time to complete the transport operation. Figure 5.7(b) shows how the 2nd target object slows down during the initial part of the operation to avoid colliding with the 1st one. In certain situations, although some of the obstacles may no longer be sensed in the final scene, they forced one or more of the target objects to follow a circuitous path to avoid collisions. Again in some other situations, it can be observed that the followed paths of the target objects intersect with the drawn obstacles. This does not represent lack of success on the part of the planner to effectively avoid obstacle collisions. Instead, they occur due to two reasons. Firstly, the Brownian motion of the obstacle may bring it to a significantly different location from where it was when the target object moved past it. Secondly, all the obstacles are not lying in the same horizontal plane. That is why, the focal plane intersection circles play an important role in ascertaining whether, indeed, geometric collisions took place. Of course, the target objects have to maintain a larger separation from all the trappable objects to prevent undesired trapping or de-trapping based on the corresponding effective obstacle region radius values.

It took about 17.5 s to complete the transport operation. The 3rd target object reached its goal location earliest, followed by the 1st and the 2nd ones. All the primary traps are kept switched on until the operation is completed. This prevents unwanted drifting of the target objects that have already reached their respective goal locations.


2D planning animation in 3D simulator
1
1
2
2
3
7
7
3
4
6
3
4
5
5
6
2D planning animation in 3D simulator
Obstacle pushed
away from goal

(a) Initial scene

(b) Final scene

Figure 5.8: Simultaneous transport of seven 2.01 μm diameter target objects

Figure 5.8 shows another case where seven target objects are transported simultaneously. The initial coordinates of the target objects are (15, 60), (10, 55), (15, 50), (10, 40), (20, 20), (65, 20), and (70, 65) respectively. The coordinates of the goal locations are (45, 70), (35, 60), (35, 40), (40, 30), (50, 30), (55, 40), and (55, 60) respectively. All units are in microns. The common z-coordinate is same as in the previous case. Although the 2nd goal location lies closest to the 1st target object (in terms of Euclidean distance), the goal assignment algorithm assigns it to the 2nd target object. Correspondingly, all other goal locations are assigned to the identically numbered target objects to minimize the expected time to complete the transport operation. It can also be seen from Figure 5.8(b) that if an obstacle occupies a goal location and it does not diffuse away from that location on its own, then a secondary trap is used to push it away to any location in the workspace not occupied by some other goal location. This transport operation is completed in 5.5 s and the sequence of target objects in terms of non-decreasing time to reach the assigned goal location is given by (7, 3, 6, 2, 5, 4, 1). The diffusion lengths are much smaller in this case than the previous one due to the shorter time interval between the beginning and end of the operation.

5.4.2 Performance Characterization

As in the case of the single-particle planner, additional experiments have been performed in the simulator to investigate the performance of the multi-particle transport algorithm. This includes enumerating the change in plan computation time with respect to the number of target objects as well as the variation in operation completion time with respect to the number of obstacles. The number of target objects and obstacles are expected to vary significantly from one experiment to the other. So this characterization will enable one to identify the limit of effectiveness of the planning approach and provide insight as regards potential improvements. All the experiments are run on an Intel Core2 Quad processor PC with 2.83 GHz clock speed and 8 GB of RAM. As in the case of the simulated trajectories, only trappable objects (silica microspheres) are considered to be present in the system. Sizes of the transportable and non-transportable objects and other physical parameters (workspace dimensions, laser power etc.) are also same as in those two cases. The target objects are always transported in the z = 5.55 μm plane.

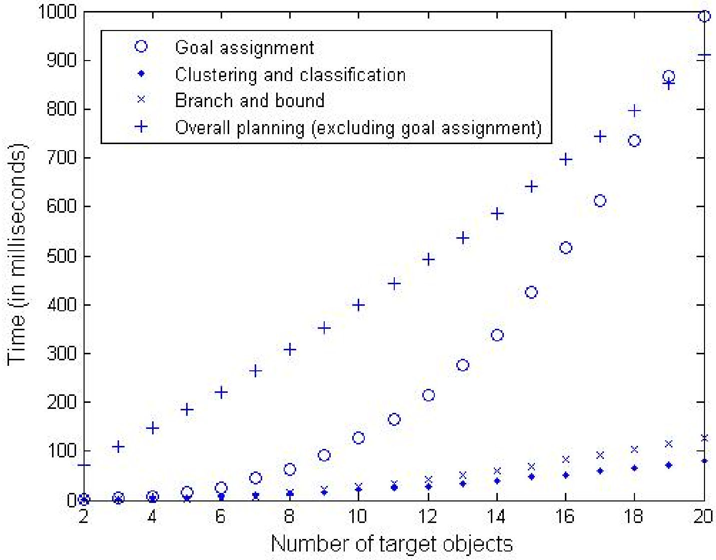


Figure 5.9: Planning approach computation time variation with respect to the number of target objects

Figure 5.9 plots the computation time of different steps of the planning approach versus the number of target objects present in the workspace. The total number of transportable objects is always selected as fifty. Twelve other 10 μm diameter and six other 15 μm diameter non-transportable objects are also kept in the workspace. Ten simulation runs, each with a randomly generated initial scene (X-Y positions of all the objects and goal locations), are used to obtain every data point. The goal assignment algorithm is invoked only once at the beginning of every transport operation. However, all the other steps in the planning approach are called repeatedly after every Δt_c. The average time taken to compute the output over the entire duration of the operation is recorded for each of the planning approach steps. The mean of this averaged value taken over ten simulation runs is used for generating the plot.

It can be seen from the Figure that the goal assignment computation time increases as the cubic power of the number of target objects. Although the theoretically possible worst-case bound is inferior to this result, in practice many entries in the expected transport time matrix table do not have to be considered for obtaining feasible assignments. Even if they are considered, often the algorithm terminates before entering the maximum bipartite graph matching step. As mentioned in Section 5.2, the clustering (and consequently classification) step computation time increases as the quadratic power of the number of target objects. The depth-first branch and bound computation time also increases in a similar manner, although the actual values are greater than those for clustering and classification. From the standpoint of practical usefulness, it is most important to know the overall planning time taken to compute all the collision-free paths since this will directly affect the attainable controller update rate. The trend is almost linear due to the fact that most of the time is spent on computing the individual paths for each of the target objects sequentially. Individual particle transport algorithm takes about 35 ms on an average for the current workspace state. A controller update rate of 2 Hz (total computation time of less than 500 ms) is obtainable as long as not more than twelve target objects are present. This planning time does not include the one-time effect of assigning goals optimally.

Figure 5.10 shows the variation in the time taken to transport six target objects from fixed initial locations to specific goal locations under varying workspace conditions. The initial locations of the target objects are selected as (5, 5), (5, 10), (5, 15), (5, 20), (5, 25), and (5, 30) respectively. The corresponding goal locations are (85, 50), (85, 55), (85, 60), (85, 65), (85, 70), and (85, 75) respectively. All the units are in microns. Eleven workspace states are considered beginning from a relatively sparse setting (twenty five transportable objects, seven 10 µm obstacles, and four 15 µm obstacles) to progressively more crowded settings. In every successive workspace state, the number of transportable objects is increased by five, the number of 10 µm obstacles is increased by 1, and the number of 15 µm obstacles is kept constant or increased by one alternatively. The average value from ten simulation runs is used to record every data point. X-Y coordinates of all the object centers are assigned following a uniform, random distribution the first time a particular simulation run is performed for a given workspace state. However, these values are stored and re-used in the remaining nine runs.

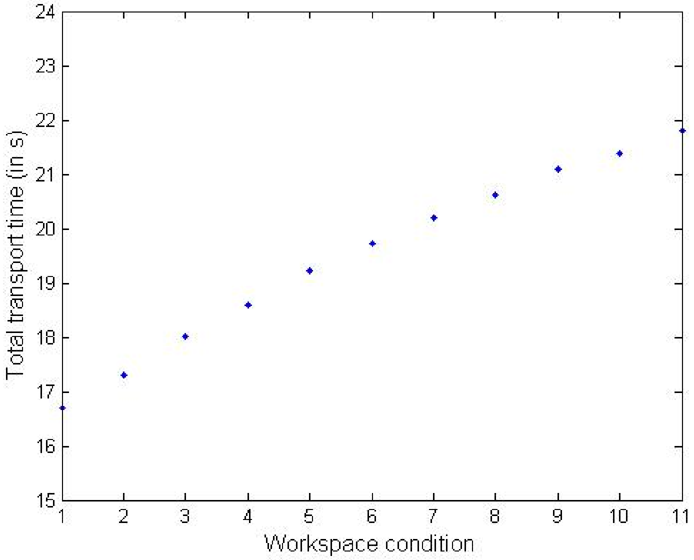


Figure 5.10: Transport time variation with respect to the number of obstacles present in the workspace

The trend resembles a linear one because a linear increase in the number of obstacles increases the possibility of additional collision avoidance almost linearly when the number of objects in the workspace is not very high and almost all of them are present near the focal plane, close to the bottom plate. However, this is no longer true for crowded scenes, when at least few obstacles will be positioned quite far above the focal plane (following the exponential distribution mentioned in Section 4.3). These obstacles are usually not potentially colliding. Hence, the slope of the linear trend decreases with a significant increase in the number of obstacles.

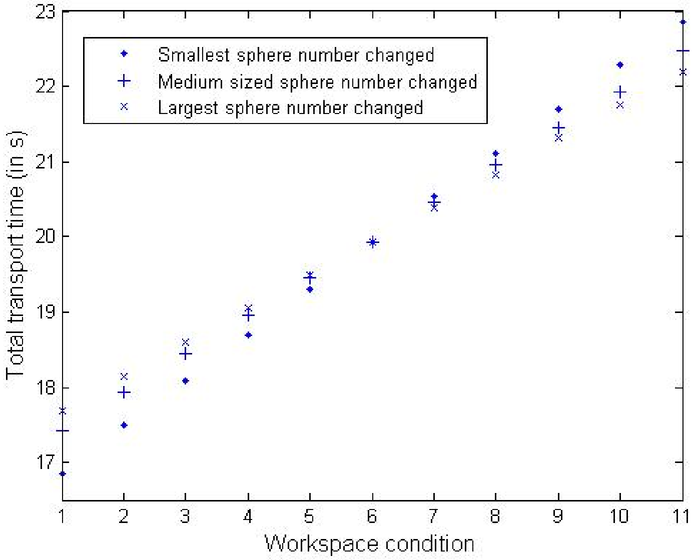


Figure 5.11: Transport time variation with respect to differential change in the number of obstacles present in the workspace

Figure 5.11 shows the variation in the time taken to transport six target objects when different number of small (2.01 µm diameter), medium (10 µm diameter), and large (15 µm diameter) sized obstacles are present in the workspace. The initial locations of the target objects and goal location coordinates are same as in the previous case. The number of runs and assignment of initial positions of the obstacles are also same as in the last case. Three data trends are shown in this Figure. Workspace # 6 represents the condition when 50 small objects, 12 medium objects, and 7 large objects are present in all the three data plots. The first trend (shown by '.' signs) represents transport time variation with respect to change in the smallest sized spheres only, keeping the numbers of other two sphere sizes constant. The second trend (shown by '+' signs) represents the variation with respect to change in the number of medium sized spheres only keeping the two other numbers constant, whereas the last trend (shown by 'x' signs) represents the variation with respect to change in the number of largest sized spheres only, keeping the other two numbers fixed. The number of smallest sized sphere is always increased by five and the number of medium and largest sized sphere is always increased by one between one workspace condition and the next higher numbered one.

Broadly speaking, all the three trends are more or less linear as observed in Figure 5.10. However, the slopes are different; change in 2.01 μm diameter sphere number has the maximum effect on transport time, followed by the change in 10 μm and 15 μm diameter sphere numbers respectively. This can be explained by considering the counteracting effects of diffusion and trapping probability region size based upon object radius. Since diffusivity is inversely proportional to radius, diffusion length varies inversely as the square root of radius. On the other hand, trapping probability radius and width both increase with object size, although this dependency does not follow a simple mathematical rule. Now, more object diffusivity means a greater possibility of unintended collision which will increase the average transport time. Similarly, larger trapping probability region means a smaller space is available to transport the target objects; this also increases the average transport time as the target objects have to follow more circumvented routes. However, the first phenomenon (diffusion) has a greater effect on increasing the transport time as can be seen from the differences in the slopes. Another observation from this Figure is that the slope decreases towards the higher numbered workspace conditions in case of the smallest sphere (as also seen in the previous Figure), whereas no such effect is observed in case of the other two spheres. This happens due to the fact that there is a greater spread in the z-distribution of the small spheres when a large number of them are present in the workspace.

5.5 Summary

In order to prototype or repair micro scale devices and assemblies or to manipulate biological objects, multiple particles need to be transported in tandem to desired locations in the workspace. Coordinated transport of multiple particles poses additional challenges in terms of computation time, and resolving collisions among the transported particles.

This Chapter describes a decoupled and prioritized stochastic dynamic programming framework for transporting multiple particles from their current locations to given goal locations in the minimum expected time. A physical problem is modeled as a path planning problem by removing the constraint that the workspace consists of only microspheres of a particular type and size and extending the mathematical framework developed in the previous Chapter. An iterative version of a maximum bipartite graph matching algorithm is developed to assign goal locations to target objects optimally in polynomial time. A three-step approach, comprising of clustering, classification, and depth-first branch and bound optimization is developed to assign priorities to the target objects for computing collision-free paths during simultaneous motion. Experimental validation in the form of manual transport of multiple spheres is carried out to demonstrate the feasibility of moving several objects simultaneously. Representative simulation trajectories are presented to highlight the optimum assignment of goal locations to target objects and simultaneous motion of multiple target objects by avoiding collisions with obstacles as well as other target objects. Finally, simulation experiments are conducted to characterize the extent of usefulness of the planning approach at different workspace states in terms of the number of target objects and obstacles.

Chapter 6

CONCLUSIONS

This chapter presents the expected intellectual contributions and the anticipated benefits to the industry and the research community from the advances reported as a part of this dissertation.

6.1 Intellectual Contributions

The current dissertation work has a significant body of intellectual contributions. They are listed as follows.

1. *Quantifying trap-particle interaction using trapping probability*: Intensive offline simulation is performed to estimate trapping probability values for 2.01 and 5.0 μm diameter silica spheres under the influence of stationary and moving optical traps. The probability contour plots enable visualization of the change in trapping behavior with respect to relative particle location, particle size, trap speed, and direction of trap motion. Significant insight is obtained from these plots; based on the results, the trapping behavior is modeled using two concentric circles at the focal plane. For more accurate modeling, circles should be replaced by two ellipse-like curves joined along the axis perpendicular to the direction of trap motion for horizontally moving traps. One of the curves is stretched out along the direction of the beam motion. This clearly reveals strong asymmetric trapping behavior in the horizontal plane. Moreover, this asymmetry is non-uniform at different cross-sections. At distances far above the focal

plane, trapping behavior is completely absent in the half-plane in which no motion takes place. In the other half-plane, ellipse-like curves can be used again but with a major difference from the previous case. Here, the lower probability contours can be represented by larger curves than the higher probability ones. Vertical trap motion opposite to the beam propagation direction shows that all the probability contours move closer to the beam axis. This dissertation investigates and explains such interesting phenomena for the first time.

2. *Effectiveness of spatial-partitioning based RBF fitting technique for querying stochastic simulation data*: The merits of fitting Gaussian RBFs to large simulation data sets inherently containing noisy data (due to random processes occurring in the environment) is clearly shown in the current dissertation. This fitting technique not only enables compact representation of large data sets, it also mitigates noise and yet captures all the local and global features. Thus, it is accurate for planning purposes. Arrangement of the fitted RBFs in the form of kd-trees allows fast data querying operations and provides lot of flexibility in terms of encoding additional parameters such as laser power, additional geometric attributes of the components under consideration etc. in the future. As diffusivity increases with decreasing particle size, the effects of Brownian motion become more pronounced. This increases the complexities in the pattern shapes, as well as the level of noise present in the data. This dissertation work demonstrates that Gaussian RBFs can support compact representation (in terms of the number of fitted functions) and fast query operations at those small size scales which will provide useful guidelines to future users in the simulation, modeling, and data mining communities.

3. *Faster, dynamic POMDP framework*: An improved, time-dependent version of an approximate, discrete POMDP algorithm, called the QMDP_NLTDV algorithm, is presented in this dissertation. Discussion on the optimality and convergence of this algorithm shows that a theoretically sound and practically useful motion planning approach is now available to the research community. This algorithm demonstrates the benefit of using stochastic dynamic programming in solving real-world planning problems (since it inherently models uncertainties). It reveals how the mathematical nature of the value function can be modified (made non-linear) by changing the form of the payoff function from one-step gain to overall gain in order to enhance convergence rates without compromising optimality. It also highlights the importance of developing control action pruning conditions to reduce the search space drastically so that results can be computed within a few milliseconds. The notion of dominance is important so that no control action that can possibly lead to an optimum solution is ever excluded, and any control action that can never yield an optimum solution is always excluded. Inclusion of a time parameter within the algorithm framework also negates the effect of plan execution latency on performance degradation to a large extent.
4. *Decoupled multi-particle planner with optimum goal assignment*: A decoupled approach for transporting multiple particles simultaneously is developed as a part of the dissertation work. The fact that collision-free paths for all the target objects can be computed in less than a second (and often in less than half a second) for relatively crowded scenes shows the practical utility of this approach as opposed to a much more computationally intensive (combinatorially hard) coupled method. Clustering

and classification ensure that the final optimization technique only runs on few target objects to assign the priorities in which their paths are determined. This is true because more than twenty particles are not expected to be transported in tandem in a typical workspace and based upon practical observations, more than five or six are not expected to remain clustered together. The problems of generating incomplete and sub-optimal solutions as in the case of many other decoupled approaches are avoided by employing the branch and bound optimization algorithm. Optimum goal assignment algorithm is invoked as a pre-processing step to ensure that the best possible of selection of target objects to goal locations is made from the very beginning. This is necessary so that the whole operation can be completed in the shortest possible time instead of encountering a situation where certain objects reach the goal locations quickly, and others take too long to reach their respective destinations. Underlying characteristics of the problem are utilized along with an iterative version of a bipartite graph matching algorithm to obtain results within a reasonable amount of time by avoiding combinatorial searching.

5. *Experimentally validated real-time path planner*: Both simulation and physical experiments are performed to demonstrate the practical utility of the real-time planner. Single particle transport simulation shows that it works well in crowded scenes, and is able to detect and transport the target object through corridors that were originally too narrow but widened over the course of time. Again, flexibility in choosing the objective function parameters provides an option to the user to entirely avoid collisions, or ignore them, and transport multiple objects to the goal location simultaneously. Physical experiments on fully automated transport of 2.01 μm

diameter silica particles show the capability of transporting a specific object by either circumventing or trapping freely diffusing obstacles. The planner may sometimes allow collisions to prevent a much longer, circuitous route in relatively crowded scenes. Multiple particle transport simulation highlights the optimum assignment of goal locations to target objects and simultaneous motion of multiple target objects by avoiding collisions with obstacles of varying sizes and material properties as well as other target objects. It also shows that certain target objects may be slowed down and obstacles may be pushed away from a goal location if necessary. Manual transport of multiple particles in the physical set-up demonstrates the feasibility of moving several particles in tandem. Thus, the planner provides the first step towards automating a wide variety of manipulation operations using optical tweezers. It is capable of working with both scanning mirror and holographic optical tweezers set-ups and can be attached with any basic imaging and controller system to move around particles in the XY plane in a stochastically optimum manner.

6. *Performance characterization using simulation*: Since it is very time-consuming and challenging due to hardware restrictions to evaluate the performance of the path planner under a wide variety of operating conditions in the physical set-up, simulation is used for this purpose. This provides useful guidelines and insight regarding the applicability or the extent of effectiveness of the planning approach at different situations. The performance of the single particle planning algorithm is enumerated under varying conditions of laser power, imaging uncertainty, and object number density as a function of height from the bottom plate. Increase in laser power is not found to decrease the overall transport time appreciably, whereas increase in imaging

uncertainty is found to have a significant effect in increasing the transport time. Performance degradation as a result of not considering the z-location of the object centers only arises relatively close to the bottom plate and will be further reduced if the increase in planning computation time is taken into account. Simulation experiments are also conducted to characterize the performance of the multiple-particle planning approach at different workspace states in terms of the number of target objects and obstacles. Plots show that planning can be completed in less than a second for 20 target objects in reasonably crowded scenes and the actual transport operation time varies more or less linearly with respect to the number of obstacles. Moreover, change in the number of smaller sized obstacles has a greater effect on the transport time as compared to a change in the number of bigger obstacles.

6.2 Anticipated Benefits

This dissertation work provides a step towards realizing the full benefits of micro and nanotechnology. As discussed earlier in Chapter 1, lack of automation seriously limits the rate at which new component-based devices can be invented and hence delays the introduction of these devices to the market. Due to superior strength, stiffness, electrical, and thermal conductivities, nanowires and nanotubes form the basic building blocks for many devices such as p-n junction diodes, transistors, drug delivery vessels etc. It is only in the recent past that researchers have successfully demonstrated trapping nanowires and nanotubes using optical tweezers; well-established trapping force models are not yet available. That is why this dissertation used spheres to illustrate the real-time planning approach. However, the computational framework developed as a part of the current dissertation work can be extended to deal with nanowires and nanotubes. By replacing the force models suitably, Langevin dynamics simulation can be performed for those components. Kd-trees can be effectively used to incorporate additional parameters or attributes such as length, angle of tilt etc. associated with nanowires within the Gaussian RBF fitting framework. Since they have been shown to be trapped by orienting them vertically along the beam axis, the dynamic programming framework can be used directly to transport them in the horizontal plane or along the vertical direction just by modifying the configuration space (or effective collision space) using Minkowski sums. Any in-plane orientation of a horizontally aligned nanowire or nanotube can also be incorporated within this framework by introducing an additional dimension (analogous to 2D robot orientation) in the workspace state vector.

Another anticipated benefit of this dissertation work is in the field of medicine. Pushing one or more beads against a cell membrane or wall after transporting them to the cell boundary can give an estimate of the cell stiffness. It is observed that in case of many diseases, healthy cells behave significantly differently from diseased ones in terms of such mechanical properties. This provides an alternative mechanism to chemical treatment or imaging based techniques for detecting occurrence of diseases at an early stage, or in detecting the extent of spread, and acuteness of the afflicted diseases at a later stage. Although this can be done by manual control of optical tweezers in the case of a single cell and few beads, automation is a must for rapid testing of several cells simultaneously in practice. Moreover, automated transport of drug-carrying liposomes to selective cells will help in drug discovery and delivery, especially in cancer treatment. This will open up new avenues in exploring the effect of a larger number of drugs on sample cells and provide an alternative treatment methodology to chemotherapy, where patients often suffer from various side effects owing to the death of healthy cells in the neighborhood of the diseased ones due to radiation exposure. An ensemble of trapped microspheres can be used to bring cells together and form a relatively stable and stationary cluster to investigate the effect of cells to certain pathogens over a long period of time. And last but definitely not the least, the current dissertation work will facilitate the process of positioning a larger number of cells of different types faster and with a higher degree of precision on a substrate, where they can form desired arrangements. Such arrangements will be useful for carrying out quantitative studies of inter cell communication and cell motility. Control over cell-cell interactions will provide new insights into cell signaling pathways and open up fresh opportunities for medical diagnosis and treatment.

6.3 Future Work

Although it is anticipated that the dissertation will provide the foundations for automating micro and nano scale assembly operations and manipulating biological components, further work needs to be done in the following areas to address both the problems.

1. *Developing imaging of 3D workspace*: This will require development and integration of the planning system with a 3D imaging system that can provide estimates about the *z*-locations of the object centers as well other than the *xy*-coordinates. Either optical section microscopy or confocal microscopy can be used to obtain a stack of images of the workspace at various horizontal cross-sections. A gradient function based

approach mentioned in Section 2.1 can be used to combine the information present in various images to estimate the 3D coordinate of the object centers. However, this approach needs to be generalized for biological objects. This is a challenging task due to translucency effects and irregular-shaped boundaries. Hence, extensions to the method are required which will also enable users to reliably distinguish between biological objects and non-biological ones and among different types of biological and non-biological components themselves. The latter task will require developing image processing algorithms to identify components based on differences in properties such as refractive index, density etc.

2. *Extending the framework to deal with nanoparticles*: The current planning framework needs to be extended to deal with nanometer sized particles for several useful applications. This requires using electromagnetic field-based optical trapping force calculations only unlike what is done in the dissertation. Such calculations are more time-consuming; moreover, the time-step used during simulation also needs to be reduced so that it is less than the particle relaxation time, which is directly proportional to the square of the particle radius. Thus, significantly greater time will be required to acquire trapping probability data and simulate planning operations. This may necessitate using parallel computing techniques to run multiple simulations simultaneously in several computers. Graphics Processing Units (GPUs) may also be used to achieve computational speed-up as shown in [Bali09]. Some modifications to the current multi-particle transport algorithm may also be required to reduce the planning time to compensate for the greater diffusivities of the smaller particles.

3. *Planning for non-spherical components*: As mentioned in the previous Section, many useful devices can be created out of nanowires and nanotubes. So planning needs to be performed for these components to create micro and nano scale assemblies automatically in the future. This will require use of suitable optical trapping, thermal, and viscous drag force models that are applicable for such cylindrical or flat components. Although established models exist in the literature for the latter two forces, researchers are still working on developing experimentally-validated trapping force models for non-spherical components. Thus, planning may entail working on development and validation of trapping force models in conjunction with physicists from *ab initio* principles. Once force models are available, the planning approach needs to be modified significantly to deal with additional parameters such as length, orientation etc. This will increase the dimensionality of the configuration space and alter the shape of the extended obstacles formed by taking the Minkowski sums of the target components and the diffusing obstacles. Increased dimensionality means that the size of the workspace set, control action set, state transition probability matrix, and the belief function space will also grow, thereby, increasing the planning time considerably. So, additional pruning conditions and other mathematical techniques need to be developed based on the insight about the problem to reduce the computational burden.

4. *Planning in three dimensions*: The current planner needs to be extended to perform transport operations in 3D space. This will require using 3D trapping probability data generated using offline simulation in the planning algorithm. Querying Gaussian RBF-based simplified models will then be useful as it will not be possible to use

simple shapes such as circles to represent the trapping probability data. Rotations of components such as nanowires and nanotubes may also need to be considered to enable stable trapping under different circumstances and maneuvering in narrow free spaces. However, simultaneous translation and rotation in 3D may be computationally intractable and theoretically hard to formulate. So, sets of decoupled motions, such as translation in *xy*-plane, translation along ±*z*-axis, and rotation about the 3 different coordinate axes may be considered separately and one at a time. This will reduce the dimensionality of the individual configuration spaces as well. However, even such decoupled motion planning will require significant extensions to the current problem formulation and planning approach. Moreover, adhesion of objects to each other and with the cover slip glass plate, effect of laser beam occlusion due to the presence of objects whose parallel projections intersect on a horizontal plane, and anisotropic and reduced diffusitivities near the glass surface have to be modeled to evaluate the planner performance correctly.

5. *Transporting multiple biological objects together using dynamically reconfigurable grippers*: Automating biological object transport operations involves several additional challenges. First, it includes learning and following the proper protocols for handling biological objects safely so that they can be kept in a healthy state and useful physical experiments can be performed. Second, problem modeling has to be significantly modified to take into account the compliant behavior and inhomogenities in the structure and material properties (for example, inside a cell, the nucleus and the other organelles are significantly denser with higher refractive indices as compared to the fluid portion of the cytoplasm). Accordingly, the force models

have to be changed. Third, extensions to the planning approach need to be developed in order to grasp the object in a stable configuration using particles such as silica microspheres. Multiple particles need to approach the biological object from different directions and at different locations with certain approach angles synchronously for stable griping. The number of such particles also needs to be determined adaptively at run-time for different types of biological objects so that the right balance between grasping strength and transport speed can be obtained. Last, plans have to be developed to impart fine motions to the individual gripping particles in addition to the overall motion of the object-particle cluster to take into account the effect of slight drifting of the object or one of the particles during motion.

Appendix

A. Ashkin's Ray Optics Model for Optical Trapping Forces

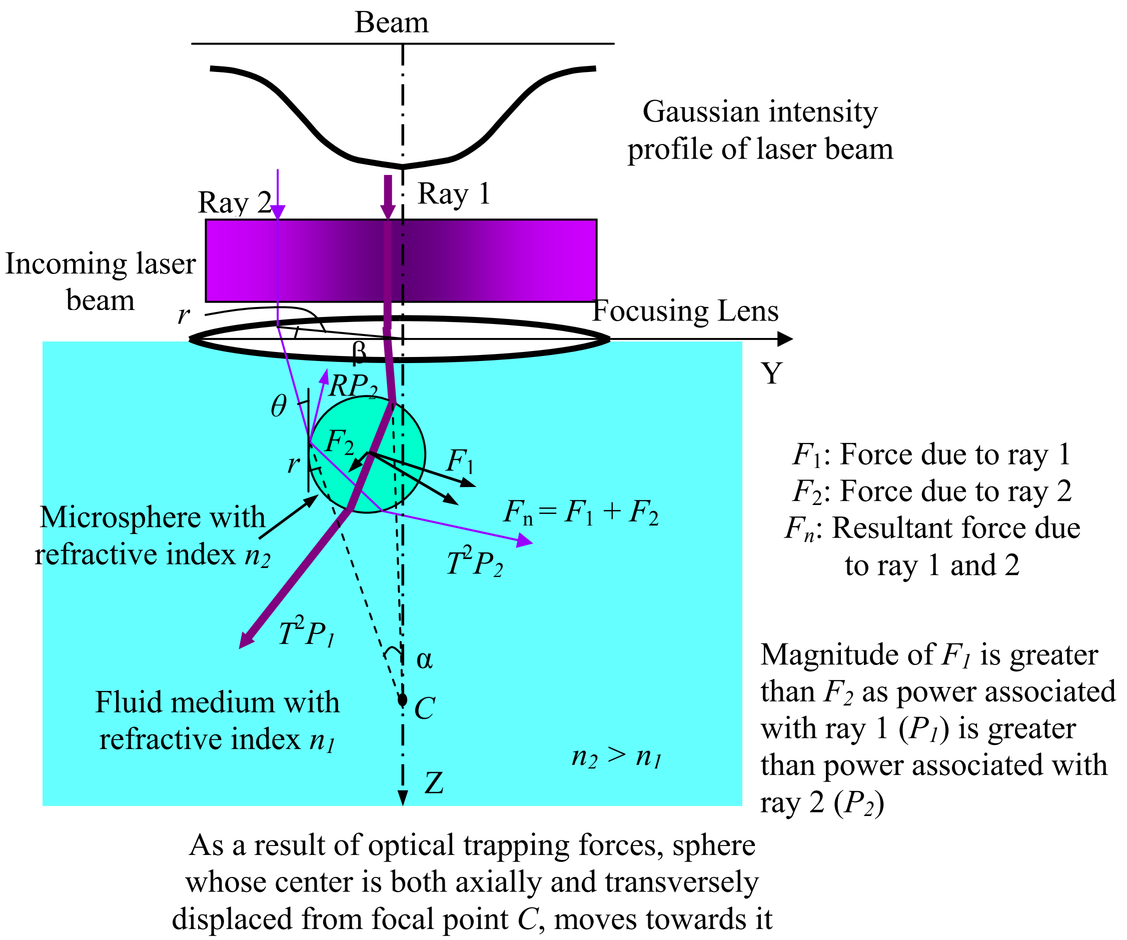

Beam
Gaussian intensity profile of laser beam
Ray 2
Ray 1
Incoming laser beam
Focusing Lens
r
r
β
Y
RP_2
θ
F_2
F_1
r
$F_n = F_1 + F_2$
Microsphere with refractive index n_2
T^2P_2
T^2P_1
α
C
Fluid medium with refractive index n_1
$n_2 > n_1$
Z
F_1: Force due to ray 1
F_2: Force due to ray 2
F_n: Resultant force due to ray 1 and 2
Magnitude of F_1 is greater than F_2 as power associated with ray 1 (P_1) is greater than power associated with ray 2 (P_2)
As a result of optical trapping forces, sphere whose center is both axially and transversely displaced from focal point C, moves towards it

Fig. A1. Schematic illustration of optical trapping forces using ray optics model

Schematic illustration of the ray tracing method for computing trapping forces is shown in Fig. A1. Assuming that we are working with a TEM$_{00}$ laser beam and it propagates along +*z*-direction, the infinitesimal forces along the axial and transverse directions can be written as [Ashk92]:

$$dF_z = \left(q_s \sin\alpha + q_g \cos\mu \cos\alpha\right) dP \tag{A.1}$$

and $dF_y = \left(-q_s \cos\alpha \cos\beta + q_g \cos\mu \sin\alpha \cos\beta + q_g \sin\mu \sin\beta\right) dP$ (A.2)

where $dP = \dfrac{2P}{\pi w_0^2} \exp\left(\dfrac{-2r_a^2}{w_o^2}\right) r_a dr_a d\beta$. (A.3)

Here, P is the laser power, w_0 is the laser beam waist radius at the lens aperture, β is the azimuthal angle and r_a is the radial variable which will be integrated over suitable ranges to give the final force values. The scattering and gradient force coefficients for a single incident ray are computed using the following equations respectively:

$$q_s = \frac{n_1}{c}\left\{1 + R\cos 2\theta - \frac{T^2\left[\cos\left(2\theta - 2r\right) + R\cos 2\theta\right]}{1 + R^2 + 2R\cos 2r}\right\} \quad \text{(A.4)}$$

and $q_g = \dfrac{n_1}{c}\left(R\sin 2\theta - \dfrac{T^2\left[\sin\left(2\theta - 2r\right) + R\sin 2\theta\right]}{1 + R^2 + 2R\cos 2r}\right)$ (A.5)

Here, n_1 is the refractive index of fluid medium, c is the speed of light in free space, θ is the angle of incidence and r is the angle of refraction. For a circularly polarized light, Fresnel transmission (T) and reflection (R) coefficients are calculated as follows:

$$R = \frac{\left(R_s + R_p\right)}{2}, \quad \text{(A.6)}$$

and $T = \dfrac{\left(T_s + T_p\right)}{2}$ (A.7)

where $R_s = \left(\dfrac{\sin\left(r - \theta\right)}{\sin\left(r + \theta\right)}\right)^2, T_s = 1 - R_s$ (A.8)

and $R_p = \left(\dfrac{\tan\left(r - \theta\right)}{\tan\left(r + \theta\right)}\right)^2, T_p = 1 - R_p$ (A.9)

Here s and p components of the electric field vector are resolved parallel and perpendicular to the plane of incidence respectively. Finally, the angles α (forward projection angle made by an incident ray passing through the focus with the horizontal plane) and μ are given by:

$$\alpha = \cos^{-1}\left(\frac{\vec{v}_1 . \vec{v}_2}{\left|\vec{v}_1\right|\left|\vec{v}_2\right|}\right), \tag{A.10a}$$

$$\text{where } \vec{v}_1 = r_a \sin\beta \,\hat{i} + r_a \cos\beta \,\hat{j} - f\,\hat{k} \tag{A.10b}$$

$$\text{and } \vec{v}_2 = r_a \sin\beta \,\hat{i} + r_a \cos\beta \,\hat{j} \tag{A.10c}$$

$$\mu = \cos^{-1}\left(\frac{\tan\alpha}{\tan\beta}\right) \tag{A.11}$$

Here *f* is the focal length of the lens. Assuming that its numerical aperture *NA* is also known, the radial limit of integration $r_{\max}$ is given by:

$$r_{\max} = f \tan\left(\phi_{\max}\right) \tag{A.12}$$

where half laser cone angle $\phi_{\max} = \sin^{-1}\left(\frac{NA}{n_{oil}}\right)$ for oil-immersion objective (A.13)

Ashkin's derivation is based on the assumption that the coordinate system is rotated such a way that the horizontal component of the sphere center lies on the *Y*-axis. Thus for any arbitrary location of sphere center, we first need to compute the relative radial distance and the angle made by the sphere center with the *Y*-axis ($\theta^{/}$). After computing the axial (F_z) and transverse (F_y) components of the optical trapping force by integrating Equations (A.1) and (A.2) respectively, the forces need to be distributed along the 3 orthogonal axes as follows:

$$F_x = F_y \sin\theta^{/}, F_y = F_y \cos\theta^{/} \text{ and } F_z = F_z \tag{A.14}$$

Bibliography

[Alle87] M. P. Allen, and D. J. Tildesley. *Computer Simulation of Liquids.* Clarendon Press, New York, NY, 1987.

[Alte05] R. Alterovitz, A. Lim, K. Goldberg, G. S. Chirikjian, and A. M. Okamura. Steering Flexible Needles under Markov Motion Uncertainty. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, Alberta, Canada, 2005.

[Amid02] I. Amidro. Scattered data interpolation methods for electronic imaging systems: a survey. *Journal of Electronic Imaging*, 11(2):157-176, 2002.

[Apay02a] M. S. Apaydin, C. E. Guestrin, C. Varma, D. L. Brutlag, and J. C. Latombe. Stochastic roadmap simulation for the study of protein-ligand interactions. *Bioinformatics*, 18:S18-S26, 2002.

[Apay02b] M. S. Apaydin, D. L. Brutlag, C. E. Guestrin, D. Hsu, J. C. Latombe, and C. Verma. Stochastic roadmap simulation: An efficient Representation and Algorithm for Analyzing Molecular Motion. In *Proceedings of the 6th annual international conference on Computational Biology*, Washington, DC, 2002.

[Ashk86] 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(5):288-290, May 1986.

[Ashk89] A. Ashkin,and J. M. Dziedzic. Optical Trapping and Manipulation of Single Living Cells Using Infrared-Laser Beams. *Berichte der Bunsen-Gesellschaft-Physical Chemistry*, 93:254-260, 1989.

[Ashk92] A. Ashkin. Forces of a single-beam gradient laser trap on a dielectric sphere in the ray optics regime. *Biophysical Journal*, 61(2):569-582, 1992.

[Ashk00] A. Ashkin. History of optical trapping and manipulation of small-neutral particle, atoms, and molecules. *IEEE Journal of Selected Topics in Quantum Electronics*, 6(6):841-856 2000.

[Azar97] K. Azarm, and G. Schmidt. Conflict-free motion of multiple mobile robots based on decentralized motion planning and negotiation. In *Proceedings of the IEEE International Conference on Robotics and Automation*, Albuquerque, NM, 1997.

[Bali05] A. Balijepalli, T. LeBrun, C. Gagnon, Y. G. Lee, and N. Dagalakis. A modular system architecture for agile assembly of nano-components using optical tweezers. In *Proceedings of the 2005 SPIE Conference on Optics and Photonics*, San Diego, CA, 2005.

[Bali06] A. Balijepalli, T. W. LeBrun, and S. K. Gupta. A flexible system framework for a nanoassembly cell using optical tweezers. In *Proceedings of the ASME International Design Engineering Technical Conferences & Computers and Information in Engineering Conference*, Philadelphia, PA, 2006.

[Bali09] A. Balijepalli, T. W. LeBrun, and S. K. Gupta. Stochastic Simulations with Graphics Hardware: Characterization of accuracy and performance. Accepted for publication in *ASME Journal of Computing and Information Science in Engineering*, 2009.

[Bane08a] A. G. Banerjee, A. Balijepalli, S. K. Gupta and T. W. LeBrun. Radial Basis Function Based Simplified Trapping Probability Models for Optical Tweezers. In

Proceedings of the *ASME Design Engineering Technical Conferences & Computers and Information in Engineering Conference,* Brooklyn, NY, 2008.

[Bane08b] A. G. Banerjee and S. K. Gupta. Use of Simulation in Developing and Characterizing Motion Planning Approaches for Automated Particle Transport using Optical Tweezers. In Proceedings of the *2nd International Virtual Manufacturing Workshop*, Turin, Italy, 2008.

[Bane09a] A. G. Banerjee, A. Balijepalli, S. K. Gupta and T. W. LeBrun. Generating Simplified Trapping Probability Models from Simulation of Optical Tweezers Systems. *ASME Journal of Computing and Information Science in Engineering*, 9(2), 2009.

[Bane09b] A. G. Banerjee, A. Pomerance, W. Losert, and S. K. Gupta. Developing a Stochastic Dynamic Programming Framework for Optical Tweezer based Automated Particle Transport Operations. Accepted for publication in *IEEE Transactions on Automation Science and Engineering*.

[Bane09c] A. G. Banerjee, W. Losert and S. K. Gupta. A Decoupled and Prioritized Stochastic Dynamic Programming Approach for Automated Transport of Multiple Particles using Optical Tweezers. Accepted for publication in *ASME 3rd International Conference on Micro and Nanosystems*, San Diego, CA, 2009.

[Barr91] J. Barraquand, and J.-C. Latombe. Robot motion planning: A distributed representation approach. *Inernational Journal of Robotics Research*, 10(6):628-649, 1991.

[Bart88] J. P. Barton, D. R. Alexander, and S. A. Schaub. Internal and near-surface electromagnetic fields for a spherical particle irradiated by a focused laser beam. *Journal of Applied Physics*, 64(4):1632-1639, 1988.

[Bart89] J. P. Barton, D. R. Alexander, and S. A. Schaub. Theoretical determination of net radiation force and torque for a spherical particle illuminated by a focused laser beam. *Journal of Applied Physics*, 66(10):4594-4602, 1989.

[Bekr07] K. E. Bekris, and L. E. Kavraki. Greedy but Safe Replanning under Kinodynamic Constraints. In *Proceedings of the IEEE Inrternational Conference on Robotics and Automation*, Rome, Italy, 2007.

[Benn01] M. Bennewitz, W. Burgard, and S. Thrun. Optimizing schedules for prioritizing path planning of multi-robot systems. In *Proceedings of the IEEE International Conference on Robotics and Automation*, Seoul, Korea, 2001.

[Bent04] A. K. Bentley, J. S. Trethewey, A. Ellis, and W. C. Crone. Magnetic manipulation of Copper-Tin nanowires capped with Nickel ends. *Nano Letters*, 4(3):487-490, 2004.

[Berg00] M. de Berg, M. van Kreveld, M. Overmars, and O. Schwarzkopf. *Computational geometry: algorithms and applications*, *Second Edition*. Springer-Verlag, Berlin, 2000.

[Berg08] J. van der Berg, S. Patil, J. Sewall, D. Manocha, and M. Lin. Interactive Navigation of Multiple Agents in Crowded Environments. In *Proceedings of the Symposium on Interactive 3D Graphics and Games*, Redwood City, CA, 2008.

[Boor99] V. Boor, M. H. Overmars, and A. F. van der Stappen. The Gaussian Sampling Strategy for Probabilistic Roadmap Planners. In *Proceedings of the 1999 IEEE International Conference on Robotics and Automation*, Detroit, MI, 1999.

[Bran01] M. S. Branicky, S. M. LaValle, K. Olson, and L. Yang. Quasi-Randomized Path Planning. In *Proceedings of the 1999 IEEE International Conference on Robotics and Automation*, Seoul, Korea, 2001.

[Bruc02] J. Bruce, and M. Veloso. Real-time randomized path planning for robot navigation. In *Proceedings of the IEEE/RSJ International Conf. Intelligent Robots and Systems*, Lausanne, Swizerland, 2002.

[Carr01] J. C. Carr, R. K. Beatson, J. B. Cherrie, T. J. Mitchell, W. R. Fright, B. C. McCallum, and T. R. Evans. Reconstruction and representation of 3D objects with radial basis functions. In SIGGRAPH '01: *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, New York, NY, 2001.

[Cast05] K. Castelino, S. Satyanarayana, and M. Sitti. Manufacturing of two/three-dimensional micro/nano-structures by integrating optical tweezers with optical assembly. *Robotica*, 23(4):435-439, 2005.

[Ceci06] J. Cecil, D. Powell, and D. Vasquez. Assembly and manipulation of micro devices-a state of the art survey. *Robotics and Computer-Integrated Manufacturing*, 23(5):580-588, 2007.

[Chiv95] P. N. Chivate, and A. G. Jablokow. Review of surface representation and fitting for reverse engineering. *Computer Integrated Manufacturing Systems*, 8(3):193-204 1995.

[Chos06] H. Choset, K. Lynch, S. Hutchinson, G. Kantor, L. Kavraki, and S. Thrun., *Principles of Robot Motion: Theory, Algorithms, and Implementation*. The MIT Press, Cambridge, MA, 2006.

[Coll2D] http://www.plasmaphysics.org.uk/programs/coll2d_cpp.htm

[Corm97] T. H. Cormen, C. E. Leiserson, and R. L. Rivest, *Introduction to Algorithms*. The MIT Press, Cambridge, MA, 1997.

[Curt02] J. E. Curtis, B. A. Koss, and D. Grier. Dynamic holographic optical tweezers. *Optics Communications,* 207(1-6):169-175, 2002.

[Dhol07] K. Dholakia, W. M. Lee, L. Paterson, M. P. MacDonald, R. McDonald, I. Andreev, P. Mthunzi, C. T. A. Brown, R. F. Marchington, and A. C. Riches. Optical Separation of Cells on Potential Energy Landscapes: Enhancement with Dielectric Tagging. *IEEE Journal of Selected Topics in Quantum Electronics*, 13(6):1646-1654, 2007.

[Dosh08a] F. Doshi, and N. Roy. The Permutable POMDP: Fast Solutions to POMDPs for Preference Elicitation. In *Proceedings of the 7th International Conference on Autonomous Agents and Multiagent Systems*, Estoril, Portugal, 2008.

[Dosh08b] F. Doshi, J. Pineau, and N. Roy. Reinforcement Learning with Limited Reinforcements: Using Bayes Risk for Active Learning on POMDPs. In *Proceedings of the 25th International Conference on Machine Learning*, Helsinki, Finland, 2008.

[Ferr05] E. Ferrari, V. Emiliani, D. Cojoc, V. Garbin, M. Zahid, C. Durieux, M. Coppey-Moisan, and E. D. Fabrizio. Biological samples micro-manipulation by means of optical tweezers. *Microelectronic Engineering*, 78-79:575-581, 2005.

[Frie98] M. E. J. Friese, T. Nieminen, N. Heckenberg, and H. Rubinsztein-Dunlop. Optical alignment and spinning of laser-trapped microscopic particles. *Nature*, 394(6691):348-350, 1998.

[Frie05] G. Friedman, and B. Yellen. Magnetic separation, manipulation and assembly of solid phase in fluids. *Current Opinion in Colloid & Interface Science*, 10(3-4):158-166, 2005.

[Gala03] P. Galajda, and P. Ormos. Orientation of flat particles in optical tweezers by linearly polarized light. *Optics Express*, 11(5):446-451, 2003.

[Gayl07] R. Gayle, A. Sud, M. C. Lin, and D. Manocha. Reactive Deformation Roadmaps: Motion Planning of Multiple Robots in Dynamic Environments. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, San Diego, CA, 2007.

[Gijs04] M. A. M. Gijs. Magnetic bead handling on-chip: new opportunities for analytical applications. *Microfluidics and Nanofluidics*, 1(1):22-40, 2004.

[Gosa03] A. Gosavi. *Simulation-Based Optimization: Parametric Optimization Techniques and Reinforcement Learning*. Kluwer Academic Publishers, Norwell, MA, 2003.

[Gosh02] A. A. Goshtasby. Grouping and parameterizing irregularly spaced points for curve fitting. *ACM Transactions on Graphics*, 21(4):855-873, 2002.

[Gous82] G. Gouesbet, and G. Grehan. Sur la generalization de la théorie de Lorenz-Mie. *Journal of Optics*, 13:97-103, 1982.

[Grie03] D. G. Grier. A revolution in optical manipulation. *Nature***,** 424(6950):810-816, 2003.

[Hard90] R. L. Hardy. Theory and applications of the multiquadric-biharmonic method. *Computers and Mathematics with Applications*, 19(8-9):163-208, 1990.

[Hsu02] D. Hsu, R. Kindel, J. C. Latombe, and S. Rock. Randomized Kinodynamic Motion Planning with Moving Obstacles. *The International Journal of Robotics Research*, 21(3):233-255, 2002.

[Juba07] D. Juba, and A. Varshney. Modelling and rendering large volume data with Gaussian radial basis functions. Technical Report, Number UMIACS-TR-2007-22, University of Maryland, College Park, MD, 2007.

[Kane06] J. M. O'Kane, B. Tovar, P. Cheng, and S. M. LaValle. Algorithms for planning under uncertainty in prediction and sensing. In *Autonomous Mobile Robots: Sensing, Control, Decision-Making, and Applications*. Marcel Dekker, 2006.

[Kavr96] L. E. Kavraki, P. Svestka, J. C. Latombe, and M. H. Overmars. Probabilistic Roadmaps for Path Planning in High-Dimensional Configuration Spaces. *IEEE Transactions on Robotics and Automation*, 12(4):566-580, 1996.

[Kavr98] L. E. Kavraki, and J. C. Latombe. Probabilistic Roadmaps for Robot Path Planning. In *Practical Motion Planning in Robotics: Current Approaches and Future Challenges*. West Sussex, UK: John Wiley, 1998.

[Koni95] K. Konig, H. Liang, M. W. Berns, and B. J. Tromberg. Cell-Damage by Near-IR Microbeams. *Nature,* 377(6544): 20-21, 1995.

[Koni96] K. Konig, H. Liang, M. W. Berns, and B. J. Tromberg. Cell damage in near-infrared multimode optical traps as a result of multiphoton absorption. *Optics Letters*, 21: 1090-1092, 1996.

[Kim06] T.H. Kim, S.Y. Lee, N.K. Cho, H.K. Seong, H.J. Choi, S.W. Jung, and S.K. Lee. Dielectrophoretic alignment of gallium nitride nanowires (GaN NWs) for use in device applications. *Nanotechnology*, 17(14):3394-3399, 2006.

[Lato91] J. C. Latombe, *Robot Motion Planning*. Boston, MA: Kluwer Academic Publishers, 1991.

[LaVa97] S. M. LaValle, and R. Sharma. On motion planning in changing, partially predictable environments. *The International Journal of Robotics Research*, 16(6):775-805, 1997.

[LaVa98] S. M. LaValle and S. A. Hutchinson. An objective-based framework for motion planning under sensing and control uncertainties. *The International Journal of Robotics Research*, 17(1):19-42, 1998.

[LaVa00] S. M. LaValle. Robot motion planning: A game-theoretic foundation. *Algorithmica*, 26(3):430-465, 2000.

[LaVa06] S. M. LaValle. *Planning Algorithms*. New York, NY: Cambridge University Press, 2006.

[Leve02] P. Leven, and S. Hutchinson. A Framework for Real-Time Path Planning in Changing Environments. *The International Journal of Robotics Research*, 21(12):999-1030, 2002.

[Li02] H. Li, and R. Bashir. Dielectrophoretic separation and manipulation of live and heat-treated cells of Listeria on microfabricated devices with interdigitated electrodes. *Sensors and Actuators B*, 86(2-3):215-221, 2002.

[Lim07] D. Lim, Y. S. Ong, Y. Jin, and B. Sendhoff. A study of metamodeling techniques, ensembles, and multi-surrogates in evolutionary computation. In *Proceedings of the Genetic and Evolutionary Computation Conference*, London, England, 2007.

[Liu96] Y. Liu, G. J. Sonek, M. W. Berns, and B. J. Tromberg. Physiological monitoring of optically trapped cells: Assessing the effects of confinement by 1064-nm laser tweezers using microfluorometry. *Biophysical Journal,* 71(4):2158-2167, 1996.

[Liu06] S. Q. Liu, S. K. Ong, Y. P. Chen, and A. Y. C. Nee. Real-time, dynamic level-of-detail management for three-axis NC milling simulation. *Computer-Aided Design*, 38(4):378-391, 2006.

[Lume97] V. J. Lumelsky, and K. R. Harinarayan. Decentralized motion planning for multiple mobile robots: The cocktail party model. *Autonomous Robots*, 4(1):121-135, 1997.

[Lume06] V. J. Lumelsky. *Sensing, Intelligence, Motion*. Hoboken, NJ: John Wiley & Sons, Inc., 2006.

[Mase07] E. Masehian, and D. Sedighizadeh. Classic and Heuristic Approaches in Robot Motion Planning – A Chronological Review. *Proceedings of World Academy of Science, Engineering and Technology*, 23:101-106, 2007.

[Meye04] E. Meyer, H. J. Hug, R. Bennewitz. *Scanning Probe Microscopy: The Lab on a Tip*. Springer-Verlag, Berlin, Heidelburg, Germany, 2004.

[Miss06] P. E. Missiuro, and N. Roy. Adapting Probabilistic Roadmaps to Handle Uncertain Maps. In *Proceedings of the 2006 IEEE International Conference on Robotics and Automation*, Orlando, FL, 2006.

[Moka07] B. Mokaberi, J. Yun, M. Wang, and A. Requicha. Automated Nanomanipulation with Atomic Force Microscopes. In *IEEE International Conference on Robotics and Automation*, Rome, Italy, 2007.

[Mont94] D. C. Montgomery, and G. C. Runger. *Applied Statistics and Probability for Engineers*. John Wiley & Sons, Inc., New York, NY, 1994.

[Mull05] A. A. Mullur and A. Messac. Extended radial basis functions: more flexible and effective metamodeling. *AIAA Journal*, 43(6):1306-1315, 2005.

[Neum99] K. C. Neuman, E. H. Chadd, G. F. Liou, K. Bergman, and S. M. Block. Characterization of photodamage to Escherichia coli in optical traps. *Biophysical Journal*, 77(5): 2856-2863, 1999.

[Neve06] A. A. R. Neves, A. Fontes, L. Y. Pozzo, A. A. Thomaz, E. Chillce, E. Rodriguez, L. C. Barbosa, and C. L. Cesar. Electromagnetic forces for an arbitrary optical trapping of a spherical dielectric. *Optics Express*, 14(26): 13101-13106, 2006.

[Niem03] T. A. Nieminen, H. Rubinsztein-Dunlop, and N. R. Heckenberg. Calculation of the *T*-matrix: general considerations and application of the point-matching method. *Journal of Quantitative Spectroscopy & Radiative Transfer*, 79-80:1019-1029, 2003.

[Niem07] T. A. Nieminen, V. L. Y. Loke, A. B. Stilgoe, G. Knöner, A. M. Brańczyk, N. R. Heckenberg, and H. Rubinsztein-Dunlop. Optical tweezers computational toolbox. *Journal of Optics A: Pure and Applied Optics*, 9, S196-S203, 2007.

[Optt07] http://www.physics.uq.edu.au/people/nieminen/software.html

[Ong08] S. E. Ong, S. Zhang, H J. Du and Y. Q. Fu. Fundamental principles and applications of microfluidic systems. *Frontiers in Bioscience*, 13, 2757-2773, 2008.

[Ouya08] P. R. Ouyang, R. C. Tjiptoprodjo, W. J. Zhang, and G. S. Yang. Micro-motion devices technology: The state of arts review. *The International Journal of Advanced Manufacturing Technology*, 38(5-6):463-478, 2008.

[Ozka03] M. Ozkan, T. Pisanic, J. Scheel, C. Barlow, S. Esener, and S. N. Bhatia. Electro-optical platform for the manipulation of live cells. *Langmuir*, 19(5):1532-1538, 2003.

[Pate05] L. Paterson, E. Papagiakoumou, G. Mile, V. Garces-Chavez, S. A. Tatarkova, W. Sibbbett, F. J. Gunn-Moore, P. E. Bryant, A. C. Riches, and K. Dholakia. Light-induced cell separation in a tailored optical landscape. *Applied Physics Letters*, 87: 123901-1-3, 2005.

[Pate07] L. Paterson, E. Papagiakoumou, G. M. V. Garces-Chavez, T. Briscoe, W. Sibbett, K. Dholakia, and A. C. Riches. Passive optical separation within a 'nondiffracting' light beam, *Journal of Biomedical Optics*, 12(5):54017-1-13, 2007.

[Peas08] M. Peasgood, C. M. Clark, and J. McPhee. A Complete and Sacalable Strategy for Coordinating Multiple Robots Within Roadmaps. *IEEE Transactions on Robotics*, 24(2):283-292, 2008.

[Peng05] J. Peng, and S. Akella. Coordinating multiple robots with kinodynamic constraints along specified paths. *International Journal of Robotics Research,* 24(4): 295-310, 2005.

[Peng07a] T. Peng, A. Balijepalli, S. K. Gupta, and T. LeBrun. Algorithms for on-line monitoring of micro-spheres in an optical tweezers-based assembly cell. *ASME Journal of Computing and Information Science in Engineering*, 7(4):330-338, 2007.

[Peng07b] T. Peng, A. Balijepalli, S. K. Gupta, and T. W. LeBrun. Algorithms for extraction of nanowire attributes from optical section microscopy images. *ASME Computers and Information in Engineering Conference*, Las Vegas, NV, 2007.

[Pere83] T. Lozano-Perez. Spatial planning: a configuration space approach. *IEEE Transactions on Computers*, 32(3):108-120, 1983.

[Plew04] J. Plewa, E. Tanner, D. Mueth, and D. Grier. Processing carbon nanotubes with holographic optical tweezers. *Optics Express*, 12(9):1978-1981, 2004.

[Pons07] A. J. Pons, A. Karma, S. Akamatsu, M. Newey, A. Pomeranc, H. Singer, and W. Losert. Feedback control of unstable cellular solidification fronts. *Physical Review E*, 75(2):021602-1-021602-5, 2007.

[Pren07] S. Prentice, and N. Roy. The Belief Roadmap: Efficient Planning in Linear POMDPs by Factoring the Covariance. In *Proceedings of the International Symposium on Robotics Research*, Hiroshima, Japan, 2007.

[Rodr07] S. Rodriguez, J. M. Lien, and N. M. Amato. A Framework for Planning Motion in Environments with Moving Obstacles. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, San Diego, CA, 2007.

[Rohr01] A. Rohrbach, and E. H. K. Stelzer. Optical trapping of dielectric particles in arbitrary fields. *Journal of Optical Society of America A*, 18(4):839-853, 2001.

[Rohr02] A. Rohrbach, and E. H. K. Stelzer. Trapping forces, force constants, and potential depths for dielectric spheres in the presence of spherical aberrations. *Applied Optics*, 41(13):2494-2507, 2002.

[Rohr05] A. Rohrbach. Stiffness of optical traps: quantitative agreement between experiment and electromagnetic theory. *Physical Review Letters*, 95(16):168102-1-168102-4, 2005.

[Sadu08] L. Sadum. *Applied Linear Algebra: The Decoupling Principle*. American Mathematical Society, Providence, RI, USA, 2008.

[Sanc05] A. J. Sanchez-Salmeron, R. Lopez-Tarazon, R. Guzman-Diana, and C. Ricolfe-Viala. Recent development of micro-handling systems for micro-manufacturing. *Journal of Materials Processing Technology*, 167:499-507, 2005.

[Schw86] J. Schwartz, J. Hopcroft, and M. Sharir, Eds. *Planning, Geometry, and Complexity. Robot Motion Aspects*, Norwood, NJ: Ablex Publishing Corporation, 1986.

[Sett01] M. Setti. Survey of Nanomanipulation Systems. In *Proceedings of the 1st IEEE Conference on Nanotechnology*, Maui, Hawaii, 2001, pp. 75-80.

[Sinc04] G. Sinclair, P. Jordan, J. Courtial, M. Padgett, J. Cooper, and Z. J. Laczik. Assembly of 3-dimensional structures using programmable holographic optical tweezers. *Optics Express*, 12(22):5475-5480, 2004.

[Sime02] T. Simeon, S. Leroy, and J.-P. Laumond. Path coordination for multiple mobile robots: A resolution complete algorithm. *IEEE Transactions on Robotics and Automation*, 18(1): 42-49, 2002.

[Simp02] T. W. Simpson, A. J. Booker, D. Ghosh, A. A. Giunta, P. N. Koch, and R. J. Yang. Approximation methods in multidisciplinary analysis and optimization: a panel discussion. *9th AIAA/ISSMO Symposium on Multidisciplinary Analysis & Optimization*, Atlanta, GA, 2002.

[Sud07a] A. Sud, E. Andersen, S. Curtis, M. Lin, and D. Manocha. Real-Time Path Planning for Virtual Agents in Dynamic Environment. In *Proceedings of the 2007 IEEE Virtual Reality Conference*, Charlotte, NC, 2007.

[Sud07b] A. Sud, R. Gayle, E. Andersen, S. Guy, M. Lin, and D. Manocha. Real-time Navigation of Independent Agents Using Adaptive Roadmaps. *In Proceedings of the*

ACM Symposium on Virtual Reality Software and Technology, Newport Beach, CA, 2007.

[Sun01] C. K. Sun, Y. C. Huang, P. C. Cheng, H. C. Liu, and B. L. Lin. Cell manipulation by use of diamond microparticles as handles of optical tweezers. *Journal of the Optical Society of America B*, 18(10):1483-1489, 2001.

[Sves98] P. Svestka, and M. Overmars. Coordinated path planning for multiple robots. *Robotics and Autonomous Systems*, 23:125-152, 1998.

[Svob94a] K. Svoboda, and S. Block. Optical trapping of metallic rayleigh particles. *Optics Letters*, 19(13):930-932, 1994.

[Svob94b] K. Svoboda, and S. M. Block. Biological Applications of Optical Forces. *Annual Review of Biophysics and Biomolecular Structure*, 23: 247-285, 1994.

[Swop82] W. C. Swope, H. C. Andersen, P. H. Berens, and K. R. Wilson. Computer simulation method for the calculation of equilibrium constants for the formation of physical clusters of molecules: application to small water clusters. *The Journal of Chemical Physics*, 76(1):637-49, 1982.

[Thru05] S. Thrun, W. Burgard, and D. Fox. *Probabilistic Robotics*. Cambridge, MA: The MIT Press, 2005.

[Turk02] Turk, G. O'Brien, J. F. Modelling with implicit surfaces that interpolate. *ACM Transactions on Graphics*. 21(2):855-873, 2002.

[Ukit06] H. Ukita, T. Saitoh, and N. Sakahara. Resolving Discrepancy between Theoretical and Experimental Optical Trapping Forces Using Effects of Beam Waist and Trapping Position Displacement due to Gravity. *Optical Review*, 13(6): 436-442, 2006.

[Vin06] L. J. D. Vin, A. H. C. Ng, J. Oscarsson, and S. F. Andler. Information fusion for simulation based decision support in manufacturing. *Robotics and Computer-Integrated Manufacturing*, 22(5-6):429-436, 2006.

[Wang07] G. G. Wang, and S. Shan. Review of metamodeling techniques in support of engineering design optimization. *ASME Journal of Mechanical Design*, 129(4):370-380, 2007.

[Wei96] P. Wei, and M. Y. Wang. Parametric structural shape and topology optimization method with radial basis functions and level-set method. In *Proceedings of the ASME Design Engineering Technical Conferences & Computers and Information in Engineering Conference,* Philadelphia, PA, 2006.

[Weil05] M. Weiler, R. Botchen, S. Stegmaier, T. Ertl, Y. Jang, D. S. Ebert, and K. P. Gaither. Hardware-assisted feature analysis and visualization of procedurally encoded multifield volumetric data. *IEEE Computer Graphics and Application*s, 25(5):72-81, 2005.

[Weis89] M. Weissbluth. *Photon-Atom Interactions*. Academic Press, Boston, MA, 1989.

[Wrig93] W. H. Wright, and G. J. Sonek. Radiation trapping forces on microspheres with optical tweezers. *Applied Physics Letters*, 63(6):715-717, 1993.

[Wrig94] W. H. Wright, G. J. Sonek, and M. W. Berns. Parametric study of the forces on microspheres held by optical tweezers. *Applied Optics*, 33(9):1735-1748, 1994.

[Wu96] Y. Wu. An Obstacle-based Probabilistic Roadmap Method for Path Planning. M.S. Thesis, Texas A & M University, 1996.

[Wu05] X. J. Wu, M. Yu, and W. Q. Xia. Implicit fitting and smoothing using radial basis functions with partition of unity. In *Proceedings of the 9th International Conference on Computer Aided Design and Computer Graphics*, Hong Kong, 2005.

[Wu06] X. J. Wu, M. Yu, and Q. Xia. 3D reconstruction methods based on radial basis functions for laser scanned data point sets. *Computer-Aided Design & Applications*, 3(1-4):145-153, 2006.

[Yan06] P. Yan, M. Ding, and C. Zheng. Mission-Adaptable Route Planning in Uncertain and Adversial Environment. 15(5):803-821, 2006.

[Yi06] C. Q. Yi, C. W. Li, S. L. Ji and M. S. Yang. Microfluidics technology for manipulation and analysis of biological cells. *Analytica Chimica Acta*, 560(1-2):1-23, 2006.

[Yu04] T. Yu, F. Cheong, and C. Sow. The manipulation and assembly of CuO nanorods with line optical tweezers. *Nanotechnology*, 15(12):1732-1736, 2004.

[Zuck07] M. Zucker, J. Kuffner, and M. Branicky. Multipartite RRTs for Rapid Replanning in Dynamic Environments. In *Proceedings of the IEEE International Conference on Robotics and Automation*, Rome, Italy, 2007.