HIGH THROUGHPUT PHOTOPATTERNING AND INTERACTIVE MANIPULATION OF MICROPARTICLES AND MICROORGANISMS

by

Christopher Oliver

A dissertation submitted in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
(Mechanical Engineering)
in The University of Michigan
2014

Doctoral Committee:

Associate Professor A. John Hart, Massachusetts Institute of Technology, Co-Chair

Associate Professor Nikolaos Chronis, Co-Chair

Assistant Professor Allen P. Liu, Member

Associate Professor Edwin Olson

DEDICATION

This edition of my dissertation is dedicated to Lauren Oliver my wife. I have never met someone who so single mindedly supports another.

ACKNOWLEDGEMENTS

Little of what was accomplished in this work was done in a vacuum and as such I'd like to acknowledge all of the contributors and colleagues that made this work possible.

- I of course thank my committee for taking the time to mentor and polish my thinking. Prof. Chronis, Prof. Liu, Prof. Olson and my advisor Prof. Hart.
- I would like to acknowledge all the members of the Mechanosynthesis group at the University of Michigan and Massachusetts Institute of Technology.
- Members who require additional thanks include John Lewandowski, and Dr. Erik Polsen for their work on simulating particles and designing the roll-to-roll system.
- I was fortunate to work with many undergraduate students. I thank, Levon Cimonian, Lilly Chen, Ilias Anagnostopoulos-Politis, Felix Sun and Malena Ohl. Special consideration goes to Adam Stevens while working robotic lithography.
- Dr. Eleni Gourgou and Dr. Daphne Bazopoulou contributed much needed knowledge on *C. elegans* and the issues faced by researchers in their field. I appreciate their experimentation and writing contributions.
- I thank Prof. Bhatia, Arnout Schepps and Dr. Kelly Stevens for their knowledge in the design and manufacture of liver tissue. As well for supplying hepatocytes necessary for this work. Further, we thank Prof. Tim Swager and Dr. Elizabeth Sterner for assistance with ATR-FTIR measurements.
- Electron microscopy was performed at the University of Michigan Electron Microbeam Analysis Laboratory (EMAL) and at the Center for Nanoscale Systems, Harvard University.
- I was financially supported by the Microfluidics in Biomedical Sciences Training Program through the National Institute of Health (5T32-EB005582) and by the Air Force Office of Scientific Research Young Investigator Program (FA9550-11-1-0089), and a Mechanical Engineering Department Fellowship from the University of Michigan.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF APPENDICES

ABSTRACT

HIGH THROUGHPUT PHOTOPATTERNING AND INTERACTIVE MANIPULATION OF MICROPARTICLES AND MICROORGANISMS

By
Christopher Ryan Oliver

Chair: Professor John Hart

Recent advances in soft material microfabrication technologies are enabling wide-ranging studies of cellular and organism behavior in vitro; however, these methods are generally time-consuming, challenging to implement by non-experts, are limited to planar features, and cannot be reconfigured within live environments. As a result, it is not possible to manufacture realistic artificial tissue constructs, nor to perform dynamic experimentation with model organisms.

This thesis describes an integrated hardware and software platform, based on micro-scale light shaping and high-speed machine vision algorithms that enables real-time, dynamic photo-patterning in response to microscale environmental changes. An optofluidic lithography system designed for the purpose of in-flow polymerization of hydrogel microstructures achieved diffraction limited resolution (r = 0.7μm) with a maximum distortion of the projection of 160nm. This enables continuous production of poly(ethylene-glycol) diacrylate(PEG-DA) microparticles (20-100μm, CoV5-15%). A new pillared microfluidic device design increased throughput up to 1500-fold, capable of synthesizing 2.5×10^6 particles per minute.

Biocompatibility of hydrogels was validated for model organism *C. elegans,* and hepatocytes. Dynamic assays where structures were built during live culture affirm that proximity of pillared structures increased the swimming speed of *C. elegans* and showed that worm behavior can be influenced by sequential photopatterning of free-floating structures.

A software architecture was designed to enable use of machine vision to in flow, by photopolymer encapsulation in response to image-based decision events. We then evaluated the sensitivity, specificity, RMSE and computational time of candidate machine vision algorithms, and find the Speeded Up Robust Feature (SURF) method was the most robust though Thresholding was 3 orders of magnitude faster than SURF. Using this capability, we sorted poly(styrene) micro particles by size via selective encapsulation (TPR=100% and SPC=99.999%, Mean error 4.7 pixels); and print patterns of hepatocyte aggregates with single cell resolution (<20μm) onto polymer substrates.

Last, the thesis describes the design and testing of a six-axis robotic dynamic lithography system for patterning large area curved surfaces. Looking forward, platforms combining micro- and nanofabrication processes with image-driven artificial intelligence algorithms could widely expand capabilities for scalable biofabrication and automation of science, including for custom fabrication of cell-based assays and in vitro organ mimics.

CHAPTER I. INTRODUCTION

Motivation

The goal of my doctoral research (Figure 1) has been to converge expertise in laboratory robotics and automation by using advanced lithography to understand how to manipulate and assemble micro-scale building blocks as a tool for bioengineering research.

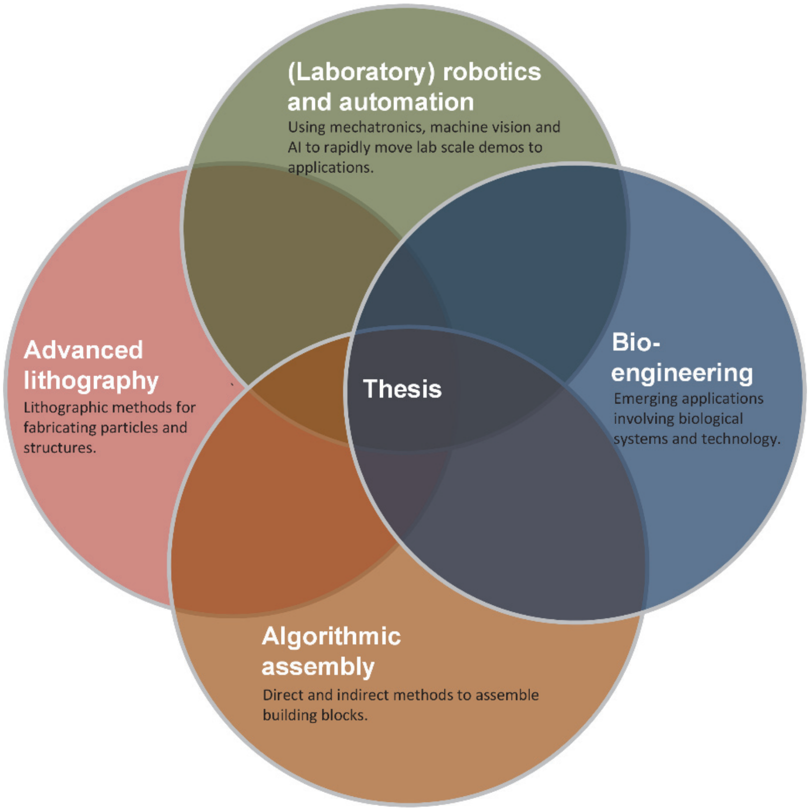

(Laboratory) robotics and automation
Using mechatronics, machine vision and AI to rapidly move lab scale demos to applications.
Advanced lithography
Lithographic methods for fabricating particles and structures.
Thesis
Bio-engineering
Emerging applications involving biological systems and technology.
Algorithmic assembly
Direct and indirect methods to assemble building blocks.

Figure 1. Intersection of fields and research interests converging in this thesis.

Study of many biological systems using *in vitro* models requires the assembly of microscale building blocks, such as biological cells or polymer microstructure in two- and three-dimensional systems. For example in tissue engineering, to create an organ on a chip the liver can be discretized into building blocks [1]. Then the building blocks can then be arranged onto a substrate to be cultured into functional tissue-like material (Figure 2).

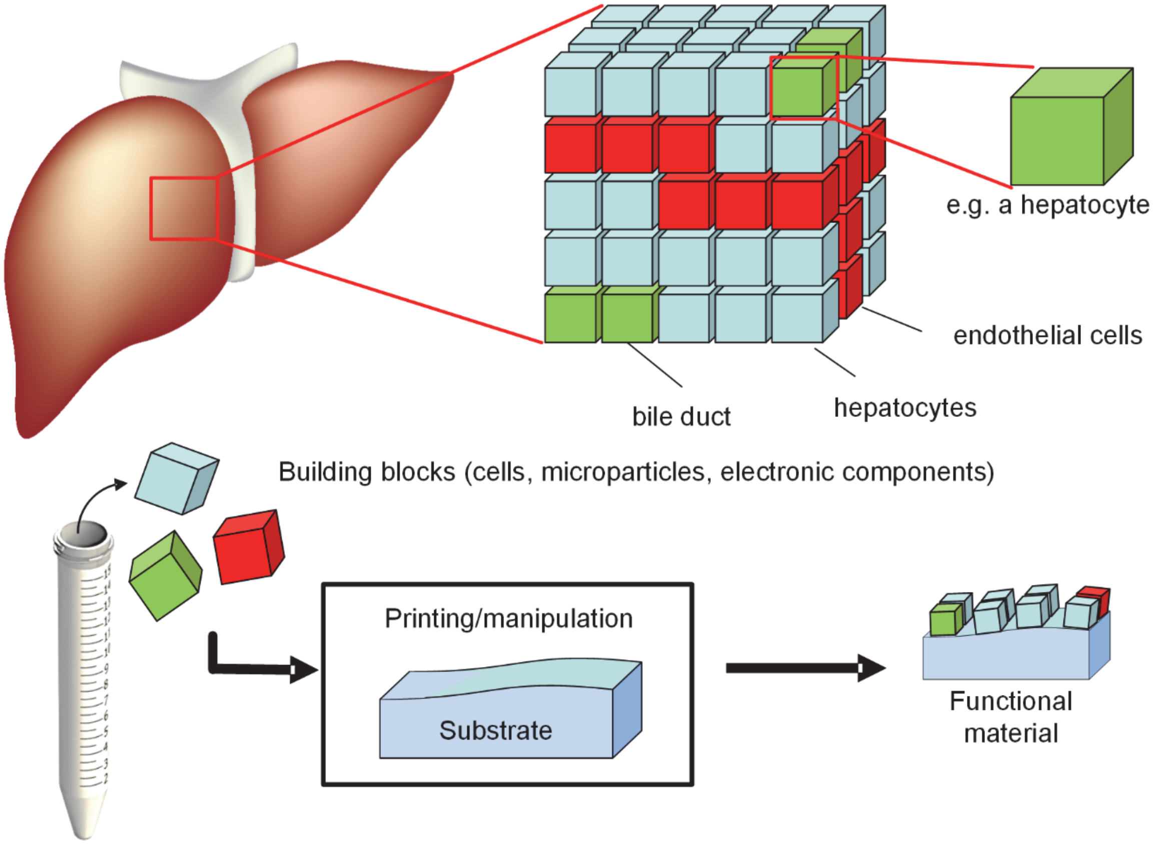


Figure 2. Motivational concept of organizing building blocks into functional systems such as in tissue engineering.

One method of organizing cells onto a substrate is to pattern topographic features that capture cells onto a device using soft lithography. Microfabrication of soft materials was introduced by George Whitesides when he introduced the idea of "soft lithography" in which PDMS is cast from a mold into microstructures such as channels and pillars [2], [3]. This method requires the fabrication of a master mold and subsequent casting and fabrication of a device over several days. This has been employed by Bhatia et al. to capture and localize cell aggregates (cells cultured in spherical polymer particles) [4]. Moreover, while soft lithography has been robustly applied to address many biologically relevant problems it requires expert knowledge and is not reconfigurable within live environments. Therefore, it is highly desirable to have a template-free method of organizing cell-scale building blocks, with arbitrary configuration, in 2D and 3D. This enables rapid testing of arrangements of cells and helper cells to find optimal configurations that function similar to the original tissue.

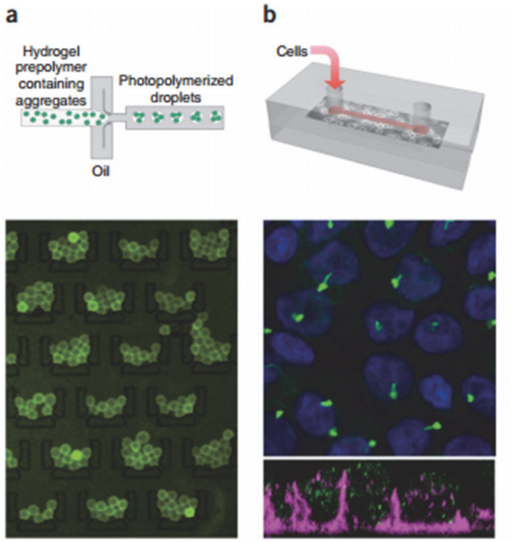

a
Hydrogel prepolymer containing aggregates
Photopolymerized droplets
Oil
b
Cells

Figure 3. An example of soft lithographic *fabrication of organ-on-a-chip. a) Fabrication of cell aggregates using photopolymerized droplets in a microfluidic device. b) Localization of cells by features patterned onto a soft lithographic microfluidic device [4]. copyright 2014 Nature Publishing Group.*

In order to enable assembly of arbitrary building blocks, a method is needed to synthesize the building blocks at scale. Inspiration is provided by the seminal work of Patrick Doyle who has developed a method of lithographically patterning microparticles inside of a microchannel using a mask. The mechanics of this process are discussed in 0, but the use of light to arbitrarily control the shape of synthesized microparticles and to encapsulate cells inspires the idea of using light do manipulate them.

The microfluidic lithography technique has been extended to direct the motion of particles to pre-planned locations via rails in the fluidic channel. Kwon et al. patterned rails into the microfluidic device that accept a fin on top of a pre-fabricated particle to direct a collection of them into grid patterns where the particles interlock due to 2D features on their surface. This form of bottom up construction is inspirational for the work in this thesis, because to meet the grand challenge one the goals is to organize cell laden particles into 2D patterns.

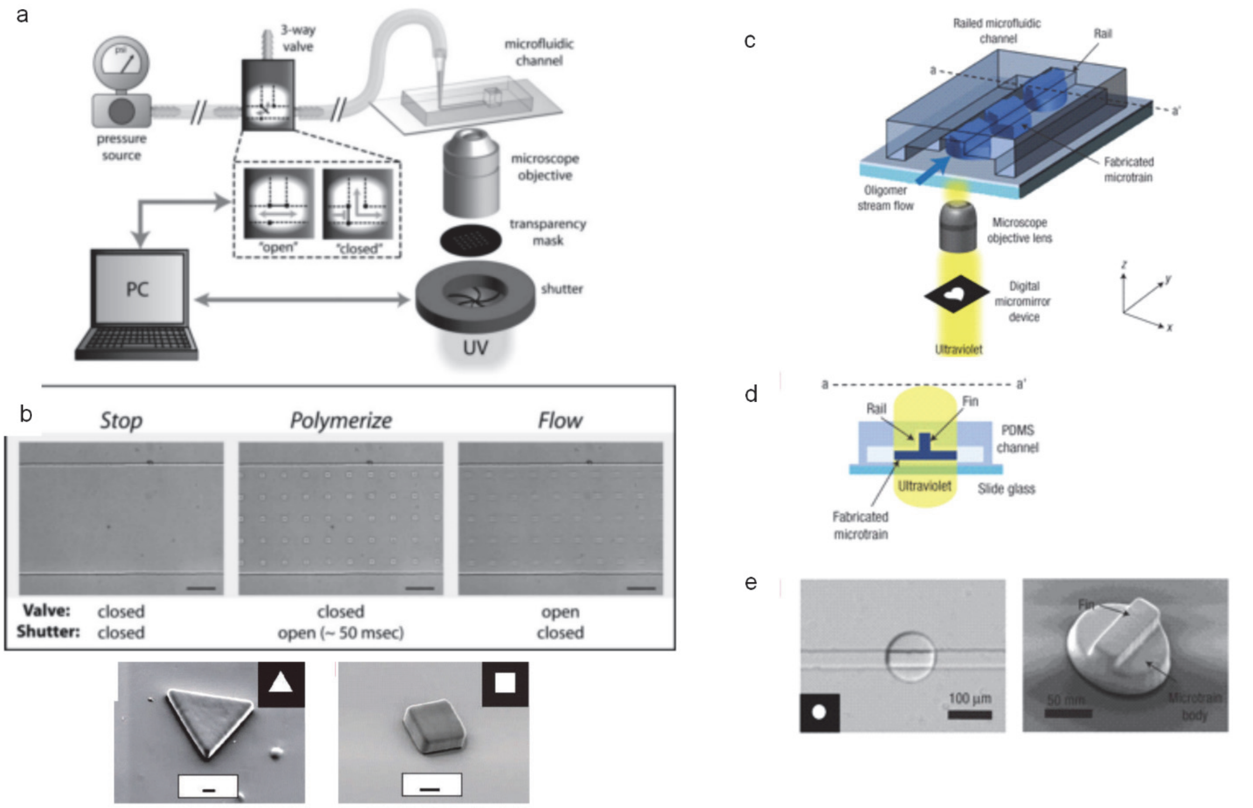


Figure 4. Microfluidic lithography prior work. *a) Microfluidic flow lithography as published by Doyle and colleagues showing fabrication of microparticles inside a fluidic channel when a photopolymer is exposed to light. b) The process of stopping the flow when the shutter is opened and flowing finished particles out of the device. c) Schematic of railed lithography as described by Kwon et al. d) Side view of the channel showing the rail and particle fin. e) Micrograph and SEM of particle in the device and illustrating the fin that aligns with the rail.*
copyright 2007 Royal Society of Chemistry. copyright 2008 Nature Publishing Group.

However, railed lithography is unable to select an arbitrary particle and relies on some other method of buffering particles to be arranged. By comparison macro manufacturing in which pick and place systems use machine vision to localize and manipulate products such as pancakes offer more flexibility and have been adapted to meet microscale positioning requirements. An example system using a flexure picker to organize microbeads was demonstrated by Chen et al. with micron level accuracy [5]. However, pick-and-place methods do not scale favorably to the micro-scale because the machine can only manipulate one particle at a time limiting the throughput. In addition, a mechanical gripper is not well suited to manipulating biological components that are easily damaged by shear forces and once gripped will tend to stick to the end effector rather than release onto the substrate because of their "sticky" surface.

The combination of the motivation we've discussed and the inspirational work in various fields has led us to consider the shaping and steering of light driven by computer aided decision making and machine vision to address the challenge and other relevant applications. Texas Instruments manufactures a device called a Digital Light Processing (DLP) or Digital Micromirror Device (DMD) that controls the orientation of micromirrors to shape light into structured patterns. It is the same technology found in commonly used office projectors to generate an image. We will discuss the design and structure of this device in more detail later in this work. Here, consider simply that it enables the ability to control thousands of beams of light and direct them according to real time feedback from an image processing unit. In the same way multiple robots can be placed on an assembly line we imagined using the light beams as end effectors to photopolymerize on demand. Moreover, the wavelengths in the electromagnetic spectrum enable the concept of light as a manipulator high flexibility in manufacturing. By dividing the spectrum into ranges (i.e.$360\ nm \pm 30\ nm, 450\ nm \pm 30\ nm$) portion of the spectrum can be used for different function such as imaging, photopolymerization and heating.

Thesis outline

With this motivation in mind, the goals of this thesis are to:

- Design and build a modular maskless photopatterning system.
- Develop software and algorithms enabling adaptive and interactive photopatterning in microfluidic environments.
- Explore initial applications of dynamic photopatterning including interacting with *C. elegans in vivo*, sorting/encapsulating polymer microparticles and printing microparticles and hepatocyte cell aggregates into predefined shapes.
- Practice my expertise and interest in laboratory automation through the realization of a robotic lithography system to apply dynamic photopatterning to arbitrary curved surfaces.

The chapters are structured such to highlight each level of complexity and the underlying engineering challenges as the work approached the goal of dynamically manipulating particles, objects and organisms. This includes the application of the author's expertise in developing large scale automation to solve microscale problems.

Chapter 2 describes the design and construction of a maskless lithography system, and presents a detailed analysis of the optical design and system performance. The goal is to design a system that is diffraction limited (r = 0.7 μm) and can achieve on-demand patterning of photopolymers in fluidic channels or on substrates.

Chapter 3 discusses use of the maskless lithography system to manufacture of microparticles synthesized from Poly(ethylene) glycol diacrylate (PEG-DA) hydrogels, which are intended to be used for future work on self-assembling microparticles. A library of microparticles (20-100 μm) is synthesized with a polydispersity (CoV) between 5-15%. Moreover, a pillared microfluidic device is shown to achieve a 40-1500 fold increase in throughput and may be able to make 2.5×10^6 microparticles per minute.

Chapter 4 demonstrates application of the maskless lithography system to study the behavior of *C. elegans* worms in assays fabricated *in vivo*. In order to enable use of the system for in vivo photo-patterning, hydrogel biocompatibility and performance is evaluated and we search for a platform that can pattern directly onto agar. Assays are performed to study locomotion and decision making in response to rewards. Additional assays demonstrate the ability to fabricate moving parts and perform interactive experimentation using researcher feedback.

Chapter 5 presents algorithms and software enabling dynamic lithography to utilize computer aided decision making in response to changes in the microenvironment. Design problems for this system are modeled and discussed including the algorithm for detecting and generating masks to project onto particles, latency in communication between system components and control over projection accuracy. Then, we demonstrate the use of this to encapsulate/sort micro beads flowing in a fluidic channel. We then detail the variables that control throughput using this system based on Little's Law.

Chapter 6 improves the performance of dynamic lithography by quantitatively studying the performance of selected established detection and tracking algorithms. We quantify the detection and tracking root mean square error (RMSE), sensitivity, specificity and computational cost of each method. These metrics are collected against ground truth simulated videos that enable us to examine independently the properties common to microscopy images (blur, signal-to-noise, particle size, particle velocity, particle dynamics, particle complexity and others). This comprehensive approach gives insight into what parameters can be controlled, and for those we cannot control how detrimental they may be to performance. Further we break down the functions involved in the algorithm and report how computational cost changes with increasing particle density.

Chapter 7 applies dynamic lithography towards the grand challenge of flexible tissue manufacture, starting with hepatocyte aggregates as the printing "ink". First, biocompatibility of the hepatocytes to both substrate and photo-polymer is investigated. Then we design an algorithm to pattern cells and other objects. Finally, we demonstrate and characterize the performance of this new capability by printing arrangements of particles and multiple types of cells into predefined patterns. This is done towards organizing tissue like material.

Chapter 8 presents a robotic system for three-dimensional (3D) photopatterning. Fabrication of structures on the surface is achieved by positioning and focusing the projector using a six axis robot before exposure of the surface. The design of the system is presented and resolution of the projection system is measured. Finally, we demonstrate the ability to pattern a metal ball and discuss challenges caused by error propagation in the motion system.

CHAPTER II. A MODULAR MASKLESS LITHOGRAPHY SYSTEM

Abstract

The capability to fabricate polymer microparticles on demand, and to create direct-write microfeatures on surfaces, enables applications such as multiplexed diagnostics. This chapter presents the design and construction of a modular maskless lithography system. First, the system was designed to accommodate a fixed mask; then, it was adapted to a maskless configuration using a Digital Light Processing chip (DLP). The performance of the system is modeled and measured including the imaging, illumination and projection of the shaped light. The system is then used to fabricate structures from polyethylene glycol diacrylate (PEG-DA) on glass. The modularity of its configuration is exemplified by integrating a roll to roll system to pattern microstructures on copper tape moving under the system.

Introduction

The origin of the word "lithography" traces back to ancient Greece when it meant to "Write with a stone [6]." Modern photolithography may be better described as "Write with light," as modern methods have enabled the scale-up and continued miniaturization of semiconductor fabrication technology [7]. Photolithography is based on the concept of transferring a pattern onto a light sensitive polymer by projecting light through a mask. The light is shone through the mask which project the pattern and initiates polymerization. The polymer used in this process is a photo sensitive polymer, commonly called a photoresist.

Despite the versatility and scalability of modern photolithography it typically requires a fixed mask which makes the cost high for producing new patterns. This is a practical challenge to laboratories that are not looking for mass-production quantities.

In the early 2000's, researchers sought to improve the flexibility of lithography using direct write microscope methods. Love *et al.* have adapted a Zeiss microscope for patterning resists in the laboratory [8]. Then with the advent of spatial light modulators (DMD) [9] maskless lithography was implemented by Mei et al. to fabricate semiconductors and flat panel displays [10]. Many other methods have been developed, that are centered on the idea of how to best project light onto a photosensitive material (i.e. photoresist). Many of the methods are shown in Figure 5 [11]–[17].

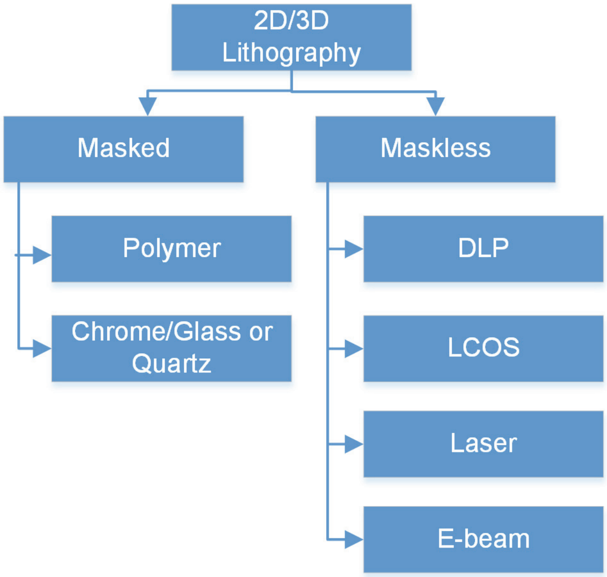


Figure 5. Tree of lithography technology. Masked based methods use polymers or glass type materials. Masked methods include DLP, Liquid crystal on Silicon (LCOS), laser and electron-beam (e-beam).

In this chapter, we discuss the design and construction of a modular maskless lithography system. Previous contributions in the field have built DLP based maskless lithography systems by modifying off-the-shelf microscopes [18] or designing the entire optical train and achieved 4 µm resolution. We sought to design a system that uses off the shelf components but has greater modularity than a modified microscope. We present the design principles to achieve diffraction limited, high resolution optical performance, and discuss tradeoffs in design that degrade performance. Aberrations in the system are modeled and when possible characterized. The system was first designed to accommodate a standard quartz mask then integration of the maskless components is introduced. We characterize the projection, illumination and imaging performance independently, and demonstrate the system fabricating microstructures in a hydrogel resist called Poly(ethylene glycol) diacrylate (PEG-DA) on glass and copper with a resolution of 10 µm. To highlight the modularity of the design (various masks, camera, objectives, stages) we also combine it with a roll-to-roll reel of tape. The structures of PEG-DA pillars are patterned continuously along the tape using the roll-to-roll system.

The design process

It is composed of a mask/DMD, CCD (Charge-coupled device) camera sensor and a light source (Figure 6).

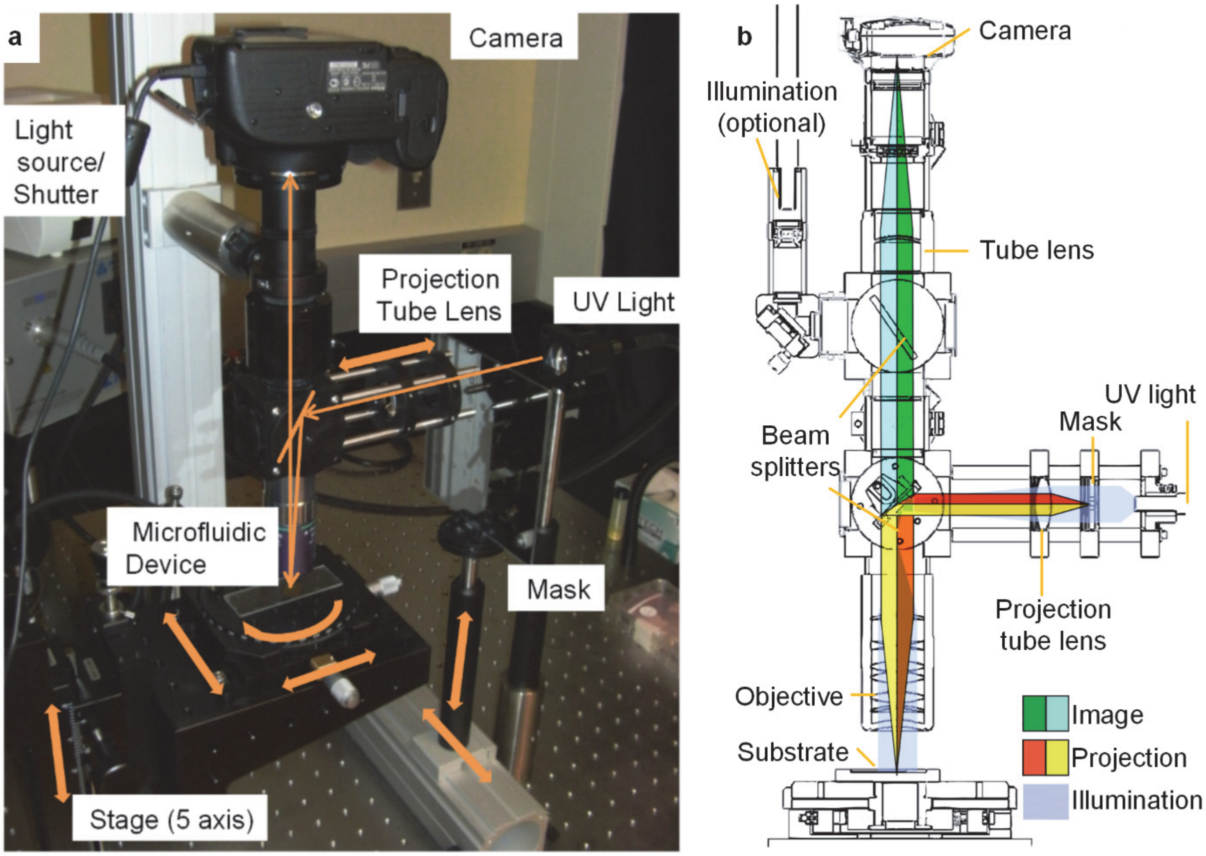


Figure 6. Photograph of the first masked version of the lithography system

This system performs three optical processes:

- Projection
- Imaging
- Illumination

The system must project the image formed by the mask accurately. At the same time, imaging the substrate is important to be able to focus the projection and observe changes in the photo-polymer. Finally, two forms of illumination play two very different roles. The first is to illuminate the substrate for the camera, while the second is to polymerize the resist. In both cases uniform, parallel beams are desired.

Design of the projection system

As shown in Figure 7 the paraxial layout establishes the focal lengths and overall size of the system. Here f_1 is the focal length of the objective, f_2 is the focal length of the tube lens and d is the distance between the objective and the tube lens.

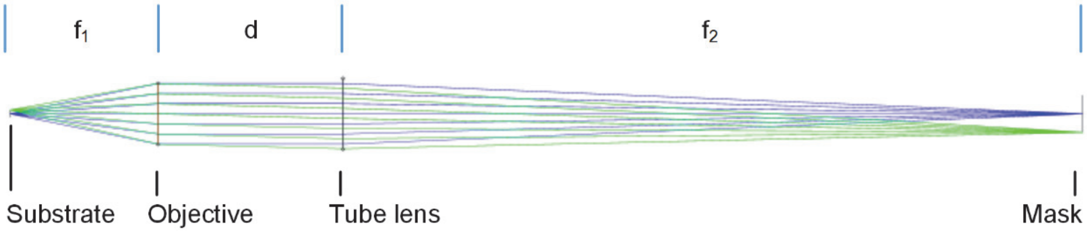


Figure 7. Paraxial design of the projection system, excluding the illumination. Schematic of designed system is shown in Figure 6.

We chose to use off the shelf optical components for accessibility and transferability to other laboratories. For that reason the system is built around Mitutoyo long working distance objectives to access a range of resolutions. These lenses have long working distances (>15 mm) making them ideal for microfluidic or roll to roll integration. Their specifications are shown in Table 1.

Table 1. Mitutoyo objectives and their imaging properties.

Objective (Magnification)	Focal length (mm)	Numerical aperture	Working distance (mm)	Depth of focus (µm)
5	40	0.17	34	14
20	10	0.42	20	1.6
50	4	0.55	13	0.9

The system is designed as an infinity corrected system, which means that the object space is focused to infinity. It is composed of two lens groups, the objective and tube lens, with a stop in between them and an entrance pupil controlled by the objective. They are arranged such that the objective is one focal length away from the substrate and the mask is one focal length away from the tube lens. In this case the objective must project an image from the mask onto the substrate, but the distance between the objective and tube lens is not related to the magnification or focus. The objective both images the light reflected from the substrate and projects the light from the tube lens. For infinity corrected systems the magnification of the system takes on a special relationship where magnification (M) is equal to the focal length of the tube lens (f_2) divided by the objective focal length (f_1).

$$M = \frac{f_2}{f_1} \qquad \text{Equation 1}$$

Constrained by the numerical aperture (NA) of the objective, we calculate the tube lens focal length necessary to achieve a specific mask size and field of view (FOV) projection. The matching focal length of the tube lens for this objective is 200 mm. However, we chose a lens with a focal length of 150 mm instead to limit illumination energy lost to the optical path.

This choice is also coupled to the projection resolution, as characterized by the modulation transfer function (MTF). The modulation transfer function is used to describe how well a lens system reproduces an image [19]. We calculate the quality by measuring the profile of the intensity across a patterned image; a bar pattern with decreasing spacing between adjacent bars. For any pair of bars along the profile the modulus of the intensity values, representing the contrast, can be calculated as:

$$\%Contrast = \frac{I_{max} - I_{min}}{I_{max} + I_{min}} x100 \qquad \text{Equation 2}$$

By relating contrast to the spacing between pairs of bars (line pair) in the image and object we calculate the MTF.

$$MTF = \frac{\%Contrast\ image}{\%Contrast\ object} \qquad \text{Equation 3}$$

Practically, we apply the MTF to identify a quantity called the line pair resolution. The line pair resolution is defined as the smallest line pair spacing that is visible in the projected image. Because the MTF describes contrast we chose an MTF value of 30% to describe a line pair that is fully resolved. This means that during exposure the space between two lines will receive 30% of the illumination intensity as is projected on the line. Therefore, we optimized the lens focal values and relative positions to achieve an MTF value of 30% when the line pair spatial resolution is 1 µm or less.

Next, we simulated the performance using the prescription of a real lens (Thorlabs, LBF254-150) rather than an ideal lens that was initially assumed. Using ZEMAX software we evaluated the negative effects such as a reduction in contrast, resolution and an increase in aberrations. Two fields are examined, 1) the on axis or zero field and 2) the edge of the sample 0.5 mm from center or the off axis/extended field. The effect on the MTF is shown in Figure 8b, where the zero field remains diffraction limited with a resolution of 700 nm but the extended field shows a resolution of 1.1 µm. This is because aberrations are introduced due to the geometry of the tube lens.

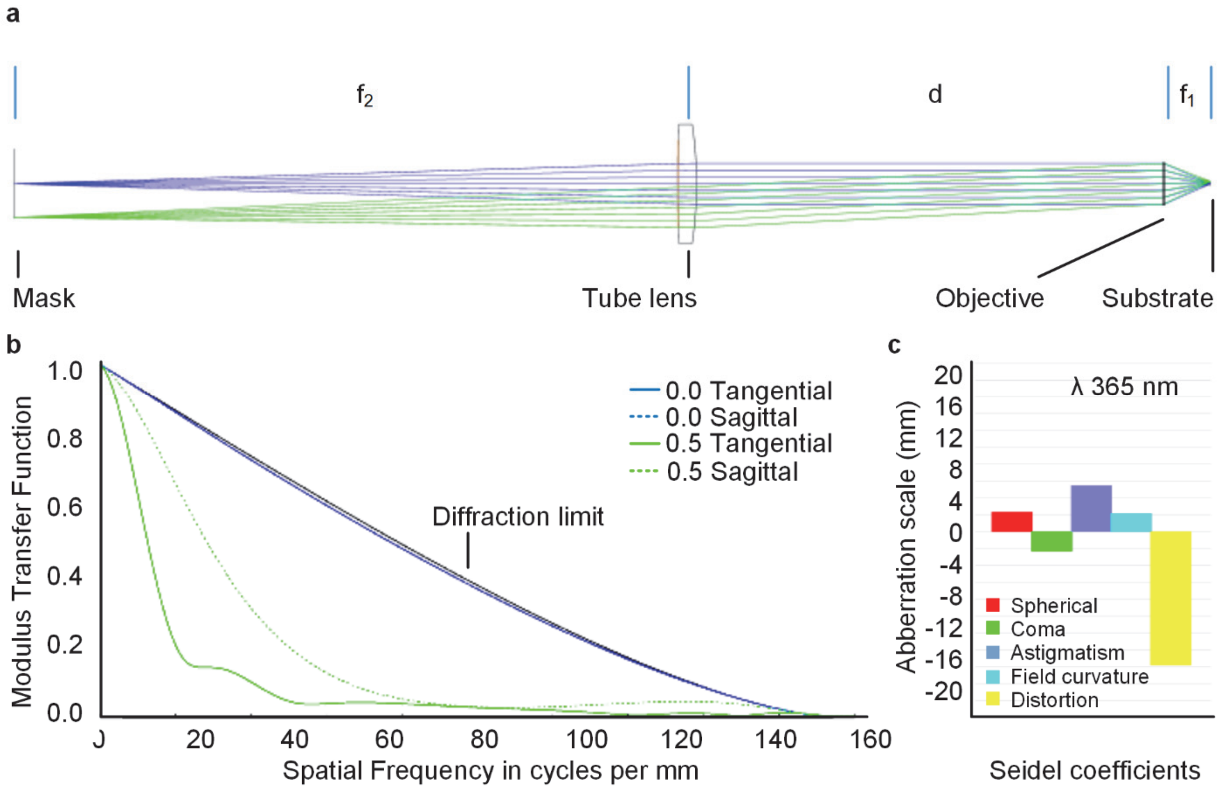


Figure 8. Modeling the projection system. a) Ray trace when using a real lens. b) The modulation transfer function (contrast between line pairs) compared to the spatial frequency of a sinusoidal input function in mm. The tangential and sagittal ray traces are shown for both fields. The diffraction limit is annotated. c) The Seidel aberrations coefficients as a sum of all surfaces in the system. A schematic of optical system is shown in Figure 6.

The common monochromatic aberrations are defocus, spherical aberration, coma, astigmatism, field curvature and distortion. These are described by the Seidel polynomial (Equation 4). Each term of $\mathrm{W}_{ijk}$is the wave front aberration coefficient in µm. H^i is the fractional image height, ranging between 0 and 1. ρ^j is the fractional pupil radius, which also ranges between 0 and 1. Finally, $cos^k\theta$ is the other pupil coordinate given by θ.

$$S = \sum_{i,j,k} W_{ijk} H^i \rho^j cos^k \theta \qquad \text{Equation 4}$$

From the Seidel polynomial the first five terms correspond to the aberrations in the image as shown in Table 2.

Table 2. Seidel coefficients for each aberration [19].

Aberration	Seidel coefficient
Defocus	$w_d = W_{020}\rho^2$
Spherical aberration	$w_s = W_{040}\rho^4$
Coma	$w_c = W_{131}H\rho^3 cos\theta$
Astigmatism	$w_d = W_{222}H^2\rho^2 cos^2\theta$
Field curvature	$w_{fc} = [W_{200}H^2]\rho^2$

The system suffers from several aberrations so we calculated the Seidel aberration coefficients as shown in Figure 8c to consider the relative contribution of each aberration to image quality. In order of descending coefficient value they are spherical aberration, coma, astigmatism, field curvature and distortion. Color based aberrations are avoided by the use of a single wavelength for polymerizing the photopolymer.

To confirm the effect of described by the Seidel coefficients ray tracing was used to visually observe the effect each aberration would have. The distortion for the off-axis field as is shown (exaggerated 10-fold) in Figure 9 indicates pincushion distortion is present with a maximum value of 1.19%. Figure 9b shows the transverse ray plot indicating how the spherical aberration is presented. At the far field 5 µm of spherical aberration occur, this can be reduced to 2.5 µm at the best mid-focus as marked. The spot diagrams shown in Figure 9c and Figure 9d show the effect of astigmatism and coma respectively. While astigmatism is present we also observe that for the zero field the system is diffraction limited as the spot is within the airy disc.

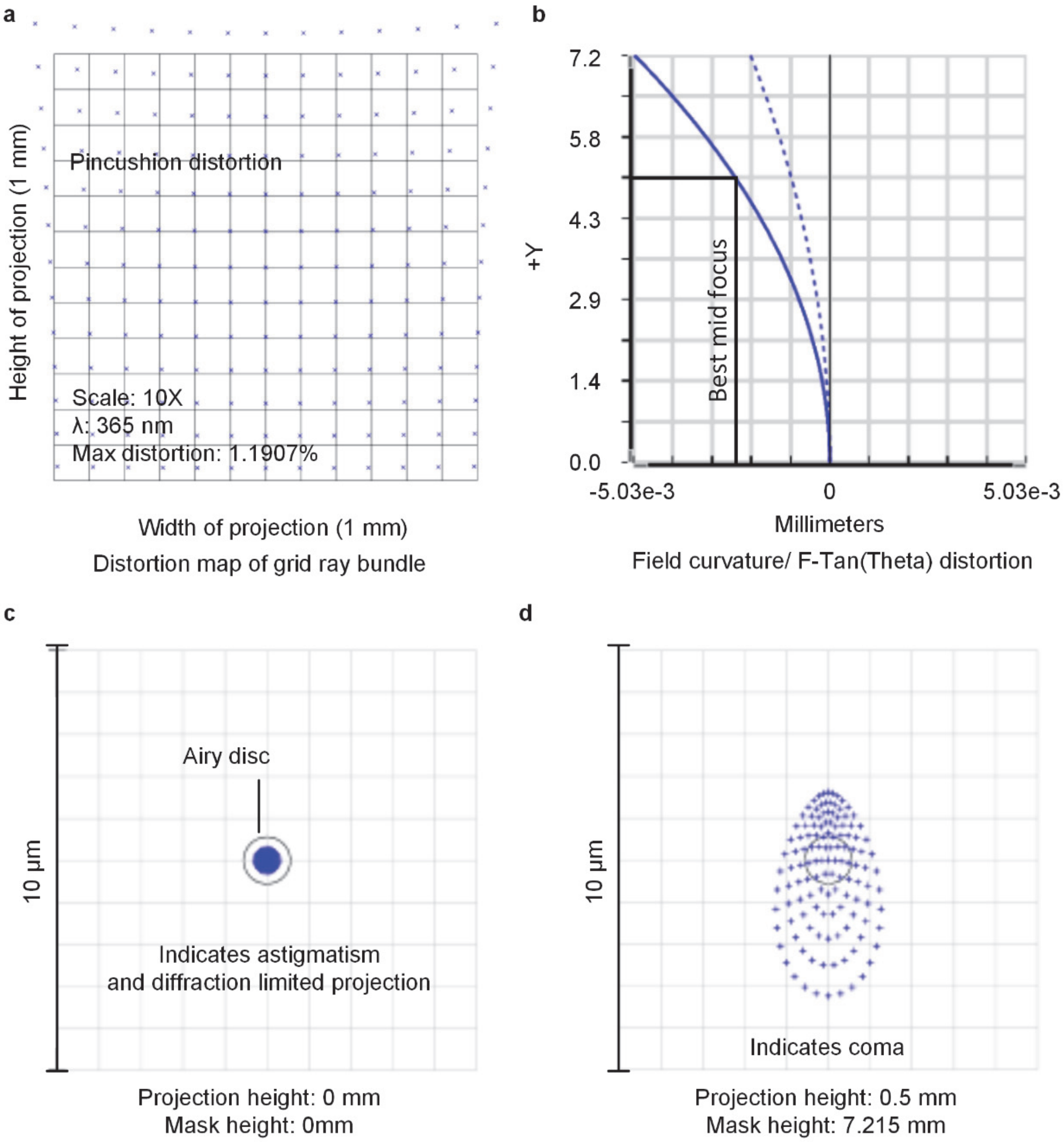


Figure 9. Ray trace plots of the projection system. a) Plot of ray distortion traveling through the system. b) Transverse ray plot. c) Ray spot plot for the zero field. d) Ray spot plot for the far field (7.215 mm).

The system has a filter for the UV light source. As discussed, the design is intended for either 365 nm or 405 nm wavelength light. Illumination of the mask is supplied by a 75 Watt short-arc lamp source (Dymax Bluewave 75). The purpose of the illumination is to provide a uniform exposure of substrate and initiate polymerization. In addition to uniformity we require that it has sufficient radiance and control over exposure time. Short-arc mercury lamps produce a wide range of wavelengths at varying intensities. The radiance for this light source at different wavelengths is shown in Table 3 as measured from the end of the optical light guide. In addition, the lamp has a mechanical shutter that can operate at 200 Hz.

*Table 3. Bluewave 75 specifications.**

	Wavelength (nm)	Intensity (W/cm^2)
Total	280-450	19
Visible	400-450	9
UVA	320-395	9
UVB	280-320	1.5

*Data supplied by manufacturer (Dymax)

This lamp produces > 9 W/cm^2 in the UVA and a total of 19 W/cm^2 across the spectrum. We therefore inserted a band pass filter (Thorlabs FGUV) before the tube lens to select ∓30 nm on either side of 355 nm wavelength. The total output from the lamps fiber optic end is 9 W/cm^2 at this wavelength. Taking into account optical losses, the high radiance will aid the system in achieving fast polymerization.

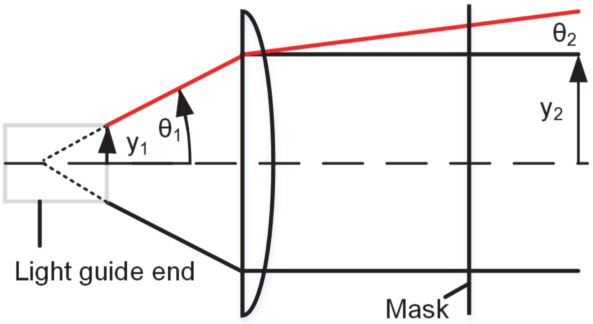


Figure 10. Schematic of the collimating lens in relation to the light guide and mask to determine the divergent angle of the beam and optimal focal length of the lens.

To achieve sharp projection of the mask, it is necessary to collimate the output from the optical light guide (Figure 10). Given the mask size of 20 mm, the illumination can be collimated into a beam larger than the mask. The light guide supplied by Dymax has a divergence angle of 0.425 radians and an initial beam diameter(y_2) of 2.5 mm. Using Equation 5 and Equation 6 we can calculate the focal length (f_1) of the collimating lens and the divergence angle θ_2 of a 25 mm output beam [20].

$$y_2 = \theta_1 f_1 \qquad \text{Equation 5}$$

$$\theta_2 = {}^{y_1}/_{f_1} \qquad \text{Equation 6}$$

The focal length of the lens is 58.8 mm and the divergence angle is 0.0425 radians. Thus we selected a Thorlabs aspheric condenser lens (ACL7560) with a focal length of 60mm. An aspheric was chosen because it efficiently collimates the light while introducing fewer aberrations. Using this, the illumination is focused to infinity when the mask image is in focus and vice versa. This prevents the system from imaging the lamp source and maximizes the depth of field.

A chrome on quartz mask was designed to study shape and size of structures and later particles. The arrangement of the mask in the system is shown in Figure 6 and a detailed view of the mask is shown in Figure 11.

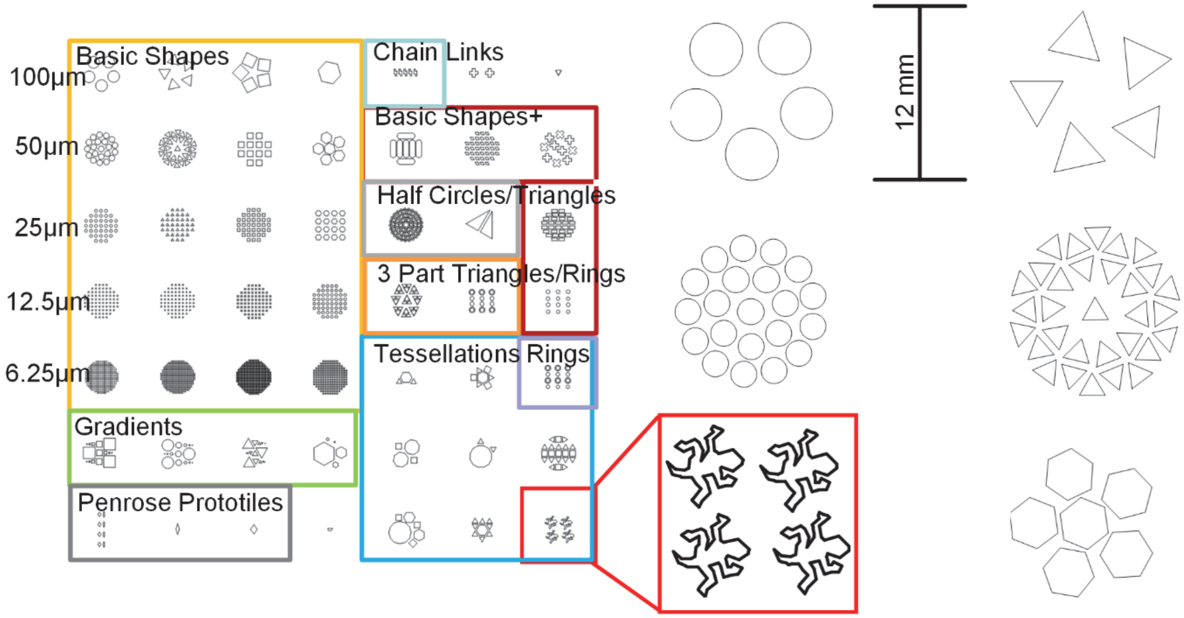
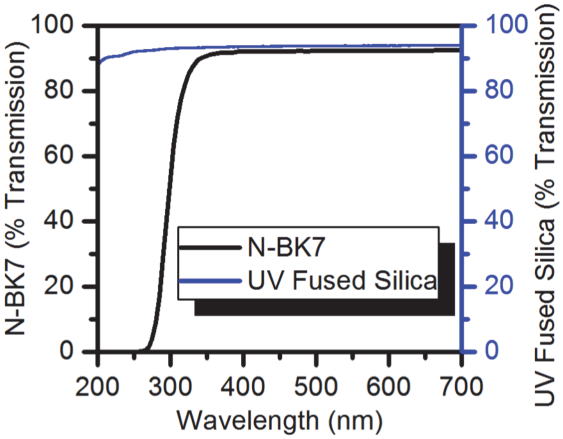


Figure 11. An overview of the mask with an array of possible shapes. The sizes next to the basic shapes correspond to the size the particles should be if projected with a 20X objective. The right hand side shows six enlarged examples of shapes on the mask.

Glass and surface coating choice is another important consideration. When possible UV fused silica was chosen or uncoated NBK-7 was chosen. The transmission data for the UV fused silica is shown in Figure 12 and provides > 90% transmission below 365 nm, while the BK-7 exhibits significantly decreased transmission of light at wavelengths below 365 nm.



*Data provided by Thorlabs Inc.

Figure 12. Optical transmission data for N-BK7 and UV fused silica as provided by the manufacturer.

The camera looks through the same objective to image the projection and aid in focusing on the substrate. A beam splitter from thorlabs (DMLP-505) sends 10% of the reflected image towards the camera. Due to the infinity corrected design the beam splitter can be inserted into the projection optical train without changing the focal lengths or introducing significant aberrations.

Along the imaging path after the beamsplitter, a tube lens (200 mm Mitutoyo) focuses the image onto a CCD. The CCD is a Nikon DSLR 5100. Interestingly, the SLR has a sensitivity to NUV light, not found in other sensors and can image the UV projection. Many other cameras filter out light below 400 nm making it difficult to focus the UV projection. The design for this is shown in Figure 13.

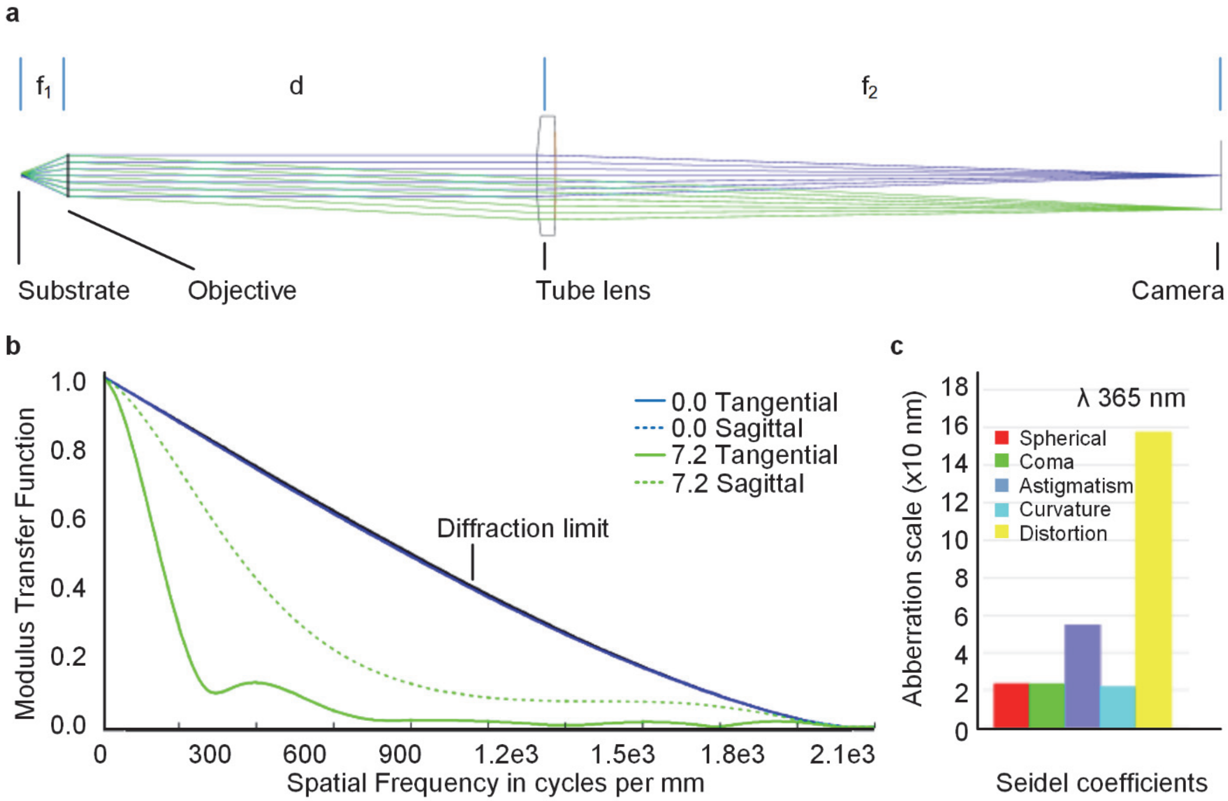


Figure 13. Modeling the imaging design. a) Layout of the camera system including the objective and tube lens. b) The modulation transfer function for this arrangement. c) The expected aberrations.

For this system we chose an MTF value of 30% predicting an imaging resolution of 0.7 µm which matches the expected value from the NA of the lens. The camera has a sensor size of pixels with (4,928x3264) pixels, where each pixel has a pitch of 4.78 µm. The sensor has a diagonal length of 28.9 mm. Therefore given a magnification of 20X the FOV is 1.3 mm diagonal and each pixel represents 0.23 microns. This changes according to the objective selected as shown in Table 4.

Table 4. Objective field of view and resolution.

Objective	µm/pixel	FOV width (µm)	FOV height (µm)
2X	2.37	11354	7520
5X	1.86	9176	6078
20X	0.23	1138	753
50X	0.09	452	300

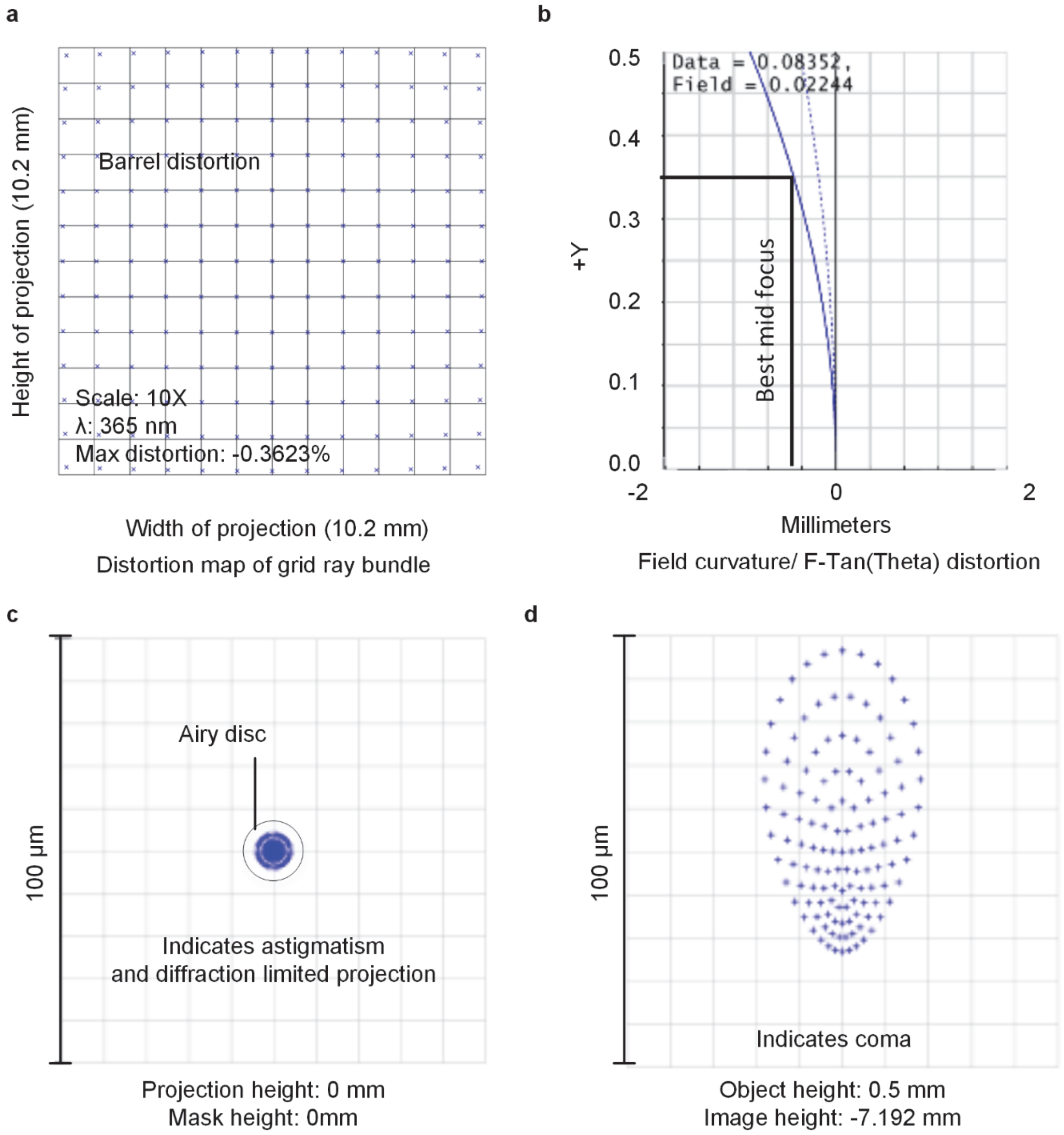


Figure 14. Ray tracing plots. a) Distortion plot for rays traveling through the system. Maximum distortion is -0.3623. b) Field curvature as measured across the field. c) Spot diagram showing the astigmatism present in the system. d) Spot diagram for the off axis field showing coma for the imaging system.

Figure 14a shows that the imaging system has some distortion but unlike the projection system it is barrel distortion with a maximum predicted value of 1 µm.

The imaging system produces the opposite type of distortion as the projection system, barrel distortion, with a maximum predicted distortion of 1 µm. The transverse ray plot indicates spherical aberration. Examining the far field the location of best focus shifts 1 µm from the zero field and a best focus position adjusted 0.5 µm as indicated. This is 2 µm away from the best focal distance indicated for the projection system. The projection is significantly more important however, so in use the focus is adjusted to focus the mask on the substrate at the expense of a small amount of imaging degradation. The spot diagram, Figure 14c and Figure 14d, shows that the imaging system is diffraction limited for the airy disc associated with its optical path, this correlates with the MTF plotted previously. Finally, coma is also present in the imaging system for the off axis field as is seen by the lengthening of spot of rays. It appears to follow the same pattern as for the projection but across 100 µm instead of 10 µm.

%. Figure 9b shows the transverse ray plot indicating how the spherical aberration is presented. At the far field 5 µm of spherical aberration occur, this can be reduced to 2.5 µm at the best mid-focus as marked. The spot diagrams shown in Figure 9c and Figure 9d show the effect of astigmatism and coma respectively. While astigmatism is present we also observe that for the zero field the system is diffraction limited as the spot is within the airy disc.

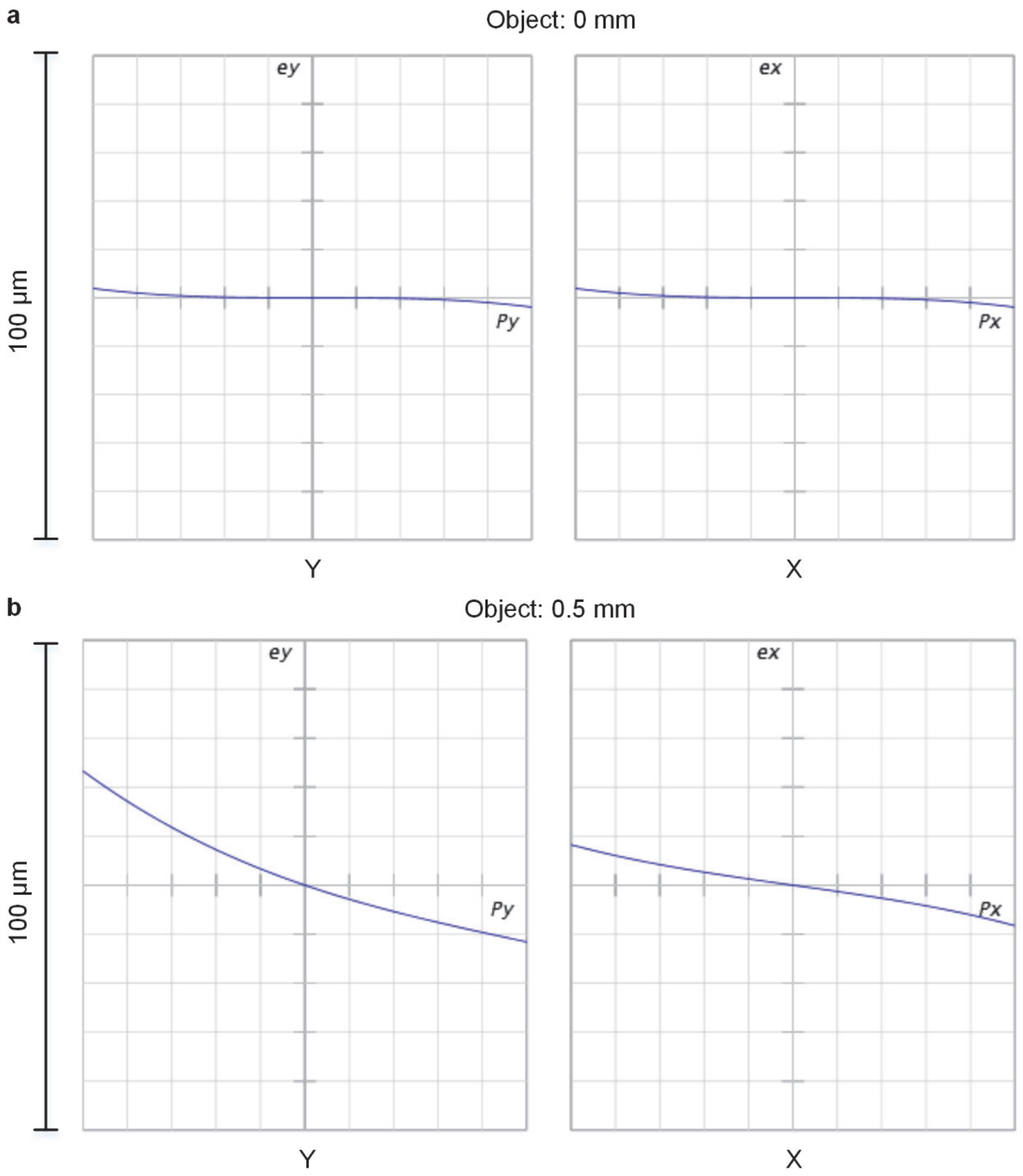


Figure 15. Ray fan plot showing spherical aberration in the system for both the transverse and sagittal directions.

Figure 15 shows the ray fan plot indicating spherical aberration in the system. However, because there is no focal shift that could reduce the spherical aberration it is taken as the best choice. Additional lens elements would be required through lens splitting and re-optimization to reduce the spherical aberration.

Now, all of the necessary components for the optical systems prescription have been described. From here on we discuss the necessary arrangement to replace the stationary mask with a digital micromirror device (DMD, W4100). This device is a spatial light modulator which enables the user to dynamically change the mask via digital control. An exemplary DMD chip and the micromirrors that are arrayed on it is shown in Figure 16a and 12b. The mirrors are MEMS devices that tilt (± 10˚) on an axis under an applied voltage. Due to their design they can switch states (ON/OFF) rapidly without wear.

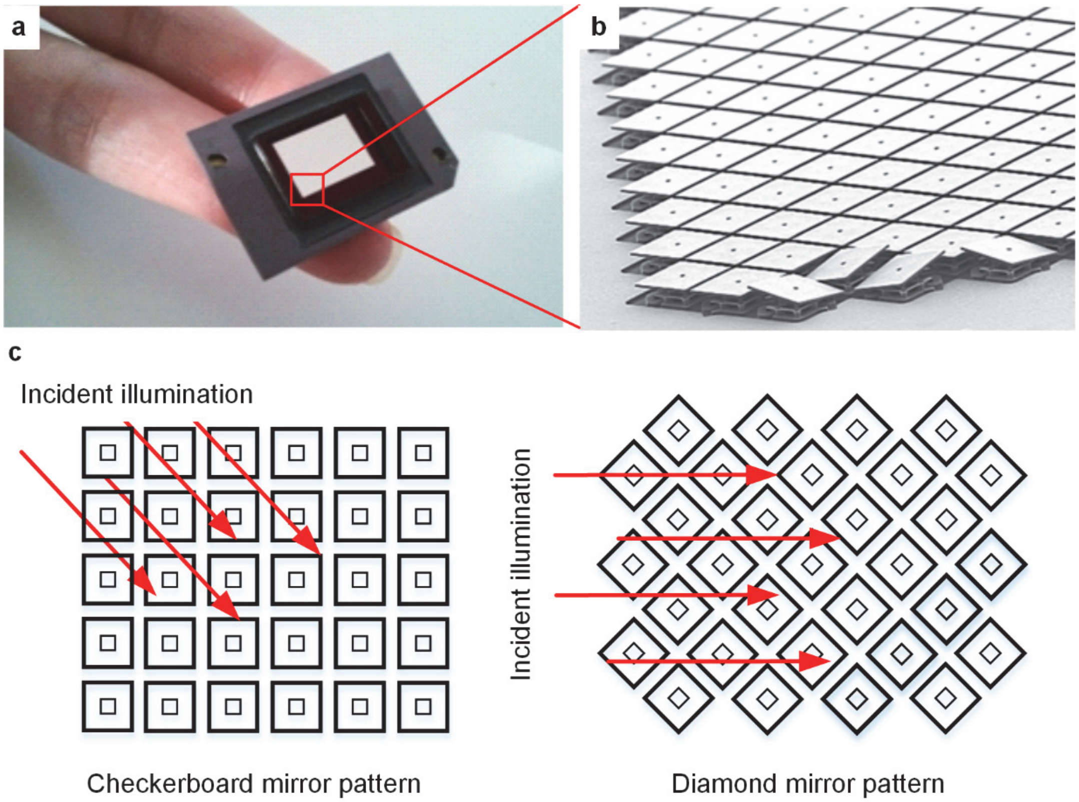


Figure 16. DMD chip design. a) DMD chip held by researcher [21]. b) SEM image of DMD mirrors. Note how they tilt on a diagonal axis when actuated. c) Two common configurations of the DMD mirrors that affect system design.

The mirrors can be arranged in one of two ways. As shown in Figure 16c the mirrors can be arranged in a checkerboard pattern or a diamond pattern. The main difference is the direction the incident light approaches the edge of the DMD from. For this reason diamond mirror patterns simplify the design of the system but require interpolation because the pixels are not oriented with image pixels from the screen. This system uses the checkerboard pattern so the angle of incidence is accounted for in the design of the system housing.

We designed a housing that positions a UV-capable DMD chip in place of the mask. The housing accepts the fiber light guide and steers the collimated beam onto the DMD such that it reflects along the axis of the projection system. This is achieved using a total internal reflectance

(TIR) prism or by positioning the light guide along the diagonal of the DMD, as shown in Figure 17a. Figure 17b shows a custom 3D printed housing to hold the DMD chip and a TIR prism. Finally, Figure 17c overlays the optical paths that represent the imaging, projection and illumination discussed previously onto the schematic of the system.

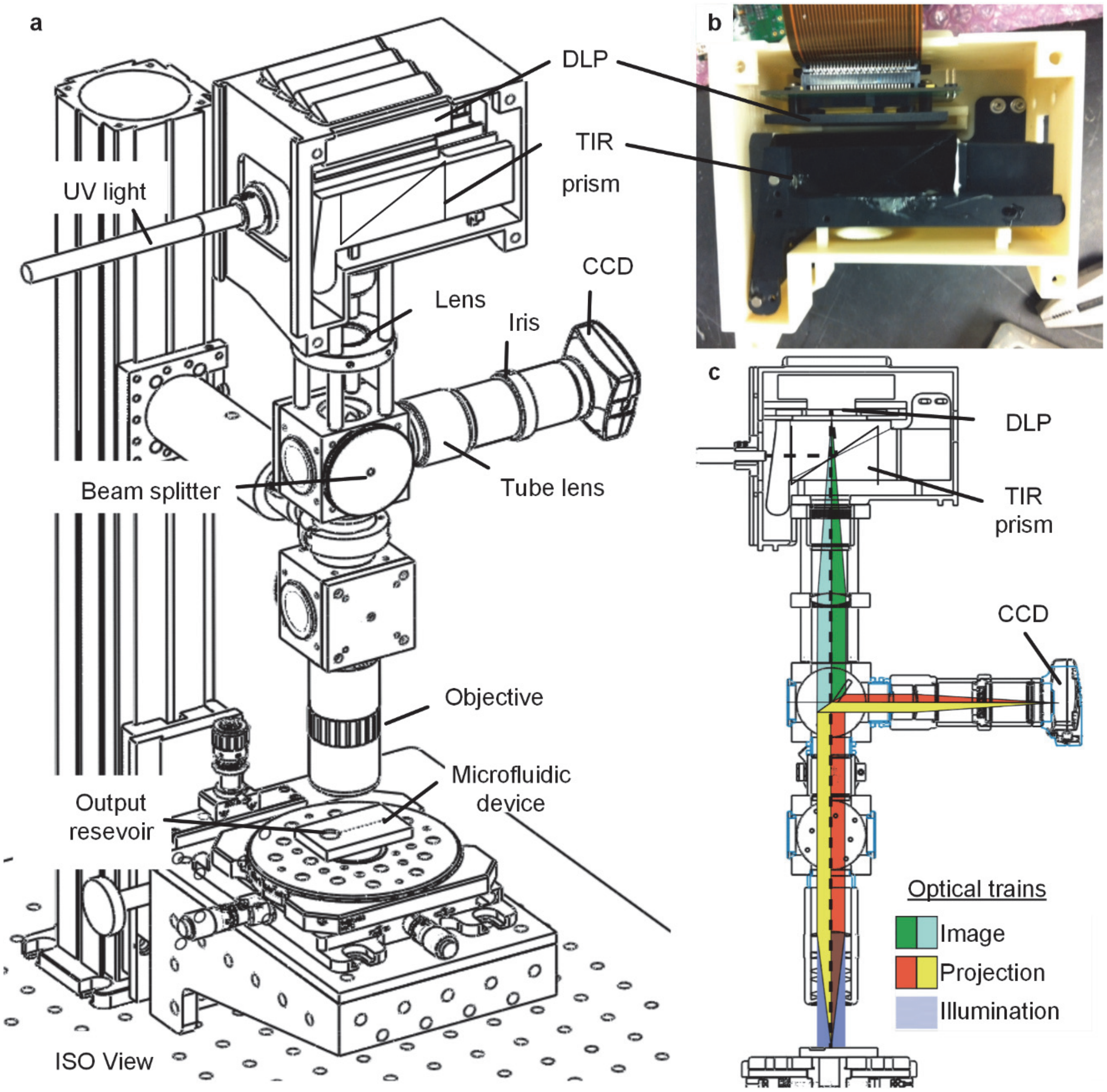
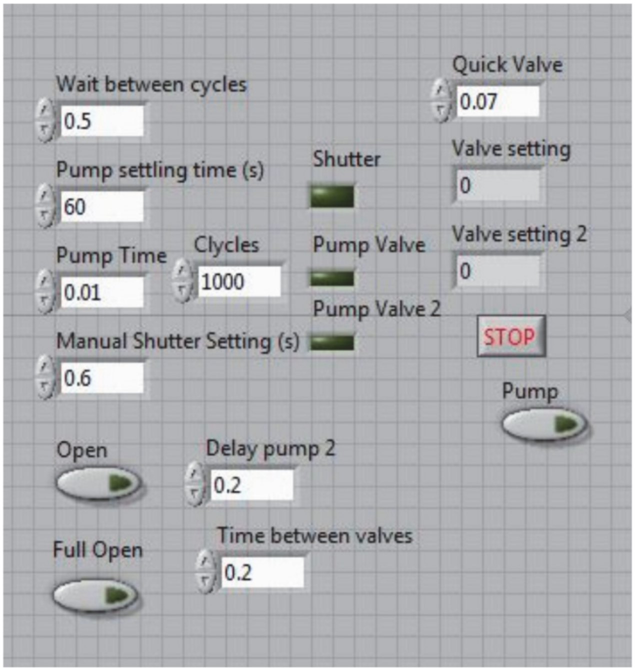


Figure 17. Overview of the maskless lithography system. a) Schematic of the UV DLP optofluidic lithography system. b) 3D printed housing for the DMD and TIR prism. c) An overlay of the optical trains discussed with the schematic.

The system is controlled via Labview and a National Instruments (NI-6008) DAQ card. The UV light is actuated by closing a resistance circuit using a 5V relay switch controlled Labview. Exposure time is controlled either via Labview or by a timer built into the BlueWave 75.



Wait between cycles 0.5

Pump settling time (s) 60

Pump Time 0.01

Clycles 1000

Manual Shutter Setting (s) 0.6

Delay pump 2 0.2

Time between valves 0.2

Quick Valve 0.07

Valve setting 0

Valve setting 2 0

Figure 7. Screenshot of LABVIEW interface for controlling exposure and in future work flow in a microfluidic device.

Validation of Performance

Performance of the imaging system is characterized in two ways. First an image from the system is compared to an image from a Keyence VHX-5000. This is shown in Figure 18 and to the author the results look comparable.

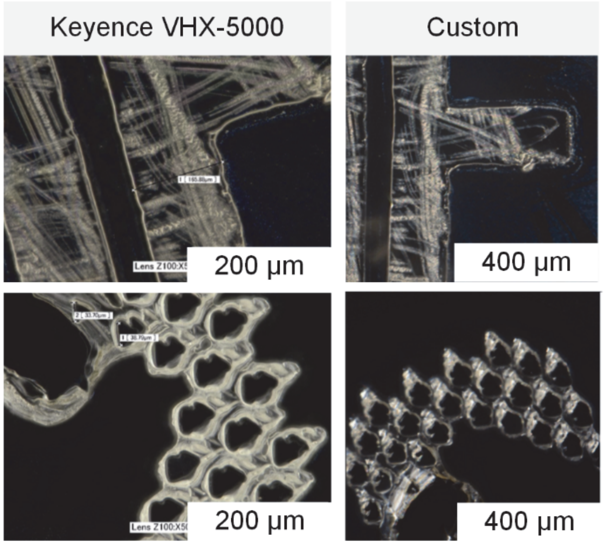


Figure 18. Comparison of micrographs taken using a Keyence VHX-5000 (500X) and our custom built microscope.

The second method used was to test the resolution. No test slide was available with 0.7 µm resolution so monodisperse Poly(styrene) beads were used. Images were taken of 0.5 µm beads as shown in Figure 19. The individual beads are easily resolvable.

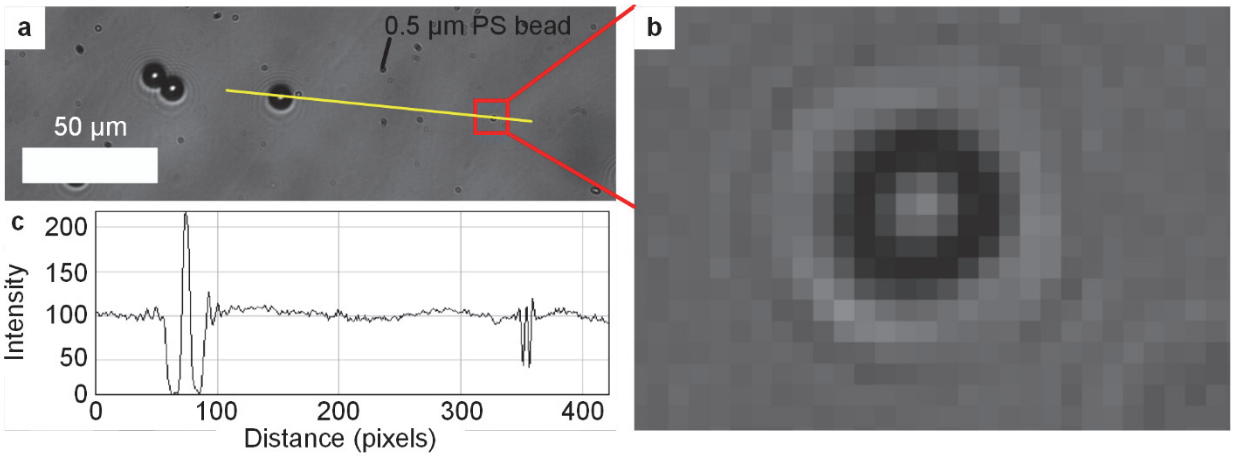


Figure 19. Characterizing the resolution of the imaging system. a) Resolution test target using 10 µm and 0.5 µm PS beads. b) Shows a close up of the 0.5 µm bead. c) An intensity profile from the image taken along the yellow line.

Illumination was tested by measuring the radiance (W/cm^2) from the light source and at the substrate. This is shown in Table 5. The projection of the DMD at the substrate did not fill the sensor area on the radiometer. The radiometer sensor area was approximately 5 mm^2 while the projection was over a 1-5 mm^2 area. We attempted to correct for this but some error may exist in the reported values such that the radiance is lower than the real radiance. From the light guide tip a radiance of 747 W/cm^2 was measured, but for the 20X objective the system actually produced a higher radiance (8.39%) because the beam was focused sufficiently to overcome losses.

Table 5. Measured radiance at various locations in the system.

Measured location	Radiance (W/cm2)	% Change
Light guide tip	747.38	0%
Collimated beam 25mm from lens	636.37	-14.85%
At substrate using 5x objective	111.39	-85.10%
At substrate using 20x objective	810.10	8.39%

The performance of the projection system was measured by polymerizing features out of polyethylene glycol diacrylate (PEG-DA). PEG-DA is a hydrogel photopolymer resist. The photopolymer solution was prepared using 0.5% v/v photoinitiator 2-hydroxy-2methyl-1-phenyl-

1-propanone (Darocur 1173, Sigma Aldrich) in poly(ethylene glycol) (400) diacrylate (PEG-DA, Sigma Aldrich), **Error! Reference source not found.**.

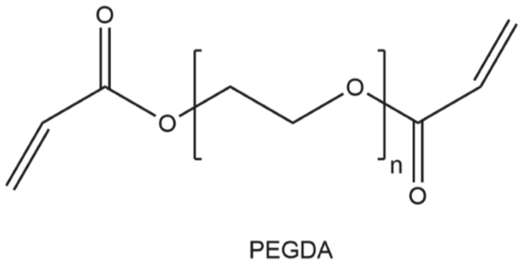


Figure 20. Chemical structure of PEG-DA

Using the system PEG-DA pillars of 10 µm width was fabricated. The same process was performed for the maskless system, and the ultimate resolution was measured as 5 µm (5X). The experiment was performed by projecting a checkerboard pattern onto PEG-DA that had been spun onto a glass slide with a thickness of 50 µm. We expect the reason the polymerization resolution does not match the optical resolution is because of a combination of diffraction in the thick resist and polymer chain kinetics.

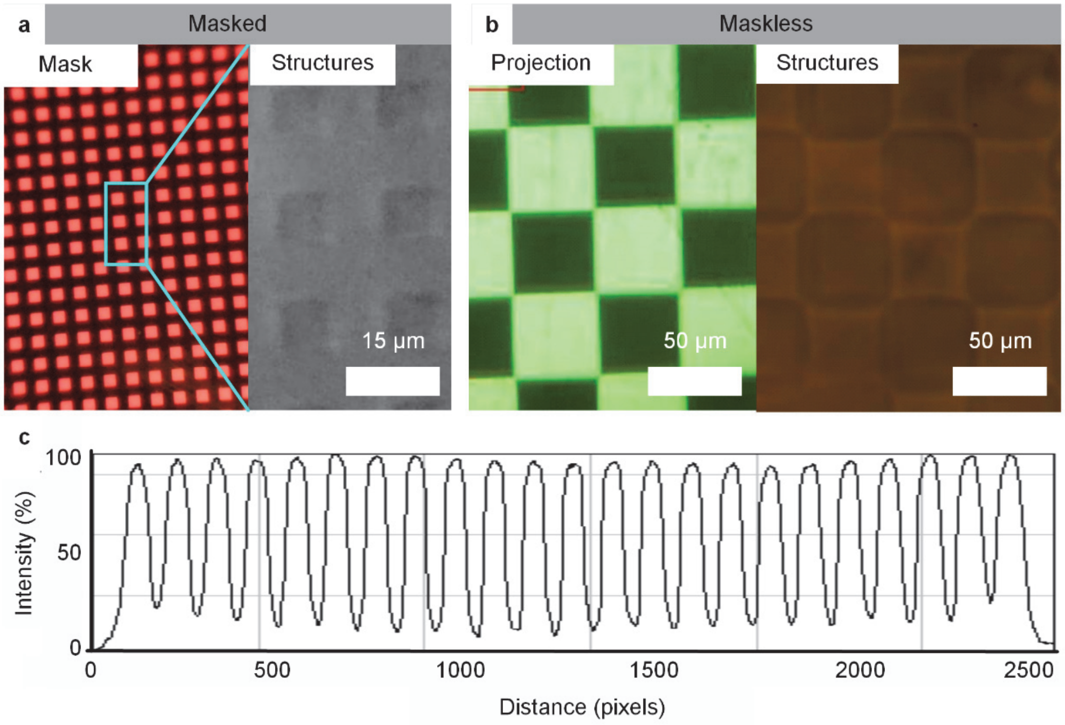


Figure 21. Measuring projection performance. a) Smallest feature size in PEG fabricated using the masked configuration. Shows 10 µm column. b) Similar resolution target for the maskless system. c) The intensity of illumination across the projection shown in a).

Finally, we examined the illumination intensity over the projection and found that the illumination varied by 7% across the mask. Aberrations also contributed to outer edges showing less contrast between the projected squares. Reduced contrast between projected features causes unintended polymerization due to the lack of a clearly defined edge.

Similar to the expected results from modeling the system, the projection suffers from aberrations near the edge of the mask (Figure 22). This practically limits the area of the DMD that can be used when patterning features less than 10 µm in size. We quantified this area as equaling ~80% of the DMD area.

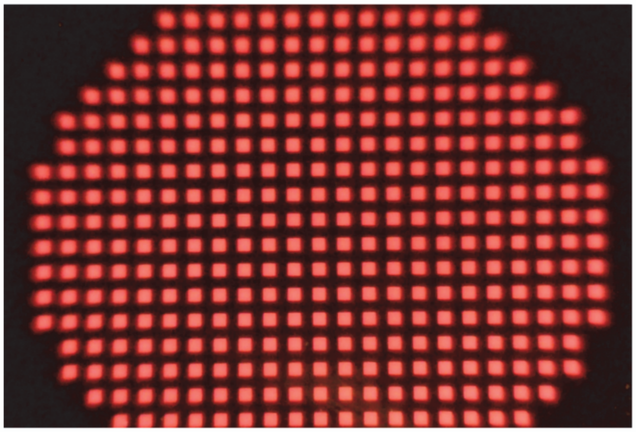


Figure 22. Grid array of squares 25 µm in length to observe aberrations across the field of the projection.

Now we show (Figure 23) patterned features on stationary substrates for both the masked and maskless designs. The masked features show an array of triangles of 5 different sizes which range from 100 µm to 6.25 µm. The maskless system shows similarly sized objects being patterned one after another on a stationary substrate. In this case we patterned the evolution of man to illustrate the progression of the pattern. Both examples were patterned from a solution of PEG-DA spun onto a slide.

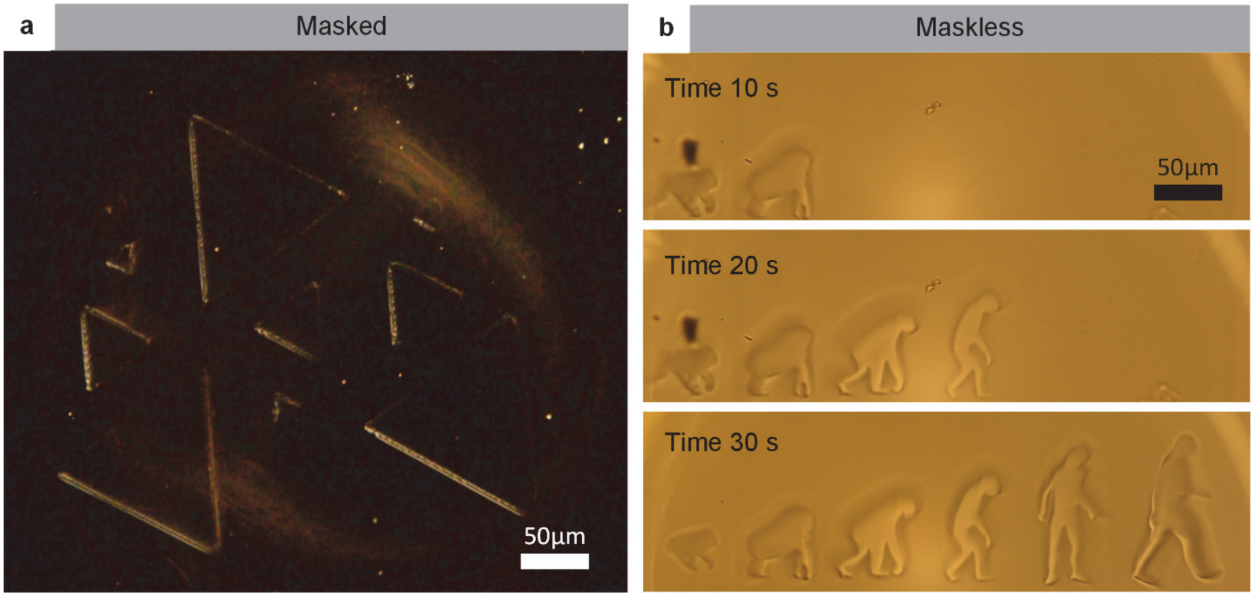


Figure 23. Examples of PEG-DA features on glass. a) Example features patterned on a stationary substrate using the masked system. b) Example features patterned on a stationary substrate using the maskless system. Both sets of features are fabricated in PEG-DA.

Finally, we demonstrate the modularity and ease of integration of the system with other components. Here a roll to roll module developed by Erik Polsen was inserted into the system as shown in Figure 24 [22]. The roll-to-roll module is comprised of a stepper motor, supply and uptake wheel to translate tape across a concave support. A blade (Figure 24c) is used to spread a uniform thin layer of PEG-DA onto the substrate and then it is patterned by the lithography system. An example of the system doing this is shown in Figure 24d where Tetris like shapes are being patterned onto copper tape.

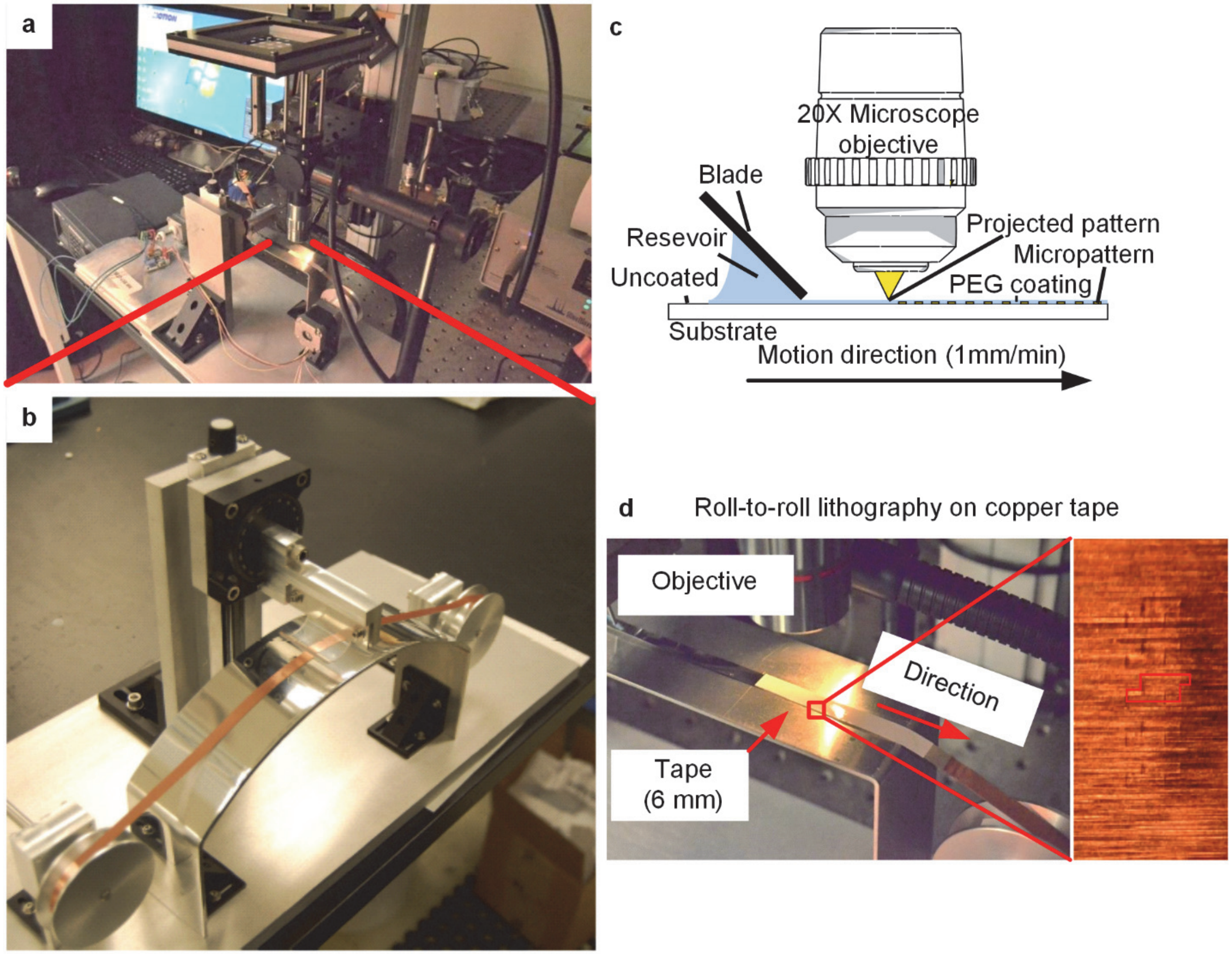


Figure 24. Use of the lithography system to produce features on a roll-to-roll copper tape. a) Photograph of a roll-to-roll module designed by Dr. Erik Polson integrated into the masked system. b) A photograph of the roll-to-roll system. c) A schematic of the process of applying PEG-DA to the substrate while it is moving and patterning features onto it sequentially. d) Exemplary results showing 5 objects being patterned on the copper tape.

Discussion and conclusions

Several key observations become apparent. First, it is possible to build a high quality lithography system that is more flexible than post-engineering an off-the-shelf Zeiss scope. Further, the system cost ¼ of a similar Zeiss system that does not include a maskless capability. When designed as an infinity corrected system, the change in magnification of the mask when projected is the ratio of the focal lengths for the field of view and tube lens.

The maskless capability to shape the light on demand opens a number of research opportunities that were not capable with a masked system. For example, the system can now pattern different shapes as the substrate translates in a continuous fashion [23]. Also, as shown later, it can respond to changes in the substrate such as the presence of a particle.

Unlike a premade imaging microscope such as Zeiss or Nikon the design of this system enables modularity for integration of multiple components. The structure is comprised of COTS components (Thorlabs) so interfacing additional lenses, filters, light sources, beam splitters, camera etc. is straightforward given design objectives. Further, the configuration of the system can be modified to adapt to constraints of a particular application. As shown, the same system was configured either as an upright and inverted arrangement. An upright configuration is best when patterning opaque substrates, while the inverted arrangement improves focus and feature resolution in microfluidic channels mounted to glass slides.

CHAPTER III. SYNTHESIS AND ANALYSIS OF HYDROGEL MICROPARTICLES BY OPTOFLUIDIC LITHOGRAPHY

Abstract

This chapter begins with a review of particle synthesis by microfluidics and other methods. We build upon the maskless lithography design presented in 0, by integrating a microfluidic device into the system and a custom flow controller. These are used to fabricate microparticles by polymerizing PEG-DA in flow. The system is tested by fabricating a library of particles and measuring their geometry and polydispersity (20-100 μm side length, CoV 5-15 %). The role of system design on throughput is described and because shallow channels take a long time to stop flowing it is challenging to rapidly fabricate small particles. To address this issue a new microfluidic device which uses a thick channel to achieve fast stop times around pillars is used to fabricate small particles quickly. The particles can be fabricate on top of the pillars at a rate of up to 2.5 Million particles/minute. I would like to acknowledge Levon Cimonian's contributions to preparing samples and performing experiments to characterize this system.

Introduction

Rapid synthesis of microparticles with arbitrary control of shape and surface chemistry enables more efficient diagnostics and improved tissue engineering. For example, multiplexed diagnostic assays have improved the ability to rapidly identify multiple analytes such as proteins in a sample. To achieve improved throughput smaller microparticles need to be fabricated rapidly without loss of quality [24]–[26]. Many methods have been developed to synthesize microparticles including traditional lithography [27], droplet based microfluidics [28], emulsions, jetting [29], [30] and stop flow lithography [31]. Each method has advantages and disadvantages, yet there is significant need for improvement in the manufacture, modeling and application of micro and nano particles for known and yet to be imagined fields. For rapid manufacture of particles with a wide variety of shapes and functions, microfluidic lithography is one of the most promising techniques[24].

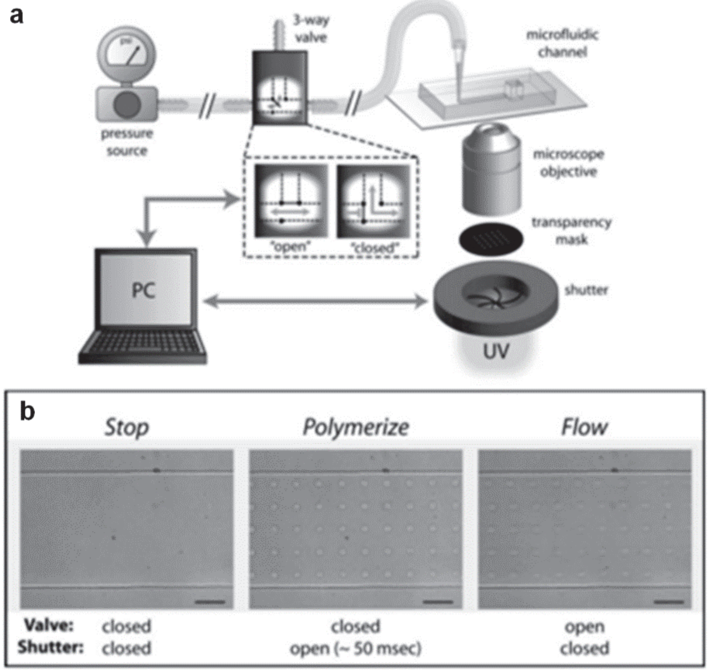


Figure 25. Schematic of stop flow lithography as first presented by Doyle et al [32].

As introduced in Chapter I, seminal work on microscope based lithography has been demonstrated in a number of forms and a review on this published by Love et al. [8] may have been the inspiration for the development of a technique described as stop flow lithography, which was first demonstrated by Doyle et al. in 2007 (Figure 25) to fabricate hydrogel microparticles. The basic method uses a UV light source focused through a mask, located at the field stop of a standard microscope, to cure a polymer flowing in a microfluidic channel. The seminal paper discussed continuous flow lithography but they later found that throughput could be improved by rapidly stopping the flow during exposure of the polymer in the channel. They have since demonstrated the ability to produce particles having 2D/3D shape and varying chemistry or color. For example, barcoded multiplexed assay contain a chemically patchy region for sensing analytes and a barcode region that is rapidly read by a laser to identify the particle [33], [34].

Prior to the invention of stop flow lithography spherical microparticles were fabricated using microfluidic droplet based techniques. T-shaped junctions had been used to fabricate hydrogel microparticles by creating droplets inside of a sheath flow. The droplets were then exposed to UV light polymerizing the hydrogel sphere.

Oxygen inhibition was essential to the success of this method. Oxygen, which diffuses through the PDMS microchannel, inhibits polymerization of PEG-DA creating a thin self-lubricating layer between the particle and the channel [35]. This layer, which is approximately 5 µm thick, both prevents particles from adhering to the channel and enables them to slide out of the channel after fabrication. Moreover, they recently published a non PDMS device that doesn't rely on oxygen diffusion through the PDMS channels to provide lubrication to the particles but instead flows lubricating fluids on either side of the photopolymer [34].

Doyle also controlled surface chemistry of the particle by flowing two or more immiscible fluids through the channel using hydrodynamic focusing in both the horizontal and vertical configurations [36] to create Janus, opaque and compartmentalized particles [37].

Kwon's group from Seoul National University has developed additional methods in optofluidic lithography; they pioneered the use of DLP maskless lithography in a stop flow device [18] similar to what is shown in this thesis. This also includes railed lithography as discussed in Chapter I.

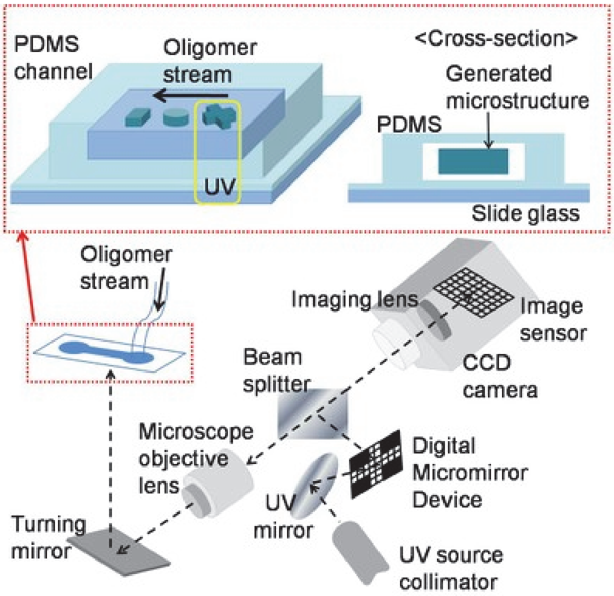


Figure 26. Kwon's maskless stop flow system [18].

That work was followed by a method for fabricating 3D structures using an air bladder to control the height of the channel [38]. The limitation of this technique is that undercuts are not allowed, and additional materials must be fabricated on a smaller area than the previous layer or else they will cure through the entire thickness of the channel.

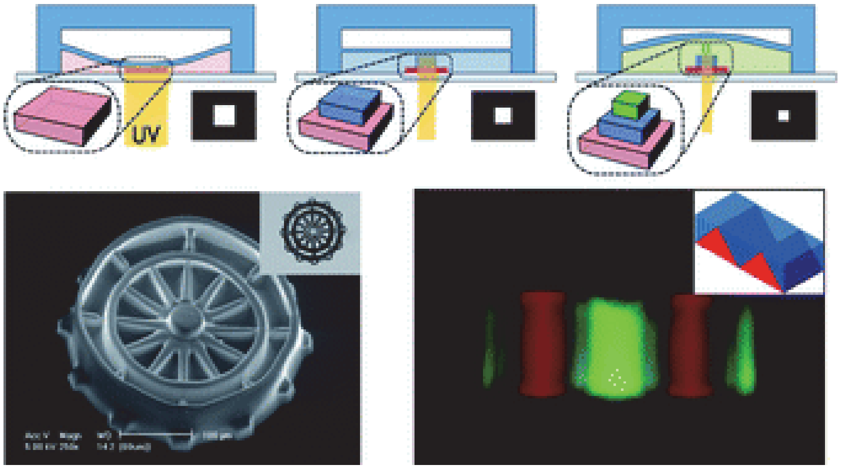


Figure 27. 3D features in microparticles enabled by air bladder to change the height of the channel during exposure [38].

Finally, Kwon's group contributed to multiplex microparticles for bioassays and a very novel photonic crystal using multi-colored ink [39]. The particle polymer uses a novel ink that changes color depending on the spacing between nanoparticles within the polymer. The spacing is controlled by a magnetic field and locked into place when the PEG-DA is polymerized. They then applied this to particles in the form of barcodes surrounded by a functional/sensing area

[40]. The sensing area interacts with biomolecules and fluoresces when specific molecules are captured. The barcoded region is then read to determine what the drug or molecule used in that particle was. This creates a unique, multiplexed assay for drug testing.

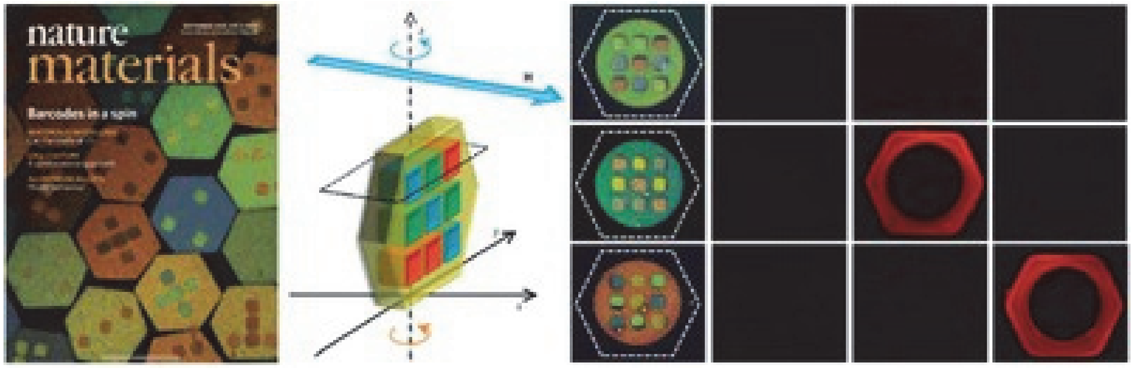


Figure 28. Binel labs color changing particles based on orientation of nanoparticles [39], [40].

Prof. DeSimone from the University of North Carolina invented and commercialized PRINT (Particle Replication in Non-wetting Templates) [41]. This is based on the use of a template with a non-wetting top surface. The photopolymer is spread onto the template and wets cavities imprinted on the surface of the mold. This is then exposed to UV light to cure the polymer. Careful preparation of the template allows for nanometer resolution particles and they have scaled-up this method using a roll-to-roll system via a startup company, Liquidia.

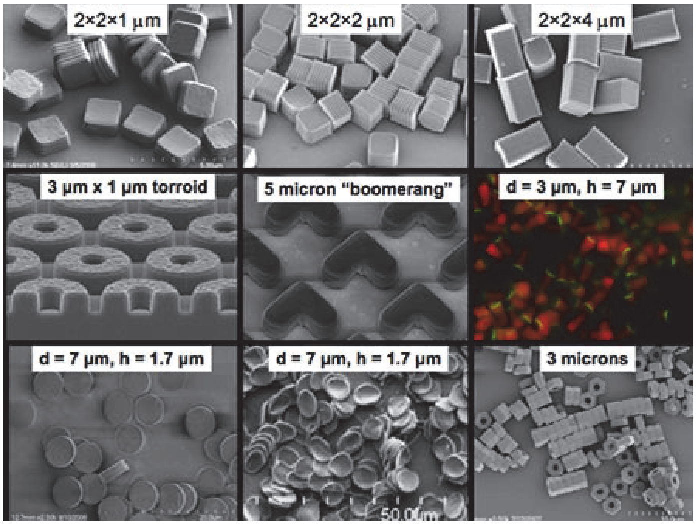


Figure 29. Examples of the PRINT method of fabricating microparticles [42].

These particles are being applied to a variety of applications including antifouling coatings, delivery vehicles for drugs and cells and photovoltaics [11].

While these are the most directly comparble methods of microparticle fabrication others do exist. Many of these techniques rely on chemical fabrication. For example DNA folding has also shown the ability to create scalable particles with nanometer scale features. For many applications it suffers from a slow reaction rates and low temperature solubility [43]–[45]. This method was first pioneered by Rothemund who found that stapled DNA strands could be used to pin down DNA into a certain shape as reviewed by Lin et al [46].

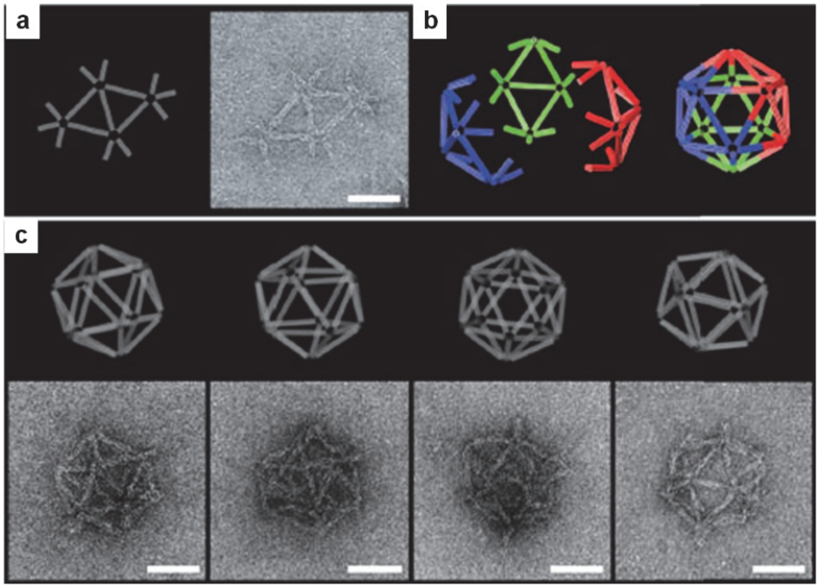


Figure 30. DNA microcapsule folding [46].

Finally, considerable work has also been done to synthesize particles from the bottom up by self assembly, electrojetting and emulsions. One promising bottom up method using nanoparticles to create structures by adding spheres to a central structure, the control of this process can create many basic polyhedrons that can then be used for secondary assembly processes [47]. Electrojetting has been employed in the fabrication of Janus particles and scaffolds [29]. Emulsions have been used to fabricate a number of particles [47]–[51].

Each method has specific advantages and disadvantages in throughput, complexity, control over shape, chemical anisotropy, biocompatibility and others. In our case the throughput, resolution and control over shape is why the optofluidic approach was the method to study. We aim to use the particles fabricated by this method to design self-assembling system at the microscale and to study the physics behind self assembling particles but did not do so in this thesis.

Synthesis and analysis methods

In 0, a maskless lithography system was discussed with a detailed optical design. To enable the system to fabricate microparticles a microfluidic device is added to the system as well as a system to control the fluid flow. A top level overview of how this works is shown in Figure 31. First, the microfluidic device is placed on top of an inverted configuration of the projection system. The system is focused onto the photopolymer within the channel. Then software, written in LabView alternates states between exposing the channel with structured light, to polymerize particles into defined shapes, and pumping the particles and uncured polymer into a reservoir. Excess oligomer is washed off and the particles are re-suspended.

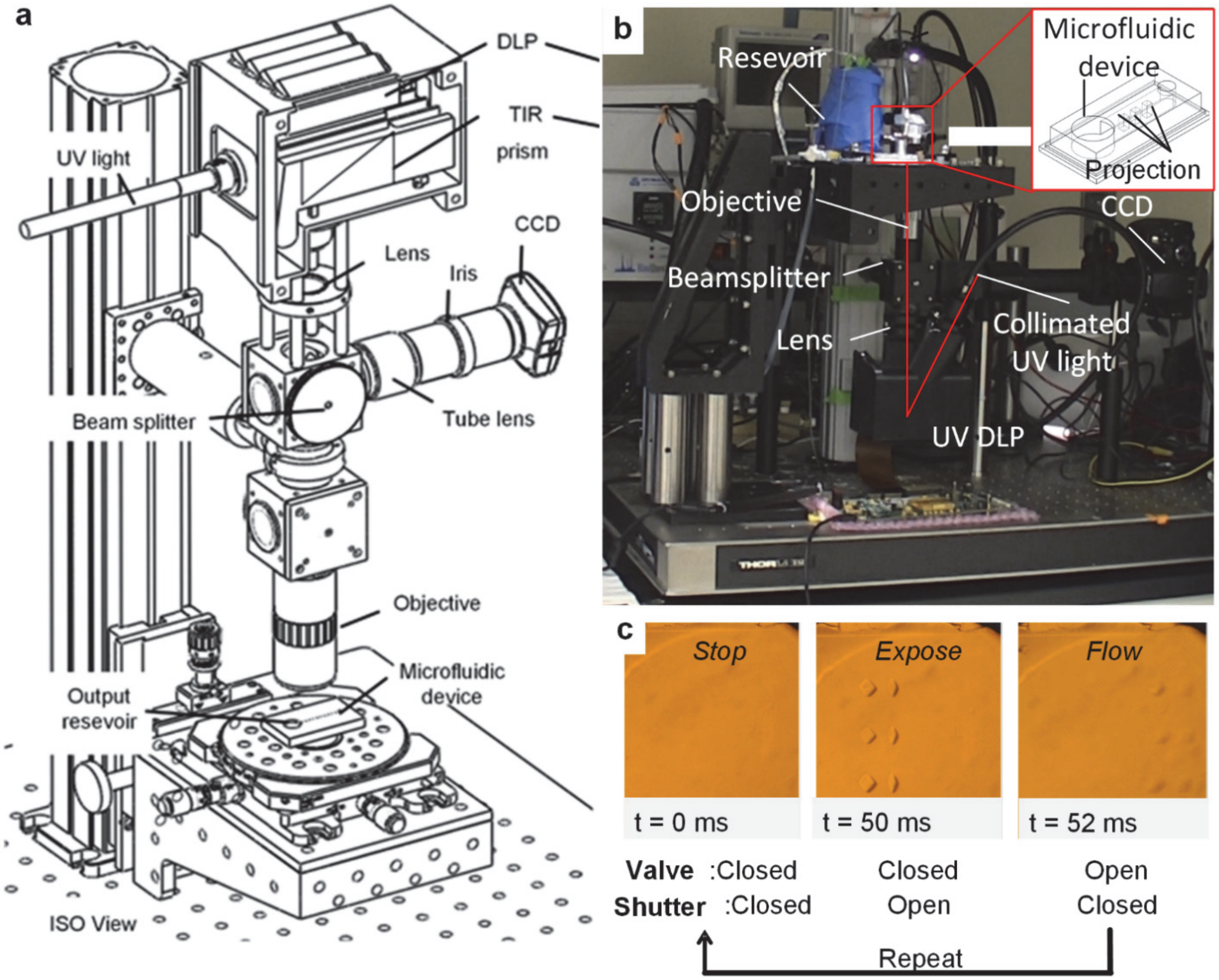


Figure 31. Overview of microfluidic synthesis system. a) Schematic of the optical system with a microfluidic device placed on the stage. b) Photograph of system used to synthesize microparticles in the inverted configuration. c) The process to polymerize microparticles begins when the flow is stopped, then the DMD image is projected polymerizing the particles which then flow out of the channel. This is repeated to synthesize any number of particles.

The microfluidic device was designed using a straight channel design with varying widths on a single mold. The widths on the device included 75 µm, 200 µm, 500 µm and 1000 µm. Example devices are shown in Figure 32.

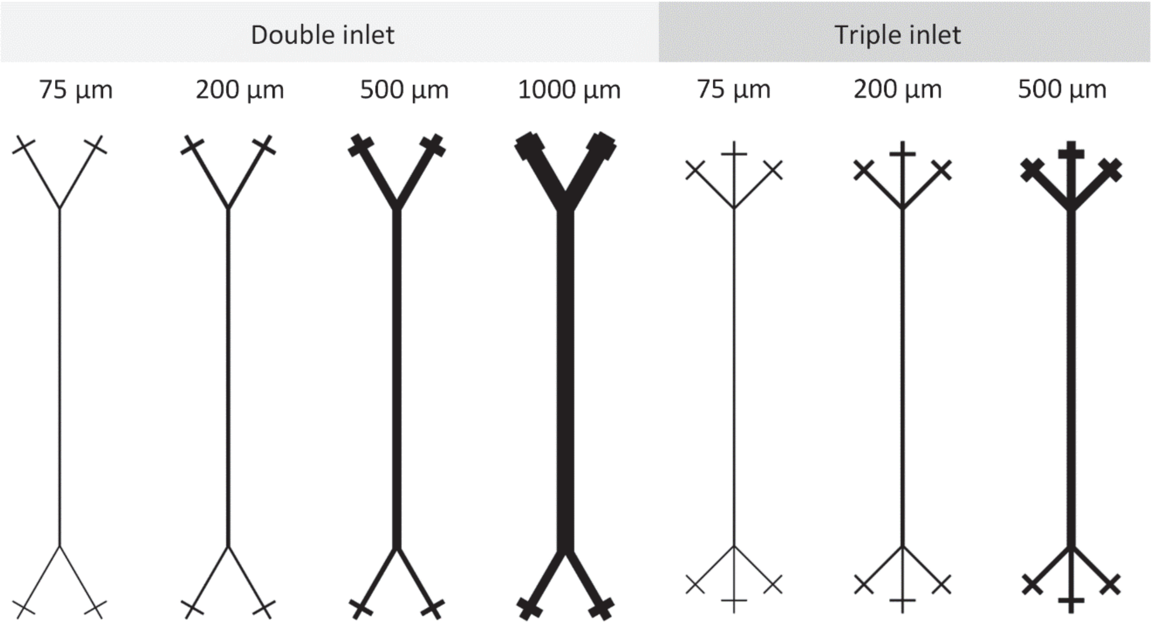


Figure 32. Microfluidic device designs for double and triple inlets used for laminar flow mixing.

Two poly-dimethylsiloxane (PDMS) microfluidic devices were fabricated by soft lithography methods, Figure 33 [3], [52] to synthesize particles of two different heights. First, photoresist is applied to a Si wafer which is then deep RIE etched to create a permanent mold. This mold is used to cast multiple devices in PDMS. After curing for 1 hour at 85 C˚ the PDMS is removed from the mold and placed atop a PDMS (1:20) coated glass slide. The PDMS was applied to the glass slide using a spin coater at 500 rpm for 30 seconds and then 3000 rpm for 30 seconds. The resulting layer for each mold was measured as 5 µm and 45 µm thick. The assembled device was allowed to cure at room temperature overnight.

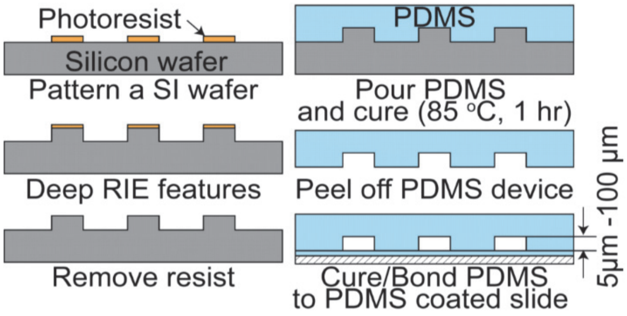


Figure 33. Fabrication process for making particle synthesis microfluidic devices.

To synthesize particles, it is necessary to flow a photopolymer solution that is sensitive to the projected light. The same hydrogel solution used in Chapter I, PEG-DA is used in this system. Flow in the microfluidic device was achieved by a pressure pump in which a reservoir (BT-4502, BC-211 Western Analytics) was pressurized to 3 psi, Figure 34. The flow was controlled by pulse width modulation of a solenoid valve (The Lee Co. INKX0514300A) using a spike/hold driver (The Lee Co. IECX0501350A). The valve was attached to a nozzle with an 80 µm opening (The Lee Co. INZA4630912T). The nozzle was inserted into the microfluidic device and held in place with a clamp. Pressure was regulated using a high precision 0-5 psi regulator.

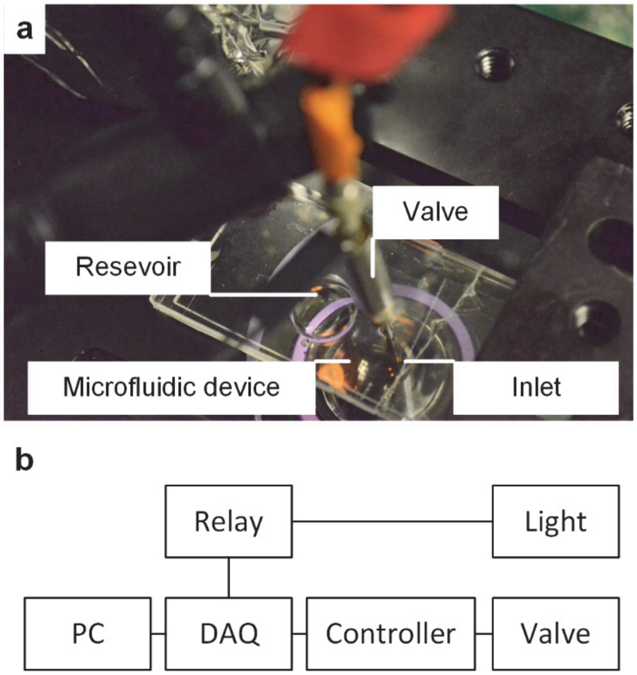


Figure 34. The pumping system. a) Detailed view of the solenoid valve and nozzle inserted into the microfluidic device. b) Control schematic for the pressure system and the light.

Stop times for the microfluidic flow were calculated by pulsing the flow for 0.25 ms and measuring via video the time required for PS beads in the flow to reach 1% of their maximum velocity. Velocities were calculated using position data measured from the ImageJ particle tracker plugin.

Measurements of microparticle dimensions were performed using an FEG scanning electron microscope (Phillips XL30 SEM). The particles were collected from solution via a syringe filter (XX3002500 Millipore) and rinsed twice using water and isopropyl alcohol or until all excess unpolymerized monomer had been washed away. Because the particles are not conductive they were coated in gold for SEM imaging using a sputter system for 10 seconds.

Results and discussion of particle synthesis

To characterize the performance of the system a library of particles was synthesized with standard shapes (hexagons, squares and triangles) across a variety of side lengths (100 µm, 50 µm and 20 µm). A matrix showing SEM images of the PEG particles is shown in Figure 35.

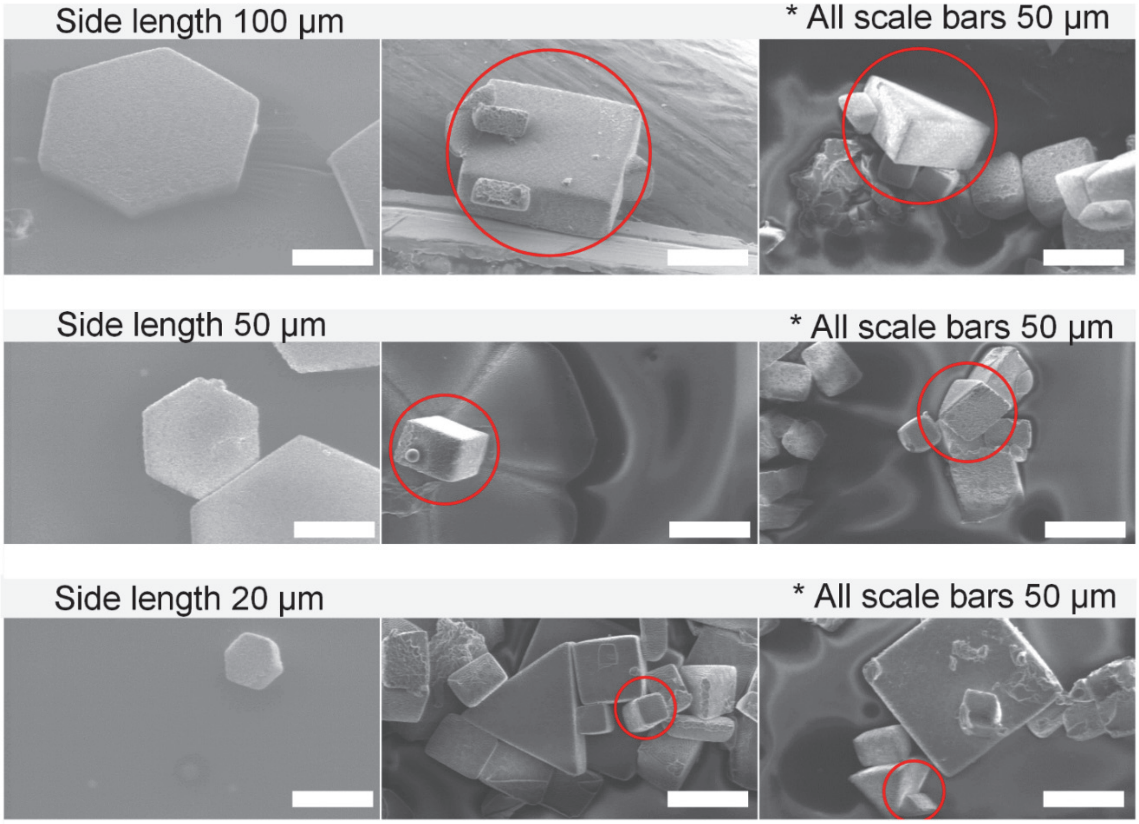


Figure 35. SEM images of PEG-DA particles (from left to right: hexagon, square, triangle) synthesized using the microfluidic lithography system. Columns illustrate shape and rows illustrate size. Red circles are to highlight the example particle.

The monodispersity of the particles was characterized by measuring their side length, corner radius, thickness and corner angle and then calculating the coefficient of variation (CoV) for each metric, Figure 36. The side lengths for 100 µm particles ranged from 89 µm - 104 µm with an average of 96 µm. The 50 µm particles had an average side length of 41 µm with a maximum of 49 µm. Finally, the 25 µm particles ranged from 19 µm – 25 µm with an average of 22.4 µm. The side lengths showed less variation as the length of the size of the particle decreased. The angle between edges of the hexagonal particle was measured and should be measure 120 ˚. As measured in the SEM this was the most uniform metric measuring 121˚ with a standard deviation of < 2.5˚. The error is likely due to measurement bias introduced by the researcher while annotating the SEM images. The thickness of the particles was measured and as expected was well controlled by the height of the fluidic channel. For 45 µm deep channels the average particle thickness differed by less than 1 µm between the different sizes. The thicknesses ranged from 41 µm - 42.5 µm for 100 µm side lengths, 39 µm - 42.7 µm for 50 um side lengths and 41 µm - 43 µm for 25 um side lengths. The radius of particle corners was 4.1 µm for 50 µm and 20 µm particles, while 100 µm particles has a corner radius of 5.05 µm. However, for 100 µm side length particles the range of for the corner radius measured from 2.5 µm – 9 µm. By comparison the 50 µm and 20 µm particles had a smaller range from 2 µm -5.5 µm. Having characterized the basic measurements the

Having measured the important metrics that describe the intra-particle variation we now discuss the monodispersity. No universal definition of monodispersity for microparticles exist, but criteria for monodispersity of spherical microparticles as measured by the CoV range from 3.04% (NIST) – 16% (Quasi-monodisperse) [53]. The reason no singular definition exist is that the necessary monodispersity is application dependent. Doyle et al. defined monodisperse as a CoV of 10% and we've chosen to adopt this standard for this discussion because we are using a similar synthesis method.

For side length measurements, the CoV ranged from 5.2-15% as the particle size increased. The angles between edges of a hexagonal particle were measured with a CoV of 1.7%. The height of the particles as measured in a channel fabricated from a 30 µm deep mold showed a CoV of 5%. Finally, the radius at the corner of the particle was measured in the SEM and showed a maximum CoV of 50%. Monodispersity was achieved for all metrics except sharpness. We were concerned with the poor performance of the corner radius, because sharp corners are expected to be necessary for the application of microparticles that self-assemble into diagnostic devices. However, the average corner radius was only 4.6 µm. Unlike the other metrics this is approaching the resolution limits of the system. Therefore, it is sensitive to manual adjustments to focus and future work should automate the focal adjustment to improve the CoV. The second possibility is the polymerization time was not tuned to ensure full polymerization at the corners so variation during polymerization is effecting corner radius. Moreover, contributions from uneven illumination may contribute to uneven polymerization across the field of view but based on measurements taken in Chapter I, it seems that tuning polymerization time and focal adjustments would improve the monodispersity of the microparticles corner radius. Further improvement could also be found by removing the particles synthesized during the first five cycles when the system focus and flow rates are being adjusted. This certainly contributes to variation since in this case. Finally, we hypothesize that shrinkage occurs in the hydrogel between synthesis and measurement. Evidence for this is found by observing the surface texture of the particles in Figure 35. Ripples appear on the surface of the particles that are indicative of volumetric shrinkage. If this is the case it is could explain why the average particle edge length is smaller than the expected length and may introduce variations into the particle population. Causes may come from ambient conditions or could have been caused by pressure difference in the evacuated SEM chamber.

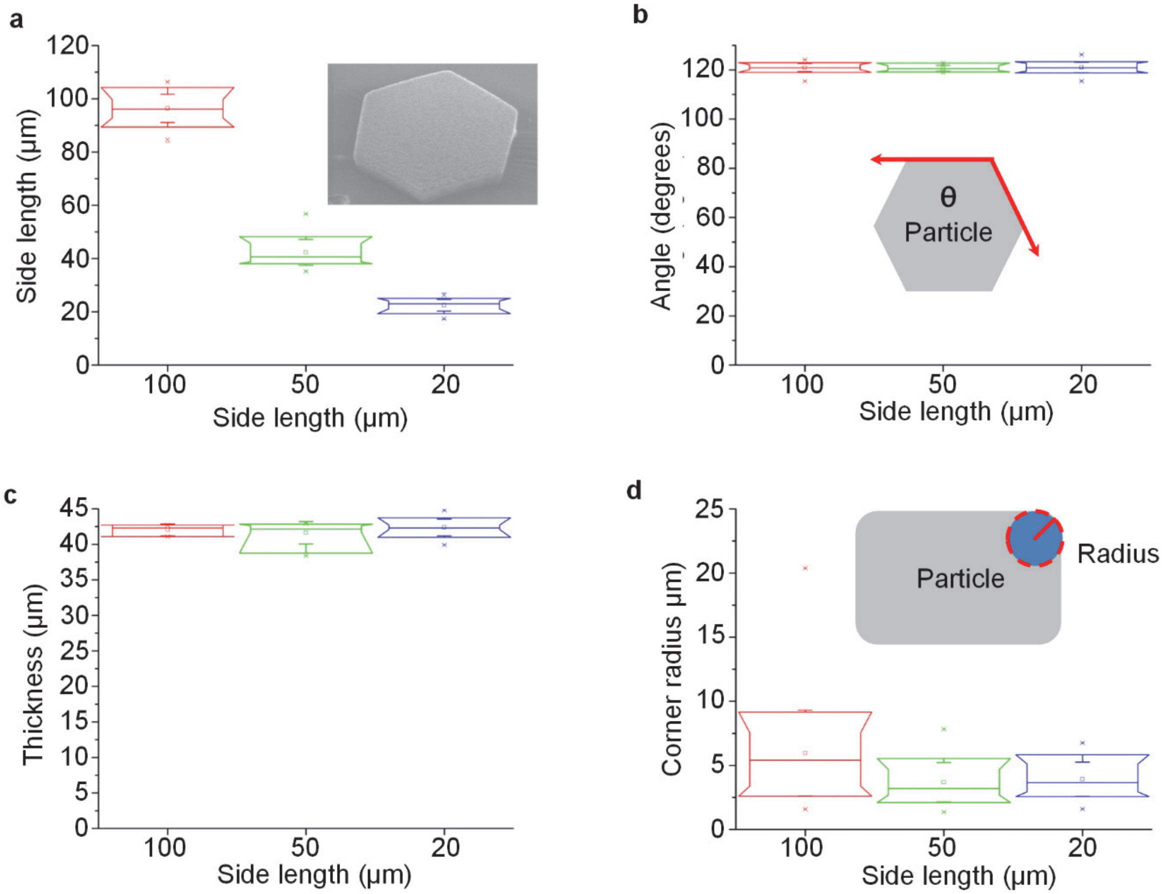


Figure 36. Measurement of particle monodispersity. a) Particle dimensions in microns as measured for three particle sizes. b) The angle between edges of a hexagonal shape for 100 µm, 50 µm and 20 µm particles. c) Height measurements for each particle size synthesized on a microfluidic device fabricated from a 30 µm deep mold. d) Plot of corner radius by particle size. The corner radius was measured in SEM by annotation.

Next, particles were fabricated to demonstrate the ability to rapidly control particle shape using the DMD discussed in Chapter II (Figure 37). In these examples we show two gear sets being fabricated where the first is synthesized, pushed out of the system and then the second mask is immediately projected to fabricate a different set of gears. This is followed by a set of fluorescent particles that have the Michigan M patterned as a cut-out in their center. Fluorescent dye was added to the solution before synthesis. Finally, a number of hexagonal and lizard shaped particles are assembled in a microfluidic channel after synthesis.

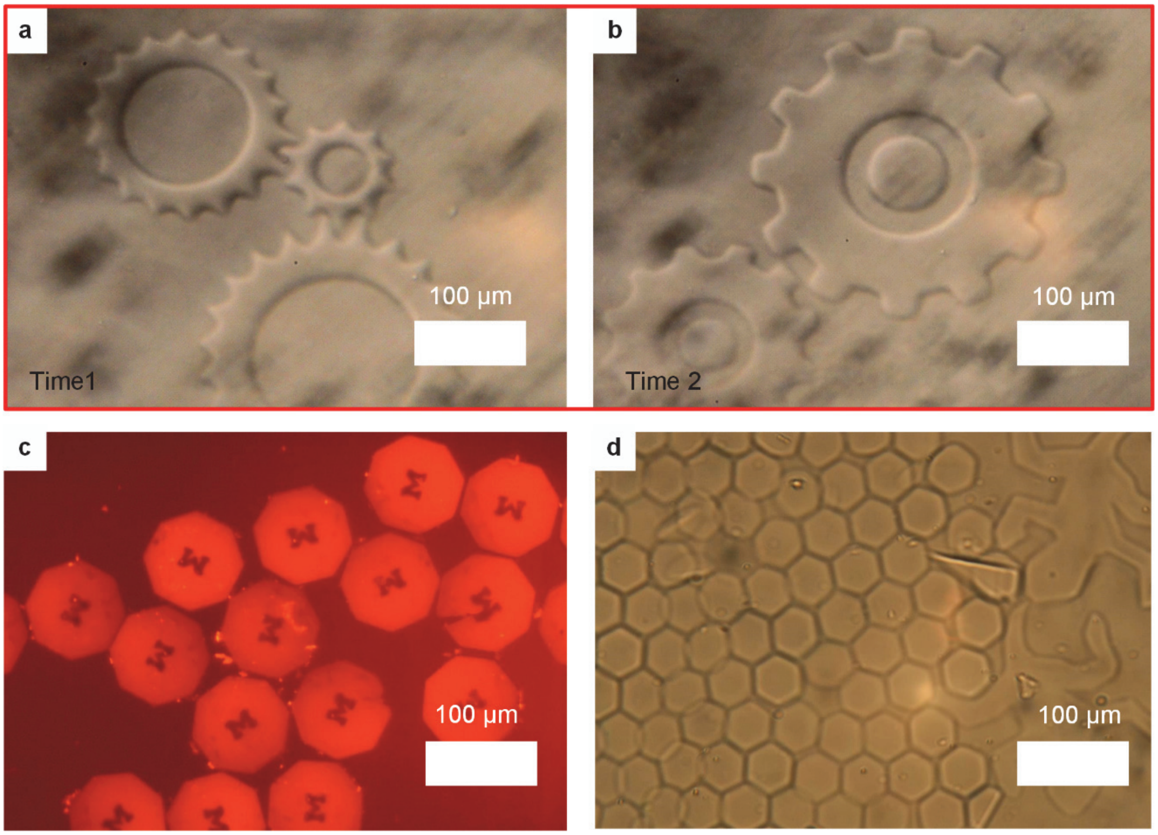


Figure 37. Examples of particle fabrication. a) Example set of gear-shaped particles fabricated in PEG-DA using the maskless system. b) After expelling the previous gears a new type of gear set was fabricated to demonstrate the maskless capability. c) Examples of fluorescent Michigan particles synthesized with the system. d) Hexagonal and Lizard shaped particles packing into a fluidic device.

Other functional types of particles can be fabricated such as the example in Figure 38 where iron nanoparticles were suspended in the PEG-DA solution [54]. After polymerization the synthesized composite microparticle was removed and suspended in water. The orientation of the particle could be controlled by moving a magnet around the edge of the dish. This type of particle could be oriented during analysis of multiplexed assays or as will be discussed in Chapter VII during assembly into tissue like structures.

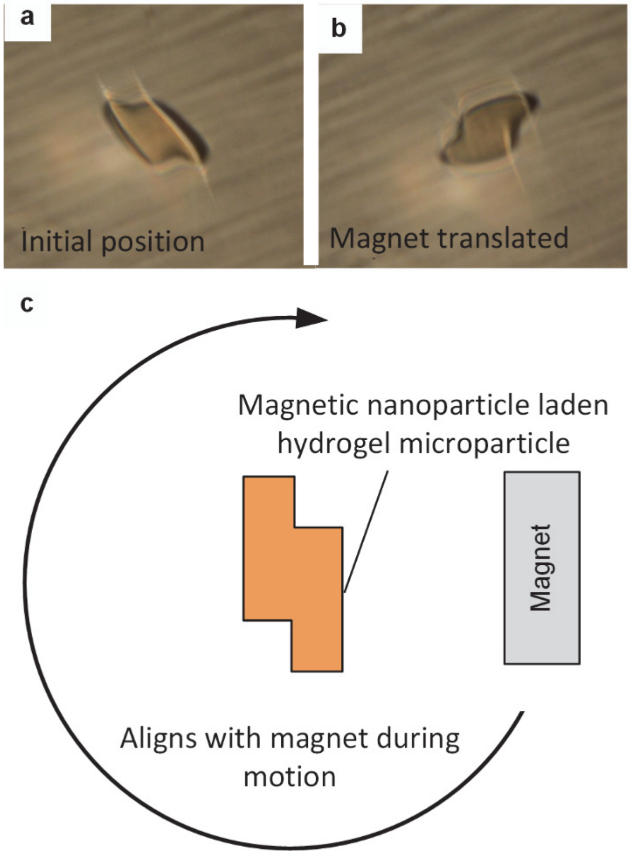

a
b
Initial position
Magnet translated
c
Magnetic nanoparticle laden hydrogel microparticle
Magnet
Aligns with magnet during motion

Figure 38. Iron nanoparticle laden hydrogel microparticle example. a) The starting location of the particle. b) The location after rotating a magnet around the particle to reorient it. c) Schematic describing how the magnet was moved in a circle around the particle.

Throughput of microfluidic lithography

In order to use these particles in either application requires high throughput synthesis. As discussed in a report by Transparency Market Research the multiplexed diagnostic industry is expected to reach 9.8 B by 2019. This valuation is dependent upon increasing the density of particles used in multiplex diagnostics. To do so requires complimentary advances in decreased particle size and increased throughput. In stop flow lithography throughput is governed by the cycle time and the number of particles that can be fabricated per cycle. Where the cycle time is a combination of exposure time (α), pump time (β) and flow stopping time (τ). Therefore, the number of cycles per time unit is the inverse of their sum. The throughput per cycle is a described by the ratio of the exposable area of the substrate (F) and the particle cross sectional area (p). The resulting term is multiplied by the percentage fill factor that describes what proportion of the DMD is filled with particles. The throughput is accordingly:

$$T = \left(\frac{1}{\alpha+\beta+\tau}\right)\left(\frac{F\omega}{p}\right)$$ Equation 7

All of the values shown are independent of the microfluidic channel design except for the flow stopping time. Because the microfluidic channel is elastic it deforms when the flow creates a pressure gradient when pumped. This results in a delay in the flow velocity dropping back to zero. This deformation is dependent on the properties of the liquid, the microfluidic device material and its dimensions. For example, as stiffness decreases the channel will deform more due to the same pressure drop and take long for the flow to stop. The relationship between microfluidic channel dimensions and stop times was analyzed in detail by Dendekuri et al. from which they derived Equation 8. The stop time is approximately equal to the viscosity of the liquid (μ) multiplied by the length of the channel squared (L) and the width (W) divided by Young's modulus of the PDMS (E) and the channel height (H).

$$\tau^t \sim \frac{\mu L^2 W}{EH^3}$$ Equation 8

This can be combined with Equation 7 to relate the throughput to the design of the microfluidic device. We assume that the channel dimensions will be greater than the FOV or if they are not then multiple channels will be used and the exposable area will be calculated by adjusting ω.

$$T = \left(\frac{1}{\alpha+\beta+\frac{\mu L^2 W}{EH^3}}\right)\left(\frac{F\omega}{p}\right)$$ Equation 9

To calculate the throughput of the system according to Equation 9 we experimentally find the exposure time, pump time and stop time. The other values are calculated based on the system design. Mechanical limits of the shutter that is used to actuate the UV light limit α to 0.05 s which was also found to be a good exposure time for particle fabrication. The pump time (β) is controlled by the high speed valve and actuation was measured as fast as 0.002 s. However, to give the flow time to push fabricated particles from the field of view a pump time of 0.025 s was more appropriate.

Next, the stopping time of the flow in the channel was measured while changing channel length, pump time and channel width as shown in Figure 39. Channel height was 5 µm for all experiments because our goal was to produce small particles and because according to Equation 9 this would have a long stop time which we want to try to overcome by tuning the other device parameters. First we examined the effect of changing channel length. The channel length was varied from 0.25 cm to 1 cm for a 500 µm wide, 5 µm tall channel with a pump time of 0.025 s and flow pressure of 3 psi. As the channel length increased beyond 0.75 cm the stop time increased to 20 seconds, however the shortest stop time was 1 second for a channel length of 0.25 cm. We then tuned the channel width in the same manner as shown in Figure 39c. The stop time was shortest for a 200 µm wide channels at 13.2 seconds. Lastly, we examined the effect of changing the pressure pump time or pressure pulse duration from 0.01 s, 0.05 s and 0.1 s. The fastest stop time was the longest pulse (0.1 s) and required 15 seconds to slow to a stop.

Finally, to finish deriving the expected throughput the FOV is calculated according to a 20X objective with a FOV of 1 mm x 1 mm. The fill factor (ω) is calculated using the MTF from Chapter II. Sufficient spacing between the particles is necessary such that the contrast between two particle projections is 30% or greater. The contrast is worse at the edge of the FOV so the far field Tangential MTF is used indicating that the spacing between projection must be 2.5 µm or greater. Assuming square particles we apply a safety factor of 100% and give a minimum spacing (R) of 5 µm. Therefore p is equal to $(l + R)^2$ where l is the length of the particle. The fill factor can be chosen arbitrarily to reduce the number of particles printed.

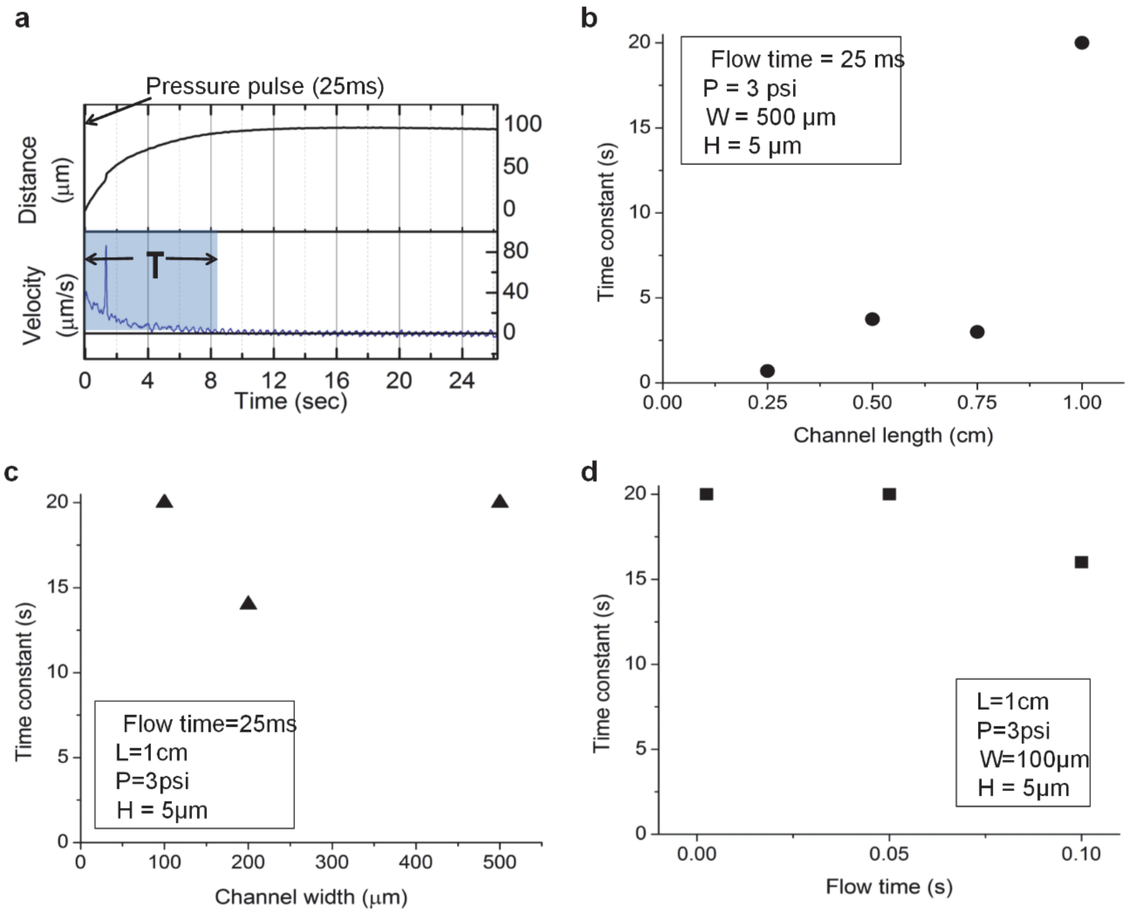


Figure 39. Flow stopping time constant values. a) Stop time constant measurement method. b) Stop time constant measured while changing channel length. c) Stop time constant measured while changing channel width. d) Stop time constant measured while changing the pressure pulse or flow time.

Now the throughput is calculated for height (1 μm, 10 μm, 20 μm, 40 μm, 60 μm, 80 μm) versus particle size with an assumed channel length of 0.25 cm, width of 1 mm and pump time 0.1 s. Finally, the FOV is set to 1 mm^2, exposure time of 0.05 s and a fill factor of 1. From the throughput calculation we observe that between channel heights of 1 μm and 10 μm throughput decreases two orders of magnitude. Therefore, we need to identify a way to achieve a similar stopping time as 80 μm tall channels but when making 5 μm thick microparticles.

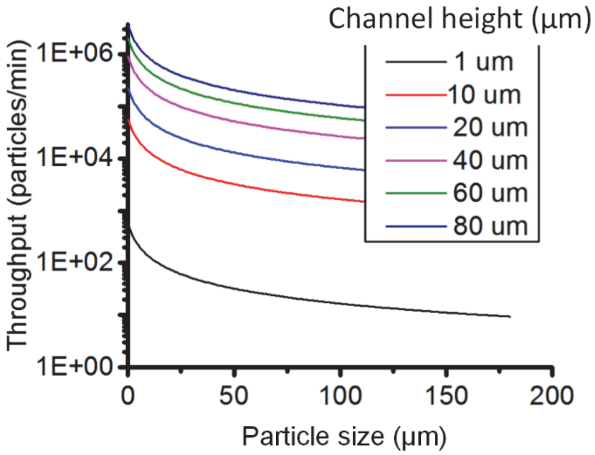


Figure 40. Throughput of the system as a function of particle size and channel height as it relates to the stopping time.

Increased throughput due to microfluidic channel design and scaling

To overcome the limitations to stoppage time and therefore increase throughput, we hypothesized that by fabricating particles on pillars inside a microfluidic channel the stopping time on the pillar would equal that of the full depth of the microfluidic channel. This assumption imagines that the pillar acts like a small microfluidic channel with a short length and width. So, we attempted to fabricate a microfluidic device as seen in Figure 41 with 30 µm wide circular pillars and a 5 µm gap between the top of the pillar and the channel.

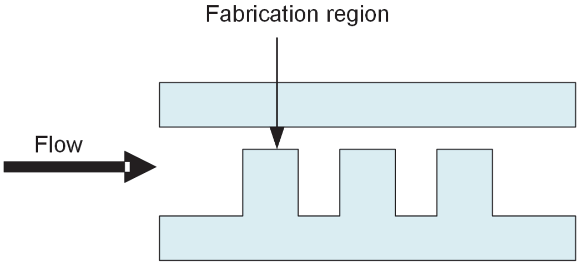


Figure 41. High-level schematic of how pillars are positioned inside the fluid flow to enable fabrication on top of the pillars.

However we found it difficult to fabricate 1 mm wide microfluidic channels with pillars that were 5 µm below the top of the channel. To address this issue we added a diagphram to the device so that we could pneumatically control the height of the channel and increase the thickness of the particle if needed. To do so we developed the device design shown in Figure 42.

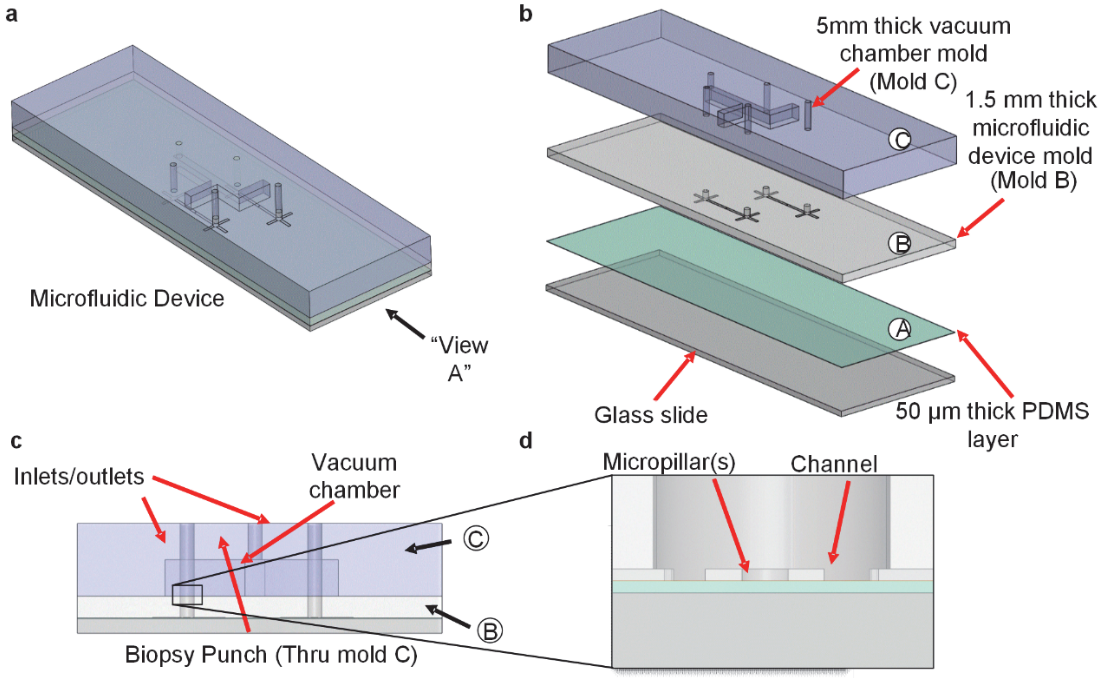


Figure 42. Fabrication of the pillared microfluidic channel. a) Isometric view of the microfluidic device. b) Exploded view showing the layers of the device and diaphragm connection. c) A side view of the position of the pillars.

The fabrication of this device was performed in two steps. First, the pillared portion of the device is fabricated. The fabrication procedure for the pillar portion of the microfluidic device is shown in Figure 43. The dry film resist was from Dupont model MX5000 with a resist thickness of 10 µm.

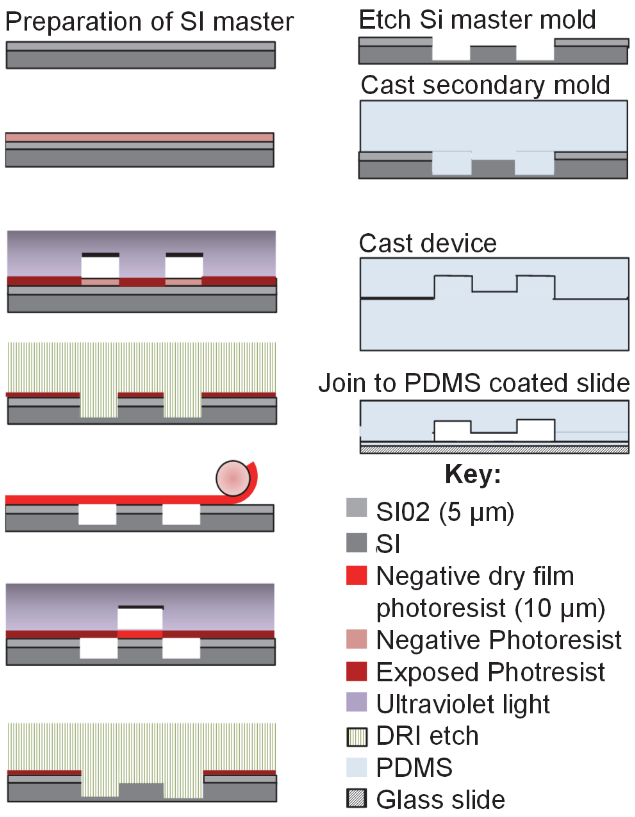


Figure 43. Process for fabricating high throughput microfluidic device.

During fabrication, the master Si mold was imaged to determine the quality of the etching process. It was observed as shown in Figure 44e that some debris from the etching process remained but they were sufficiently small that no attempt was made to remove them beyond a wash. Several of the designs can be seen in the SEM image, including ellipses, circles and tear drops. Our reasoning was that the tear drops may deform less than the circles when the flow is pulsed.

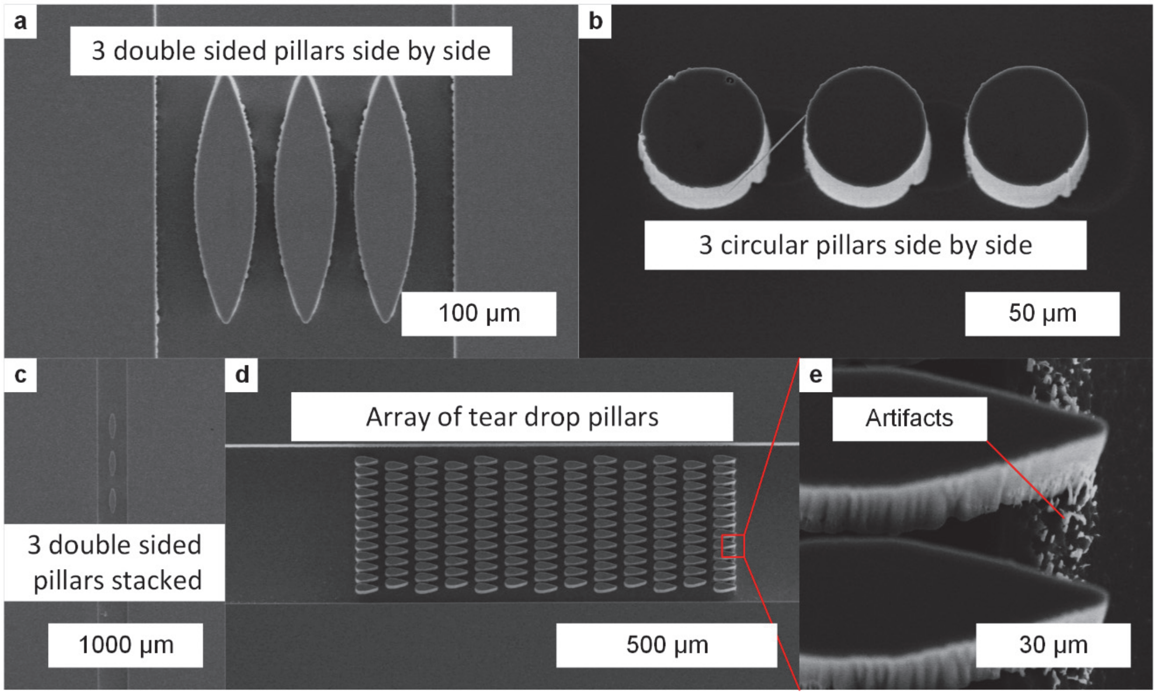

a
3 double sided pillars side by side
100 µm
b
3 circular pillars side by side
50 µm
c
3 double sided pillars stacked
1000 µm
d
Array of tear drop pillars
500 µm
e
Artifacts
30 µm

Figure 44. SEM images of the master mold. a) SEM image SI mold of 3 elongated pillars side by side in a 200 µm channel. b) SEM image of 3 circular pillars side by side in a 200 µm channel. c) SEM image of 3 elongated pillars placed along the channel. d) An array of tear drop pillars designed for high density throughput. e) A close up of the artifacts that remain after DREI etching the mold.

Next the master mold is replicated in PDMS to enable oxygen to diffuse into the pillars inhibiting the particles form adhering to the device. A glass slide is placed on top of the Si wafer that contains the micro pillar device array. Next, a vacuum chamber design around the microchannels is traced; this acts as a guide when preparing the negative mold for the vacuum chamber (mold C). Then, a 3M double-sided tape is used to cover the vacuum chamber design previously drawn. This glass slide/tape assembly is what will be used to cast Mold C from. The mold is coarse but sufficient. After checking the orientation of the channel; the assembly is set aside. A set of glass slides is cleaned with IPA then attached attached to the spin coater using double sided tape. The oven is pre-heated to 80 °C. PDMS is mixed at a 1:5 volume ratio of curing agent to base. The final volume of PDMS, sufficient to fill the mold prepared in step 1 to a height of 1.5 mm and the mold prepared in step 2 to a height of 5 mm, was 15 ml. The mixture is placed into a vacuum oven to degas. Then, the PDMS is mixed at a 1:20 volume ratio of curing agent to base. Once mixed, the mixture is dispensed onto the glass slides, then spun at 2000 RPM for 30 seconds, and poured into the molds to create mold B and C. Mold C is placed into an 80 °C oven for 10 minutes, then set aside the oven. The glass slide is placed into the oven for 10 minutes, and Mold B for 20 minutes. While the glass slides and mold B are in the oven, mold C is carefully de-molded and the tape is removed as well. A $\emptyset 1mm$ biopsy punch is used to create the vacuum chamber inlet. Next, the glass slides is removed from the oven and set aside for 10 minutes. After ten 10 minutes mold B is removed from the oven. Now, mold B is de-molded. Now, Mold B and C are aligned and adhered to one another. A $\emptyset 1mm$ biopsy punch is used to create the microfluidic inlet/outlet holes.

Finally, the assembly (mold B and mold C) is adhered to the PDMS coated glass slides. Control over the glass slide curing is important to prevent uncured PDMS from flowing into the channels during this step. The entire chip assembly is placed into the oven at 80 ℃ for 8-12 hours. The results of this process are shown in Figure 45b and is linked to the mask design for clarity in Figure 45a. One vacuum inlet controls the diaphragm on two channels.

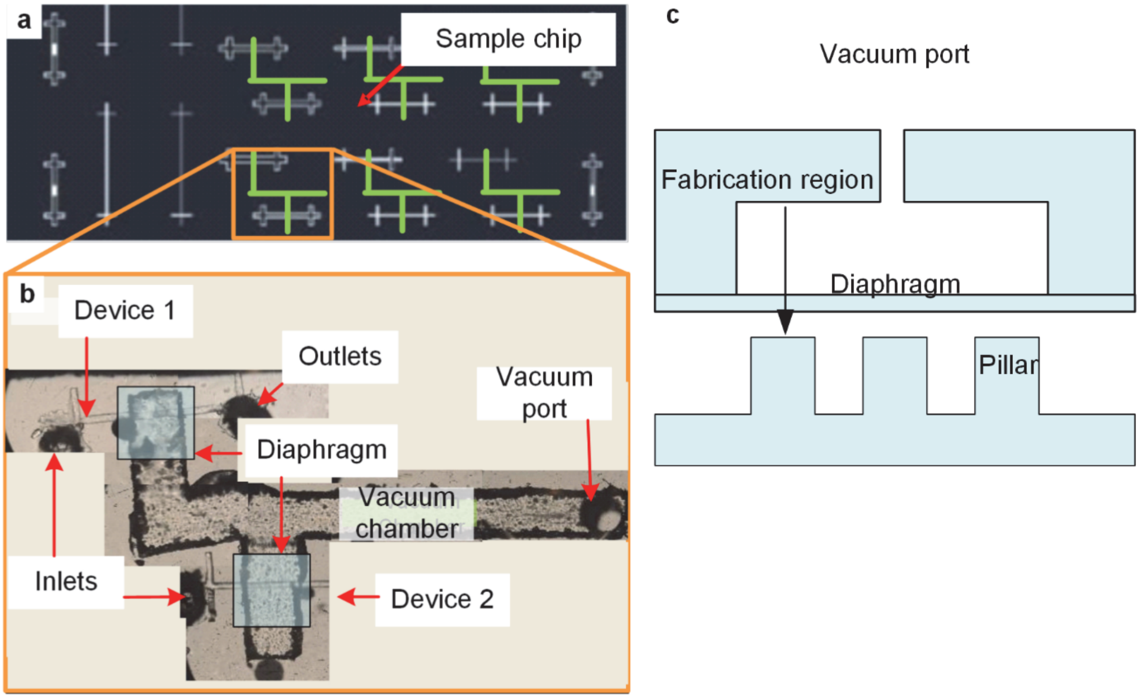


Figure 45. The diaphragm design and assembly. a) Layout of pillared diaphragm driven microfluidic device. b) Many images stitched to show the full device. c) Schematic showing relationship of the diaphragm to the pillared channel.

Various devices were fabricated and tested. Device type 11 (device numbers are listed with their properties in 0) is shown in Figure 46a. Using this device, the stop time was measured using 2.5 µm PS beads mixed into the PEG-DA solution. The pressure was pulsed for 25 ms and then the stop time was measured as described previously. Then, 10 µm microparticles (enclosed in the red square) were fabricated on top of pillars, Figure 46b. The top image shows the pillar without a particle and the bottom image shows the pillar with the particle that has stopped on top of it.

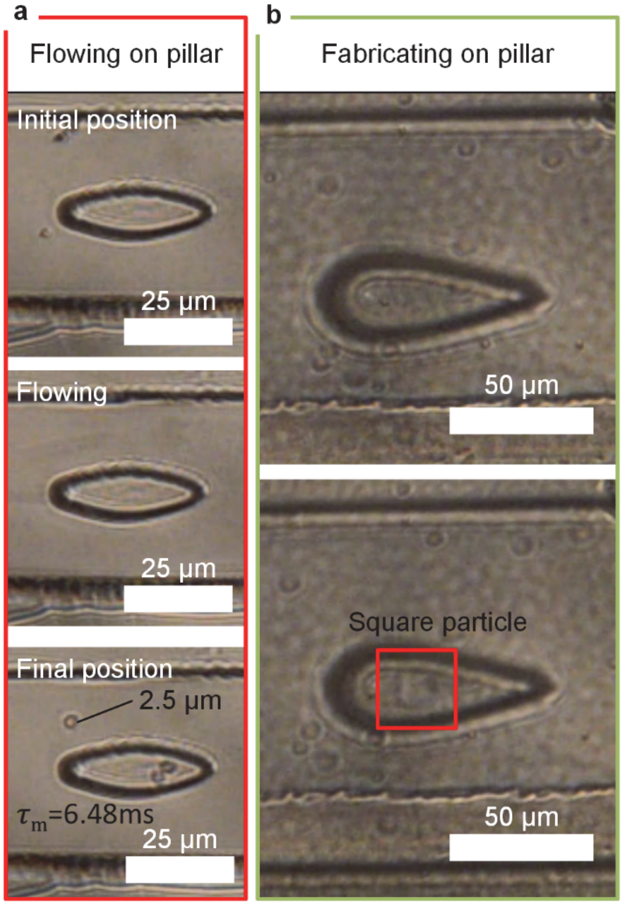


Figure 46. Measuring the performance of the new microfluidic device. a) Verification of stop time using 2.5 μm beads in solution. The beads were flown in the device and imaged as they stopped on top of the pillar. b) Demonstration of synthesizing particles on top of the pillar structure.

Plotting the stop time for a device with the parameters of the full depth of the pillared channel we see that the stop time for this device is very good with a value of 6.48 milliseconds. Comparing the full height channel (5 μm) and the pillared experiment (5 μm gap in 80 μm channel) we observe that using the micropillared device may manufacture up to 2.5E6 particles which is 1500X the throughput of the original design. We believe this confirms our hypothesis that the stop time on pillars acts in the same manner as the chip it is housed on.

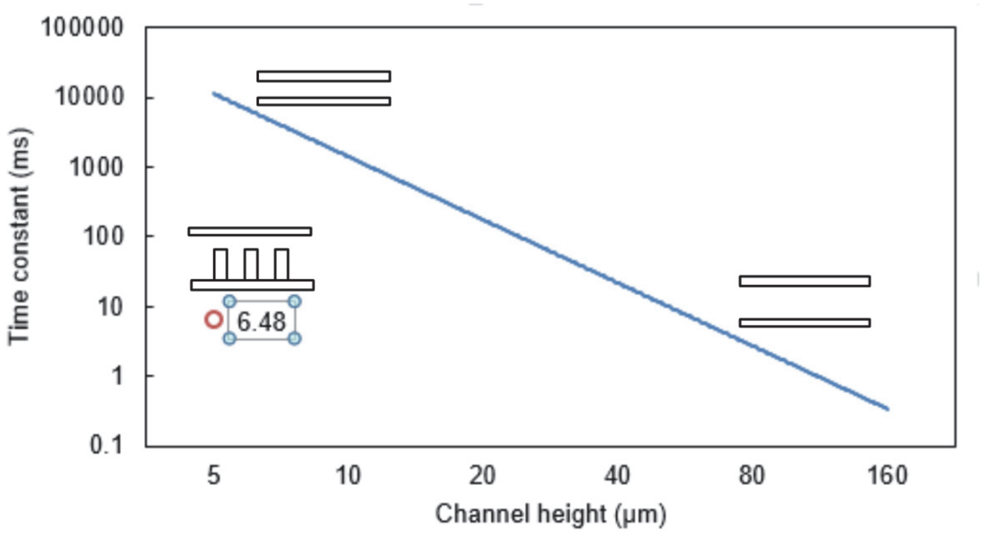


Figure 47. Comparison of the time constant for full depth channels and pillared channels which perform as in the same manner as an 80 µm deep channel.

Conclusion

This chapter demonstrated the application of our maskless lithography system to rapidly fabricate PEG-DA microparticles with arbitrary 2D shape and to improve throughput of small particles. Maskless microfluidic lithography is an attractive tool to manufacture polymer hydrogel microparticles. For example, it can be used to encapsulate an object such as a cell within a functional shape (barcode, sensing). In addition, the method has demonstrated the capability of monodisperse synthesis of microparticles. We observed CoV of 1.7%, 5%, 10%, 50% for edge length, height, angle and corner radius respectively. Using the system we demonstrated examples of functional microparticles (fluorescing and magnetic particles). Future efforts should study the role of shape and chemistry to enable assembly of these particles on a liquid surface.

Throughput of microparticle synthesis was limited by stop time of the flow as a function of the microfluidic channel height. By redesigning the microfluidic device to use fabricate microparticles on top of pillars in the channel, it was shown that the stop time can be reduced by three orders of magnitude for thin particles. This method is expected to grow continuously in sophistication as collaborators in the field have produced insightful and creative methods of adapting this technique. In particular, it is hoped that the method can be used to further understand the micro-self assembly of microparticles patterned with hydrophobic regions by flowing immiscible photopolymers through the microfluidic device [55], [56].

CHAPTER IV. THE STUDY OF C. ELEGAN NEMATODES BEHAVIORAL RESPONSE TO DIRECT-WRITE ENCAPSULATION, MANIPULATION AND IN-SITU SYNTHESIS OF STRUCTURES

Abstract

The nematode *Caenorhabditis elegans* has proven to be a striking model for bio-engineering research. This can be attributed to conserved molecular pathways and physiological mechanisms that provide insight into more complex organisms. New technologies are needed to improve immobilization, isolation, manipulation and breeding of these nematodes to learn about how their behavior and physiology change with time. However, most tools and microfluidics in general must be designed and fabricated according to a fixed intent before the experiment begins, and cannot be modified *in situ*. Here, taking advantage of *C. elegans* as a model organism, we demonstrate a new method, dynamic lithography, used to manipulate worms and modify their microenvironment *in situ*, in a rapid and inexpensive manner. This method, is used to fabricate microfeatures onto agarose gel seeded with *C. elegans* in real time creating a new assay in as little as 12 minutes. To demonstrate the use of this platform we show a classic mobility study reaffirming that proximity of pillared structures increases the swimming speed of *C. elegans* up to 89%. In accordance with other studies we found that the *C. elegans* can solve a hydrogel maze when directed by food twice as well as when no reward is present. Other examples highlight the paradigm of *in situ* fabrication and manipulation using a stylus pad and/or moving parts. I would like to acknowledge the contributions of Dr. Eleni Gorgou and Dr. Daphne Baszopoulou for providing their expertise, time and animals to this collaboration. I also thank Prof. Tim Swager and Elizabeth Sterner for assistance with ATR-FTIR measurements.

Introduction

Microfluidics, and in particular methods to pattern soft materials like PDMS and hydrogels, has revolutionized methods to manipulate cells and small model organisms such as *C. elegans*, *Drosophila* and zebrafish, *in vitro*. However, conventional approaches enable only the fabrication of fixed assay designs, which cannot be changed during experimentation, unless specialized features are designed into the device ahead of time, such as gates and valves requiring complex external hardware [57].

Specifically, regarding *C. elegans*, microfluidic based assays have been successfully used for worm immobilization and imaging, behavioral assays, laser microsurgery and high throughput drug screening [58]–[60]. Despite their elegant and effective design, they are static devices, allowing only one reversible modification. This modification is usually in the form of a pressure change or a stimulus delivery. To achieve this, they have to be connected to external hardware, have integrated synthetic valves (e.g. valves [57], [60]) or have a more complicated multi-layer design. Moreover, multi-layer design requires alignment of features between layers of PDMS or other systems and, being a manual method, suffers from poor reproducibility. Often, these restrictions are imposed by the fabrication method (soft lithography) which, in spite of the boost it has given to developing highly impactful tools for biological studies (e.g enabling micro-scale manipulation of the nematodes via clamps), does not allow for much flexibility in real-time modification and device adaptation. Studies that aim to map learning and behavioral response to brain plasticity, could significantly gain from a tool that would select individual *C. elegans* which show a unique learning capacity.

While, PDMS microfluidic assays are being applied to a growing number of *C. elegans* studies, the standard practice is to use agar plates. Agarose gel is made of porous superstructures made of helical agarose molecules. The initially liquid agar gels between 35-42 ^{o}C. It is accepted that this material does not stress the worms significantly, however, no method of fabricating complex structures made of agarose gel exists.

There is also promise due to the decreasing cost and increasing performance and availability of additive manufacturing techniques, such as fused deposition modeling (FDM) and stereolithography (STL) [61]. However, limitations in resolution and optical clarity have limited its adoption. Moreover, the biocompatibility of common 3D stereolithography resins is still under investigation. The availability of mail-order 3D microfluidics is attractive because it reduces the need for in-house fabrication knowledge but this does not resolve the need to order a new device for each iteration of the experimental process or add flexibility to an individual experiment.

Here, we propose and demonstrate a method for reducing the complexity and time necessary to fabricate *C. elegans* assays as well as enable fundamentally new assay capabilities that rely on real time interaction with the worms. We exhibit how this new technique can be successfully used for dynamic micro-fabrication during the performance of a live *C. elegans* assay.

We aim to provide a platform for biocompatible assay production *in situ*, through the combination of benchtop maskless fabrication, biocompatible material selection, integration of standard agarose gel plates and user feedback.

Materials and Methods

Materials

The photopolymer used for the majority of experiments was 0.5% v/v photoinitiator 2-hydroxy-2methyl-1-phenyl-1-propanone (Darocur 1173, Sigma Aldrich) in poly(ethylene glycol) (400) diacrylate (PEG-DA, Sigma Aldrich) in 80% aqueous solution. Agarose gel was prepared using standard methods outlined in Brenner et al [62]. Other photoinitiators used include Phenylbis(2,4,6-trimethylbenzoyl)-phosphine oxide (Sigma Aldrich) and a custom mixture of 0.01 mM Eosin Y (Sigma Aldrich) and 0.1% TEA (Sigma Aldrich). The structure of the agar gel is shown in Figure 48 to aid in interpreting the results of the Attenuated Total Reflectance – Fourier Transform Infrared Spectroscopy (ATIR-FTIR) data.

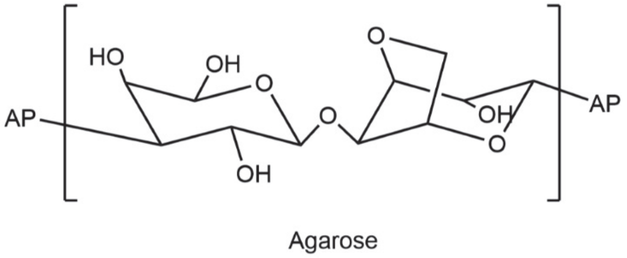


Figure 48. Chemical structure of the agar used in these experiments to aid in interpreting the ATR-FTIR data.

ATR-FTIR measurements of diffusion

Diffusion of the PEG-DA polymer into the agarose gel was measured by ATR-FTIR (Thermo Scientific Nicollet 6700, fitted with an ATR-FTIR sampling accessory with a Germanium crystal). A 500 µm thick sample of agarose was placed on top of the crystal and 10 μl of PEG-DA was pipetted onto the surface of the agarose. A baseline spectrum was taken and re-measured periodically over 20 minutes as shown in Figure 51. Bands were measured against known the location of known chemical expressions. Because the system identifies molecules at a specific depth in the sample we assume that the development of a band is due to diffusion.

Survival assay

Synchronous populations of nematodes were established by allowing 20 adult hermaphrodites to lay eggs for a limited time interval (4–5 h) on nematode growth medium (NGM) plates seeded with *Escherichia coli* OP50. Progeny were grown through the L4 larval stage, and then transferred to fresh NGM-seeded plates with PEG-DA-coated walls (20% PEG-DA in water) at groups of 10–20 nematodes per plate and kept at 20^0C. The first day of adulthood was used as day 0. Animals were transferred to fresh plates every 2–4 days thereafter and were examined every day for pharyngeal pumping and movement in response to a gentle tap with a thin platinum wire or pipet tip, until death. Lifespan was defined as the number of days between day 0 and the last day on which the worm was scored as alive. Worms that died due to internally hatched eggs, desiccation, an extruded gonad or crawling on the edge of the plates, were incorporated into the date set as censored.

Statistical analysis

We used the Origin software package (OriginLab Corporation, Massachusetts, USA), to carry out statistical analysis and to determine lifespan values. The Kaplan Meier estimator was used to evaluate differences between survivals and determine the *p* value.

C. elegans tracking using image processing

Custom C++/Matlab code was written to identify the worm in each frame built using the OpenCV toolkit. Image differencing between frames was used to identify where the nematode had moved and the centroid of the highlighted region on the image was used to mark its position.

Results and discussion

Dynamic fabrication of PEG-DA microstructures on agar plates

For *C. elegans* assays we fabricate microstructures of a common hydrogel (PEG-DA) on substrates of agarose gel (see Methods). The key advance over previous methods of *in vivo* assay fabrication in the use of a maskless lithography system, coupled with custom image analysis software to achieve precise localization of *C. elegans* in assays and track their behavior. The system as described is shown in Figure 49, uses a digital micromirror device (DMD) to generate diffraction limited image projections through a microscope. A ultra-violet (UV) 10W Mercury lamp, at 365nm, is projected off the DMD and through a Mitutoyo objective (2X, 5X, 20X, 50X). The projection and subject is imaged using a Nikon D5100 SLR camera through the same optical train via a beam splitter. Figure 49b shows an example of the resulting structures formed around a *C. elegans* nematode as seen through the CCD. The procedure of fabricating an assay takes three steps. First, an agarose gel plate is dried under Nitrogen for 30 seconds. Then 5 µL of photo-polymer (PEG-DA/H_2O/Photoinitiator) is dispensed using a pipette. The droplet may or may not be covered with a glass cover-slip based on assay intent. When covered a shim of PS Beads or cover glass is used to control the height of the features from 1-1000 µm. Finally, the plate is inserted into the microscope and the pattern is projected onto the photo-polymer curing it in place and adhering it to the agarose gel. Because the fabrication step occurs at the same time as imaging additional exposures can fabricate additional features, and can be stitched to span a larger area. The microstructure design was produced using standard drawing software (Adobe Illustrator).

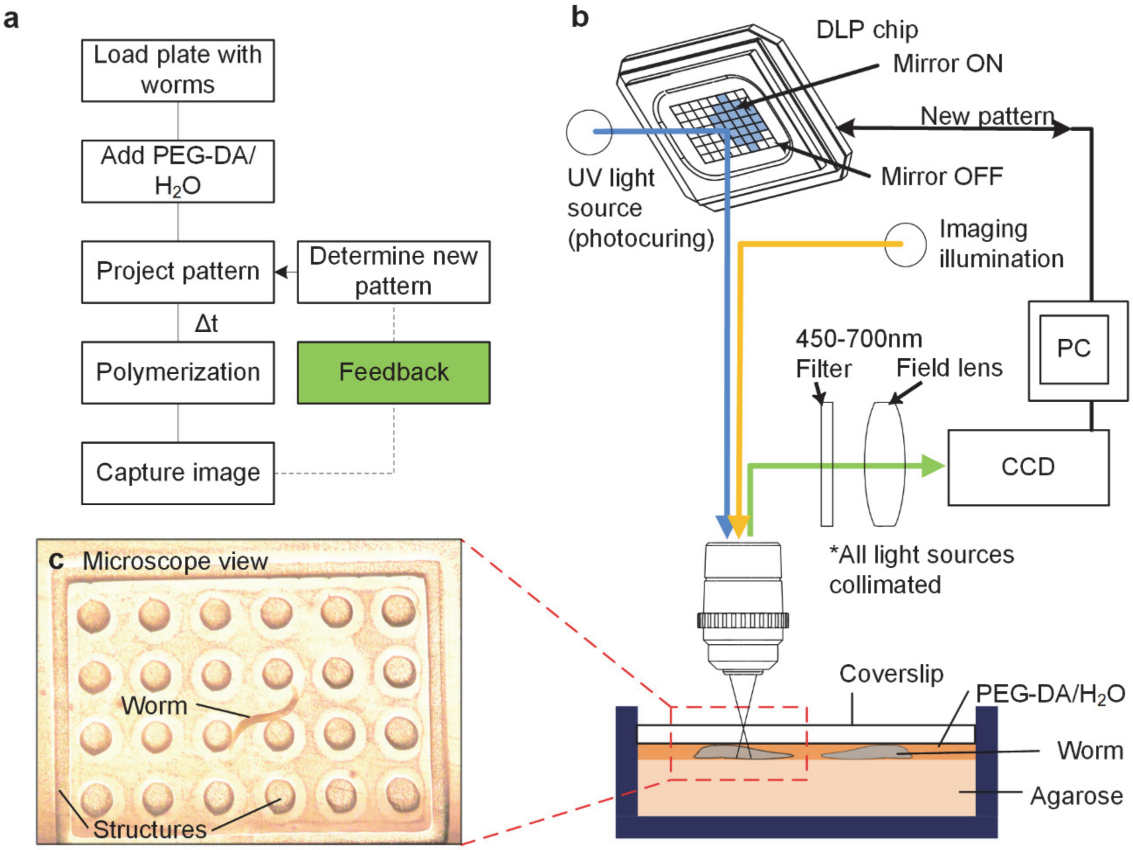


Figure 49. Overview of C. elegans patterning process. a) Flow chart of the patterning process. b) Schematic of the experimental setup for patterning agar plates. Lithography system is a modified microscope comprised of a UV DLP mask and light source for projecting image of desired structure. Real-time video feedback is provided by a CCD. The system is controlled by Labview. c) An example of a structure formed around a C. elegans worm in situ.

The material system comprised of PEG-DA/H_2O/Photoinitiator on Agarose was chosen as a biocompatible platform for the assays. We initially investigated several bio-adhesives and monomers including PEG, Pluronic-F127, Methacrylate and PDMS. We chose PEG-DA for its ease of polymerization, biocompatibility, and hydrogel properties (e.g. oxygen diffusion, softness). It was also attractive to use agarose, which has long been the gold standard growth medium for *C. elegans* worms. The last key ingredient is the Photoinitiator, which initiates polymerization of the PEG-DA monomer into solid structures, and must be selected carefully based on its wavelength sensitivity and biocompatibility. Biocompatibility can be affected by the formation of free radicals during exposure to light or its hydrophobicity. We tested three candidates, including 2-hydroxy-2methyl-1-phenyl-1-propanone (Darocur 1173), Phenylbis(2,4,6-trimethylbenzoyl)-phosphine oxide and a custom mixture of 0.01 mM Eosin Y and 0.1% TEA. The latter two initiate in white light while the former initiates in NUV. Titrating concentrations of Phenylbis(2,4,6-trimethylbenzoyl)-phosphine oxide into PEG-DA yielded no solution that would photopolymerize and be biocompatible. The Eosin Y solution would not adhere to the agarose gel but did show promise as a biocompatible photopolymer with *C. elegans* to be studied later. Finally, the NUV Photoinitiator showed the ability to both adhere and be compatible with living *C. elegans*. Results are shown in Figure 50, for a 2X, 5X, 20X and 50X objective across a range of feature heights from 1 µm to 1000 µm. The exposure time for each experiment was controlled using a shutter with a mechanically limited minimum exposure time of 50 ms. The length of time of the exposure determined the thickness of the cured PEG-DA layer. The optimal exposure time for each objective and thickness was chosen for the highest resolution line pair measured. As shown in Figure 50c as thickness (feature height) decreased the exposure time increased due to oxygen inhibition at very small heights [35].

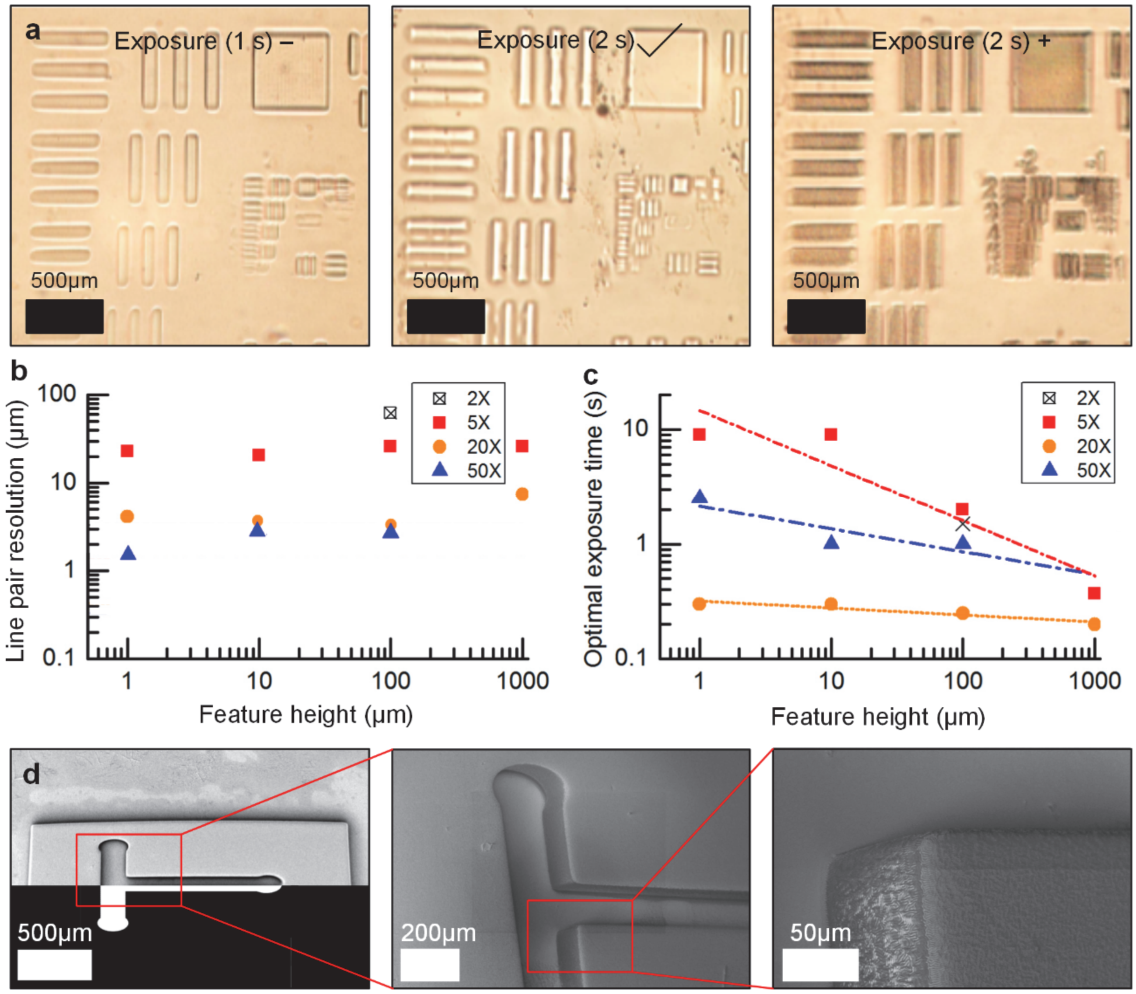


Figure 50. Resolution and exposure data for PEG-DA on agar. a) Optical images of fabricated test pattern (USAF 1951 pattern) used to determine line pair resolution, with examples (from left to right) of under-exposed, properly exposed, and over-exposed results. b) Relationship between line pair resolution and feature height for different objective magnifications indicated in the legend. c) Relationship between optimal exposure time and feature height, determined using smallest line pair fabricated in each case. The lines are linear fits using the data points. d) SEM Images of T-mazes as used in Figure 8, including (from left to right) the full maze in the upper half and the projected mask overlayed in the lower half, the sub-image shows one leg of the maze, and the right sub-image shows the corner of a wall feature.

Unfortunately, we found that features smaller than 50 μm did not remain adhered to the PEG-DA upon rinsing. As a result, this determined the minimum feature size we could use in the *C. elegans* assays, and this is still significantly smaller than the worm body length (~1000 μm). This limit is related to the strength of the PEG-DA and how the photopolymer diffuses into the agarose gel, thereby influencing the adhesion strength between the two materials. Diffusion of the PEG-DA polymer into the agarose gel was measured by ATR-FTIR A baseline spectrum was taken and re-measured periodically over 20 minutes as shown in Figure 51. Figure 51b compares the first and last measurement showing the development of a band at 1714 wavenumbers due to the presence of an ester carbonyl. Our interpretation is this band development indicates diffusion because this group can only be found in the PEG-DA monomer which must have diffused into the agarose to be detected. Polymerization of diffused PEG-DA enables adhesion to the substrate proportional to the structure cross sectional area.

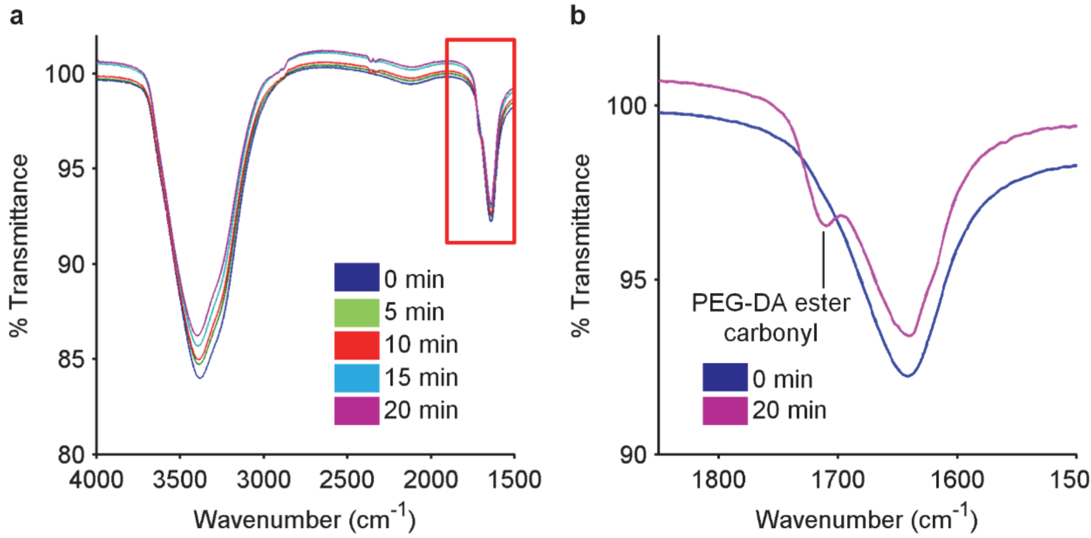


Figure 51. Results of FTIR measurement showing diffusion of PEG-DA into agar.

Figure 50d, shows SEM images of an example structure fabricated out of PEG-DA on agarose gel alongside the mask projected. Each successive frame shows a close-up detail of the corner of a T structure imaged at 45˚. The corner radius of the maze fabricated with a 5X objective is 10µm.

Survival of C. elegans

Before performing live worm assays using the dynamic lithography method, it was essential to evaluate the viability of the worms in the presence of the PEG-DA solution on agar plates. To this end, and since PEG-DA is a relatively new material for *C. elegan* assays, we tested its effect on worms in various concentrations. Our results show that, diluted to 20% PEG-DA and 80% water does not have any effect on population longevity. Increased concentrations of PEG-DA, 30%, 50%, 75% and 100% appear to induce stress on worms, expressed in reduced motility, increased curling and signs of dehydration, resulting in death after 3-10 minutes. These results, in combination to the fact that the suitable photo-initiator, as explained above, is insoluble at PEG-DA concentrations below 15%, led us to choose 20% as the appropriate concentration.

The viability test was performed on NGM plates non seeded with *Escherichia coli* OP50. Each plate was seeded with adult worms of various ages and a droplet of the material under investigation was placed onto them. We incubated the worms within the PEG-DA solution droplet at 20 ^{o}C and checked on the worms every 1 min.

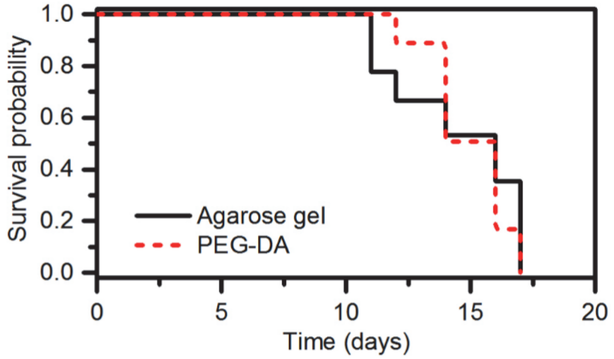


Figure 52. Worm viability comparing agarose gel substrate and PEG-DA/agarose substrate.

Despite our observation showing no apparent effect of 20% PEG-DA in *C. elegans* short-term viability, we performed a survival assay to verify that this is also the case with the worm's lifespan. To this end, we used NGM plates, seeded with *Escherichia coli* OP50, rinsed in PEG-DA, on which we built simple structures from photopolymerized PEG-DA (i.e. rings, rectangular). We then placed worms on them, as described in the Methods section. This way we tested the effect of both cured and traces of non-cured PEG-DA. As shown in Figure 52, the presence of PEG-DA has no effect on long-term viability of *C. elegans*. Worms also show normal locomotion patterns and motility, no dehydration signs and normal life cycle. These results encourage us to use 20% PEG-DA in our technique and also permit us to use it for behavioral studies, since no major modification of their behavior or activity was observed during their exposure to it.

Manipulation of C. elegans by dynamic confinement

To demonstrate the capability to build locomotive assay designs *in situ*, we first designed experiments that could be compared to previous studies [63] using static assays. Specifically we fabricated pillars and confinement channels. Locomotive assays are useful for identifying how the dynamics of the worm change with its environment. Starting with an agar plate loaded with worms, a single worm was isolated from the worm population in a 5 by 3 mm box that was 100 μm deep, shown in Figure 53a by a single exposure (1 s) from the lithography system. A second exposure generated an array of pillars 100 μm in diameter with a spacing of 100 μm between pillar centers 4 seconds after the first assay. The array of pillars mimics the period length of the worm's motion enabling it to move much faster than swimming freely. The path of the worm in the pillars over a duration of 200 seconds is shown in Figure 53b. At another time, a third exposure confined the worm within a corrugated microchannel approximately 100 μm wide.

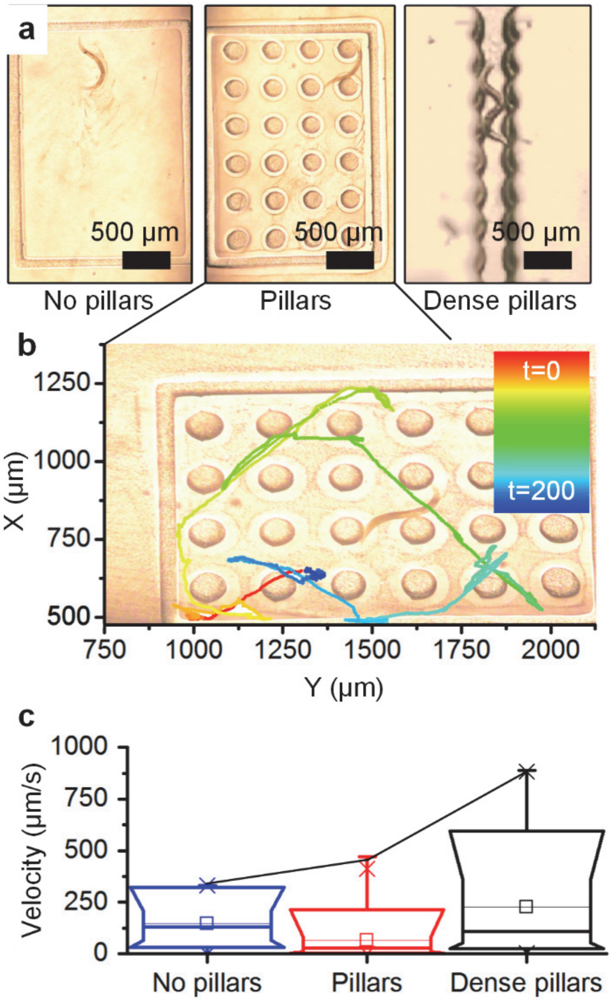


Figure 53. In situ photopolymerization of microstructures resulting in physical confinement of a C. elegans. a) Three exemplary assays built sequentially (from left to right): an open frame, an array of micropillars (100 μm diameter), and a rippled microchannel created by gradually increasing the density of polymerized micropillars. b) Tracking of the worm motion over a time period of 200s. c) Box-whisker plots of velocity in each configuration, showing that sequential confinement increases mobility. The whiskers represent the max velocity of the worm in each scenario and are connected to highlight the trend.

By comparing the average speed of the worm in each assay design (Figure 53c), we conclude that the worm swims faster in confinement. For example, comparing pillars spaced with 100 microns between them and overlapping pillars we found an 89% increase maximum velocity.

We observed that the worms take advantage of the pillars to offer leverage during locomotion. When the pillars were overlapped the maximum swimming speed of the worm increased from 330 µm/s, free swimming, to 890 µm/s corroborating physical simulation and experimentation by Majmudar et al. [64]. The periodicity is easily tunable by simply changing the image projected to study a broad range of locomotive behaviors. The total time to develop three independent assays around a freely swimming worm is reported as 30 seconds.

Free-floating structure and mechanical elements

Free floating features cannot be fabricated in traditional PDMS systems. By controlling the exposure time and focal plane we are able to specify an exposure depth from the surface, thereby creating free floating features. Here, we demonstrate the fabrication of a lever fabricated around a pin. Video was taken of the *C. elegans* interacting with the lever and rotating it until it hit a wall (Figure 54). Next, the worm wedged its body between the outer wall and the hinge. Once in that position it straightened its body, pushing the hinge up and around the pin. This motion was repetitive, symmetric and touch reactive. The rotating lever is an example while other types of mechanisms can be fabricated as well.

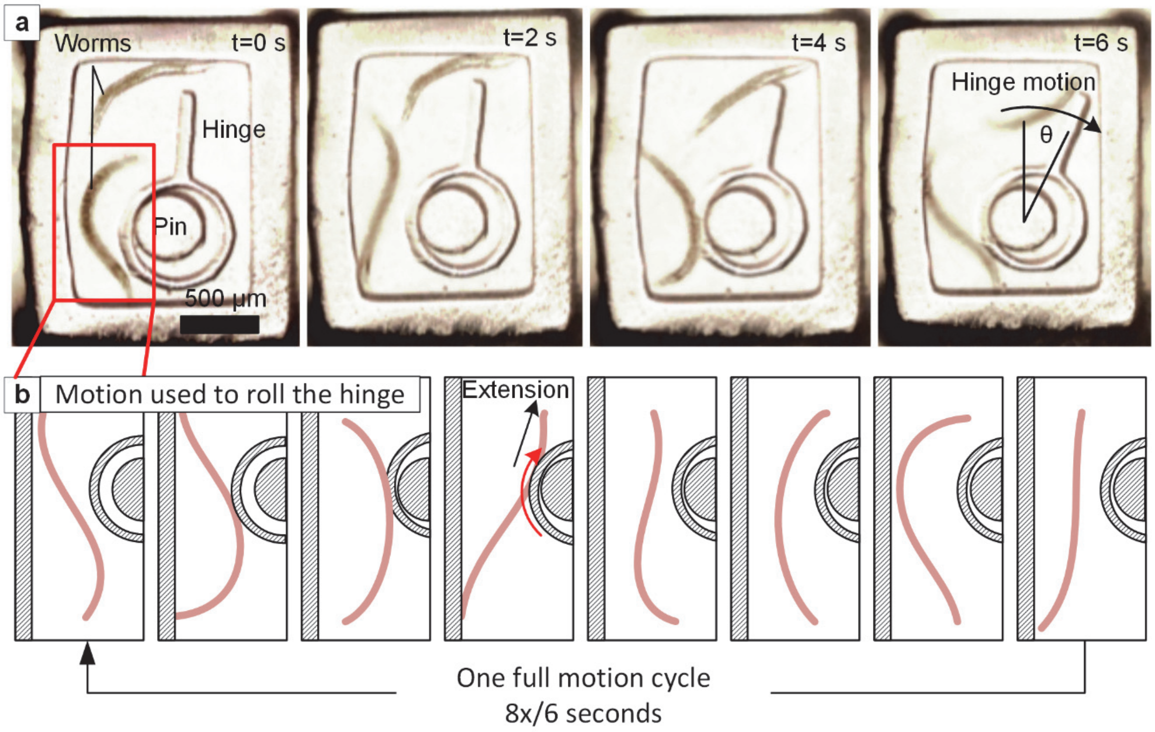


Figure 54. Worm interaction with a simple hinge-pin mechanism patterned within a millimeter-scale area. a) Sequential frames show the worm rotating a hinge around a pin. b) Schematics of worm motion as observed in the experiment, which is also shown in Video S2. When the worm contacts two points spanning from the frame to the hinge, it extends its body, exerting force on the hinge and causing it to rotate.

Free-hand drawing of features within the assay

In addition to projecting fixed geometric patterns during the assay process, the system can generate freeform input from the researcher. One example is the use of a tablet to draw an array of pillars (Figure 55b). We integrated a touch pad into the system so that the researchers touch would correspond to projection and fabrication of microstructures (Figure 55a). In this case the input from the tablet is converted in real-time into a projected mask using a custom LabVIEW software. The latency between the researchers moving their finger/pen across the screen until the light is projected is >250ms. The limit to the speed at which this process can be driven is coupled to the latency in response of the computer to recognize the input, transfer of the input to the projector, kinetics of the DMD mirrors and shutter latency. The rate-limiting step is currently the transfer of the image to the projector due to the manufacturer software for this module. If this were not the case, the fundamental limitation would be the photopolymerization rate or the reaction time of the researcher.

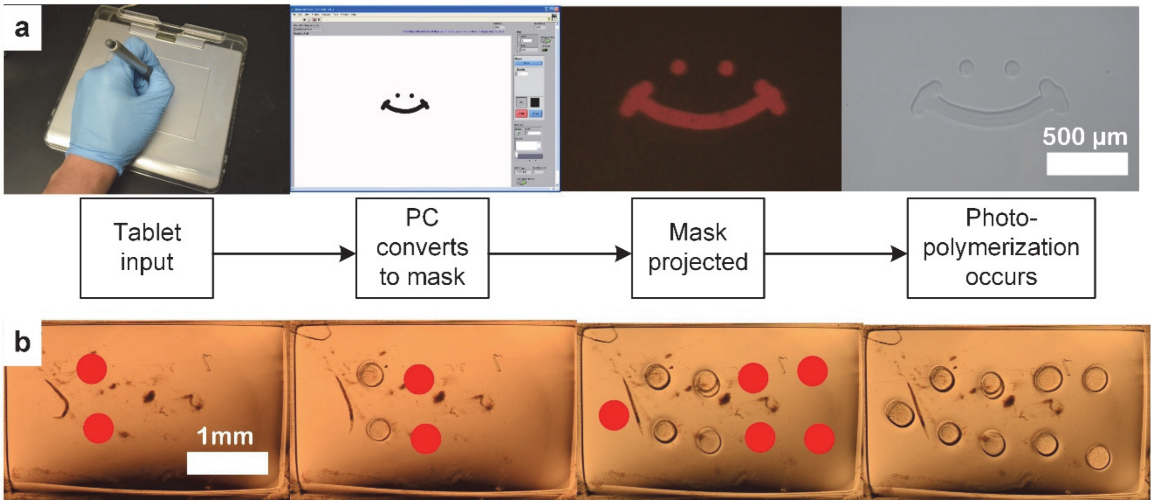


Figure 55. Tablet-based method for real-time input by hand drawing. a) Schematic of sequence, resulting in projection of input to create PEG-DA features on the substrate. b) Optical microscope images of an array of dots, drawn sequentially by hand, subsequently built on the chip. The system projects the features drawn by the researcher after a time delay of 0.25 seconds.

Direct-write encapsulation and imaging of nematodes follows naturally as the examples given allude sufficiently towards how a user can encapsulate specific nematodes with the touch pad and then perform separation or imaging of them.

Dynamic maze fabrication

In *C. elegans* population studies, locating individuals with exceptional fitness compared to the rest of the population is a necessary first step to selectively breed traits, given that phenotypes within *C. elegans* populations are stringently conserved. Here we demonstrate how individuals can be tracked in custom-made mazes to determine their fitness regarding their ability to locate food.

More specifically, we demonstrate how a maze can be fabricated around a *C. elegans* worm suspended in the photopolymer, or placed *ex situ*. To begin, forty seven adult hermaphrodite worms were placed into the entrances of an array of already built T-shaped mazes and tracked until 80% of their body was inside of a circle region at either terminal end of the maze (Figure 56a). We recorded whether they chose left or right maze leg and found that the choice occurred with equal probability (Figure 56b). Our results verify that the choice is random, corroborating previous maze studies using PDMS assays [3]. Next, we placed food (*Escherichia coli* OP50) on the left end of each maze and we recorded the choice made by forty two adult hermaphrodites.

Using the above mentioned criterion of a worm's final choice, we found that the worms would choose the leg containing the food almost twice as often as the leg without food (Figure 56b).

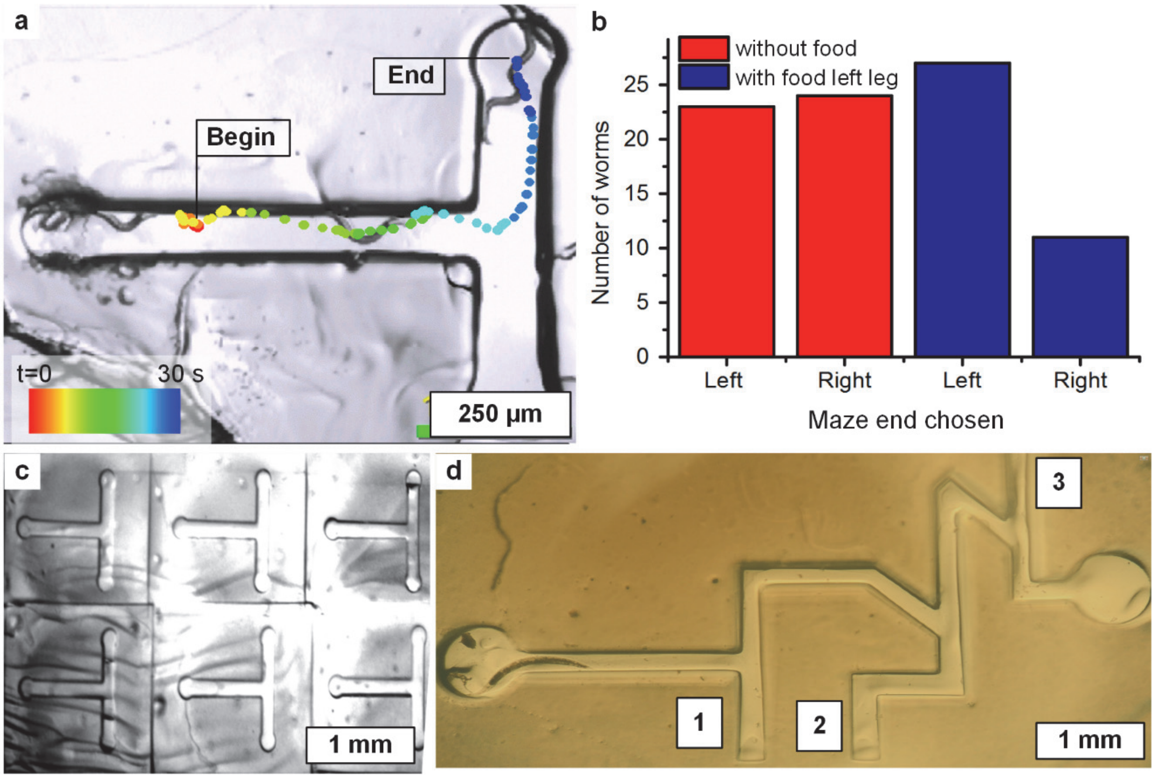


Figure 56. Maze assays used as a test case of C. elegans decision making behavior. a) T-shaped PEG-DA microchannel, with colored dots showing position of worm during a 30 second period. b) Control group of worms with and without food, and trained worms. c) Array of T-shaped mazes. d) Increasingly complex maze used to examine individuals with increased ability to track food placed in the green dots. Numbers indicate locations that correspond to failed decisions.

Given the fact that, based on the above mentioned results, not all worms located food correctly, we postulated that some worms would have the ability of locating food more effectively than others. To explore this further and taking into account that training the worms to solve complex mazes has been the subject of previous works (Pandey et al. [65]) we fabricated a second, more complicated maze to determine if the worm will continue to choose the correct leg of the maze to find more food (Figure 56d). Preliminary results are indeed implying an ability of certain individuals to locate food correctly (data not shown), however the full study is currently in progress, and our future experiments involve also the identification of genes that participate in the worm's ability to locate food.

We would like to also note that the exploratory behavior of *C. elegans* inside a maze reported in previous studies [59] was observed during the described experiments. This, in our opinion, occurred as in our case a third dimension in space (height) was available to explore due to the absence of a top cover, as used in the study of Qin and Wheeler, 2007.

This gave the worms the chance to climb up the walls of the maze corridors, verifying their ability to navigate in all space dimensions, in accordance with the conditions of their natural environment, the soil. Individuals that climbed their way out of the maze, even in initial stages, were not put back into the maze, so as to eliminate the possibility of them being already accustomed to the structure.

Our approach offers also the option to use feedback from the imaging system to decide the fitness of an individual by tracking worms in mazes and thus can provide an impetus to map learning and behavioral response to brain plasticity fitness in learning.

Conclusion

As tools and software for *in situ* analysis of assays become more capable, the possibility opens to dynamically alter the physical conditions of the experiments. Moreover, a bench top process eliminates the need for mask fabrication, master fabrication, or stamp/molding and enables rapid testing and iteration of assay designs using direct write lithography of Poly-ethylene glycol diacrylate (PEG-DA) onto agarose gel.

A comparison of dynamic lithography to soft lithography, lithography and stereolithography

Soft lithography has many advantages over traditional lithography for biologist and chemists. These are covered in detail by Xia et al. [52], and include the ability to fabricate many devices from a single master mold. However, the master mold is still fabricated via molding photo-lithographically defined features. An adapted process diagram shown from Xia, Figure 1a, shows how Dynamic lithography's process differs from soft lithography. Dynamic lithography removes the mask fabrication, master fabrication and molding step to fabricate features. Mitigating these issues enables the direct-write fabrication of photoresists microfeatures including hydrogels onto the substrate of choice. Moreover, Table 6 details how the mechanism of structure/pattern formation differs as well as functional differences. Compared to mail-order stereolithography this method will improve resolution due to a broader range of available chemistries and better resolution (30X).

Table 6. Comparison of photolithography, soft lithography and dynamic lithography.

	Photolithography	Soft lithography	Mail order stereolithography	Dynamic lithography
Definition of patterns	Rigid photomask (patterned Cr supported on a quartz plate)	Elastomeric stamp or mold (a PDMS block patterned with relief features)	Laser or Digital micromirror device	Digital micromirror device (array of 7 µm mirrors that are addressable pixels)
Materials that can be patterned directly	Photoresists (polymers with photosensitive additives) SAMs	Photoresists Sams Unsensitized polymers	Proprietary photopolymers	Photoresists (polymers with photosensitive additives) (PEG-DA)
Current limits to resolution	ca. 250 nm		30 µm	ca. 250 nm
Minimum feature size	ca. 100 nm	10 -100 nm	300 µm	1 µm
Dynamic patterning	N/A	N/A	N/A	Image, machine vision or researcher input

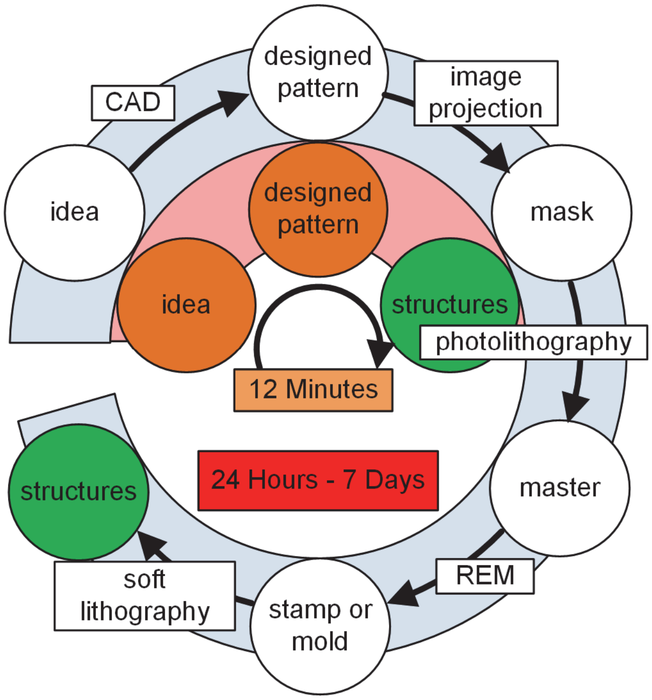


Figure 57. Comparison of soft lithography to dynamic patterning method.

We have presented a bench top process for fabricating, via direct-write lithography, micro-features onto NGM plates for assaying and encapsulating *C. elegans* in a rapid process. This, in our opinion displays numerous advantages compared to traditional lithography and soft lithography methods via enabling simultaneous rather than sequential assay fabrication and experimentation. First, using soft lithographic methods the expected time to develop a new assay is 7 days to 24 hours depending on the use of an industrial mask fabricator or in house plotter, however using our process we expect a complete new assay to be developed in 10-15minutes [52], [66], [67]. Moreover, high precision assays, with feature sizes nearing 1 μm require clean room access or industrial mask printing, which we also claim to eliminate the need for. This method is a platform for *in-situ* interrogation of *C. elegans* behavior by connecting the researcher to the micro-environment of the worm. Furthermore, it removes complicated interconnects for delivery the worm to the device by building the device around the worm. The capability of the system to fabricate free-floating structures could be used to observe a more complex behavior. The paradigm shift of dynamic feedback can be used to incorporate new features into a running assay or encapsulate a phenotype of interest for further processing. Moreover, this technique could be used in combination with real time worm tracking systems after selected neurons have been tagged with fluorescent markers, in order to observe which neurons and neuronal circuits participate in solving a maze, decision making and readjusting locomotion routes. Not only will advances in scaling assay patterning and integrating optogenetics improve the potential application of this method but so will advances in material synthesis. Recently, reported techniques to use Pluronic F127 as a thermally switchable encapsulant for *C. elegans* has the opportunity to expand the breadth of solutions available to this method [68].

Furthermore, this method may be useful for other model organisms such as manipulating Zebrafish or *Daphnia* [69], [70]. Zebrafish are another model organism used to study vertebrate development and disease. This method would be well suited to manipulating swimming organisms such as Zebrafish but they are much larger reaching 25 mm in length. The platform could be adapted to large area patterning however, polymerization or manipulation may occur too slowly. Moreover, Daphnia is used to understand genomic response to environmental stressors. This method could be adapted to control the introduction of stressors, but increased knowledge about how the system would affect genomic response would need to be studied first.

Towards the potential advances outlined we are currently using this methodology to select worm phenotypes based on problem solving capability in progressively challenging mazes. Selection is made by collecting and breeding worms that have completed the challenge to breed the trait. It is our vision that this method can also be used by other labs to rapidly deploy new assays that require many iterations, feedback, moving parts or incorporate machine learning for large data sets.

CHAPTER V. HIGH-SPEED OBJECT MANIPULATION BY DYNAMIC LITHOGRAPHY USING MACHINE VISION

Abstract

The previous chapter discussed the use of microfabrication in vitro via human directed manipulation of the micro-environment. To achieve automated adaptive and selective manipulation of the microenvironment based on feedback from machine vision, we can build upon previous microfabrication methods. Batch microfabrication, for example enables parallel processing of microscale objects but lacks the ability to dynamically adapt to live environments. Overcoming these limitations is will enable manufacturing of, realistic artificial tissue constructs with vascularization.

The goal of this chapter is to build an integrated hardware and software platform, based on micro-scale light shaping and high-speed machine vision algorithms that enables real time, dynamic photo-patterning in response to changes in microscale environments.

To enable the system to respond without human intervention, software was written to use machine vision to dynamically manipulate objects in flow, by photopolymer encapsulation. In order to track motion of particles while accounting for system latency a Kalman filter and image encoding algorithm are implemented to improve projection accuracy. The Kalman filter properties are derived by a genetic algorithm to match the motion characteristics of each experiment.

As a demonstration of this technique, micro-particles are sorted by size (TPR=100% and SPC=99.999%) and the mean encapsulation error of 4.7 pixels is measured. This demonstrates the promise of this method to effectively solve sorting and by extension general manipulation of particles. Looking forward, the realized platform combining light-driven polymerization and image-driven artificial intelligence algorithms could enable many dynamic fabrication capabilities for interacting with cellular and living systems.

I thank undergraduate researcher Lilly Chin and Felix Sun for their assistance writing the software used to track and detect objects.

Introduction and design of the dynamic lithography system

We introduce the term "dynamic lithography" to refer to the main goal and accomplishment of this chapter: adapting a DLP based photopatterning system to include hardware and software that enables real time computer driven decision making to interact with a dynamic microenvironment, for example to encapsulate a particle in flow (Figure 58). This differs from the methods we have previously discussed in that those methods either required foreknowledge about the system or researcher input while this method must analyze the image from the camera and automatically generate a new mask to project. This is challenging because ideally the system will act in in real time in response to a stimulus, such as the presence of a particle. Work by Kwon et al. have demonstrated similar functionality to apply protective coating LEDs when applying a phosphor coating [71], [72]. This requires tight integration of all the components in the system (optics, software, chemistry). To detail the requirements, we breakdown the design into the optics and its housing and the software algorithm with additional detail taken to discuss the individual steps of the algorithm.

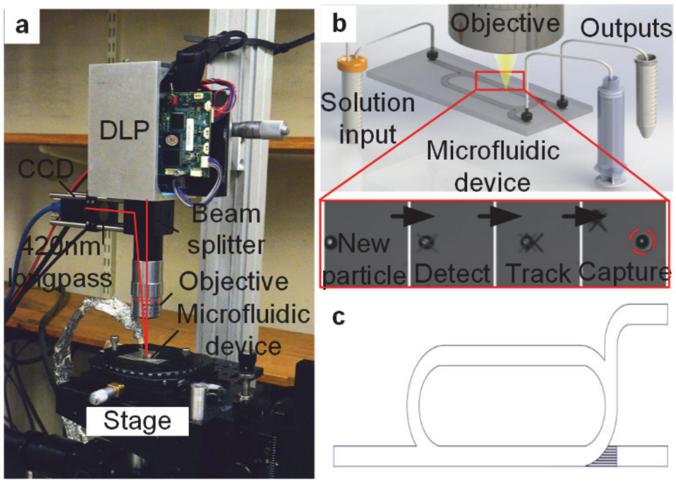


Figure 58. Overview of dynamic lithography. a) Dynamic lithography system. b) Schematic of microfluidic device and basic process of manipulating a particle. c) Layout of the microfluidic device.

Optics and housing

The problem of designing the system can again be described as three optical processes: projection, illumination and imaging. However, for this application the requirements are different than for manufacturing microparticles with arbitrary shape. First, we identified design requirements to enable the system to process many particles at a high rate.

Table 7. Design parameters for dynamic lithography system.

Design requirement	Value	Units
Camera frame rate	60-300	fps
Latency	16	ms
FOV	5	mm
Camera resolution	5	µm
DLP framerate	1440	Fps
Wavelength	405	nm
Depth of field	20	µm

The TI DLP system chosen was the Lightcrafter 4500, designed by Wintech which has an HDMI based communication method and can stream video at 60 frames per second. The DLP has a resolution of 912 x 1140 pixels with a pitch of 7.6 µm. The cover glass on the DMD chip acts as a bandpass filter between 405 nm to 750 nm wavelengths. The mirrors are in a diamond pattern (Figure 16 from Chapter I) which enables illumination along the mounting plane as shown in Figure 16. This system has a built in controller designed for development and experimentation. Settings are configured via USB connection to a PC and controlled by either the manufacturer software or custom C++ interface.

The camera chosen was a Ximea 3 USB camera. This camera communicates via USB 3.0 which allows 5 Gbit/s of data. Using this the maximum frame rate achievable for full sensor (2048x2048) video is 90 fps in 8 bit capture. It has a pixel pitch of 5.5 µm. The size of the sensor is 25.4 mm along its diagonal. The sensor is CMOS with a global shutter.

The objective is interchangeable, and a good compromise of FOV, resolution and depth of field was found when using the 5X Mitutoyo objective. This specifications of this objective state a FOV of 4mm, a resolving power of 2.0 µm (0.14 NA) and a depth of field of 14.0 µm. It also has a long working distance of 34 mm.

Depth of field can vary depending on how it was calculated because in a real system there is no definite location at which the object immediately becomes unfocused. Rather it is gradual therefore, we calculate the depth of field to understand what parameters the manufacture used to determine the specification. The depth of field (DOF) is given by the sum of wave and geometric optical depths of field as in Equation 10 [73]. Where d represents the depth of field, λ is the wavelength of light, n is the refractive index of the medium, and NA is the objective numerical aperture. The variable e is the smallest resolvable feature by a detector placed in the image plane with a lateral magnification M. Note how the first term relates to the contribution of the numerical aperture of the objective while the second term weights the circle of confusion. Using this equation the depth of field of the system is calculated as 28 µm at 500 nm which equals the manufacturer specification of ±14 µm. This means that *NA* which describes the half cone angle of the objective geometrically describes the

$$d = \frac{\lambda_o}{NA^2} + \frac{n}{M \cdot NA} e \qquad \text{Equation 10}$$

To image during exposure we chose a beamsplitter that would transmit the projected image to the substrate and reflect the substrate image to the camera. For this we chose a Cube-Mounted pellicle beamsplitter from Thorlabs. It has an 8:92 (R:T) reflectance and transmission ratio and is uncoated with a wavelength range of 400-2400 nm. The beamsplitter has a very thin cross section (2 µm), eliminating ghosting and removing chromatic aberration. Due to thin film interference there is a sinusoidal variation in the R:T spectrum as shown in Figure 59.

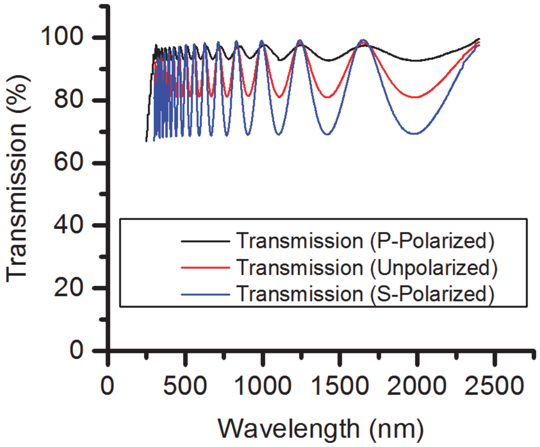


Figure 59. Transmission data for pellicle beam splitter. Data provided by manufacturer (Thorlabs)

Each of these components was assembled into a custom housing (Figure 60) with a constant distance between the DMD and the objective. The CCD adjusts along four optical cage rods to focus it onto the image plane. Finally, a longpass filter (Thorlabs FGL495) was mounted in front of the camera to filter blue light from the camera. This enables viewing with the camera in green and red light without bleaching caused by the exposure in blue light.

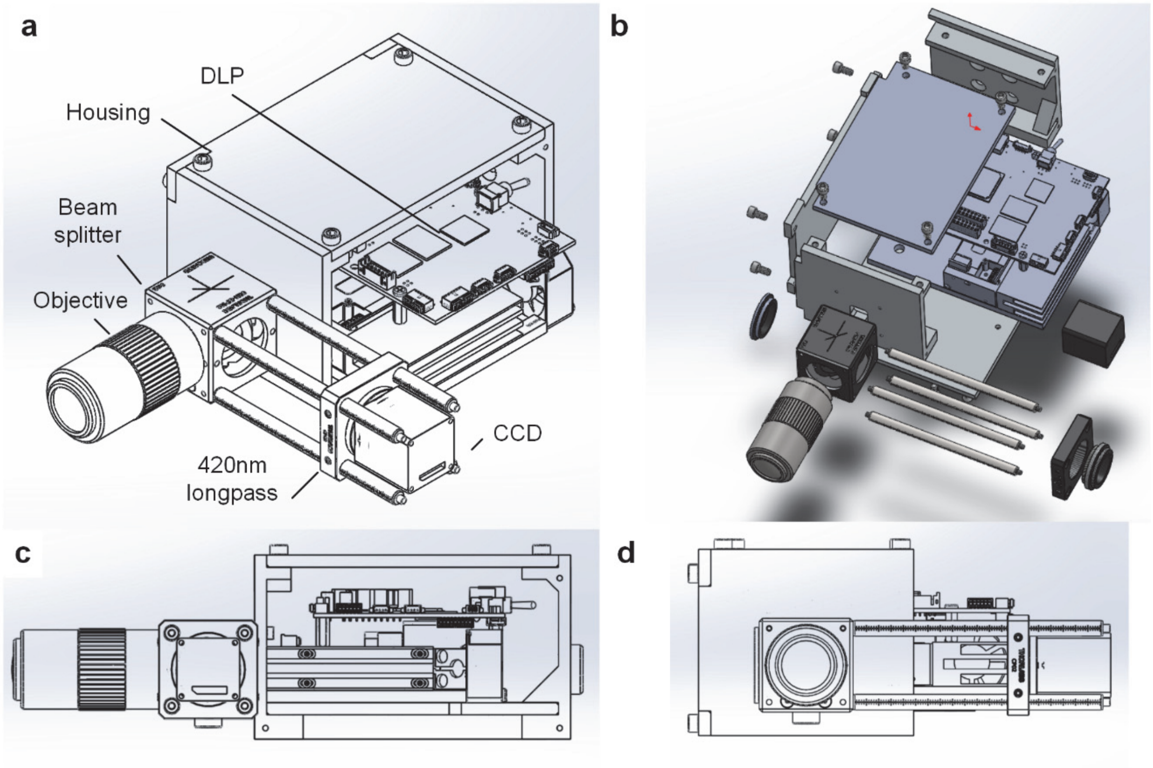


Figure 60. 405 nm dynamic lithography system. a) ISO view of the system assembled. b) Break apart view of the components in the system. c) Side view of the system showing the alignment of the optics. d) Front view of the system.

The assembled system is mounted onto an optical rail and positioned to project towards a motorized 3-axis stage (X, Y, Z: Thorlabs PT1-Z8), which has a manufacturer quoted 0.05 µm resolution, 130 µm absolute accuracy and 1.5 µm repeatability. The X and Y stages have a 25 mm travel range while the Z stage has a fine range of 25 mm and a coarse rack and pinion range of 200 mm.

This arrangement differs from the previously presented UV DLP lithography system (Chapter I) in that it has a smaller overall size, faster camera capture and projection rates. Moreover, it is better suited to "g line" resist, which denotes any photo-polymer sensitive to 435 nm illumination wavelengths because of a reflective coating on the DMD window that filters light below 400 nm.

Dynamic lithography algorithm

The goal of the dynamic lithography algorithm is to generate feedback and based on pre-defined rules decide how to process a particle in the system. A high level overview of this process can be seen in Figure 61a**.**

First, an image is captured and pre-processed to remove noise. Objects in the image are then detected and filtered further to only relevantly sized/shaped particles. Next, single particles are tracked. The software maintains a list of particle positions for 100 frames, new particles are added to the list and particles that have disappeared from view are deleted. Once the particles have been tracked, a Kalman filter and a bit plane slicing algorithm is applied in order to enable high resolution tracking and account for latency. The Kalman filter is, a linear estimator of particle position, velocity and acceleration that is updated on each iteration. While the bit plane slicing algorithm encodes predicted positions of for the particle into subframes encoded in the projection frame. The projector interprets the projection frame as 24 sub frames projected individually and sequentially to smooth the projection as it tracks a particles.

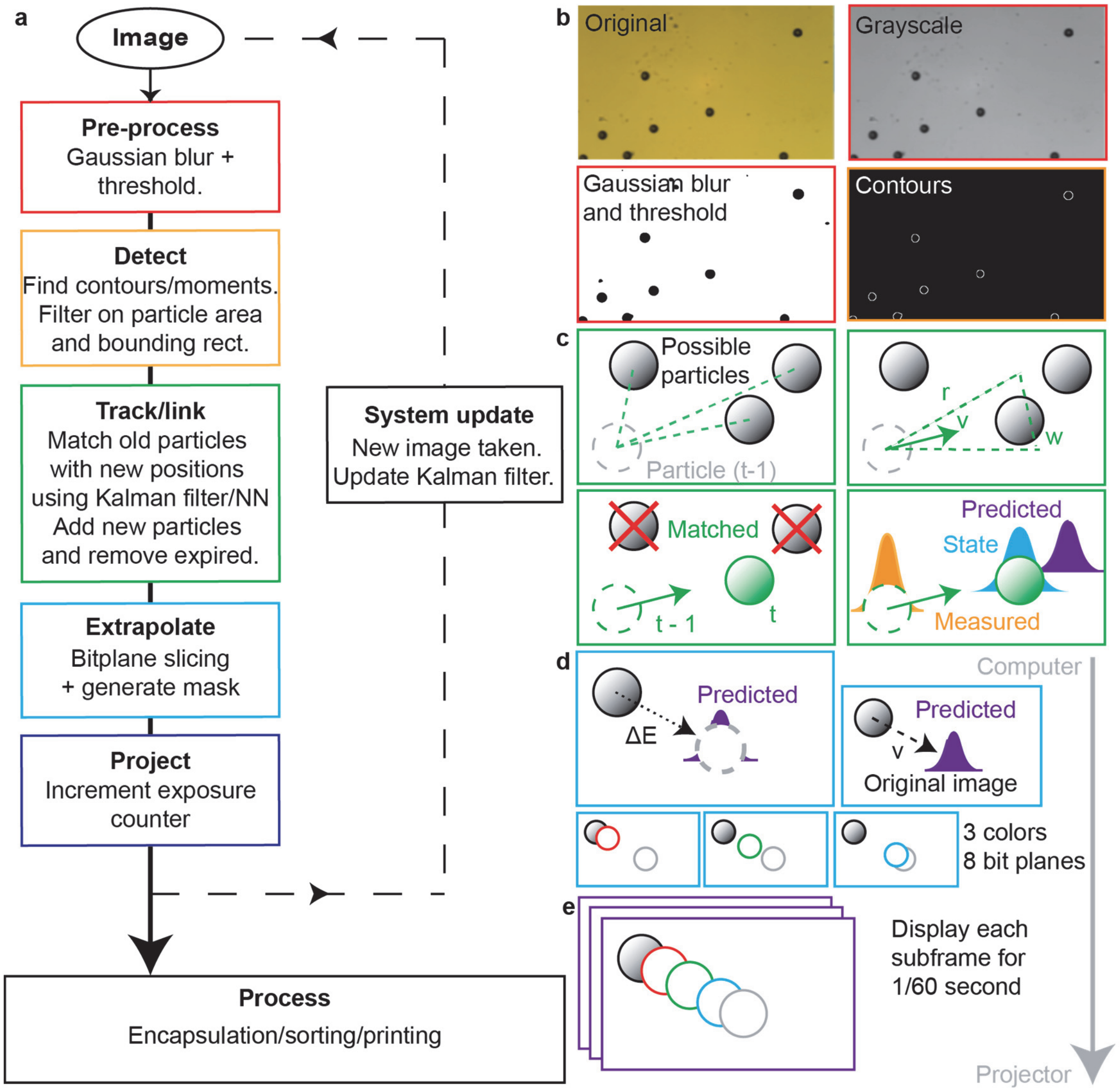

a
Image
Pre-process
Gaussian blur + threshold.
Detect
Find contours/moments. Filter on particle area and bounding rect.
Track/link
Match old particles with new positions using Kalman filter/NN Add new particles and remove expired.
Extrapolate
Bitplane slicing + generate mask
Project
Increment exposure counter
System update
New image taken. Update Kalman filter.
Process
Encapsulation/sorting/printing
b
Original
Grayscale
Gaussian blur and threshold
Contours
c
Possible particles
Particle (t-1)
r
v
w
Matched
t - 1
t
Predicted
State
Measured
d
Computer
Predicted
ΔE
Predicted
v
Original image
3 colors
8 bit planes
e
Display each subframe for 1/60 second
Projector

Figure 61. Overview of the particle tracking, detection and processing. a) Flow chart of the algorithm. b) Preprocessing steps. c) Detection sequence checking current frame against position and velocity of previous frame. d) Kalman filter prediction of state position and extrapolation of position via bit plane slicing. e) Display of subframes for high speed projection.

The particle class

Particle information is maintained through an object-oriented approach. For each particle type (C. Elegans, hepatocytes, etc.), there is a "Particle Group" class which defines properties about the particle type such as image pre-processing and drawing parameters. The ParticleGroup also contains a vector of all current particles of that type. Each particle tracks its own location and instance of a Kalman filter for further iteration. The particle class is shown in Figure 62.

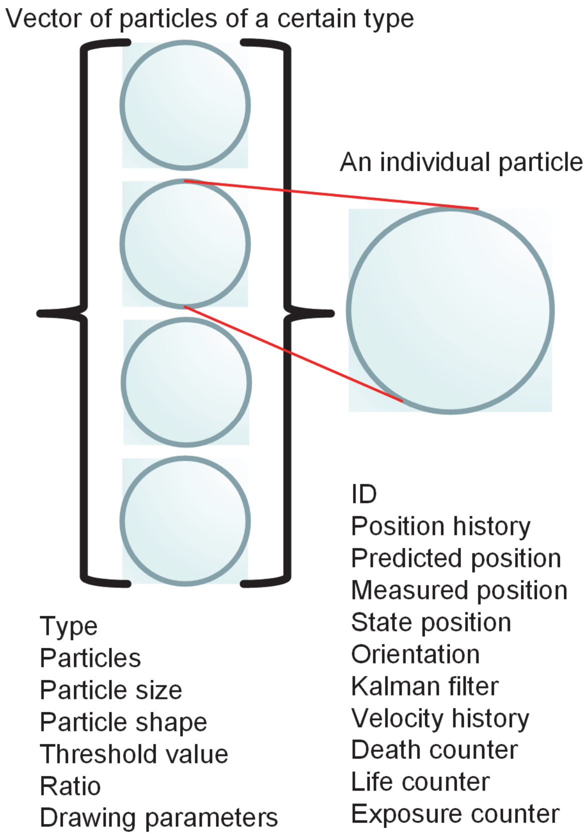


Figure 62 Diagram of the particle class object and particle group object

Pre-processing

After an image is acquires, it is pre-processed to reduce noise and improve detection by applying a Gaussian blur and morphological operations [74].

Gaussian noise along one axis is defined as:

$$G(x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{x^2}{2\sigma^2}} \qquad \text{Equation 11}$$

However it is direction independent so performing it along both the x and y directions can be accomplished by simply performing the blur again in the other direction.

$$G(x) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}} \qquad \text{Equation 12}$$

Where σ is the standard deviation of the Gaussian distribution. To understand how this is used in the blur algorithm each pixel is blurred by averaging it with its neighbors. Each neighbor is weighted according to the Gaussian distribution value corresponding to that pixel. So closer pixels add more color information than those further away. To make the algorithm more efficient a blur radius cut off can be set effectively setting σ for the distribution so that pixels further away than the radius do not contribute to the effect. Applied to a single image of 45 µm PS beads yields the progression shown in Figure 63.

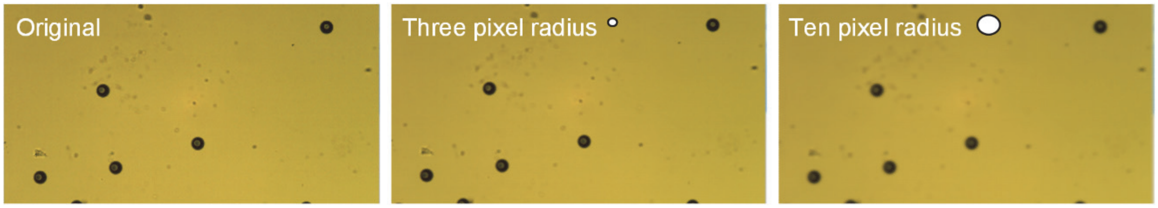


Figure 63 Effect of Gaussian blur radius on noise in an image of 45 µm PS beads

The second method we use to deal with noise are morphological operations. The two operations being used are *dilation* and *erosion*. The effect of these two operations is similar to the classic game *Minesweeper*. If a black block of pixels is present on a white background applying the dilate operation will change to black each pixel that is adjacent to a black pixel, causing the black block to enlarge. The erode operation applies the same effect to white pixels, making the black block smaller. In the present algorithm we apply erode and dilate sequentially to first remove noise and then return the size of the particles of interest to their original size. This removes noise by iteratively applying erode so that black blocks that are only a few pixels in width are erased leaving only larger blocks.

Detection

The first step in detection is to apply a threshold to the image to generate a binary representation of the particles. More details on the threshold method are discussed in the next chapter. To generate a list of detected objects, contour detection was run on the new binary image. Having maximized the edge contrast of the particles, the image moments (M_{ij}) of each particle can be found [75], according to Equation 13 $I(x, y)$is the intensity of the pixel at position x,y with indices i, j.

$$M_{ij} = \sum_{x}\sum_{y} x^i y^j I(x,y) \qquad \text{Equation 13}$$

A uniqueness theorem [76] states that if f(x,y) is piecewise continuous and has nonzero values only in a finite part of the xy plane, moments of all orders exist, and the moment sequence is uniquely determined by f(x,y). The moment of an image uniquely describes that image and the converse is also true.

This is useful because the M_{00} moment for binary region (particle after thresholding) on an image gives its value, while the centroid of a region can be found by $\{x, y\} = \{M_{10}M_{00}, M_{01}/M_{00}\}$. Next in the algorithm, the remaining particle candidates are filtered by comparing the areas found for each particle using the moments to a maximum area.

Tracking

In each frame, the algorithm first detects particles in the frame then links them to a history using a tracking algorithm, resulting in a vector of particles. Particles are tracked by matching particles from the current frame to the previous frame. Any particles that did not match are considered new particles.

Our tracking algorithm incorporates Kalman filter predictions into a nearest neighbor algorithm. This filter was chosen because it is statistically the optimal linear estimator for systems that are one dimensional and linear [77]. We assume that the motion between frames can be simplified into linear motion. At any frame, we know the predicted position of particles, estimated from the previous frame. From this, we can calculate an approximate velocity vector for each particle. Thus, we know that it is likely that in the current frame, the particle will be found along the direction of this velocity vector. So, for a specified radius r and a specified particle width w, we can construct a rectangle where the particle will most likely be for this time step. If the algorithm finds a particle in this triangle, it is matched with the older particle. This method was chosen because it improves on nearest neighbor tracking by focusing the search region to the most likely area on the image. A pictorial representation of this algorithm can be seen in Figure 62c.

After all particles on the detected particles list have been accounted for, we iterate over all particles once more. If they have not been updated in a five frames (death counter) or if they are too close to the edge of the image and will be going outside our field of view, they will be removed from our particle list so they will no longer be accounted for in our tracking algorithm.

The death counter was chosen to ensure responsive removal of particles that are no longer visible while ignoring momentary detection algorithm failures. A more detailed description of the Kalman filter is to follow.

Predicting/Extrapolating

A limitation of the HDMI communication between the PC and the projector is that the drawing rate to the projector is 60 fps. In an attempt to improve the projection accuracy between imaging frames, we implemented a method to increase the frame rate of the projector between each drawing event. Given the predicted position of the particle, from the Kalman filter, the system attempts to draw the extrapolated position of the particle extrapolated in increments of 0.7 milliseconds. However, we must send this data to the projector and sync the software. In a 24 bit RGB (Red, Green, Blue) image each pixel in the image is described by 24 bits. The first bit for every pixel can be selected and would described a binary image or a plane of image data. This is true of each bit related to a pixel. So, in this way a 24-bit image is composed of 24 1-bit binary images. Since we are projecting in RGB, each image sent to the projector is composed of 24 independent bit planes – 8 for each color. Normally, the projector displays these rapidly while changing the display color to trick our eyes into seeing multiple colors on the screen at once. However, by encoding 24 binary images into each of the 24 bit planes, we were able to increase our projection frame rate to 1,440 frames per second. (60 frames X 24 subframes). Each of the binary images or sub-images is drawn on to include the next extrapolated particle mask. Therefore each binary image contains the projection for the position of the particle at each increment in the extrapolation as shown in Figure 61d [78]. However, changes in the software's frame rate are inevitable as the number of particles in the system fluctuates. Under standard use, a clock is used to sync the projector, camera and system software. However, if the system frame rate dropped below 60 fps then the extrapolated projection would display the last subframe longer than intended. In this case the latency is used to alter the display rate of the subframes on the projector resulting in a smooth, accurate projection.

Generating masks and projecting images

The generated mask to be projected uses the state positions of the particles and any drawing data the particle class may convey. For example, it may contain information on orientation and type. In future cases this will orient the projection and change it based on the particle type detected. The applications discussed in this thesis adjust the size of the projected mask to encapsulate the detected object. The image that has been generated is then displayed on the projector as transferred via an HDMI connection.

Improving projection accuracy by accounting for latency

Latency in the projection

A significant challenge in real time tracking is introduced by latency in the system [79]. The latency is defined as the duration from when light reaches the camera sensor and the system emits patterned light according to the result of the processing algorithm. The major contributions to latency are the camera encoding the image, the transfer across the USB cable, writing the data to RAM, performing a demosiac procedure on the image, performing the machine vision algorithm and transferring another image to the projector, loading it into projector memory, and switching the mirrors to match the pattern of pixels in the image.

The effect of latency can be seen in an early version of the software (Figure 64) in which polymerized PEG-DA is trailing the detected particle by a distance e proportional to their velocity

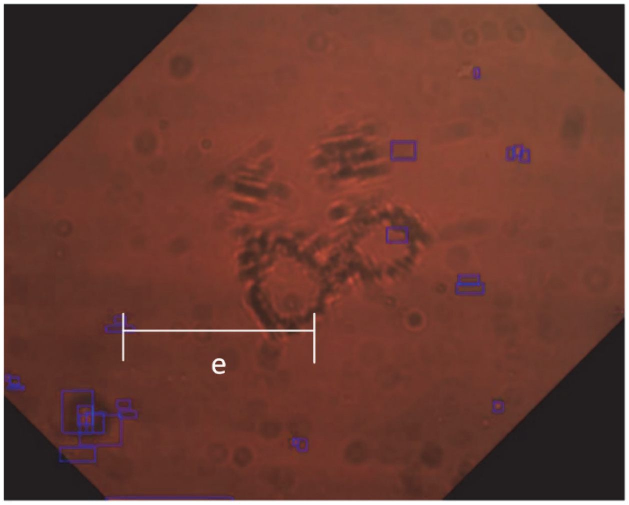


Figure 64. Early attempt at encapsulation. Latency caused delay in projection so the system missed the particles by some distance proportional to their velocity.

The Kalman filter

In order for dynamic lithography to be successful we must accurately and consistently know the position of each object of interest. However, our measurements are nothing more than approximations. Yet, given some minimum number of measurements (we may accurately determine the unknown quantity of position at time t. This problem is described by Gauss in his development of Least Squares estimators and further by Kalman when employed in recursive systems [80]. Gauss noted that because of errors in any measurement system we require redundant measurements to accurately describe the state of a particle. This is true even if we have a description of the dynamics of the particles parameters (position, velocity). Gauss stated that the parameter estimates must satisfy the observations. He referred to this as minimization of the residuals defined as the difference between observed and predicted values. Gauss further indicated that the inaccuracies of the observations or errors are unknown which set the stage to be addressed later by probabilistic modeling.

To account for latency it became apparent that we needed to calculate the latency and predict the true state of the object. To do this we employed the Kalman filter as invented by R. E. Kalman. An report by Welch and Bishop provides a foundation for our discussion of it and how it can be applied to the problem of real time tracking in a 2D image [81].

With Gauss' findings in mind, the next step is to model dynamic lithography using a Kalman filter. First, we must derive the filter according to the dynamics of the system. Assuming the state of a particle (position, velocity and acceleration) at time t evolved from its prior state at time t-1 according to the linear stochastic equation

$$x_t = A_{t-1}x_{t-1} + Bu_t + w_t. \qquad \text{Equation 14}$$

With a measurement (of particle position) $z \in \Re^n$:

$$z_t = H_t x_t + v_t \qquad \text{Equation 15}$$

The $n \times n$ matrix A in the difference equation (Equation 14) relates the state at time step t to the state at step $t + 1$, in the absence of either a driving function or process noise. The $n \times l$ matrix B relates the control input $u \in \Re$ to the state x. The $m \times n$ matrix H in the measurement equation (Equation 15) relates the state to the measurement z_t.

The random variables w_k and v_k represent the process and measurement noise (respectively). They are assumed to be independent (of each other), white, and with normal probability distributions $p(w)$ and $p(v)$.

$$p(w) - N(0, Q), \qquad \text{Equation 16}$$

$$p(v) - N(0, R). \qquad \text{Equation 17}$$

Modeling the Kalman filter

Next, we estimate the vector of an object and the latency inherent in the system. The vector represents the two-dimensional dynamics of an object moving through a medium under the lithography system. We assume that we can take measurements of the position of each particle and calculate the direction and velocity of motion.

The motion of a particle in 2D can be modeled using a Kalman filter of the following form of A. No control input is used so B has a null value.

$$\boldsymbol{A} = \begin{bmatrix} 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \qquad \text{Equation 18}$$

$$B = []$$

The Kalman filter is a two-step algorithm: first the filter estimates the process state at some time and then takes some measurement as feedback. This process is described by five standardized equations broken into a prediction set and measurement update set. The prediction equations advance the time step of the system and calculate the expected current state and error covariance estimates to obtain the a priori estimates for the next time step. The measurement update equations gather information about the position of the object in x, y coordinates and convert them into an a priori estimate of the objects position, velocity and direction to obtain a weighted and improved a posteriori estimate.

The weight of the estimate or the measurement is dependent upon the Kalman gain,K_t. After each time step (prediction and measurement update), the process repeats recursively using the previous a posteriori estimates to predict the new a priori estimates. This is why the Kalman filter enables efficient calculation of the optimal linear estimator [77], [82].

$$\hat{\boldsymbol{x}}_{t|t-1} = \boldsymbol{A}\hat{\boldsymbol{x}}_{t-1|t-1} + \boldsymbol{B}\boldsymbol{u}_{t-1} \qquad \text{Equation 19}$$

$$\boldsymbol{P}_{t|t-1} = \boldsymbol{A}\boldsymbol{P}_{t-1|t-1}\boldsymbol{A}^T + \boldsymbol{Q} \qquad \text{Equation 20}$$

Measurement Update (“Correction”)

$$\boldsymbol{K}_t = \boldsymbol{P}_{t|t-1}\boldsymbol{H}_t^T(\boldsymbol{H}_t\boldsymbol{P}_{t|t-1}\boldsymbol{H}_t^T + \boldsymbol{R}_t)^{-1} \qquad \text{Equation 21}$$

$$\hat{\boldsymbol{x}}_{t|t} = \hat{\boldsymbol{x}}_{t|t-1} + \boldsymbol{K}(z_t - \boldsymbol{H}_t\hat{\boldsymbol{x}}_{t|t-1}) \qquad \text{Equation 22}$$

$$\boldsymbol{P}_{t|t} = \boldsymbol{P}_{t|t-1} - \boldsymbol{K}_t\boldsymbol{H}_t\boldsymbol{P}_{t|t-1} \qquad \text{Equation 23}$$

From the update state variables in $\hat{\mathbf{x}}_{t|t}$ the projection x position is calculated adding the distance traveled due to latency. This is calculated using the state velocity $\dot{x}$ and $\dot{y}$ and the latency compensation coefficient (τ) of the system.

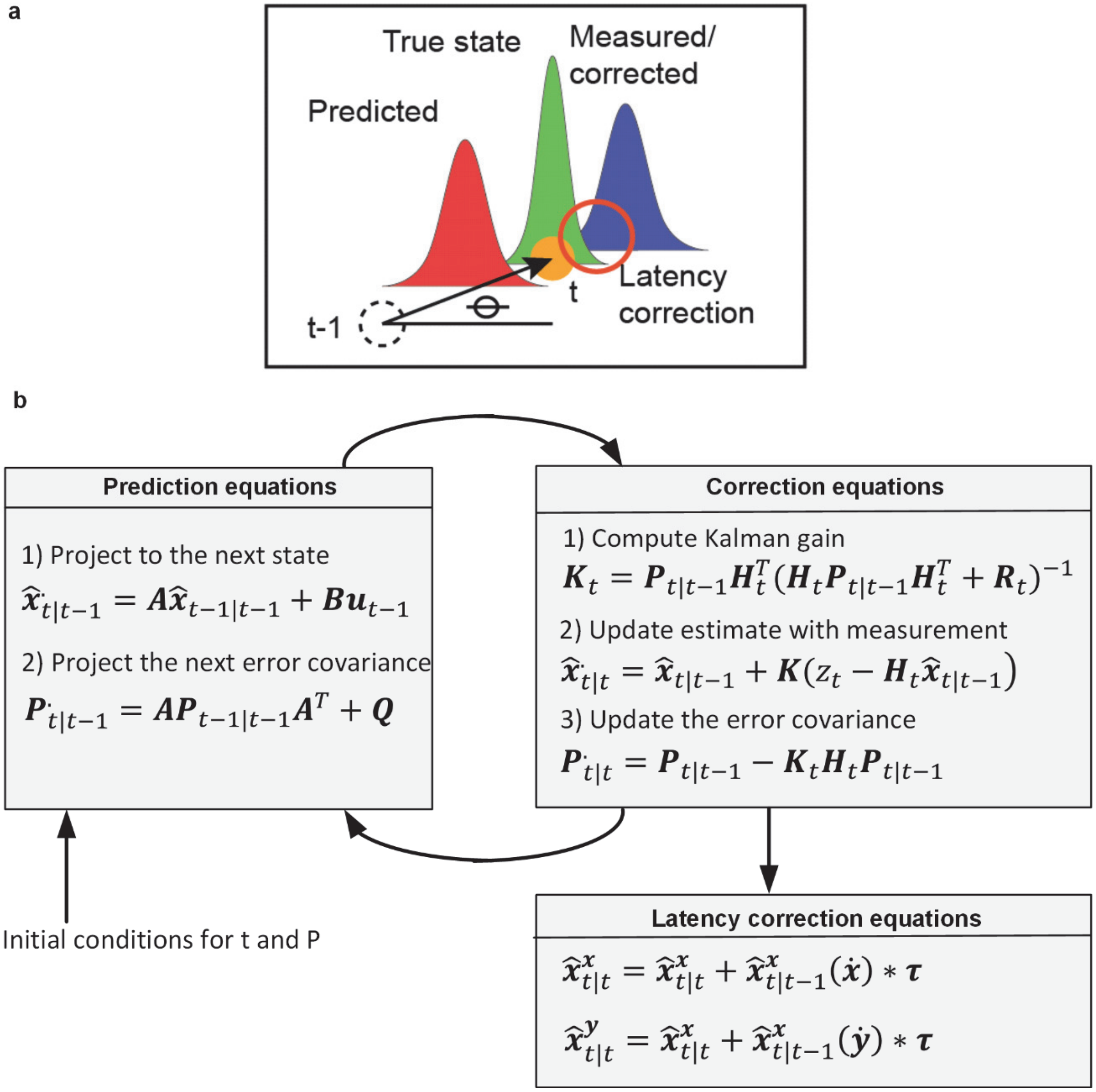

a
True state
Measured/ corrected
Predicted
Latency correction
t-1
t
θ
b
Prediction equations
1) Project to the next state
$\hat{\boldsymbol{x}}_{t|t-1} = \boldsymbol{A}\hat{\boldsymbol{x}}_{t-1|t-1} + \boldsymbol{B}\boldsymbol{u}_{t-1}$
2) Project the next error covariance
$\boldsymbol{P}_{t|t-1} = \boldsymbol{A}\boldsymbol{P}_{t-1|t-1}\boldsymbol{A}^T + \boldsymbol{Q}$
Correction equations
1) Compute Kalman gain
$\boldsymbol{K}_t = \boldsymbol{P}_{t|t-1}\boldsymbol{H}_t^T(\boldsymbol{H}_t\boldsymbol{P}_{t|t-1}\boldsymbol{H}_t^T + \boldsymbol{R}_t)^{-1}$
2) Update estimate with measurement
$\hat{\boldsymbol{x}}_{t|t} = \hat{\boldsymbol{x}}_{t|t-1} + \boldsymbol{K}(z_t - \boldsymbol{H}_t\hat{\boldsymbol{x}}_{t|t-1})$
3) Update the error covariance
$\boldsymbol{P}_{t|t} = \boldsymbol{P}_{t|t-1} - \boldsymbol{K}_t\boldsymbol{H}_t\boldsymbol{P}_{t|t-1}$
Initial conditions for t and P
Latency correction equations
$\hat{\boldsymbol{x}}^x_{t|t} = \hat{\boldsymbol{x}}^x_{t|t} + \hat{\boldsymbol{x}}^x_{t|t-1}(\dot{\boldsymbol{x}}) * \boldsymbol{\tau}$
$\hat{\boldsymbol{x}}^y_{t|t} = \hat{\boldsymbol{x}}^x_{t|t} + \hat{\boldsymbol{x}}^x_{t|t-1}(\dot{\boldsymbol{y}}) * \boldsymbol{\tau}$

Figure 65. Kalman filter algorithm. a) Visualization of the implementation of the Kalman filter. b) Schematic relating the equations to an x, y position on the screen.

Tuning using a genetic algorithm and simulation

Genetic algorithms, which is a method of iteratively finding a local optimum from a population of possible solutions have previously been applied to diverse fields such as bioinformatics, aeronautics and trading. In this section we apply the genetic algorithm to find the optimal parameters to the Kalman filter difference equation that will minimize the projection error [83]. Many optimization methods could be used but in this case we did not know how many parameters needed to be optimized. The algorithm began with examining the values of the Kalman filter including the transform matrix (A), the covariance matrices and the scalar (latency compensation coefficient). It was then determined that not all of these were important and a simpler optimization algorithm would have sufficed. The genetic algorithm works by first creating a population of possible values for four parameter inputs to the Kalman filter (error covariance, process covariance, measurement covariance and the latency delay) as shown in Figure 66. Each member of the population is assigned to an individual particle being tracked in a simulated video of particles moving. The particles have different velocities and trajectories (linear, sinusoidal). After each member of the population is assigned the Kalman filter is re-initialized and the particles are tracked for 10 frames. After all of the members of the population have been tracked for 10 frames the error in the fitness of each member is evaluated. This means that the error between the projected state position of the tracker and the ground truth position of the particle (given some latency) is measured and assigned a fitness value. The goal of the algorithm is to drive the fitness value towards zero. The particle that first reaches a fitness value of 1 pixel therefore has parameters that minimize the error between the projection and ground truth. If none of the member of the population meets the goal fitness of value of 1 pixel, then the genetic algorithm creates a new population by crossing over values between members and mutating some of the parameter values. Crossing over means that half of the parameter values from one member of the population are swapped with another member. Further, mutation randomly occurs and changes the value of a parameter for selected members. Elitism was also applied so that the top two members of the population were preserved during the cross over but not during mutation. So, the best performing member of the population persist until a better member replaces it, so performance can’t degrade between iterations. To our knowledge the Genetic Algorithm has not yet been applied to find the optimal solution to the Kalman filter for a particular problem. This enables us to be confident in the performance of the Kalman filter and to quickly adapt it to new dynamics in the particle system by running the GA again.

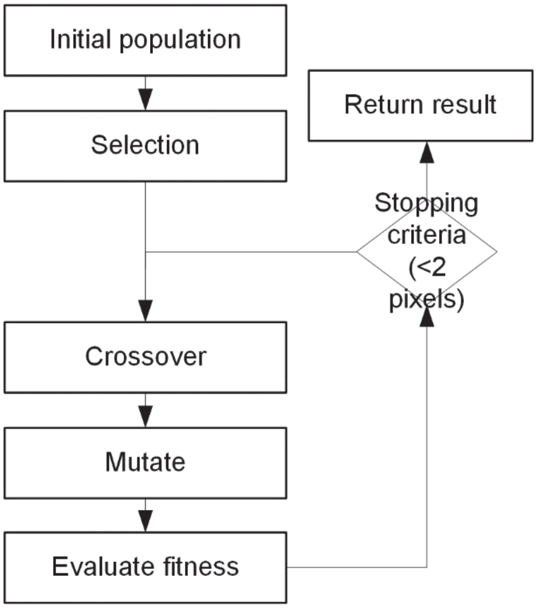


Figure 66. Genetic algorithm overview

Mapping projector space to image space

Alignment between the camera and projector is automatically mapped on system start up. This is necessary to ensure accurate projection onto the particles. The projection does not take up the entire imaging sensor. Therefore, a red rectangle is projected during an initialization process representing the extents of the projector (Figure 67). Using image processing the edges of the rectangle are found and assigned as the region of interest (ROI) in the image.

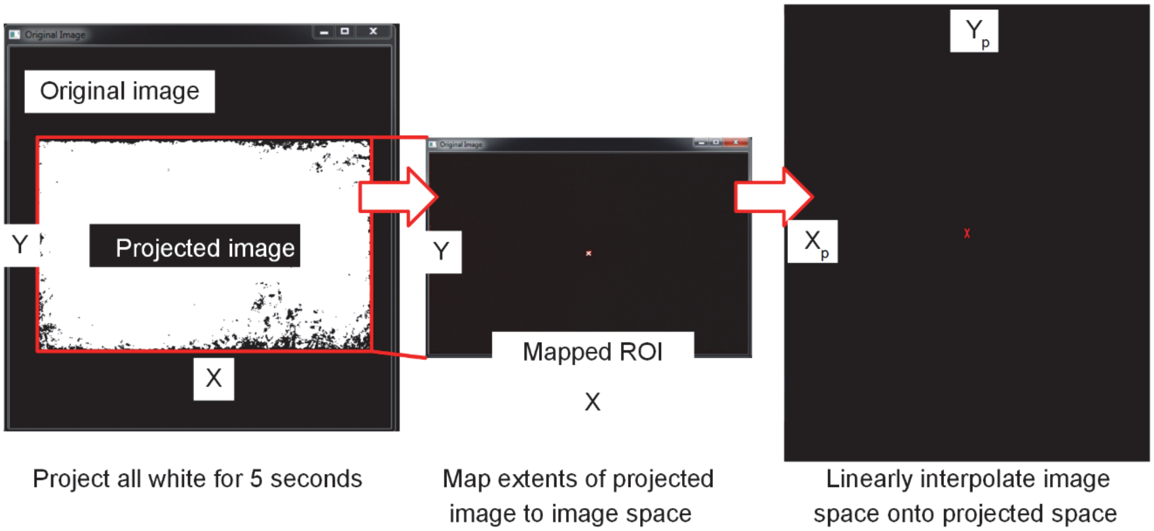

Original Image
Original image
Projected image
Y
X
Y
Mapped ROI
X
Y_p
X_p
Project all white for 5 seconds
Map extents of projected image to image space
Linearly interpolate image space onto projected space

Figure 67. Algorithm for mapping projector space to image space.

Methods

Encapsulation experiments were performed by translating a glass slide on which particles were cast in solution using the system's stage as described earlier. The stage was translated at 0.1, 0.5, 1 and 2 mm/s (its maximum velocity).

Sorting was performed using a microfluidic device with a race track pattern and two outlets. One outlet has gates (100 µm wide) that act like filters, letting particles smaller than the width of the gate exit the device. A solution of PEG-DA with a white light photoinitator (Phenylbis(2,4,6-trimethylbenzoyl)-phosphine oxide) was prepared with (5 µm, 10 µm, 45 µm and 90 µm) PS microparticles at a concentration of 200,000/ml for each particle size. The solution was pumped through the system via a syringe pump.

Fabrication of the microfluidic device

SU-8 (MicroChem 2050) master molds were fabricated to then cast PDMS microfluidic devices from [84], [85]. To fabricate the mold a 4 inch Si wafer was cleaned in a Piranha solution to remove any organic residue. Then 4ml of SU-8 2050 was spun at 2000 rpm for 30 s. According to manufacturer specifications this should give 100 µm thickness. This was followed by a Pre-bake and softbake for 3 minutes at 65 ˚C and 9 minutes at 95 ˚C respectively. The mask was then aligned and exposed for 15 s, allowed to cool for 15 s and then exposed again for 15 s with 230 mJ/cm^3 in MA-4. The mask was removed and the wafer was put in the oven for a two-step post bake with 1 min at 65 ˚C and 7 minutes at 95 ˚C. The wafer was then developed by spinning it at 200 rpm while rinsing it with developer until clean followed by IPA and a drying spin at 1000 rpm. No hard bake was performed as it wasn't necessary for this application. After drying PDMS (1:5) was cast to a thickness of 5 mm on the mold and allowed to cure for 45 minutes on a hotplate at 85 ˚C. The resulting layer was then removed and affixed onto a glass slide that had PDMS already spun onto it to 100 µm. The completed device was placed into the oven overnight to cure fully before use. The inlets and outlets were opened using a 1 mm biopsy punch.

The height of the SU-8 master mold was measured using a profilometer (KLA-Tencor P-16+). The profilometer was fitted with a 12.5 µm tip with a length of 500 µm. The SU-8 patterns on the first device measured 198 µm tall while the second measured 117 µm.

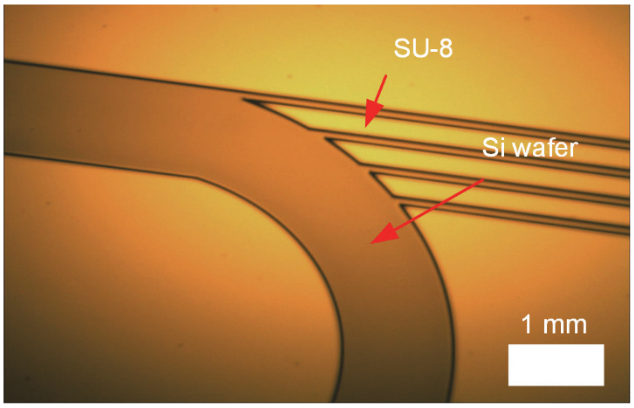


Figure 68. Image of the mold used to fabricate the microfluidic device.

Results

The resolution of imaging, projection and photopolymerization of the DLP lithography system described for high speed manipulation using white light is as follows. The imaging resolution was measured as 2 μm and the photopolymerization resolution was 5μm. The mean time to photo-polymerize a structure was 0.1 s measured by imaging encapsulated beads using exposure times from 0.01-1 second. As previously reported (0) the exposure time depends on the size of the feature, photo-initiator concentration, dilution and illumination intensity.

The effect of tuning the Kalman filter on system performance

The Kalman filter is used to track and predict the position of particles in the next frame. However, it has several parameters that need to be correctly set, as discussed in the modeling section, to function properly. The Genetic Algorithm was used to find the optimal latency compensation coefficient for system wide latencies ranging from 0-2 seconds. This was done by running the GA on a simulated video of particles whose positions in each frame are known. Then the latency was set and the GA searched for the optimal parameters the minimized the error between the projection as predicted by the Kalman filter and the simulated position of the particle. The search process is shown in Figure 69a, while the projection (rings) are properly projected in Figure 69b after the GA has run. The error was reduced to 1 pixel on average. The single circle lagging on the left is simply due to the initialization of the Kalman filter, that circle will be centered by the next frame. Moreover, by this method we mapped the relationship of velocity and latency to the four Kalman filter input parameters.

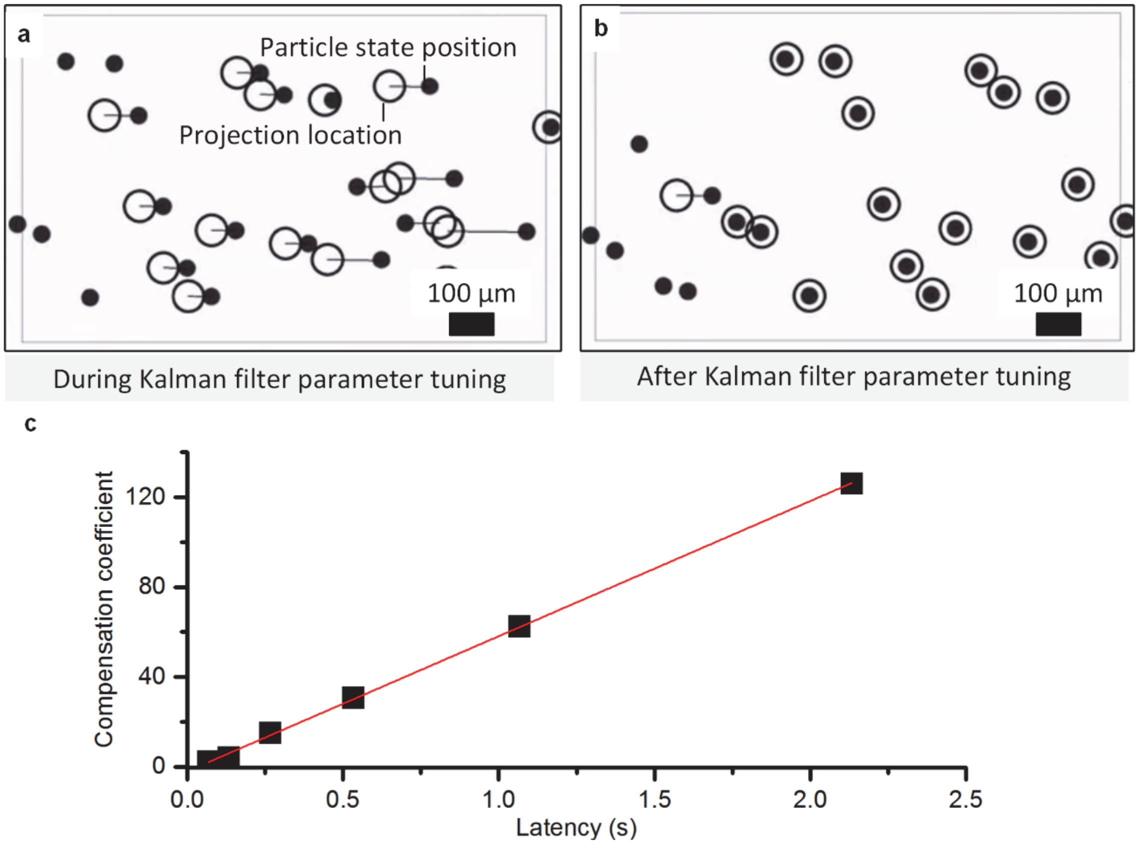


Figure 69. Results of tuning the Kalman filter. A) A view of state t showing the error between particle state position and projection position during the genetic algorithm process. B) The result of the tuning process showing a displacement error of 1 pixel on average. C) Measured relationship between latency and the compensation coefficient for the Kalman filter and the linear fit.

Verification of bit place slicing procedure

Using the bit plane slicing algorithm a test image was prepared with 16 slices. This is shown in Figure 70a. After the image was generated it was projected against a substrate at 60 fps, which was imaged with a 30fps camera as seen in Figure 70b. At this rate the image appears as a solid white image with uniform brightness. This indicates that each bit plane is sharing a uniform exposure time. Moreover, when using a 10,000 fps camera each bit plane is easily observed being projected sequentially. For example, the number 2 projected in Figure 70c is a single bit plane exposed from the full image. Finally, we translated this to generate masks using bit plane slicing for particles as shown in an exaggerated example (the latency is multiplied by 100X so the distance between subframe positions is exaggerated so that it is easier to understand) in Figure 70d.

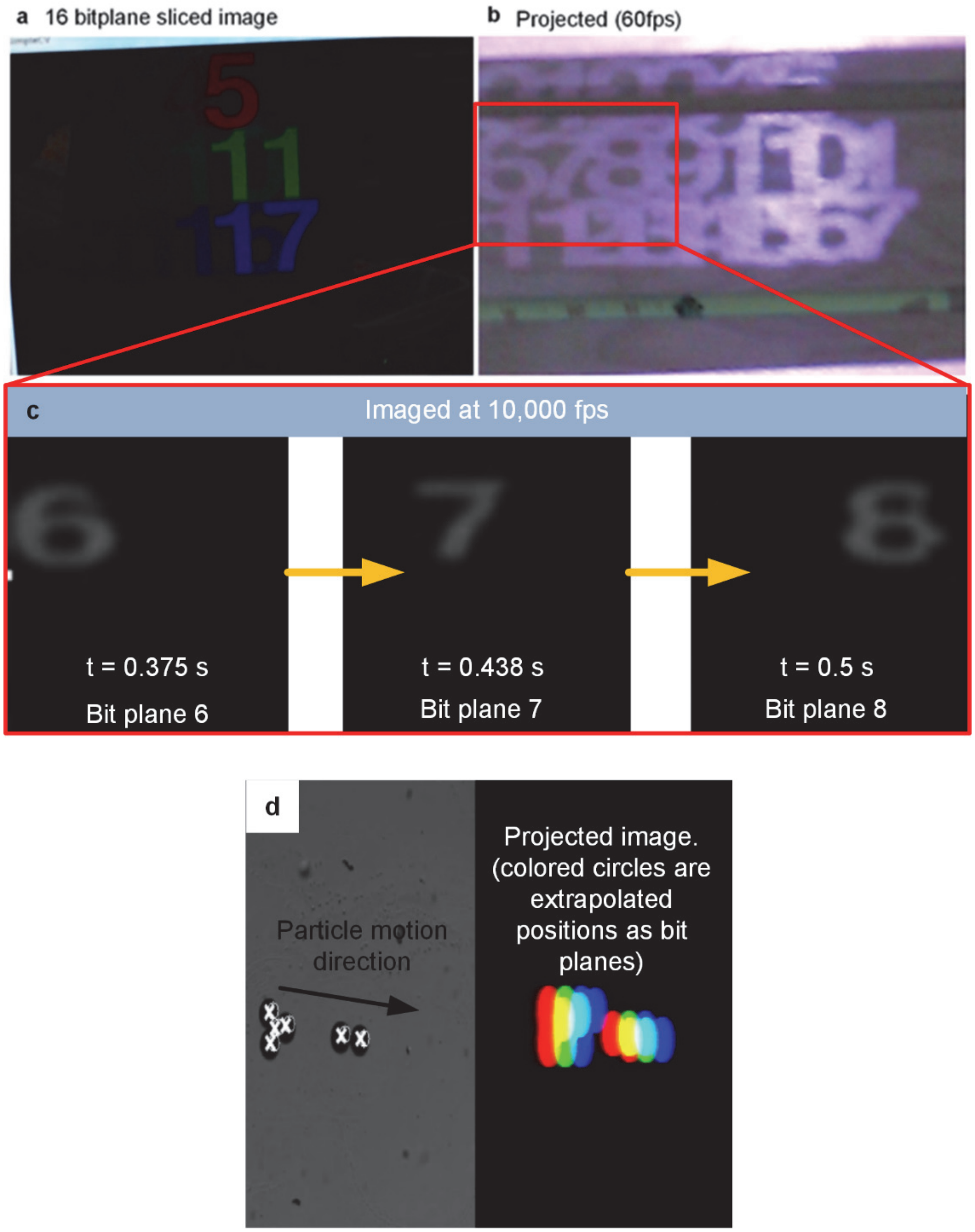

a 16 bitplane sliced image
b Projected (60fps)
c
Imaged at 10,000 fps
t = 0.375 s
Bit plane 6
t = 0.438 s
Bit plane 7
t = 0.5 s
Bit plane 8
d
Particle motion direction
Projected image. (colored circles are extrapolated positions as bit planes)

Figure 70. Bit plane slicing verification. A) Test image on monitor to verify the bit plane slicing algorithm constructed a 16 bit image from 16 binary images with colors indicating bit planes. B) View of the image projected as imaged at 30 fps second. C) Same projection viewed through 10,000 fps camera showing individual bit planes (6,7,8) as they are displayed. D) Bit plane slicing adapted to draw 24 locations to project circle sequentially in direction of particle motion.

Performance of projector space mapping

The accuracy of the projection relative to the camera using the mapping procedure was measured as having a maximum error of 1 pixel across the FOV. Each time the system starts this mapping can be run to reset the values if projections begin to drift due to temperature fluctuations or rough handling.

The frame rate of the detection and tracking code was also measured and averaged 60 fps with a deviation of 10 frames. The code was capped at 60 fps by a timer but the frame rate was measured for every frame and fed into the tracking code for the next frame to address latency.

Sorting results

Next we sought to validate the ability of the system to identify, track and project onto moving objects by performing a dynamic experiment using a sorting device as shown in Figure 58. The goal was to test how well the system could select one type of particle (90 μm) from a population flowing inside a microfluidic channel and encapsulate it preventing it from exiting the device. Then to use the results to identify the limits of particle velocity in a flow that can be successfully encapsulated. First we identified the cut-off velocity for which polymerization time limited the ability of the system to successfully encapsulate a particle. This was done by flowing particles between 0.1 mm/s and 30 mm/s as measured by machine vision. If incomplete polymerization around the particle was observed as it left the FOV then that value was determined to be outside the limits of the system. The remaining data represented the velocities for which polymerization was guaranteed. This maximum value was identified as 10 mm/s. Beyond this velocity the probability of successful encapsulation drops dramatically. In addition, the error for the measurements was calculated as an average of 4.7 pixels and a maximum of 10 pixels.

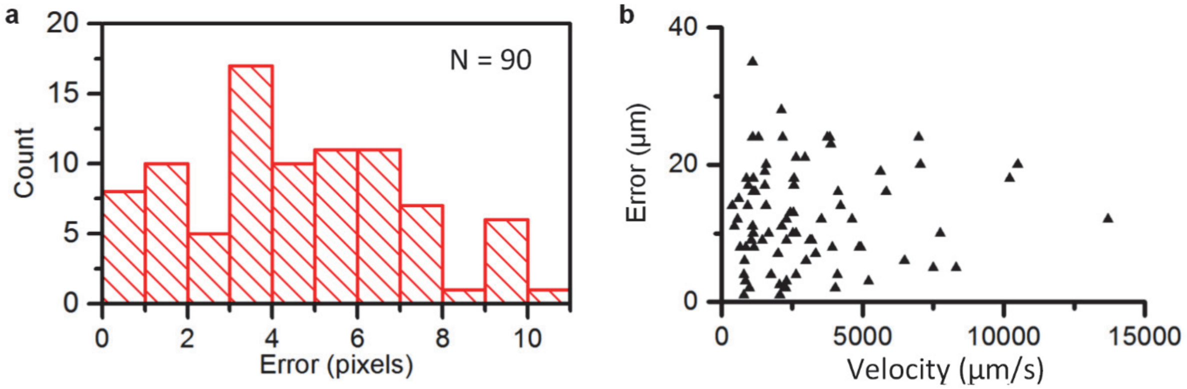


Figure 71. Particle velocity and projection error. A) Histogram of the error between particle center and projection center. B) Scatterplot of error between particle center and projection center in microns and particle velocity in microns.

The following experiment flowed a mixture of 90 µm, 45 µm, 10 µm and 5 µm PS beads through the device. Flow rates were limited to 10 mm/s. The goal was to identify the 90 µm particles and encapsulate them.

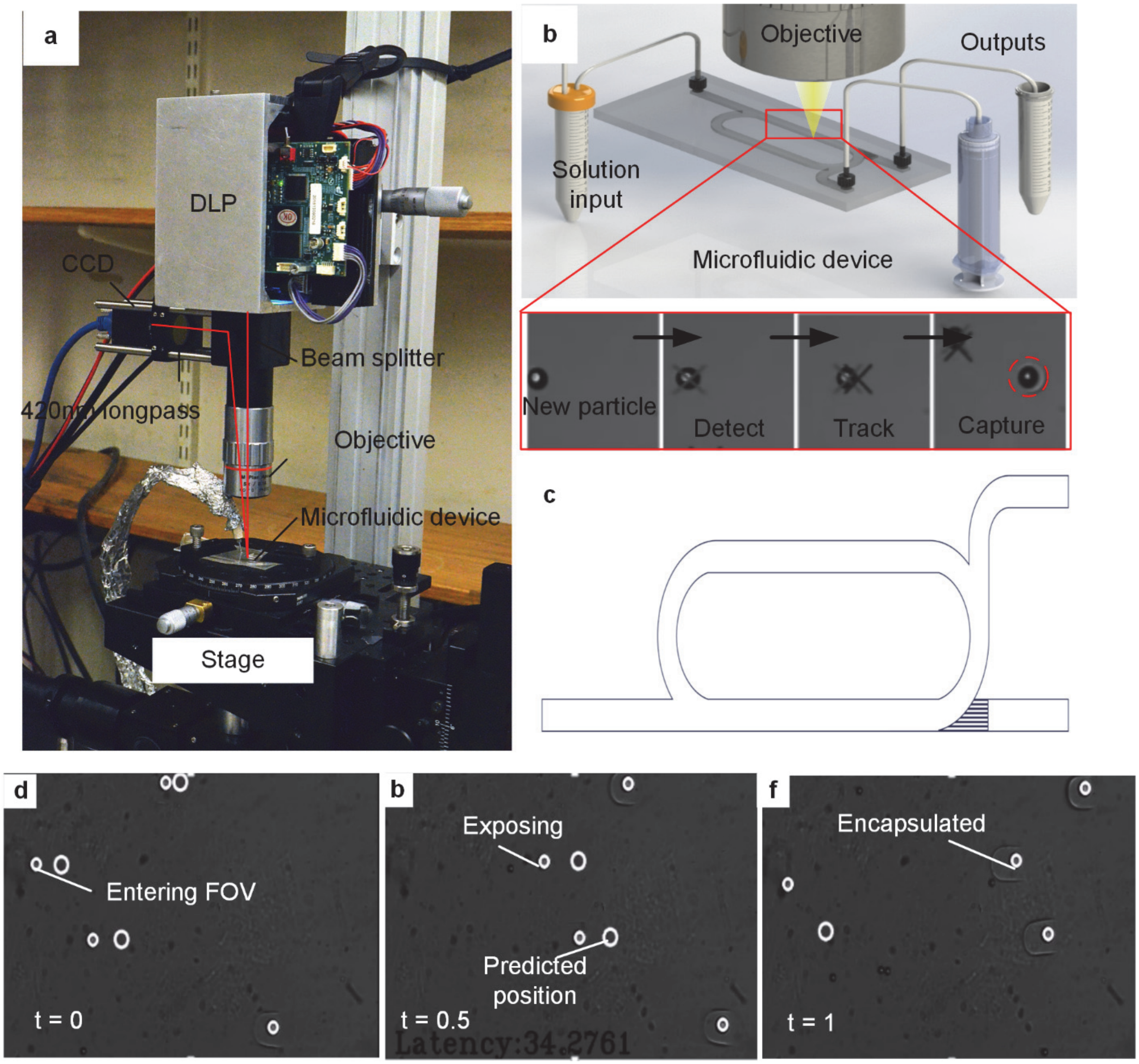


Figure 72. Method and demonstration of sorting PS beads. a-c) Reminder of the structure of the system and the microfluidic channel that is used to separate particles. The basic process is to identify, track and then encapsulate or encircle the particle. c) Shows the racetrack fluidic device they particles circle around while being processed. d-f) Two particles being encapsulated over 1 second. The small white circle is the detected position of the particle of interest. The larger white circle is the predicted position for it in the next time step.

Using the system as shown in Figure 72 the sensitivity, specificity, precision and accuracy (definitions in Table 8) were calculated for the experimental demonstration of sorting particles by size (Table 8). The sensitivity below 10 mm/s in flow was 100% with a specificity of 99.9%.

This means that the system correctly identified and sorted the beads consistently. The precision and accuracy were 98.9% and 99.9% respectively.

Table 8. Performance metrics for sorting and encapsulation methodology using dynamic lithography.

Performance measure	Calculation	Value
Sensitivity	TP/(TP+FN)	1
Specificity	TN/(FP+TN)	0.999
Precision	TP/(TP+FP)	0.989
Accuracy	(TP+TN)/(P+N)	0.999

Throughput analysis of dynamic lithography for generalized flow manipulation

For tailoring of dynamic lithography toward applications, it is necessary to have a rigorous understanding of how the density of particles, FOV, frame rate and polymerization kinetics governs speed and accuracy. We consider the system to be a manufacturing system where operations are performed on particles so we will model it using manufacturing concepts.

Moreover, modeling the throughput helps identify what governs the process limits. For example, is it the hardware or is it physically limited (e.g. photopolymerization)? The component of the system that limits throughput will be called the bottleneck and can be used to describe the sensitivity of the system to uncertainty.

The throughput of this type of system can be described as highly parallelized with each row of mirrors on the DMD acting as a processing lane or machine. We will make several assumptions for this model to be accurate:

1. Objects travel in a straight line and polymerization of one particle does not affect the motion of any other particle.
2. All rows are equivalent to single machines and act independently; consequently the line is trivially balanced.
3. Process times are random and occur according to a distribution known as a normal distribution.

From these assumptions, the throughput of the system can be modeled by first considering Little's Law [86]. According to Little's Law the steady state number of particles in a system divided by the processing time is equal to the throughput. The limiting case in which the critical number of particles in the system (P_0), having no variability, can describe the maximum throughput (r_b) with minimum cycle time to process (e.g. encapsulate) a particle (T_0).

$$r_b = P_0/t_0 \qquad \text{Equation 24}$$

The critical number of particles in the system can be described by considering the relationship between the number of pixels on the DLP and the ratio of particle diameter to pixels.

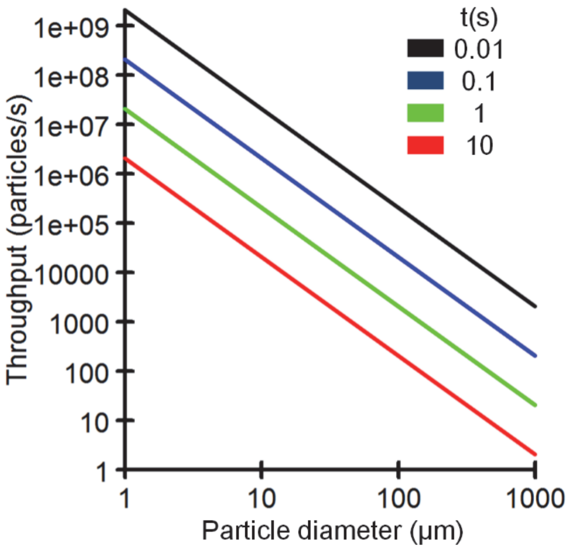
$$P_0 = \left(\frac{(pm)^2}{\pi\left(\frac{d + l_p}{2}\right)^2}\right)^2 \qquad \text{Equation 25}$$

Here p represents the number of pixels along an edge of the DLP (1104 pixels), assuming a square DLP; m is the conversion from microns to pixels for the current objective, which is divided by the equivalent particle area as defined by the particle diameter (d); and the line pair resolution of the system (l_p). Therefore, P_0 is equivalent to the steady state expected number of particles that can be in the system at any point which is visually a grid of particles separated by the minimum distance at which particles can be processed without interfering with each other. The resulting maximum throughput is equal to:

$$r_b = \left(\frac{(pm)^2}{\pi\left(\frac{d + l_p}{2}\right)^2}\right)^2 / t_0 \qquad \text{Equation 26}$$

$$r_b = bottleneck\ throughput$$
$$p = Pixels\ on\ dlp$$
$$d = Particle\ diameter$$
$$l_p = line\ pair\ resolution$$
$$t_0 = minimum\ cycle\ time$$
$$m = conversion\ from\ \mu\ to\ pixels$$

This suggests that, in the absence of variability, the throughput of the system is equal to the values shown in Figure 73 for varying processing times (0.01s, 0.05s, 0.25s and 1.25s) for particle diameters from10 µm to 100 µm.



.

Figure 73. Throughput of the system for varying diameters as a function of processing times with no variability.

The system is never in this ideal state because space between particles is required to photopolymer to surround individual particles and to allow particles to flow past on another. Therefore, we multiply P_0 by a factor f that represents the volumetric fraction of particles in the medium used. The volume fraction we will use for most of this analysis is 1M particles/ml unless the volume fraction and diameter are both explicitly stated. The relationship can be seen in Figure 74 and follows the corresponding equation.

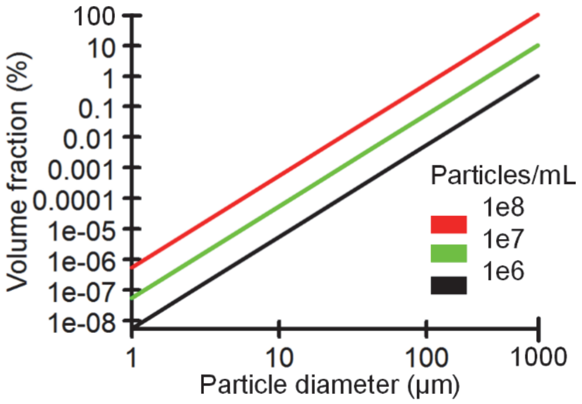


Figure 74. Volume fraction of a suspension of non-colloidal particles compared to particle diameter.

$$f = \frac{(\frac{32}{3})b\pi d^3}{V * 10^{14}}$$ Equation 27

$$f = volumetric\ fraction$$

$$b = number\ of\ particles\ per\ volume$$

$$V = volume\ in\ ml$$

$$d = the\ diameter\ of\ the\ particle\ in\ \mu m$$

Put another way the volume of solid particles in relationship to the medium is equivalent to the 2D fill percentage (area) of particles across the DLP.

$$r_b = f(\frac{(pm)^2}{\pi(\frac{d+l_p}{2})^2})^2/t_0 \qquad \text{Equation 28}$$

However, we must consider that not all variables may be deterministic but have some variability such as, the volume fraction of particles in the system and the processing time per cycle. Volumetric fraction is variable because the number of particles in view will change with some probability. We assume that the volume fraction f of particles in the medium, which will characterize an ideal, monodisperse hard-sphere suspension, is Normally distributed such that:

$$f \sim N(\mu_1, \sigma_1) \qquad \text{Equation 29}$$

Where μ_1 equals the mean fraction of particles in suspension and σ_1 equals the standard deviation. We also assume that the processing time is Normally distributed such that:

$$t \sim N(\mu_2, \sigma_2) \qquad \text{Equation 30}$$

The processing time is a summation of all the individual steps to process a particle. These are shown in

Table 9 and include grabbing an image, transferring the image, demosaicing the image, detecting particles, tracking them, drawing the mask and sending it to the projector long enough to polymerize the medium. The table shows measured values for each step. Four terms will add parameters to our model. Those are Detect, Track and Cure and Draw. Detect and Track are both dependent on the volume fraction of particles according O(N!) [87] proportion, because the algorithms used are assumed to be worst case brute-force methods. The Draw process is proportional to the number of particles or O(N). Additionally, the Cure process follows the working curve of the photopolymer which can be affected by concentration of the monomer, concentration of the Photoinitiator, the illumination radiance and depth of cure.

Table 9. A description of the processing times by task and links to their derivation.

Station name	Variable	Processing time (ms)	Source
Camera	C_t	11.1	1/fps
Transfer	W_t	11.1	1/60
Demosaic	D_t	6.2	Measured
Detect	X_t	11	Measured
Track	Y_t	10	Measured
Draw	V_t	7	Measured
Transfer	W_t	$16.\overline{6}$	1/60
Project	Z_t	0.02	Estimated
Cure	H_t	50.0	Empirical

Each of the variables in Table 9 contribute to the system processing time, therefore the total processing time can be calculated as their sum (Equation 31).

$$t = C_t + 2W_t + D_t + X_t + Y_t + V_t + Z_t + H_t \qquad \text{Equation 31}$$

As a natural consequence of the properties of Normal distributions we can then substitute the random variables into Equation 28 and the result is a random variable represented by a new Normal distribution.

The resulting mean throughput is equal to:

$$r_\mu = \frac{\mu_1 \left(\frac{(pm)^2}{\pi \left(\frac{d + l_p}{2} \right)^2} \right)^2}{\mu_2} \qquad \text{Equation 32}$$

And the resulting standard deviation is equal to:

$$r_\sigma = \frac{\sigma_1 \left(\frac{(pm)^2}{\pi \left(\frac{d + l_p}{2} \right)^2} \right)^2}{\sigma_2} \qquad \text{Equation 33}$$

Thus we plot (Figure 75) the expected steady state throughput and the effect of variability for varying system wide sigma values (1, 3, 6) with a mean processing time of 0.123 and a mean volumetric fraction of 0.1%. Here we assume that the population standard deviations are 0.0002 and 0.2 respectively. These were chosen until measured or calculated values are given but indicate the sensitivity of the throughput. The maximum processing time can then effectively be calculated as the processing time at six sigma. Therefore the lower limit throughput can be observed for a specific diameter. Perhaps a better way to state this fact is that the expected steady state throughput for the system is between the green lines. The system is constrained in that throughput can't be greater thanr_b.

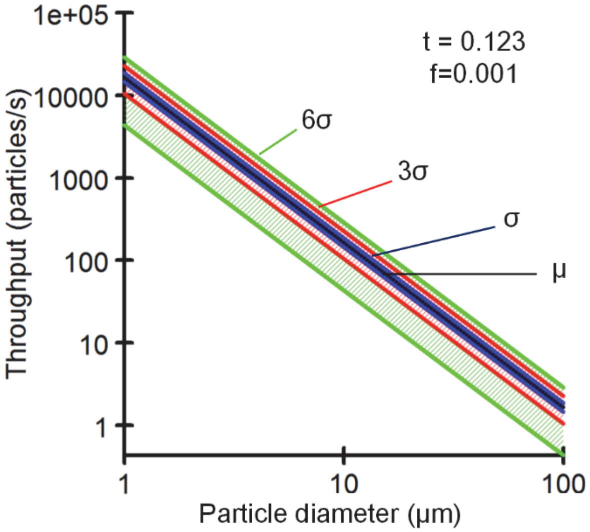


Figure 75. Throughput of the system when taking the standard deviation of the processing times and volumetric fractions into account. This represents the limits of the system under these assumption that processing time and volumetric fraction are independent.

Having established the basic relationship between system parameters we want to understand the sensitivity of the system to changes in these parameters to identify both the most likely cause of failure (partially or uncured particle) and best opportunity for improvement. Looking at our system throughput equation we have two opportunities for improvement 1: Reduce processing time 2: Increase volume fraction of particles. The second option may impact processing time as will be discussed in the next chapter. For example, if the number of particles that can be processed is limited to 10,000 (to not impact the system) then Else, we have to look at increasing the number of DLP's working in tandem, which would have the effect of increasing working area.

Now consider the processing times in greater detail. Figure 76 shows throughput compared to changes in processing time. It follows that reduction in processing time is critically important as an order of magnitude change has the inverse effect on throughput. Given the steady state measure of cycle time of 0.123 seconds the throughput of the system as shown in Figure 76 is on the order of 10,000 per second.

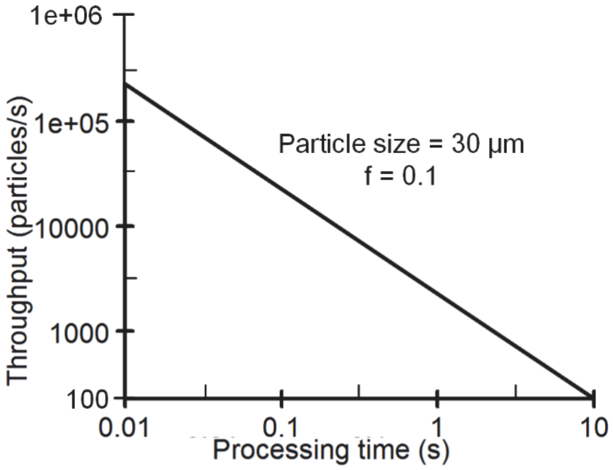


Figure 76. The throughput of the system when only varying processing time. Particle size and volumetric fraction are held constant.

From the initial discussion on processing times the curing speed of the photopolymer, detection rates and tracking rates each show the greatest potential for initial improvement. All other rates are limited by design of off the shelf components or standards such as USB3.

What this does not account for is how variability affects the likelihood of the system to perform correctly and what the rate of successful encapsulation is likely to be. This problem can be defined and searching for the probability that a particle is correctly encapsulated given some velocity or processing time. This probability is controlled by both the likelihood of correctly detecting/tracking the particle and the probability that the polymer receives the correct dosage of light. Empirical values for this were discussed in Figure 71b, where 10 mm/s velocity for a FOV of 5 mm saw a dramatic decrease in success rates from 100% to 8%.

Conclusion

A new method of integrating machine vision into lithography to create dynamic responses to micro-environments has been engineered. It was enabled by the combination of quality micro-projection system and adaptive software. Accurate projection (4.7 pixels) by the software was enabled by a fast and accurate detection and tracking algorithm, latency correction via a Kalman filter and an increased frame rate from bit plane slicing. The application of the Kalman filter and tuning via the GA to minimize the projection error provide a platform for tuning the system quickly for any motion type. During experimentation we also observed that lighting played an important role in performance as the system would detect the edges of the polymerized region as a new particle unless it was either filtered programmatically or removed by using dark field lighting.

Comparable methods to manipulate objects in flow such as microfluidic focusing or laser trapping are either inflexible or limited to very small particles (< 1 μm). The ability to program applets into the software of this system enables it to adapt to a wide variety of problems, particle sizes and shapes. Here we discussed the ability to use it for sorting, but in the coming chapters we will discuss how to print particles with it as well. In addition, the flexibility of machine vision algorithms enables the same system to adapt to particles with non-spherical shape. The mask can be generated to align itself to anisotropic particles including the addition of a barcode or feature to change its hydrodynamic properties in flow. Moreover, we can extend the *C. elegans* study to adaptively project in response to worm behavior taking into account the worm's elongated shape.

Future work could improve performance in sorting further by implementing non-linear motion into the predictive equations. In microfluidic flows that rapidly change the assumption that motion is linear frame to frame may be invalid. There is also opportunity for improvement in the performance of the photopolymer to yield faster polymerization at lower radiance levels. This would result in increased throughput of the system by reducing the processing time and enabling faster flow velocities.

In addition to improving the method and the models of what governs it, we want to apply it to more applications. Inspired by the demonstration of sorting polystyrene beads shown here one can imagine challenges that may be solved if the systems performance reaches an appropriate level. Examples include separating circulating tumor cells, encapsulating cells that approach drug laden microparticles to test efficacy, or, as we will discuss in the coming chapters photopatterning micro objects into arbitrary arrangements for tissue engineering.

CHAPTER VI. QUANTITATIVE COMPARISON OF DETECTION AND TRACKING METHODS APPLIED TO MICROPARTICLES IN MOTION

Abstract

The effect of particle properties on the performance of 5 detection algorithms and two tracking algorithms common to machine vision are measured against simulated videos representing basic properties in biological videos (Signal to Noise, particle complexity, particle density, particle velocity, blur and particle motion) . The algorithms tested include SURF, Hough detection, Caany, Otsu and Threshold for detection and nearest neighbor or Kalman filtering for tracking. Of these, the Speeded Up Robust Feature (SURF) detection was robust but Thresholding performed well under controlled conditions. Thresholding was 3 orders of magnitude faster than SURF making it a better candidate for real time processing. Finally, we evaluate each function used in dynamic lithography described in 0 to characterize its algorithmic complexity. I'd like to acknowledge the contributions from John Lewandowski in the Mechanosynthesis group in assisting in the design of this study and for his work in developing a simulation platform in Matlab to generate simulated videos to test with.

Introduction

Real-time detection and tracking of cells and organisms is important to both research and clinical methods including flow cytometry, bio-informatics, and single cell studies. Advances in microscopy have resulted in impressive capabilities to image fundamental components of cells such as the cytoskeleton. Quantitative methods to track the motion of particles (bead, cell, Zebrafish, etc.), enables the derivation of numerical quantities from the image or video sequences from which biologically relevant data can be inferred.

Before real-time image processing algorithms were available post-processing of data would create aggregate information about populations of cells or micro-objects after an experiment. During such static image analysis, many methods and parameters can be applied until a satisfactory extraction of information about the particles is achieved. Trends in analysis of large populations of objects (e.g. cell colonies) and complex structures (e.g. neurons, cytoskeleton, growing organisms) demand a comprehensive understanding of how the performance of image analysis methods can be optimized for real-time applications.

The problem is therefore: How to detect a particle spatially and then link it to a previous frame temporally. While the detection algorithms are primarily concerned with extracting the location and morphology descriptors for a particle the linking or tracking algorithm is concerned with creating a history of the particles location with time.

There is a number of software packages designed to enable particle detection. However, choosing the correct analysis method and implementing it with the correct parameters can be difficult and compromise its overall efficacy. Moreover, they are not designed to integrate into real-time systems such as probe stations, microfluidics and dynamic lithography. To do so requires an API interface to Recently published work by Chenouard et al. compared algorithms by distributing simulated videos of cell like environments to bio-imaging laboratories, each laboratory then tested and reported the performance of their own detection and tracking method [88]. The study found 14 different methods were affected by three main factors (particle dynamics, density and signal). They concluded that each situation is unique and requires special care to identify the correct particle tracking method, a conclusion that left open the possibility of further investigation into the relationships between the parameters and method performance. Therefore, our goal is to develop a comprehensive framework enabling the evaluation of a particle detection and tracking algorithm for dynamic lithography. We show that, by using simulated videos while varying particle properties we can characterize the performance of five detection algorithms (Speeded up Robust Feature Detector (SURF), CAANY, Hough, Otsu and Threshold) by the root mean square error (RMSE) of the detection, the sensitivity to the presence of particles and the number of false positives. We then compare how each performs while 7 primary factors (particle density, particle size, particle velocity, particle motion, particle complexity and image signal to noise ratio) are independently varied. The then report the computational cost of each method and how the best performing method degrades as the number of particles increase. Finally, interpretation is provided for the results indicating how to select from the tested algorithms and how to test additional methods.

Object relationship and organization in images

Figure 77 shows some considerations that describe a particle and its relationship to the image or other particles. These considerations ultimately constituted the population of parameters that we could select from to examine in the study.

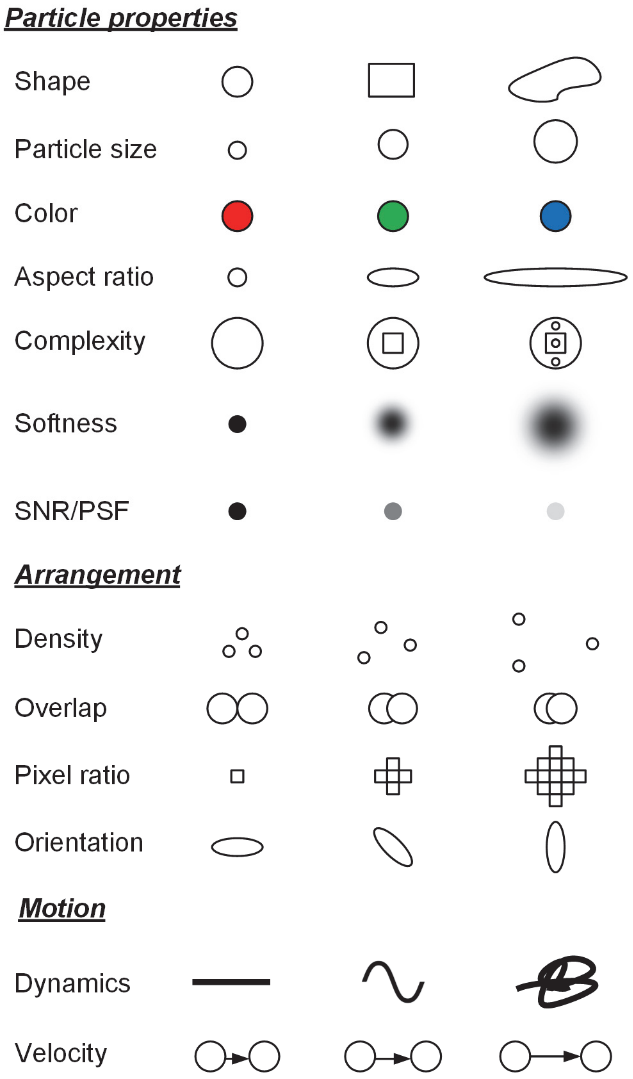


Figure 77. Particle properties their arrangement and motion characteristics

Methods

To provide an objective comparison of the detection and tracking methods, we developed ground truth videos of particles to simulate various conditions that might be found when analyzing cells in a system. The ground truth is an unambiguous description of the particle and its position at every time step. This is compared with the predicted values from each method to generate measurements of performance.

Generating data sets and ground truth by simulation

We developed a simulation framework using MATLAB's basic particle system toolbox (http://www.mathworks.com/matlabcentral/fileexchange/14314-particle-system-toolbox [89]) and the final code used to create the particles inside of the particle system can be found in the Appendix that allowed us to specify and simulate all the parameters in Figure 77 and output the simulation as videos which are easily processed by the detection and tracking code.

Within the simulation, we ensured that every particle created was tagged with a unique identification number and its true particle size to ensure that sensitivity and specificity data could be generated, aggregated and further manipulated with ease. Meanwhile, for every frame, the MATLAB code recorded the x-y position pixel coordinates of the particle (rounded to the nearest whole number) and tied them to its ID. The resulting output file consists of frame, ID, x-position, y-position, and size for each time in the simulation.

A matrix of videos was generated combing all parameters shown in Figure 78. These include particle dynamics, size, velocity, density, signal, blur and complexity.

Table 10. Matrix of test parameters for simulation study.

Parameter	Type I	Type II	Type III
Motion	Linear	Sinusoid	Brownian
Representing	Microfluidics	*C. elegans*	Self-assembly
Speed (mm/s)	0.1, 0.5, 1, 2	0.1, 0.5, 1, 2	0.1, 0.5, 1, 2
Particles (um)	5, 10, 45, 90, 90 and 45, 10 and 5	5, 10, 45, 90, 90 and 45, 10 and 5	5, 10, 45, 90, 90 and 45, 10 and 5
Size (pixels)	1276 x 794	1276 x 794	1276 x 794
Length (frames)*	1800	1800	1800
Density (low, medium, high)**	50, 500, 1000	50, 500, 1000	100, 500, 1000
SNR (levels)	1, 3, 5, 8	1, 3, 5, 8	1, 3, 5, 8
Complexity	No, Yes	No, Yes	No, Yes
Blur	None, Low, High	None, Low, High	None, Low, High

*For the fastest speed of 2 mm/s, only 900 frames were used (at 30 fps, this constituted 30 seconds). 1800 frames was the typical length due to video file writing limitations.

**For simulations with two sizes of particles, medium and high densities constituted 250 and 500 particles within the field of view and for linear and sinusoid videos there were less particles in the low scenario to ensure a minimal intersection fraction.

Each video file and corresponding data output file was labeled with a unique identifier and then processed through the detection and tracking code to output quantitative performance criteria (RMSE, sensitivity, specificity, false positives, and computational cost) and link it to the unique combination of parameters. One simulation was generated for each unique combination of parameters in Table 10, allowing for a full factorial analysis to be performed across all of these parameters. Examples that demonstrate the effect of each parameter on the video result are shown Figure 78.

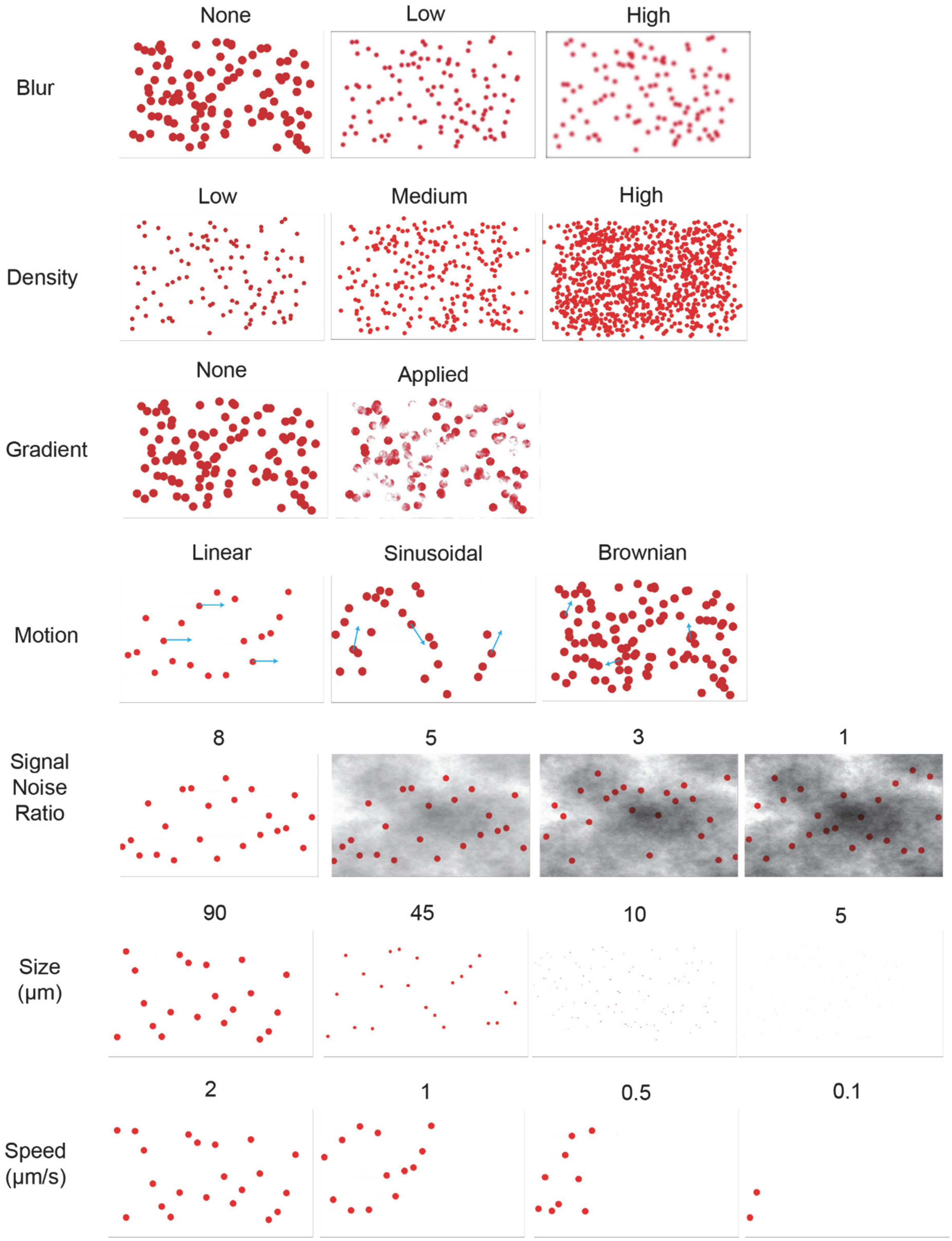


Figure 78. Properties of particle simulations

Dynamics

We decided upon three different motion types, linear, sinusoid, and Brownian, for two specific reasons. The first reason is that each of these motions correlates to an exemplary model system. For example, linear motion corresponds to detection and tracking of particles or cells captured by a laminar flow in a microfluidic device. Sinusoidal motion occurs when *C. elegans* are swimming. Brownian motion describes diffusive migration of particles. This is particularly the case when the time between frame grabs is large such that the instantaneous vector of the particle is uniformly random. This is very similar to what would happen when observing self-assembly of particles subject to Brownian motion. The second reason is for testing the capabilities and limitations of the tracking algorithms to correctly identify and predict the position of particles undergoing different motion trajectories. To test its limits with predicting a new particle's position, we wanted constant velocity in a single direction (linear), changing velocities in multiple directions (sinusoid), and arbitrary instantaneous velocity in an arbitrary direction (Brownian). We believe these three types are representative of a wide variety of motion types; other systems may be approximated by combinations of these motions. From the tracking algorithm perspective, all else being equal, theoretically these types of motion should increase in difficulty.

Length and Speed

It was important for the simulations to constitute a sufficient period of time to ensure data spanned over many sub-cases within the scenarios and to provide data to gather statistically sound results. Therefore, the majority of the simulation videos were one minute in length, constituting 1800 frames at 30 frames per second to match our experimental parameters. The various speeds used ranged from 0.1mm/s to 2.0mm/s. Based on the field of view, this velocity ultimately dictated how long the particle was in the system and how many interactions would undergo and frames it would be present in.

Particle size

The particle sizes ranged from 5 um to 90 um for four key reasons. First, the goal was to test an order of magnitude decrease in size and do so in collaboration with standard sizes of polystyrene beads. Second, 5 um (3 pixels) neared the resolution of our camera and so was a suitable lower boundary. Third, we chose similar sizes such as 10 um and 45 um (double and half the diameters, respectively) to challenge the specificity of the detection. Fourth, the pixel sizes of these particles reflect similar sizes of cells, viruses, and other microorganisms in a biological system. Specificity was only determined for simulations with two sizes, to test the algorithm's ability to differentiate sizes rather than focusing on less interesting false negatives.

Density

According to previous work, increased density is one of the largest contributors to decreased sensitivity [88]. Therefore, careful selection of the density parameter was crucial to a thorough understanding of its sensitivity. For low-density scenarios, we desired to have no overlap of particles. To achieve this we used 50 particles for linear and sinusoid and 100 particles for Brownian.

SNR

Using the GNU Image Manipulation Tool (GIMP), we established standard layers of fractal noise or "cloud fog" that could be overlaid on the simulations to decrease the signal-to-noise ratio (SNR) to desired levels. The maximum SNR (particles on a white background) that we could achieve given our system and particle intensity was 8. This would simulate particles flowing through a channel after already having been separated, diluted, and washed. Previous work has shown that an SNR of 5 or greater is necessary to achieve good detection therefore, we weighted the parameter selections to the more sensitive side of the spectrum, using values of 1, 3, 5 and 8. These values were designed and confirmed using ImageJ to analyze the average intensity of the particle in relation to the entire system, using the equation:

$$SNR = \left|\frac{\bar{S} - \bar{P}}{\sqrt{|\bar{S}|}}\right| \qquad \text{Equation 34}$$

Where SNR represents the signal to noise ratio, $\bar{S}$ is the average intensity of the system and $\bar{P}$ is the average intensity of the particle. This specific equation was implemented because it balanced typical average signal to noise calculations with influence of the density and distribution of particles and noise in the system.

Blur

The blur effect was chosen because of two common occurrences: 1. The camera is slightly out of focus and causes the subject particles to blur, and 2. The particles in question have soft edges that either blend into the background or cause the typically circular shape to look irregular.

The blur effect tested the capacity of the contour-finding tool in the detection phase of the code. Blur was applied as a Gaussian blur according to the following:

$$G(x) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}} \qquad \text{Equation 35}$$

Complexity

While the SNR parameter allowed the investigation of system-wide noise, there also exists cell-to-cell or particle-to-particle variation, meaning not every particle is identical in size, shape, color, or texture. The gradient effect, which consisted of scaled intensity values across the diameter of the particle, was overlaid on the system to investigate how particles that differed in intrinsic characteristics affected the overall performance of the algorithm.

Renderings

Adobe Creative Suite 6 After Effects was used to perform the rendering of the videos with noise, gradient, and blur effects (See SI for exact software specifications for all of the following parameters). The renderings were performed in escalating order of individual particles to particle population to particle system – blur, gradient, and then noise – so that the desired scenarios and underlying representative conditions were not confounded.

Measures of performance

Choosing how to rank the performance of the methods for objective comparison is critical and largely determined by the intended use. For example sorting particles of rare types requires a high sensitivity. This may accept a high false positive rate. On the other hand printing high value cells is more stringent requiring both high sensitivity and a low false positive rate.

We chose six measures of performance. The first is the root mean square error between the ground truth position of a particle and the detected position of a particle. This is an important measure in dynamic lithography to ensure proper curing location. In addition, sensitivity and specificity are measured to illustrate the capacity to both identify a particle and distinguish it from others. Sensitivity is defined as the number of true positives divided by the total number of positives present, while specificity is the number of true negatives divided by the total number of negatives present. False positives are reported because in conjunction with sensitivity they indicate if the method is being overly sensitive. In many applications such as diagnostics, there is a fine balance and even thresholds to maintain between the competing false positives and sensitivity. Finally, the computational cost of each method is examined to determine the potential and limits of each method in a real-time scenario.

Evaluating methods against ground truth

A program was written in C++ to load the prepared simulated videos and ground truth files into a detection experiment. The software was designed to load the list of each video and the parameter it was testing. It would iterate through each video testing all five methods and the two tracking methods while collecting metrics of performance for each frame, aggregating the results and writing them to a separate file. At the same time the history of tracked positions for every particle was saved to another file for future processing.

During each frame of the video processed the frame would be converted to grayscale then passed to a detection function which would apply the detection method generating a list of particles in the frame. Then the list of particles would be passed to a tracking function where each object is matched to a previous particle. If no match can be found then the particle is added to the list of particles as a new particle.

The list of tracked particles is then passed to class which compares the list to the ground truth for that frame of the video. To do this the distance between each particle in ground truth and each particle in the frame is calculated. The nearest match is set as a match and the ID of the ground truth particle and particle in the frame are linked. At the same time the root mean square error between ground truth and the particle in the frame is recorded to a history for that ground truth particle.

Finally, the true positives, false positives, false negatives and if applicable true negatives are recorded for each particle. If a particle in the frame is within the minimum distance to be considered a true positive then it is recorded as a true positive. For these experiments that value was 10 pixels for 45 μm and 90 μm, 5 pixels for 10 μm and 2 pixels for 5 μm. This ensured the detection point was within the bounds of the particle, meaning less than its radius... The remaining particles in ground truth that had not found a match were marked as false negatives. Finally, the particles in the frame that did not match were marked as false positives. For each particle in ground truth the sensitivity and if applicable the specificity were calculated using the collected positive/negative values.

Lastly, for each frame the computation time of the entire frame as well as individual functions was calculated and recorded by using the system clock to measure the beginning and end of the function and calculate the delta time.

The input parameters of each method were tuned by hand using the most confounded video. Then, unless otherwise noted, they were not changed to illustrate the sensitivity of each method to the changing factors in the image.

Detection algorithms

The detection algorithms used include SURF, Hough, Canny, Otsu and Threshold. These methods were both accessible and had a history of being successfully applied. Canny, Otsu and Threshold all rely on some form of threshold and may be referred to as the threshold based methods because we hypothesis that they may perform similarly.

a. SURF (Speeded Up Robust Features)

SURF is a robust feature detector presented by Herbert Bay et al. in 2006 [90]. It is built using the Hessian-matrix approximation. By doing so the integral images made by Viola and Jones [91] can be used. The integral image allows fast box type convolution. The integral image is formed by iterating each pixel and for each pixel summing all of the intensities from the input image within a rectangular region.

$$I_{\Sigma(x)=} \sum_{i=0}^{x} \sum_{j=0}^{y} I(i,j) \qquad \text{Equation 36}$$

Next the Hessian matrix is applied to find locations where the Hessian determinant is maximized to identify interest points. Given a point $\ell = (x, y)$ in an image I, the Hessian is as follows with a scale of σ:

$$\mathcal{H}(\ell,\sigma) = \begin{bmatrix} L_{xx}(\ell,\sigma) & L_{xy}(\ell,\sigma) \\ L_{xy}(\ell,\sigma) & L_{yy}(\ell,\sigma) \end{bmatrix} \qquad \text{Equation 37}$$

Where $L_{xx}(\ell,\sigma)$ is the convolution of the Gaussian second order derivative with the image I in point ℓ, and similarly for the values of L. However, approximations of the Gaussians are used when approximating the determinant of the Hessian.

$$det(\mathcal{H}_{approx}) = D_{xx}D_{yy} - (wD_{xy})^2 \qquad \text{Equation 38}$$

The relative weight w of the filter responses are used to balance the Gaussian Kernal and the approximation. The resulting determinant of the Hessian for each location in the image represents a blob response. To localize the blob points of interest non-maximum suppression is applied as described by Neubeck and Van Gool [92].

Finally, the descriptors are extracted from the resulting data and matched. First a square region is centered on the interest point, and if necessary it is oriented. The region is split up into 4x4 square sub-regions. For each subregion the Haar wavelet response is computed and summed up over each sub-region. These are saved into a feature vector in addition to the sum of the absolute values of the responses. The feature vector of values representing these four metrics is the descriptor. The list of vectors represents the detected particles. Further discussion can be given to size and orientation but we leave those to the original author's publications.

The parameters used for the SURF detection method are:

For 90 µm and 90 µm/45 µm scenarios

- Hessian: 10000
- Octaves: 2
- Octave layers: 15

For 45 µm scenarios

- Hessian: 10000
- Octaves: 2
- Octave layers: 15

For 10 µm and 10 µm /5 µm scenarios

- Hessian: 6
- Octaves: 4
- Octave layers: 64

For 5 µm scenarios

- Hessian: 1
- Octaves: 4
- Octave layers: 64

b. Hough

This method applies a Hough transformation to an image to identify circular particles. This method was invented by Richard Dude and Peter Hart in 1972, but popularized by Dana Ballard in 1981 [93]. The Hough transform extracts features from an image through a voting procedure that assumes any point in a binary image could be part of some set of circles. First the image is passed through a Canny edge detector. For all nonzero points, representing edges, the local gradient is calculated by the Sobel derivatives in the x and y directions. By parameterizing the circles in the set the algorithm can identify points that are related to the center of the circle. The candidate centers are chosen from the locations of the nonzero pixels in the output of the Canny detector. The accumulator (an array of positions and likely hoods for being centers of circles) is incremented for every point on the slope of the line some maximum and minimum distance according to the gradient. For example, if we are searching for a circle the data from the image may be imperfect and edges of a circle that are detected may have breaks due to noise.

The process the algorithm follows first creates an accumulator space (r, a, b). This space is made up of a cell for each pixel initialized to zero. Then beginning with the edge pixels in an image it increments all cells which could be the center of a circle. The increment is applied to cells that satisfy the equation:

$$r^2 = (i - a)^2 + (j - b)^2 \qquad \text{Equation 39}$$

Then iterate over all possible values of a that are found and find the possible values of b which satisfy the equation for each potential circle cell.

Finally, search for local maxima cells whose accumulated value is greater than its neighborhood. Past some threshold these are the cells that are most likely to be the center of the circle we are trying to locate.

The parameters used for the Hough circle detection method are:

For 90 µm and 90 µm/45 µm scenarios

- Minimum distance between circles: 20
- Upper threshold for internal Canny detector: 7
- Threshold for center detection: 7
- Minimum circle radius: 20
- Maximum circle radius: 30

For 45 µm scenarios

- Minimum distance between circles: 13
- Upper threshold for internal Canny detector: 7
- Threshold for center detection: 7
- Minimum circle radius: 13
- Maximum circle radius: 20

For 10 µm and 5 µm scenarios

- Minimum distance between circles: 7
- Upper threshold for internal Canny detector: 2
- Threshold for center detection: 2
- Minimum circle radius: 7
- Maximum circle radius: 12

c. Canny

Canny detection is a four-step algorithm. First a Gaussian blur is applied to reduce noise in the image [94]. Second, a gradient operator is applied for obtaining the gradients intensity and direction. The gradient operator is a Sobel edge detection operator which returns the first derivative in both directions (G_x, G_y) for each pixel. The edge gradient and direction are then calculated as follows:

$$G = \sqrt{G_x^2 + G_y^2} \qquad \text{Equation 40}$$

$$\theta = atan2(G_y, G_x) \qquad \text{Equation 41}$$

The edge direction angle is rounded to match either a horizontal, vertical or diagonal angle. Third the non-maximum suppression determines if the pixels is a better candidate for an edge than its neighbors. Finally, hysteresis thresholding creates contours from individual pixels. This examines if for a particular pixel, the pixels gradient is above the maximum threshold. If this is true, the pixel is marked as an edge pixel; if it is below the minimum threshold, it is rejected. However, it is between the two thresholds it is accepted if it is a neighbor of a pixel that exceeds the high threshold.

The parameters applied to the method include:

- Gaussian filter size: 3
- First threshold: 50
- Second threshold: 100

d. Otsu

We will describe the algorithm as shown by Al-Kubati et al. and others [95]. This method is built upon the threshold method but uses an additional step to automatically choose the threshold value that separates the pixels in an image into two classes such that the variance within the classes is minimized. In other words it tries to minimize the spread in intensity values for the collection of pixels above the threshold and the collection of pixels below the threshold. To do this with a threshold t, let $q_0\ and\ q_1$ represent the estimate of the class probabilities such that:

$$q_o(t) = \sum_{i=0}^{t} p(i)\,, q_1(t) = \sum_{i=t+1}^{l-1} p(i)$$ Equation 42

And the variances are defined by:

$$\sigma_0^2(t) = \sum_{i=0}^{t}[i - \mu_0(t)]^2 \frac{p(i)}{q_0(t)}, and\ \sigma_1^2(t) = \sum_{i=t+1}^{l-1}[i - \mu_1(t)]^2 \frac{p(i)}{q_0(t)}$$ Equation 43

The class means as shown above are defined as:

$$\mu_0(t) = \sum_{i=0}^{t} \frac{ip(i)}{q_0(t)}, and\ \mu_1(t) = \sum_{i=t+1}^{l-1} \frac{ip(i)}{q_1(t)}$$ Equation 44

Here, p represents the image histogram. Otsu, demonstrated that the problem of minimizing the inter-class variance is the same as maximizing the intra-class variance. This is written as:

$$\sigma_b^2(t) = \sigma^2 - \sigma_w^2(t) = q_0(t)[1 - q_0(t)][\mu_1(t) - \mu_0(t)]^2$$ Equation 45

Finally, Equation 42 can be maximized at the maximum value of $\sigma_b^2(t)$ yielding the threshold value t.

The process for this algorithm is as follows:

1. Calculate the histogram and probabilities of each intensity value.
2. Initialize $q_i(t)$and $\mu_i(t)$ equal to zero.
3. Step through all possible thresholds until the maximum intensity of the image is tested.
4. Update $q_i(t)$ and $\mu_i(t)$.
5. Compute $\sigma_b^2(t)$ and report the desired threshold corresponding to this value.

No parameter is given to this algorithm.

e. Threshold

This is possibly the simplest method tested as it creates a cut off value t in pixel intensity I. For each pixel if I > t then the pixel is set to 1 otherwise it is set to 0. The result is a binary image chosen by setting the intensity cutoff.

This written as t:

$$g(i,j) = \begin{cases} 1\ if\ f(i,j) \geq t \\ 0\ if\ f(i,j) < t \end{cases}$$ Equation 46

The threshold parameters used for all scenarios was:

- Inverted intensity threshold value: 30

Tracking algorithms

a. Nearest Neighbor

Classically known as the Post Office problem the nearest neighbor search method attempts to identify a particle detected in the current video frame with the closest proximity to the particle being tracked. Formally, it is described as given a set P of particle locations in space R and a query particle $s \in R$, find the closet particle in P to s. In this case the dissimilarity between members of P and s are measured in 2D Euclidean space as a distance [96]. This is a naive implementation of the algorithm such that the best match is tracked and replaced if a better is found as the entire list is searched. It has a complexity of O(N) where N is the cardinality of P which in this case is 100, 500 or 1000 particles in a frame.

The parameters used for this method are:

- Maximum search radius: 100

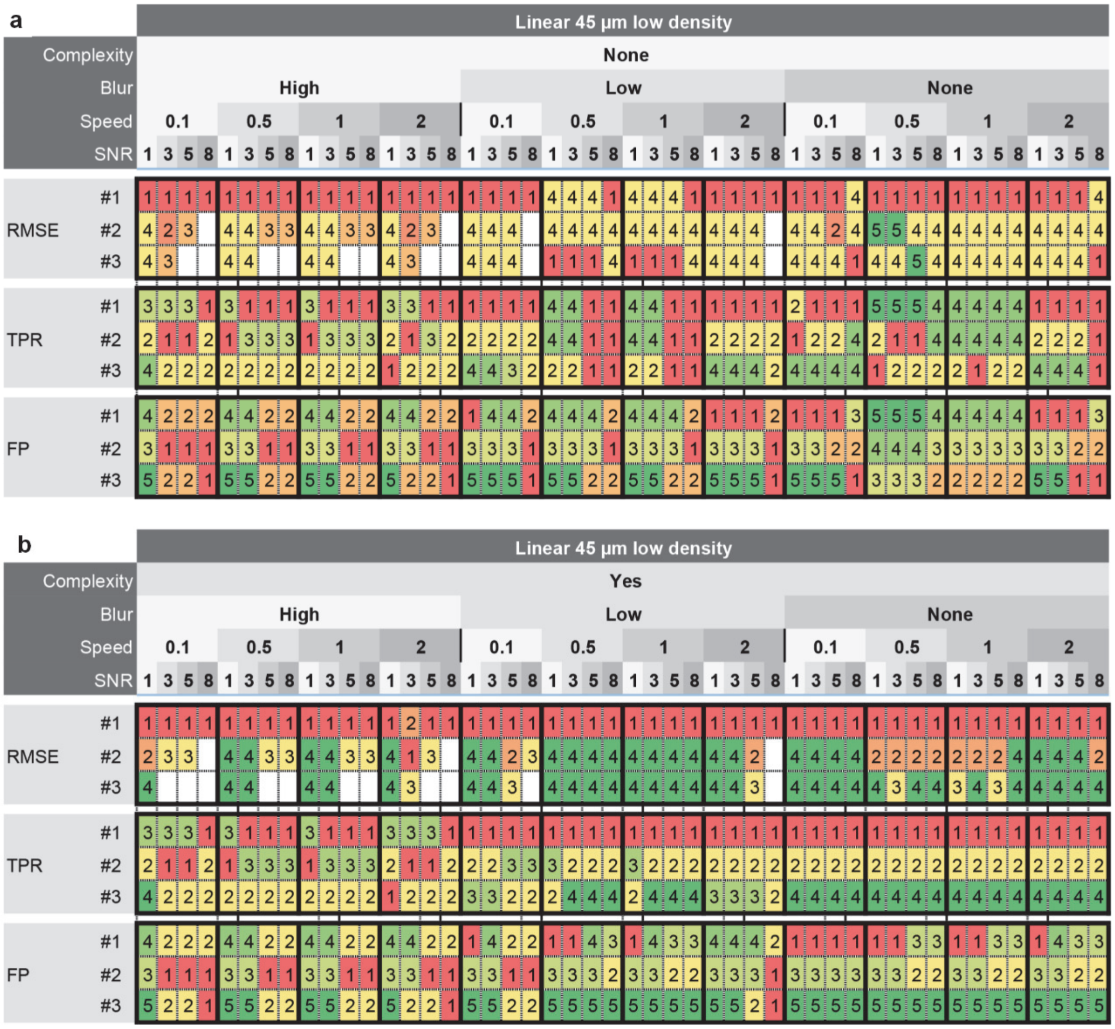
b. Kalman tracking

This method has been implemented but preliminary data suggests that the chosen implementation performs almost identically to Nearest Neighbor and insufficient time was available to collect additional data.

Ranking each methods overall performance

To convey an overall sense of how detection methods vary in success, we ranked the top three methods according to each value of the performance criteria under study. The metrics reported include the RMSE, Sensitivity (TPR) and False Positives (FP). The detection methods are coded as follows: 1:SURF, 2: Canny, 3:Hough, 4:Otsu, 5:Threshold and are also color-coded from red to green. Scenarios in which no value could be calculated were left blank (white).

Figure 79 shows the ranked results for 45 µm particles undergoing linear motion at low density. SURF performed the best across a variety of scenarios for the RMSE metric, followed by Threshold and Otsu. Sensitivity however, showed greater variety. For example, high levels of blur saw better performance from SURF, which looks for specific features/shapes rather than histogram variation. For other levels of blur, SURF and threshold methods ranked highest. Finally, false positives were consistently lowest using the threshold methods because intensity values due to the SNR varied across the image. However, Hough ranked second across a broad range of scenarios.



a

Linear 45 µm low density

	Complexity	None															
	Blur	High															
	Speed	0.1				0.5				1				2			
	SNR	1	3	5	8	1	3	5	8	1	3	5	8	1	3	5	8
RMSE	#1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
	#2	4	2	3		4	4	3	3	4	4	3	3	4	2	3	
	#3	4	3			4	4			4	4			4	3		
TPR	#1	3	3	3	1	3	1	1	1	3	1	1	1	3	3	1	1
	#2	2	1	1	2	1	3	3	3	1	3	3	3	2	1	3	2
	#3	4	2	2	2	2	2	2	2	2	2	2	2	1	2	2	2
FP	#1	4	2	2	2	4	4	2	2	4	4	2	2	4	4	2	2
	#2	3	1	1	1	3	3	1	1	3	3	1	1	3	3	1	1
	#3	5	2	2	1	5	5	2	2	5	5	2	2	5	2	2	1

	Complexity	None															
	Blur	Low															
	Speed	0.1				0.5				1				2			
	SNR	1	3	5	8	1	3	5	8	1	3	5	8	1	3	5	8
RMSE	#1	1	1	1	1	4	4	4	1	4	4	4	1	1	1	1	1
	#2	4	4	4		4	4	4	4	4	4	4	4	4	4	4	
	#3	4	4	4		1	1	1	4	1	1	1	4	4	4	4	
TPR	#1	1	1	1	1	4	4	1	1	4	4	1	1	1	1	1	1
	#2	2	2	2	2	4	4	1	1	4	4	1	1	2	2	2	2
	#3	4	4	3	2	2	2	1	1	2	2	1	1	4	4	4	2
FP	#1	1	4	4	2	4	4	4	2	4	4	4	2	1	1	1	2
	#2	3	3	3	1	3	3	3	1	3	3	3	1	3	3	3	1
	#3	5	5	5	1	5	5	2	2	5	5	2	2	5	5	5	1

	Complexity	None															
	Blur	None															
	Speed	0.1				0.5				1				2			
	SNR	1	3	5	8	1	3	5	8	1	3	5	8	1	3	5	8
RMSE	#1	1	1	1	4	1	1	1	1	1	1	1	1	1	1	1	4
	#2	4	4	2	4	5	5	4	4	4	4	4	4	4	4	4	4
	#3	4	4	4	1	4	4	5	4	4	4	4	4	4	4	4	1
TPR	#1	2	1	1	1	5	5	5	4	4	4	4	4	1	1	1	1
	#2	1	2	2	4	2	1	1	4	4	4	4	4	2	2	2	1
	#3	4	4	4	4	1	2	2	2	2	1	2	2	4	4	4	1
FP	#1	1	1	1	3	5	5	5	4	4	4	4	4	1	1	1	3
	#2	3	3	2	2	4	4	4	3	3	3	3	3	3	3	2	2
	#3	5	5	5	1	3	3	3	2	2	2	2	2	5	5	1	1

b

Linear 45 µm low density

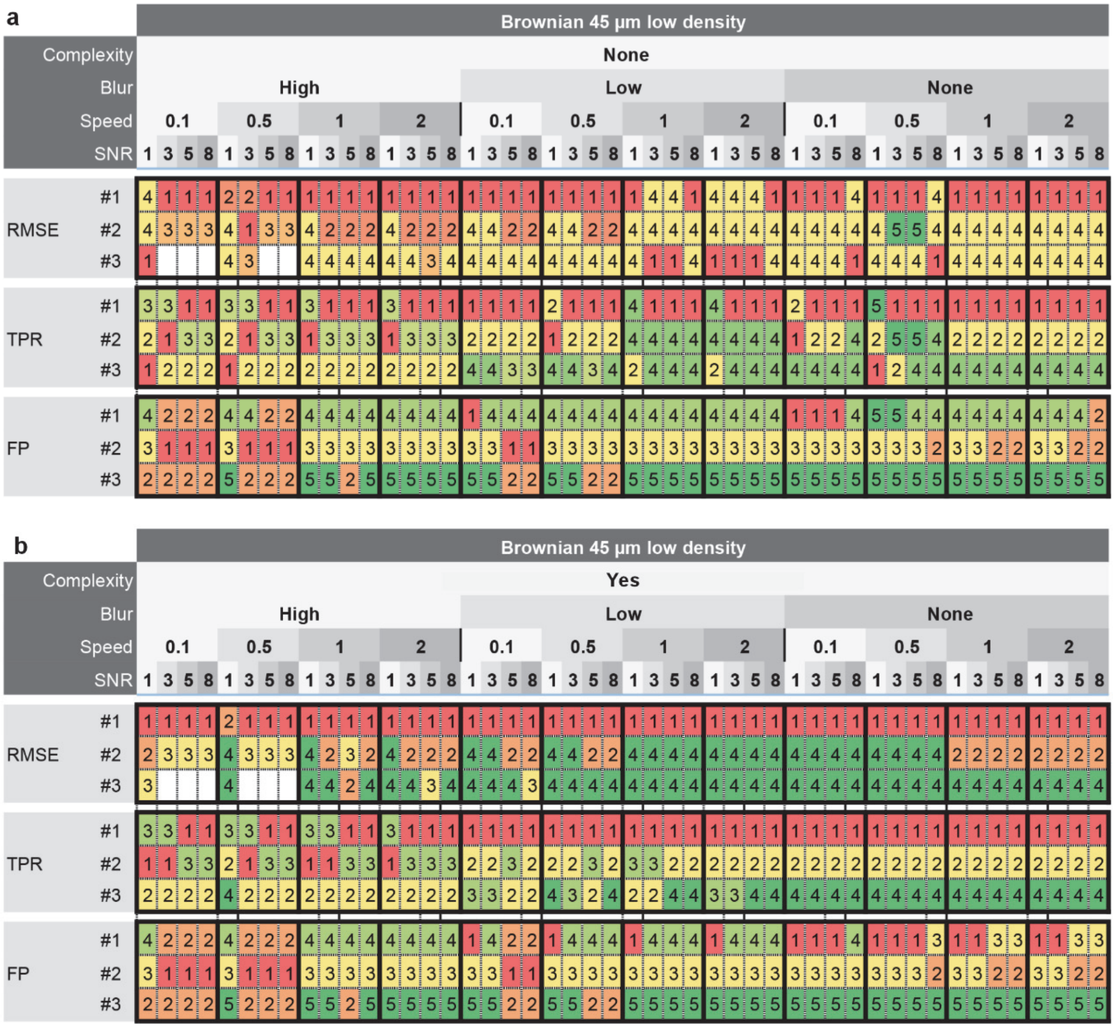
	Complexity	Yes															
	Blur	High															
	Speed	0.1				0.5				1				2			
	SNR	1	3	5	8	1	3	5	8	1	3	5	8	1	3	5	8
RMSE	#1	1	1	1	1	1	1	1	1	1	1	1	1	1	2	1	1
	#2	2	3	3		4	4	3	3	4	4	3	3	4	1	3	
	#3	4				4	4			4	4			4	3		
TPR	#1	3	3	3	1	3	1	1	1	3	1	1	1	3	3	3	1
	#2	2	1	1	2	1	3	3	3	1	3	3	3	2	1	1	2
	#3	4	2	2	2	2	2	2	2	2	2	2	2	1	2	2	2
FP	#1	4	2	2	2	4	4	2	2	4	4	2	2	4	4	2	2
	#2	3	1	1	1	3	3	1	1	3	3	1	1	3	3	1	1
	#3	5	2	2	1	5	5	2	2	5	5	2	2	5	2	2	1

	Complexity	Yes															
	Blur	Low															
	Speed	0.1				0.5				1				2			
	SNR	1	3	5	8	1	3	5	8	1	3	5	8	1	3	5	8
RMSE	#1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
	#2	4	4	2	3	4	4	4	4	4	4	4	4	4	4	2	
	#3	4	4	3		4	4	4	4	4	4	4	4	4	4	3	
TPR	#1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
	#2	2	2	3	3	3	2	2	2	3	2	2	2	2	2	2	2
	#3	3	3	2	2	2	4	4	4	2	4	4	4	3	3	3	2
FP	#1	1	4	2	2	1	1	4	3	1	4	3	3	4	4	4	2
	#2	3	3	1	1	3	3	3	2	3	3	2	2	3	3	3	1
	#3	5	5	2	2	5	5	5	5	5	5	5	5	5	5	2	1

	Complexity	Yes															
	Blur	None															
	Speed	0.1				0.5				1				2			
	SNR	1	3	5	8	1	3	5	8	1	3	5	8	1	3	5	8
RMSE	#1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
	#2	4	4	4	4	2	2	2	2	2	2	2	4	4	4	4	2
	#3	4	4	4	4	4	3	4	4	3	4	3	4	4	4	4	4
TPR	#1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
	#2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2
	#3	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4
FP	#1	1	1	1	1	1	1	3	3	1	1	3	3	1	4	3	3
	#2	3	3	3	3	3	3	2	2	3	3	2	2	3	3	2	2
	#3	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5

Figure 79. The top three best performing methods for each combination of properties and Linear motion. a) 45 µm particles moving in linear motion at a low density with no particle complexity. b) 45 µm particles moving in linear motion at a low density with particle complexity. The methods are coded as follows: 1:SURF, 2: Canny, 3:Hough, 4:Otsu, 5:Threshold.

The results are similar for 45 µm particles undergoing Brownian motion at low density, shown in the continued summary of performance in Figure 80... Again, SURF, Threshold and Otsu ranked highest for the RMSE metric. For high levels of blur, Canny outperformed the other threshold methods at higher velocities. Sensitivity had three distinct winners: SURF, Canny and Otsu. However, for higher levels of blur Hough outperformed Otsu. The fewest false positives were reported by threshold methods. Taking all three metrics into consideration, SURF is by far the most robust method 45 µm particles undergoing Brownian motion at low density, if false positives are accepted.



a

Brownian 45 µm low density

Complexity		None															
Blur		High															
Speed		0.1				0.5				1				2			
SNR		1	3	5	8	1	3	5	8	1	3	5	8	1	3	5	8
RMSE	#1	4	1	1	1	2	2	1	1	1	1	1	1	1	1	1	1
	#2	4	3	3	3	4	1	3	3	4	2	2	2	4	2	2	2
	#3	1				4	3			4	4	4	4	4	4	3	4
TPR	#1	3	3	1	1	3	3	1	1	3	1	1	1	3	1	1	1
	#2	2	1	3	3	2	1	3	3	1	3	3	3	1	3	3	3
	#3	1	2	2	2	1	2	2	2	2	2	2	2	2	2	2	2
FP	#1	4	2	2	2	4	4	2	2	4	4	4	4	4	4	4	4
	#2	3	1	1	1	3	1	1	1	3	3	3	3	3	3	3	3
	#3	2	2	2	2	5	2	2	2	5	5	2	5	5	5	5	5

Complexity		None															
Blur		Low															
Speed		0.1				0.5				1				2			
SNR		1	3	5	8	1	3	5	8	1	3	5	8	1	3	5	8
RMSE	#1	1	1	1	1	1	1	1	1	1	4	4	1	4	4	4	1
	#2	4	4	2	2	4	4	2	2	4	4	4	4	4	4	4	4
	#3	4	4	4	4	4	4	4	4	4	1	1	4	1	1	1	4
TPR	#1	1	1	1	1	2	1	1	1	4	1	1	1	4	1	1	1
	#2	2	2	2	2	1	2	2	2	4	4	4	4	4	4	4	4
	#3	4	4	3	3	4	4	3	4	2	4	4	4	2	4	4	4
FP	#1	1	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4
	#2	3	3	1	1	3	3	3	3	3	3	3	3	3	3	3	3
	#3	5	5	2	2	5	5	2	2	5	5	5	5	5	5	5	5

Complexity		None															
Blur		None															
Speed		0.1				0.5				1				2			
SNR		1	3	5	8	1	3	5	8	1	3	5	8	1	3	5	8
RMSE	#1	1	1	1	4	1	1	1	4	1	1	1	1	1	1	1	1
	#2	4	4	4	4	4	5	5	4	4	4	4	4	4	4	4	4
	#3	4	4	4	1	4	4	4	1	4	4	4	4	4	4	4	4
TPR	#1	2	1	1	1	5	1	1	1	1	1	1	1	1	1	1	1
	#2	1	2	2	4	2	5	5	4	2	2	2	2	2	2	2	2
	#3	4	4	4	4	1	2	4	4	4	4	4	4	4	4	4	4
FP	#1	1	1	1	4	5	5	4	4	4	4	4	4	4	4	4	2
	#2	3	3	3	3	3	3	3	2	3	3	2	2	3	3	2	2
	#3	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5

b

Brownian 45 µm low density

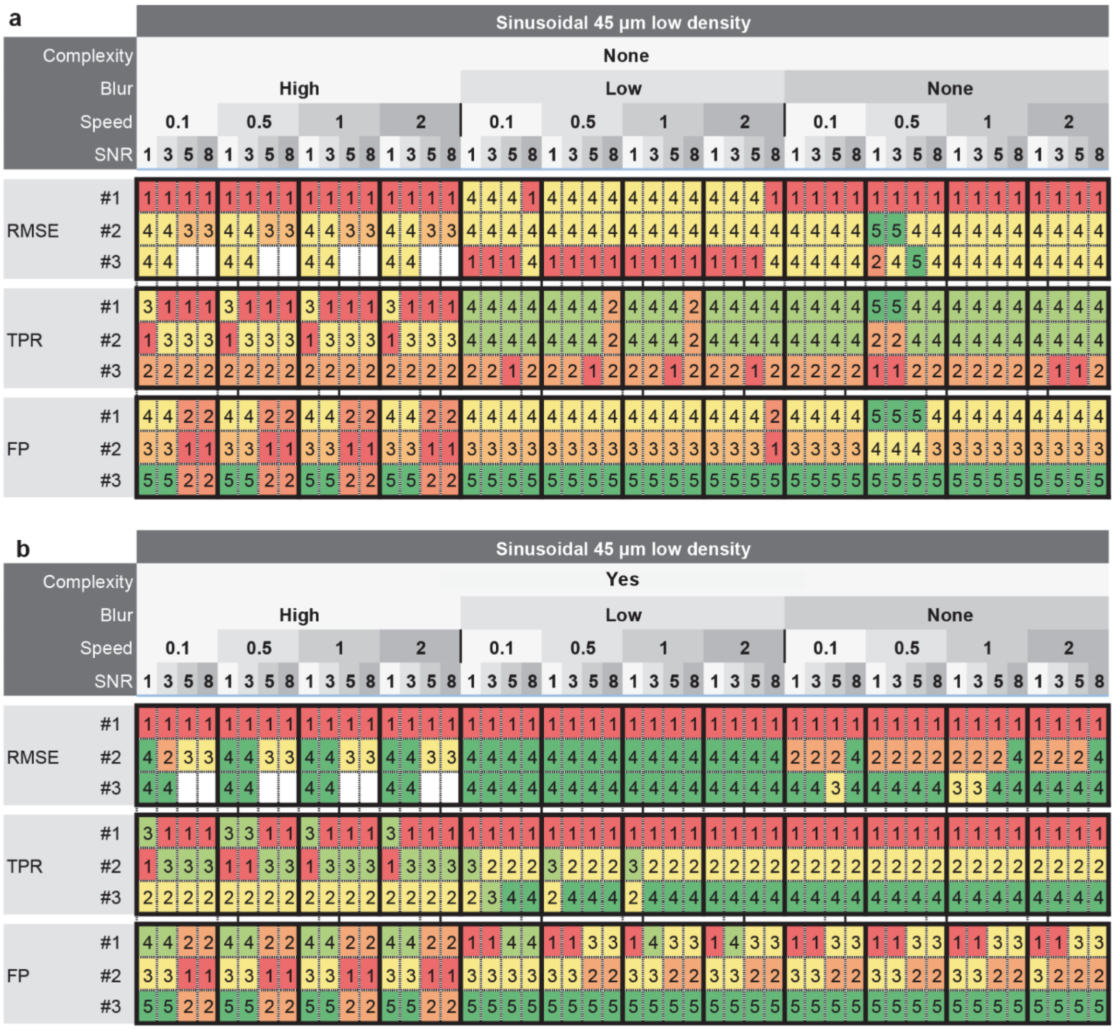
Complexity		Yes															
Blur		High															
Speed		0.1				0.5				1				2			
SNR		1	3	5	8	1	3	5	8	1	3	5	8	1	3	5	8
RMSE	#1	1	1	1	1	2	1	1	1	1	1	1	1	1	1	1	1
	#2	2	3	3	3	4	3	3	3	4	2	3	2	4	2	2	2
	#3	3				4				4	4	2	4	4	4	3	4
TPR	#1	3	3	1	1	3	3	1	1	3	3	1	1	3	1	1	1
	#2	1	1	3	3	2	1	3	3	1	1	3	3	1	3	3	3
	#3	2	2	2	2	4	2	2	2	2	2	2	2	2	2	2	2
FP	#1	4	2	2	2	4	2	2	2	4	4	4	4	4	4	4	4
	#2	3	1	1	1	3	1	1	1	3	3	3	3	3	3	3	3
	#3	2	2	2	2	5	2	2	2	5	5	2	5	5	5	5	5

Complexity		Yes															
Blur		Low															
Speed		0.1				0.5				1				2			
SNR		1	3	5	8	1	3	5	8	1	3	5	8	1	3	5	8
RMSE	#1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
	#2	4	4	2	2	4	4	2	2	4	4	4	4	4	4	4	4
	#3	4	4	4	3	4	4	4	4	4	4	4	4	4	4	4	4
TPR	#1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
	#2	2	2	3	2	2	2	3	2	3	3	2	2	2	2	2	2
	#3	3	3	2	2	4	3	2	4	2	2	4	4	3	3	4	4
FP	#1	1	4	2	2	1	4	4	4	1	4	4	4	1	4	4	4
	#2	3	3	1	1	3	3	3	3	3	3	3	3	3	3	3	3
	#3	5	5	2	2	5	5	2	2	5	5	5	5	5	5	5	5

Complexity		Yes															
Blur		None															
Speed		0.1				0.5				1				2			
SNR		1	3	5	8	1	3	5	8	1	3	5	8	1	3	5	8
RMSE	#1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
	#2	4	4	4	4	4	4	4	4	2	2	2	2	2	2	2	2
	#3	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4
TPR	#1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
	#2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2
	#3	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4
FP	#1	1	1	1	4	1	1	1	3	1	1	3	3	1	1	3	3
	#2	3	3	3	3	3	3	3	2	3	3	2	2	3	3	2	2
	#3	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5

Figure 80. The top three best performing methods for each combination of properties and Brownian motion. a) 45 µm particles moving in Brownian motion at a low density with no particle complexity. b) 45 µm particles moving in Brownian motion at a low density with particle complexity. The methods are coded as follows: 1:SURF, 2: Canny, 3:Hough, 4:Otsu, 5:Threshold.

Finally, for 45 µm particles in a sinusoidal motion at low density in Figure 20 the results mainly differ for low blur levels. For the RMSE metric SURF, Threshold and Otsu ranked highest again but Otsu surpassed SURF at low blur levels. Sensitivity had three distinct winners SURF, Canny and Otsu. However, for lower levels of particle complexity Otsu outperformed SURF. The fewest false positives at low levels of complexity were for Otsu but at higher levels the trend is less clear. It is clear that Threshold while ranked third was robustly third across methods. For this scenario either SURF or Otsu performed best depending on the metric of interest.



a		Sinusoidal 45 μm low density																																													
Complexity		None																																													
Blur		High																Low																None															
Speed		0.1				0.5				1				2				0.1				0.5				1				2				0.1				0.5				1				2			
SNR		1	3	5	8	1	3	5	8	1	3	5	8	1	3	5	8	1	3	5	8	1	3	5	8	1	3	5	8	1	3	5	8	1	3	5	8	1	3	5	8	1	3	5	8	1	3	5	8
RMSE	#1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	4	4	4	1	4	4	4	4	4	4	4	4	4	4	4	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
	#2	4	4	3	3	4	4	3	3	4	4	3	3	4	4	3	3	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	5	5	4	4	4	4	4	4	4	4	4	4
	#3	4	4			4	4			4	4			4	4			1	1	1	4	1	1	1	1	1	1	1	1	1	1	1	4	4	4	4	4	2	4	5	4	4	4	4	4	4	4	4	4
TPR	#1	3	1	1	1	3	1	1	1	3	1	1	1	3	1	1	1	4	4	4	4	4	4	4	2	4	4	4	2	4	4	4	4	4	4	4	4	5	5	4	4	4	4	4	4	4	4	4	4
	#2	1	3	3	3	1	3	3	3	1	3	3	3	1	3	3	3	4	4	4	4	4	4	4	2	4	4	4	2	4	4	4	4	4	4	4	4	2	2	4	4	4	4	4	4	4	4	4	4
	#3	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	1	2	2	2	1	2	2	2	1	2	2	2	1	2	2	2	2	2	1	1	2	2	2	2	2	2	2	1	1	2
FP	#1	4	4	2	2	4	4	2	2	4	4	2	2	4	4	2	2	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	2	4	4	4	4	5	5	5	4	4	4	4	4	4	4	4	4
	#2	3	3	1	1	3	3	1	1	3	3	1	1	3	3	1	1	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	1	3	3	3	3	4	4	4	3	3	3	3	3	3	3	3	3
	#3	5	5	2	2	5	5	2	2	5	5	2	2	5	5	2	2	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5

b		Sinusoidal 45 μm low density																																													
Complexity		Yes																																													
Blur		High																Low																None															
Speed		0.1				0.5				1				2				0.1				0.5				1				2				0.1				0.5				1				2			
SNR		1	3	5	8	1	3	5	8	1	3	5	8	1	3	5	8	1	3	5	8	1	3	5	8	1	3	5	8	1	3	5	8	1	3	5	8	1	3	5	8	1	3	5	8	1	3	5	8
RMSE	#1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
	#2	4	2	3	3	4	4	3	3	4	4	3	3	4	4	3	3	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	2	2	2	4	2	2	2	2	2	2	2	4	2	2	2	4
	#3	4	4			4	4			4	4			4	4			4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	3	4	4	4	4	4	3	3	4	4	4	4	4	4
TPR	#1	3	1	1	1	3	3	1	1	3	1	1	1	3	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
	#2	1	3	3	3	1	1	3	3	1	3	3	3	1	3	3	3	3	2	2	2	3	2	2	2	3	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2
	#3	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	3	4	4	2	4	4	4	2	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4
FP	#1	4	4	2	2	4	4	2	2	4	4	2	2	4	4	2	2	1	1	4	4	1	1	3	3	1	4	3	3	1	4	3	3	1	1	3	3	1	1	3	3	1	1	3	3	1	1	3	3
	#2	3	3	1	1	3	3	1	1	3	3	1	1	3	3	1	1	3	3	3	3	3	3	2	2	3	3	2	2	3	3	2	2	3	3	2	2	3	3	2	2	3	3	2	2	3	2	2	2
	#3	5	5	2	2	5	5	2	2	5	5	2	2	5	5	2	2	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5

Figure 81. The top three best performing methods for each combination of properties and Sinusoidal motion. a) 45 μm particles moving in Sinusoidal motion at a low density with no particle complexity. b) 45 μm particles moving in Sinusoidal motion at a low density with particle complexity. The methods are coded as follows: 1:SURF, 2: Canny, 3:Hough, 4:Otsu, 5:Threshold.

Performance of surveyed detection methods under varying imaging conditions

Having now established a sense of the overall ranking between methods, we want to also understand how each method changed in performance as the primary factors changed. Specifically, to study the effect of SNR, density, particle size, velocity, blur and particle complexity we compared how the sensitivity, RMSE and number of false positives changed as each parameter varied. The full factorial analysis allowed us to vary the parameters independently of other effects. These factors are shown for each dynamic including linear, Brownian and sinusoidal.

Effect of SNR

We measured the RMSE, sensitivity and false positives for each method across all of the SNR values and dynamics. To maintain clarity in the visualization of results encompassing a large number of situations while maintaining standardization across them, the primary effects of the factors investigated are presented for 45 µm particles with a low density, no complexity or blur applied. This can be seen in Figure 82. We first observed that the RMSE was not affected by the SNR values present in our study. This is true for all three dynamics. Hough detection was the only method to have an average RMSE greater than 4 pixels. The best performing method was SURF with a RMSE of 1.72 pixels. However, threshold based methods were all within 2 pixels. Brownian motion showed slightly worse performance at an average of 3.4 pixels RMSE, likely due to lag in the Kalman filter correction.

Sensitivity was very high for all methods except for Hough detection. Otsu was susceptible to SNR in respects to sensitivity, dropping from 0.96 to 0.22 across the SNR range of 8 to 1. The sensitivity of SURF decreased only from 0.99 to 0.95 across SNR of 3 to 1. CANNY and Threshold were invariant and stable across all values but did perform slightly worse under Brownian motion.

While false positives were consistently zero for the Threshold method, they ranged from 85-266 for CAANY within linear motion. All three performed worse for sinusoid and Brownian, respectively. Of particular interest is that Otsu performed much worse as SNR increased, with 208,063 false positives at its minimum, FP causing a dip in sensitivity. Hough also performed worse with decreasing SNR.

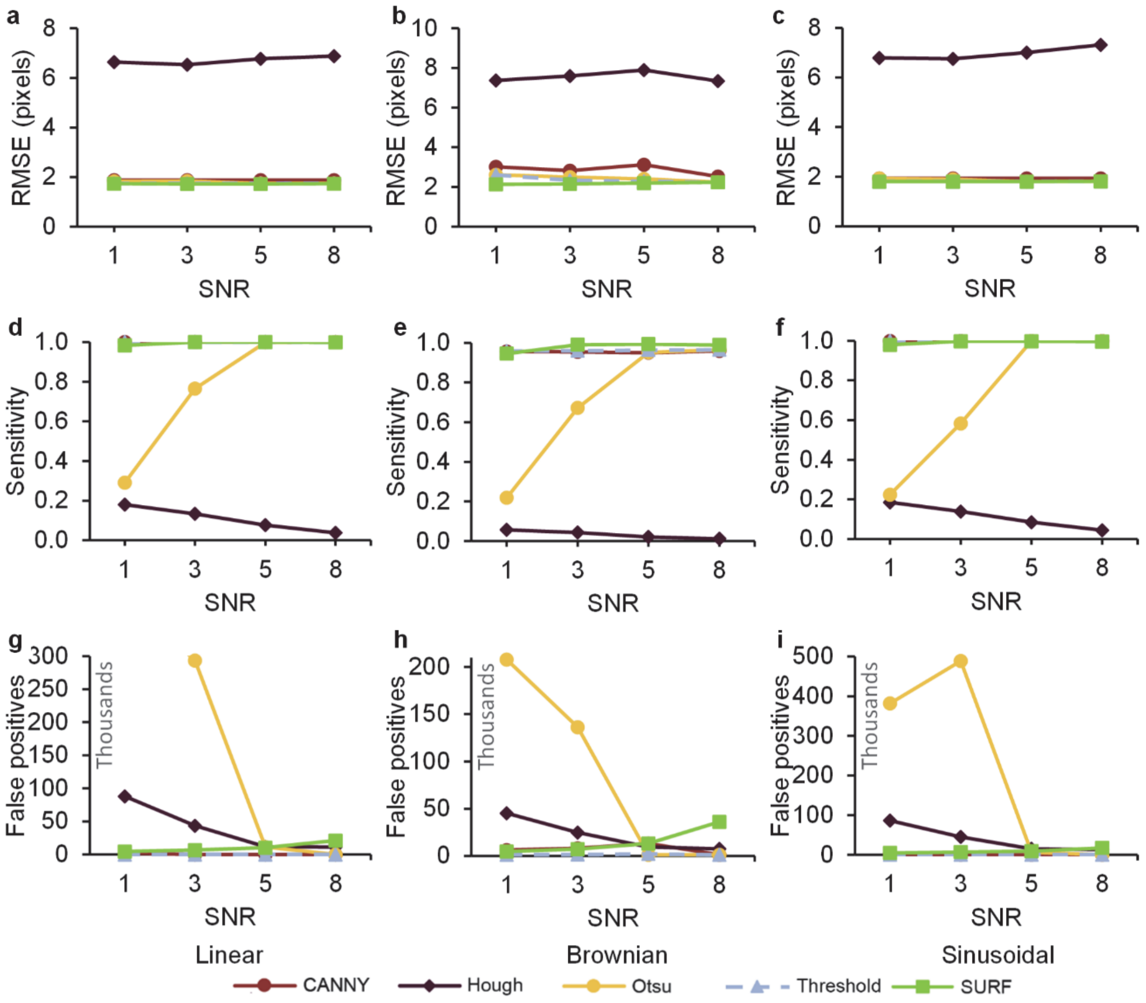


Figure 82. RMSE, Sensitivity and False positives measured for varying Signal-to-Noise Ratios (SNR) and dynamics.

Effect of density

The effect of particle density is presented for Brownian motion. RMSE performed worse with increasing density. Examination of the videos makes the cause clear; overlap increases as density increases and for uniform particles this obscures detection of the centroid because two regions become one large particle with a centroid shared between them. Also, we observed that sensitivity decreased as density increased. The rate of decrease was faster between 500 and 1000 particles. This suggests that some number of particles between these two values would be the cross-over value for balancing throughput and sensitivity. One could interpolate or test a more granular range of densities to find that exact value. False positives also increased significantly between medium and high-density runs.

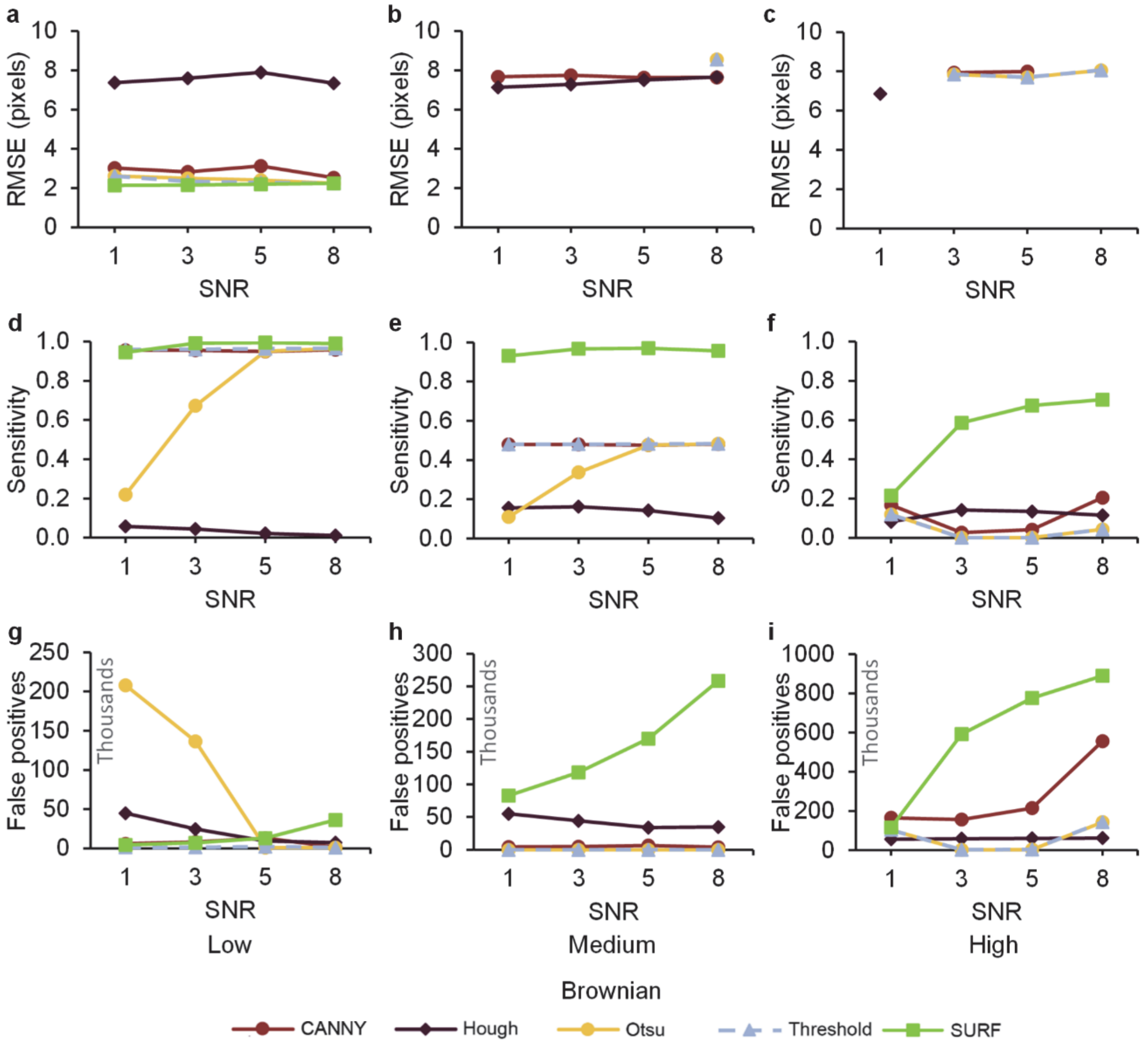


Figure 83. Influence of particle dynamics on primary output metrics (RMSE, Sensitivity, False positives) for varying SNR in simulated videos.

Effect of particle size

Primary particle size effects are shown for all four particles for each dynamic at a velocity of 0.5 mm/s and SNR of 8. Figure 84 (a) shows that no method was able to sufficiently detect 5 µm particles. During investigation it was established that these were too small to be processed by the contour-finding function we utilized. The remaining data shows that the RMSE improves as the particle size increases.

Sensitivity showed one clear and important trend; that each algorithm had a value for which the sensitivity would suddenly drop off. Because the number of false positives is not increasing with decreasing particle size, the number of true positives or correctly detected particles is confidently decreasing. The particle size thresholds sizes at which sensitivity drops are 45 µm, 10 µm, 5 µm, 5 µm and 5 µm for Hough, SURF, Canny, Otsu and Threshold respectively. However, for Brownian motion Canny, Otsu and Threshold reduce in sensitivity at 10 µm rather than at 5 µm. False positives are less common for smaller particles and increase as particle size increases for SURF and Hough. After further investigation, this is because edges of larger particles are more likely to be near one another when the particle number density is held constant. When two particles are near each other SURF and Hough often detect the region in which the two curves approach each other as one or more false positives. This happens because the shape the two particles make near each other approximately resembles a curve. This caused the total number of false positives and even the ratio of false positives to true positives to inflate dramatically if many of these situations are present and the scenario holds throughout the lifetime of the particle in the field of view.

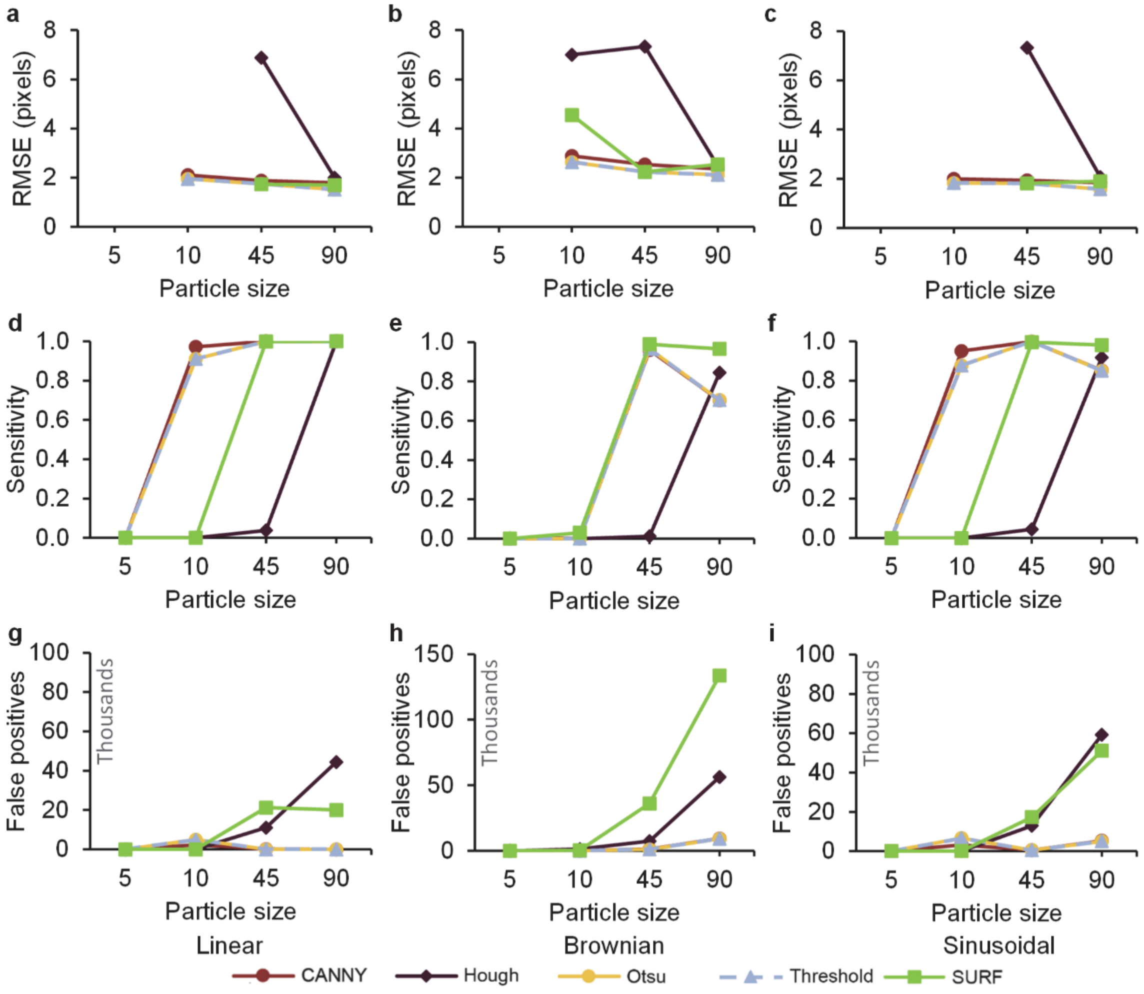


Figure 84. Influence of particle dynamics on primary output metrics (RMSE, Sensitivity, False positives) for varying particle size in simulated videos.

Effect of velocity

The RMSE across four velocities showed no trends across all dynamics and methods which was expected. Other trends presented were similar to previous results as shown in Figure 85.

Otsu, Threshold and Canny showed a reduction in sensitivity for Brownian motion as velocity increased with a distinct shift from 0.5 to 1 mm/s.

False positives increased with increasing velocity for all methods within Brownian motion. Although, it stayed constant for the other two motion types. A comparison of the video prepared for 1 mm/s and 2 mm/s showed a reduced concentration of particles near each other for 2 mm/s due to random seeding. This accounts for the unusual reduction at 2 mm/s seen in SURF, Hough. Nevertheless, we still expect false positives to increase with increasing velocity.

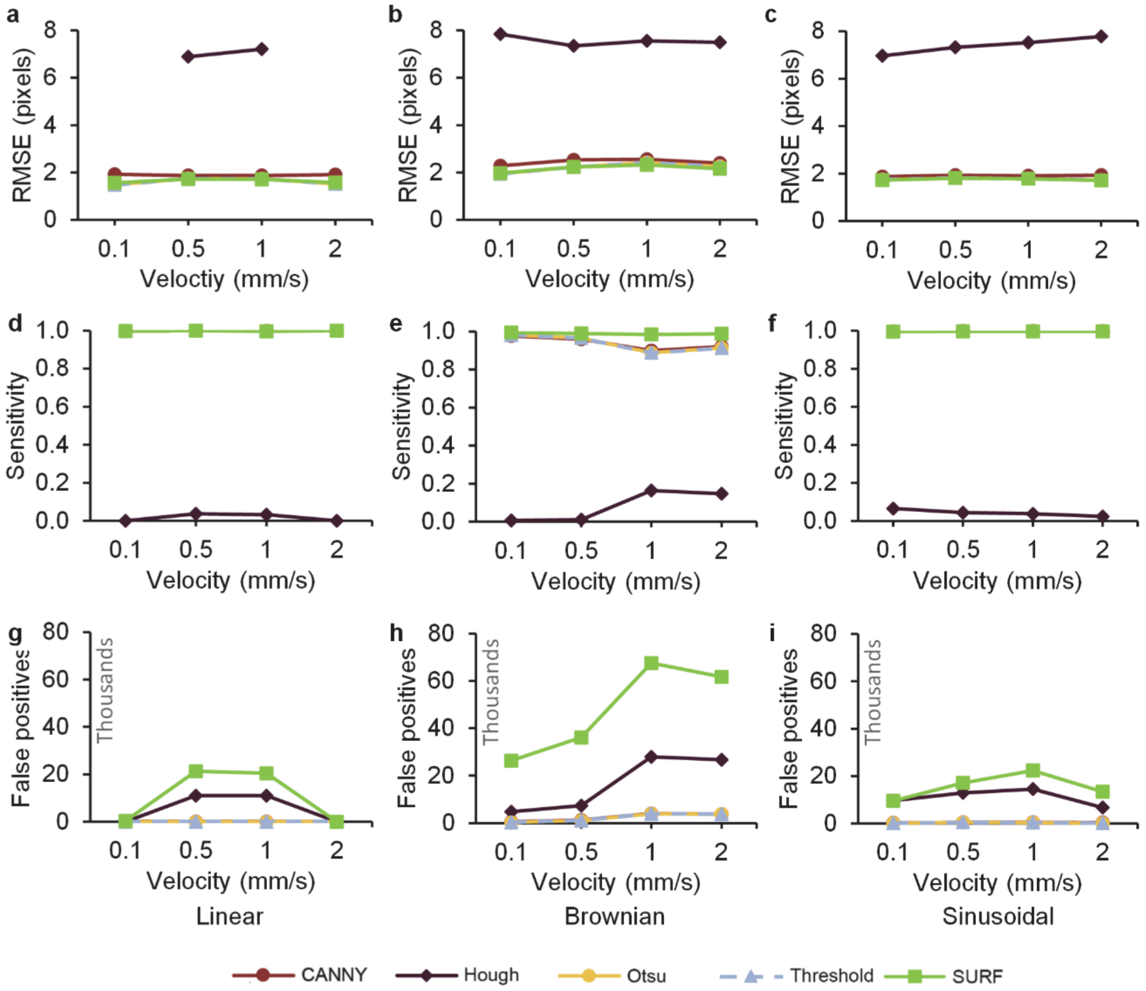


Figure 85. Influence of particle dynamics on primary output metrics (RMSE, Sensitivity, False positives) for varying SNR in simulated velocity.

Effect of blur

For linear dynamics, Hough detection improved its RMSE with increasing blur as shown in Figure 86. For Brownian motion however, increasing blur caused Otsu, Threshold and Canny to more than double in RMSE for low levels of blur and to stop detecting altogether at high levels. Sinusoidal motion saw the same three methods suddenly cutoff at high levels of blur. This seems to occur for these because the blur affects the entire particle, decreasing its signal level below the threshold value. This is easily remedied by increasing the threshold, but can also be related to the PSF or focus of the image.

The sensitivity of each method follow similar patterns with three of the five methods no longer detecting any particles at high levels of blur. Considering there is not an increase in FP the detection scheme is failing.

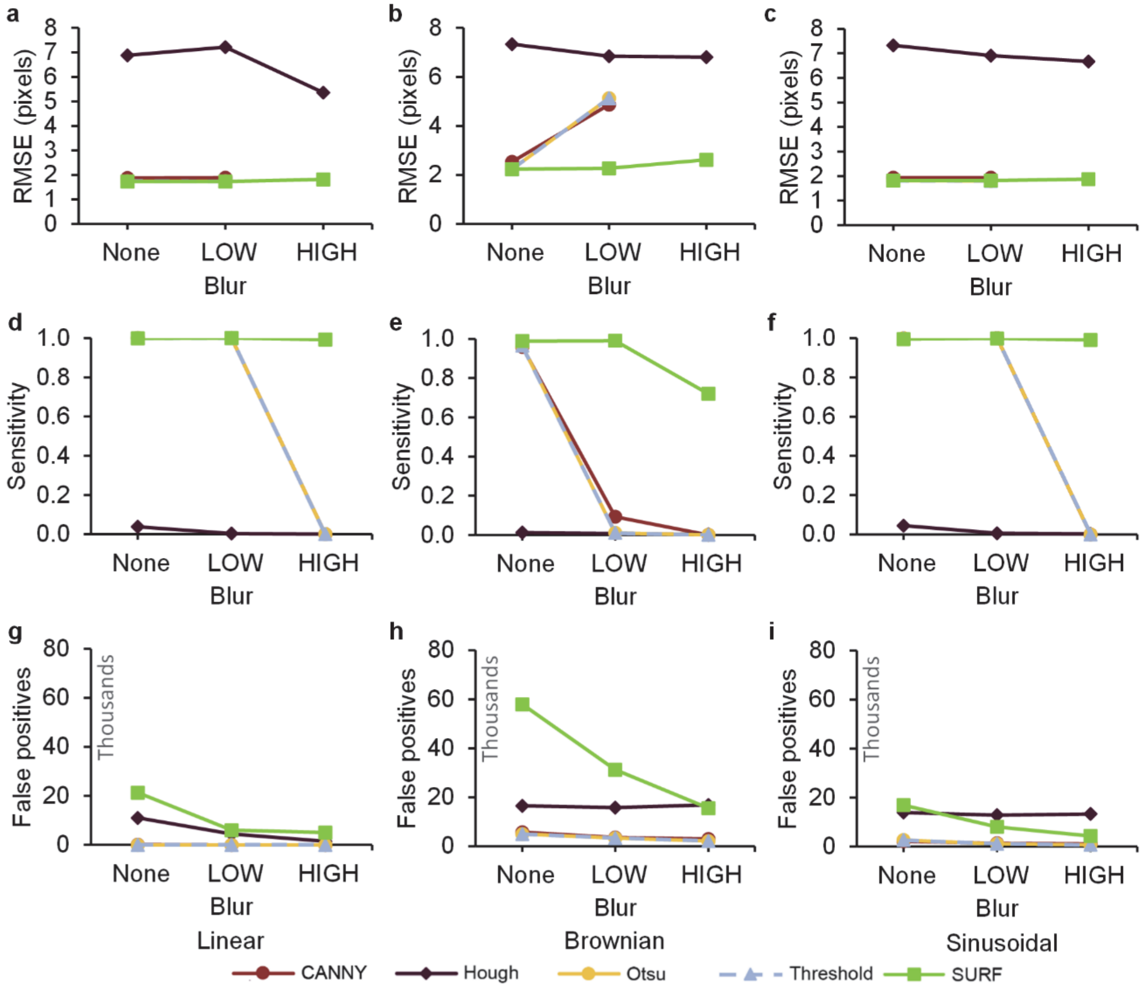


Figure 86. Influence of particle dynamics on primary output metrics (RMSE, Sensitivity, False positives) for varying blur in simulated videos.

Effect of complexity

Complexity is a categorical variable and so the trends appear overstated as shown in Figure 87. Nevertheless, RMSE increases for all methods except Hough, which stays constant. Sensitivity decreases with increasing complexity and false positives increase for threshold methods. SURF and Hough actually see a decrease in false positives. These trends unusually hold true for all dynamic styles, making it the most consistent alterations across methods within our study, and making it a candidate for further investigation in a study if further tuning of methods were not possible.

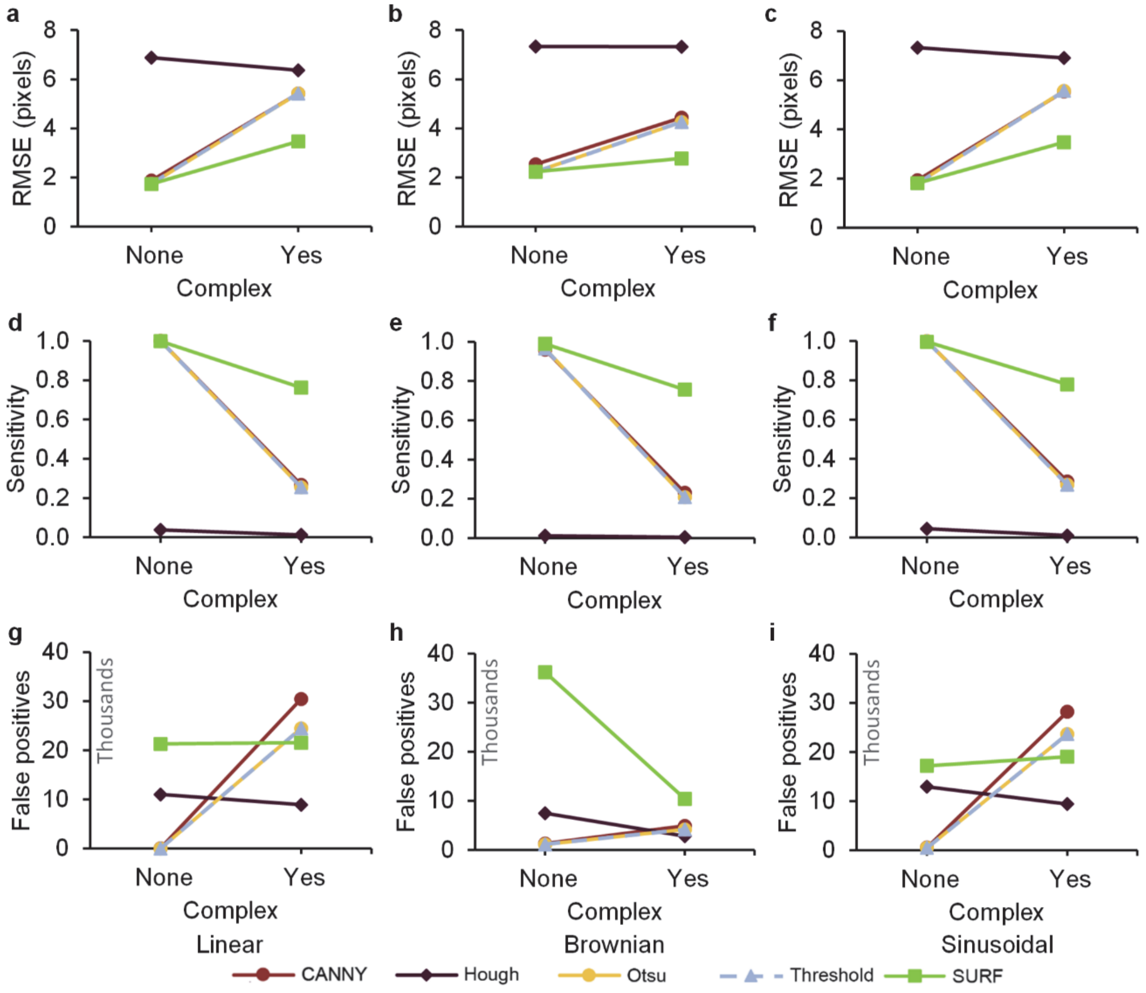


Figure 87. Influence of particle dynamics on primary output metrics (RMSE, Sensitivity, False positives) for varying particle complexity in simulated videos.

Performance of Nearest Neighbor tracking under varying conditions

Next we discuss quantitative analysis of the nearest neighbor tracking algorithm when using the five chosen detection methods. Changes in the RMSE due to the detection method and its parameter settings will have an impact on the metrics of tracking. Moreover, if a detection algorithm skips detecting a particle for several frames, the tracking algorithm is also penalized for being unable to link the particle. We assume that the particles have been detected and now only require linking so density and velocity are primarily discussed. Other factors, such as SNR could have some effect in which a particle is divided into two and the linker switches between tracking one half of the particle or the other. Because it is difficult to distinguish between false linking we chose not to report the data.

Effect of density on tracking

The effect of density on tracking performance was measured for Brownian motion, because we expected it to be the most susceptible. Sensitivity of tracking decreased with increased density for all methods, as expected. However, Hough performed better for medium density. The cause of sensitivity reducing with increasing density was due to an increased number of false positives. The Kalman filter, which predicts the motion of the particle for all methods, will predict that a particle moving in one direction will stay moving in that direction. In the Brownian case there is no reason to expect this. Therefore, at higher densities, if a particle moves into a location randomly vacated by a particle being tracked, it may ultimately be tracked and labeled as the original rather than its own as an unintended consequence. This issue presented here may be overcome by more advanced methods that memorize a property of the particle other than location and orientation. However, that is beyond the present scope.

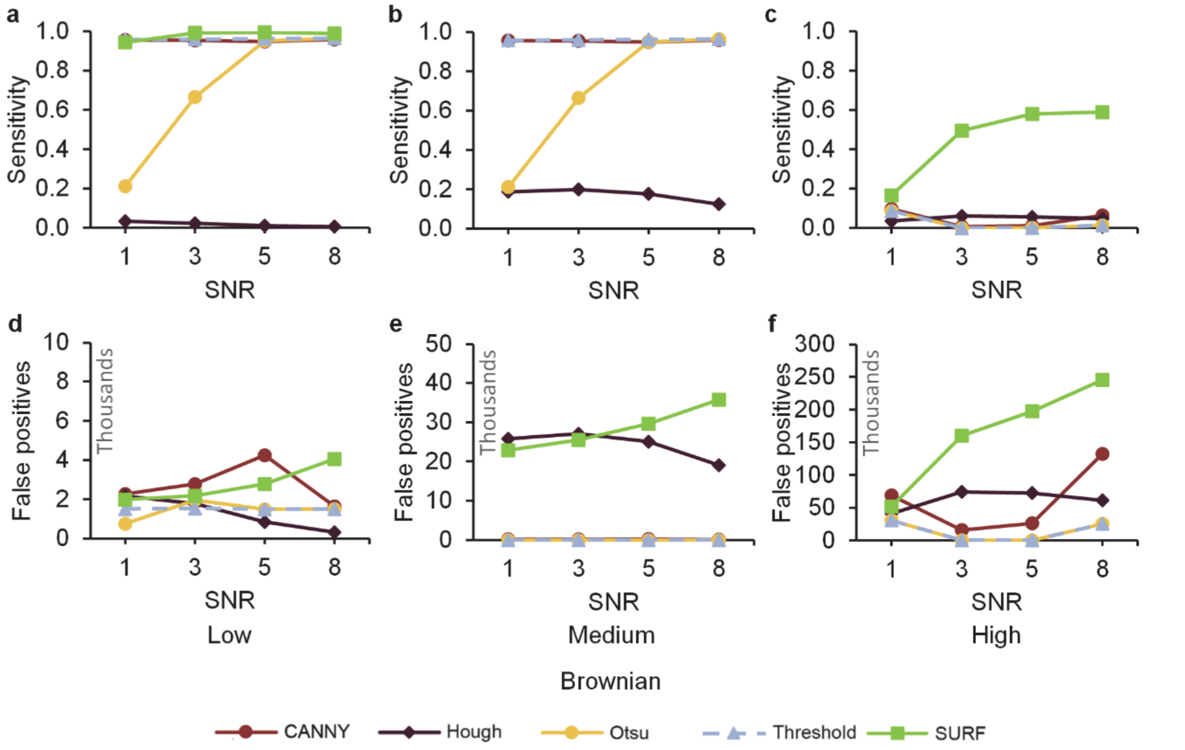


Figure 88. Sensitivity and false positives for the nearest neighbor tracking method while varying density and dynamics.

Effect of velocity on tracking

The effect of particle velocity on tracking is critically important as the distance a particle travels between measurements could lead to misidentification. Figure 89 shows the tracking sensitivity does not change by varying velocity.

Invariant tracking sensitivity across this velocity range is expected for linear and sinusoidal motion as the predictive algorithm and linking algorithm can work together to reduce mistakes. However, Brownian motion causes a reduction in sensitivity as velocity increases. This is likely due to poor modeling of the Brownian motion in the tracker leading to increased false positives.

Among all five methods, Hough showed the worse sensitivity for linking. In fact, in some cases it did not properly link any particles. This failure is likely due to the fact that the detected circles in Hough were not well centered. An examination of the particles paths would confirm.

SURF also showed an increased number of false positives with increasing velocity but it didn’t affect sensitivity because more particles were simply measured over the same video length.

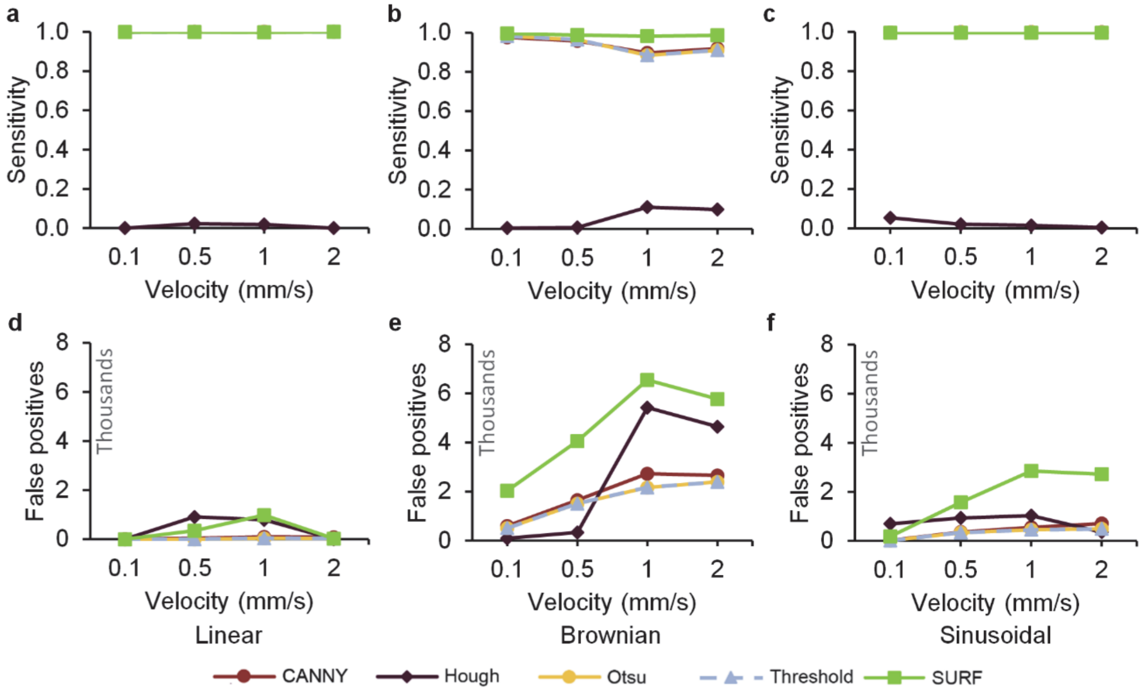


Figure 89. Sensitivity and false positives for the nearest neighbor tracking method while varying velocity and dynamics.

Is real-time realistic?

We also measured the computational cost of each method for both detection and tracking for all videos with 100 particles. All videos were analyzed on an Intel i7 2.4 GHz with 8GB of RAM and Windows 8.1 operating system. We directly compared the relative computation time required by each algorithm as shown in Figure 90. The amount of time the program spent performing each detection and tracking algorithm was recorded during every frame and then averaged across all of the frames within the sample. This data was converted into a box and whisker plot and ranked. The ranking shows that SURF requires the highest average computation time (0.174 s/frame) followed by Hough detection (0.038 s/frame), Canny (0.009 s/frame), Thresholding (6.47 e^{-4} s/frame) and Otsu Thresholding (6.14 e^{-4} s/frame). SURF's iterative voting procedure is simply a larger number of computations per pixel than threshold. The implication is that by designing your illumination to favor Thresholding, you can reduce your computational cost by two orders of magnitude. On the other hand, Kalman filtering and nearest neighbor tracking algorithms were on average in the same order of magnitude at 0.06 s/frame and 0.03 s/frame respectively. Max-min values are shown in the whisker region of the plot.

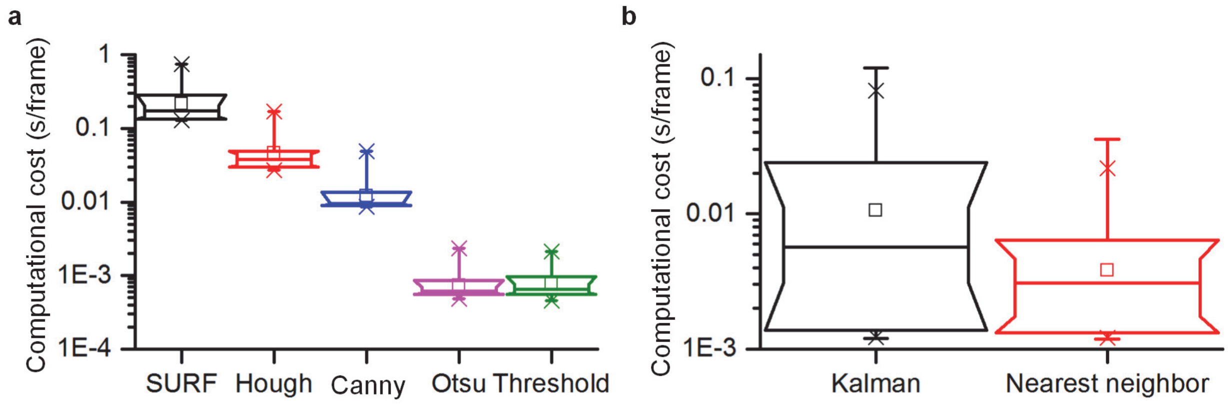


Figure 90. Computational cost of detection and tracking algorithms. a) Computational cost of detection algorithms across all simulated videos. b) Computational cost of tracking algorithms across all simulated videos. Whiskers show max-min values.

Using Threshold detection with Nearest Neighbor tracking we performed a second set of experiments to examine the change in computational cost for each function as the number of particles increase to determine how effective this method would be in real time. Figure 91 shows these results for the total time and the five functions that contribute and are aggregated to produce that total time value. Figure 91 shows that the relationship between the total time to process one frame is exponential as the particles are increased an order of magnitude from 10 to 10,000. The *threshold function* operates on the whole image not on the number of particles and therefore isn't affected by a change in particles as shown in Figure 91b.

The *finding-contours function* does increase as more contours are in the image but even at 10,000 particles it contributes only 0.02 s to the total time of 10 s. *Find image moments* function also increase exponentially but at a slower pace from 1.4x10^{-3} to 1.4. The majority of the computation time is actually spent *matching and linking* particles that have been previously found to particles detected in the current image. Finally, *adding new particles* if they weren't found in the tracking list contributes less than 1x10^{-5} s for 10, 100 and 1,000 particle densities but increases 0.5 s for 10,000 particles.

This data conveys that the brute force search method to match particles in the list is the largest contributor to limiting the implementation high throughput, high speed particle tracking methods. Future work to replace this method is suggested by implementing well known sorting methods to decrease matching time and limiting the search list to particles within a region of interest in the image. This has been tested for dynamic lithography and proven to be a useful method when the number of regions is small.

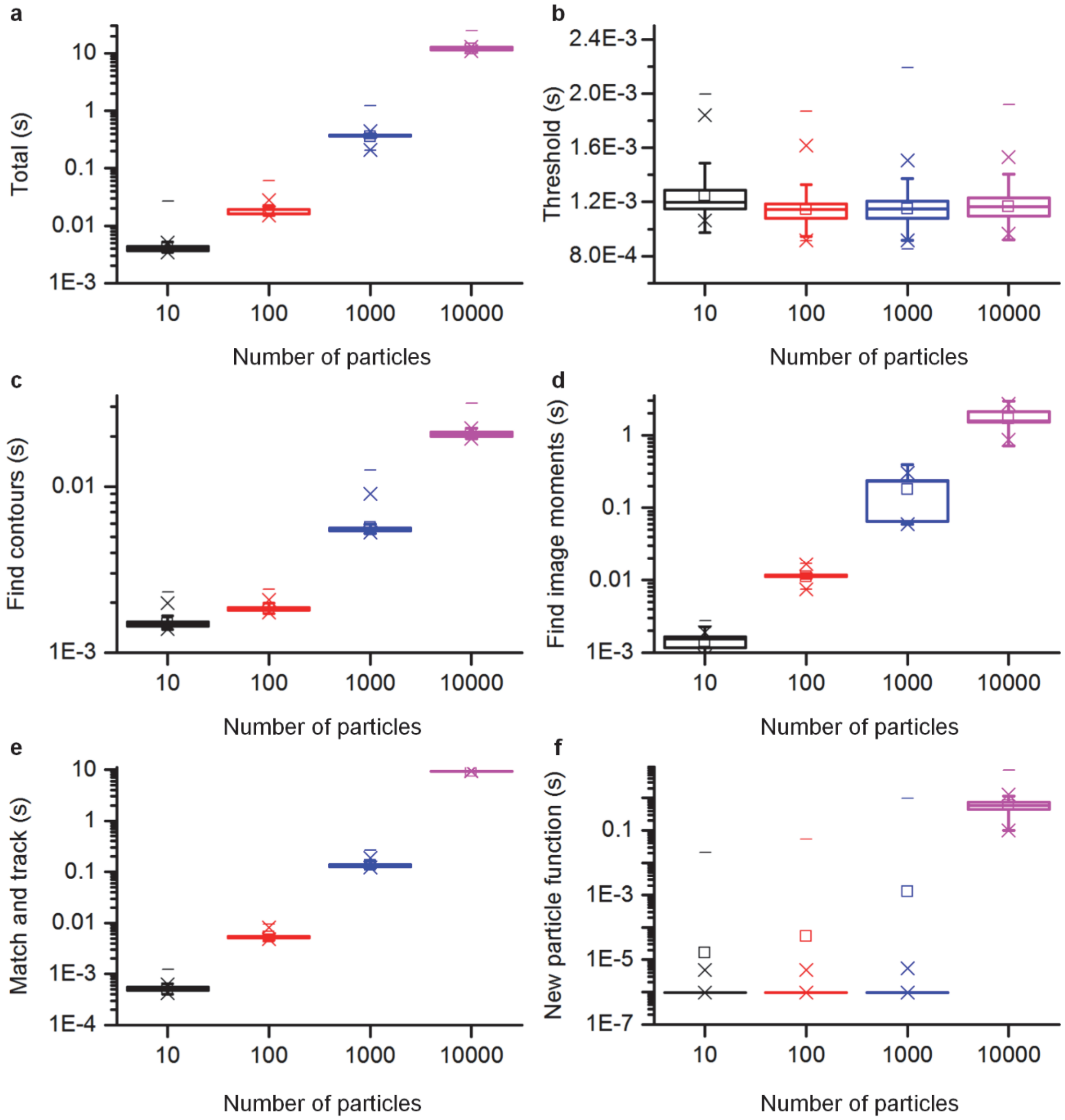


Figure 91. Box-whisker plots showing the computation time in seconds for various functions used in the detection and tracking algorithm. a) The total time for each frame. b) The computational cost of threshold. c) The computational cost of finding contours of objects in the image. d) The computational cost of finding the moments of the image. e) The computational cost of matching and updating tracking variables. f) The computational cost of creating new particle object if no match is found.

Discussion

As a survey of accessible traditional and sophisticated methods, this study could certainly include many other possibilities for examination such as multiple hypothesis testing. However, we feel that the methods presented represent a collection of both traditional and modern algorithms that form a basis of what is easily accessible. While the parameters used were tuned globally across all scenarios for each method it should be noted that post-tuning to a specific scenario region often produced better results. This is expected and should be used as justification to expand the study to then tune input parameters as needed.

Density appears to play the largest role in determining success of an algorithm in detecting objects, because particle density (i.e. number of particles per unit area) influences both detection and tracking. Particle overlap increases with density, making detection more difficult when objects are similar in size and/or shape. Second, it increases the likelihood that linking during the tracking process will mistake a particle near the correct one for the original. This is especially pronounced for Brownian dynamics where the likelihood is increased due to the random direction and magnitude of the motion of each individual particle since these algorithms take no information on motion trajectory.

Of course, each parameter in its extreme case had the potential to drastically impact performance. For example, high levels of blur quickly deteriorated performance for threshold-based methods by reducing the signal/intensity from small particles. The number of pixels that represent a particle directly impacts how detrimental blur and SNR changes are. In a scenario in which the system is out of focus equivalent to a Gaussian blur of 50 pixel radius, a 5 μm particle will no longer be visible while a 45 μm will. Based on our results, a reasonable conclusion is that if a particle is represented by fewer than 4 pixels on the image it will be difficult to accurately detect. A non-dimensional rule could also be forged stating that if the particle's representation in pixels approaches one pixel, then the edge detection algorithm used for overall particle detection will no longer function and other methods will be needed.

The relationship between particle size and pixel representation is a balance between image size, resolution, and performance. For example, the videos used in this study had a WXGA resolution, which was chosen as part of this balance. With some video formats approaching 4 MP, the performance of many algorithms would naturally be slower even though the number of pixels per object would be higher. But, because WXGA was chosen, with our optics, 1 pixel represented our 5 µm particles. If a larger format (4K) was chosen then 9 pixels would have represented the particle making it easier to detect and draw morphological data from. In real world systems this is also coupled to magnification and pixel pitch size, meaning if we increase the resolution at the same magnification then we require a smaller pixel pitch. It becomes easy to see that a balance between image size, resolution and performance needs to be struck.

Another consideration is if the simulations are sufficiently similar to real world scenarios. Figure 92 shows a comparison between 45 µm particles in three different scenarios: a white light bright field image, a dark field image, and a simulation. The dark field image and simulation are very similar, allowing one to be used as a proxy for the other.

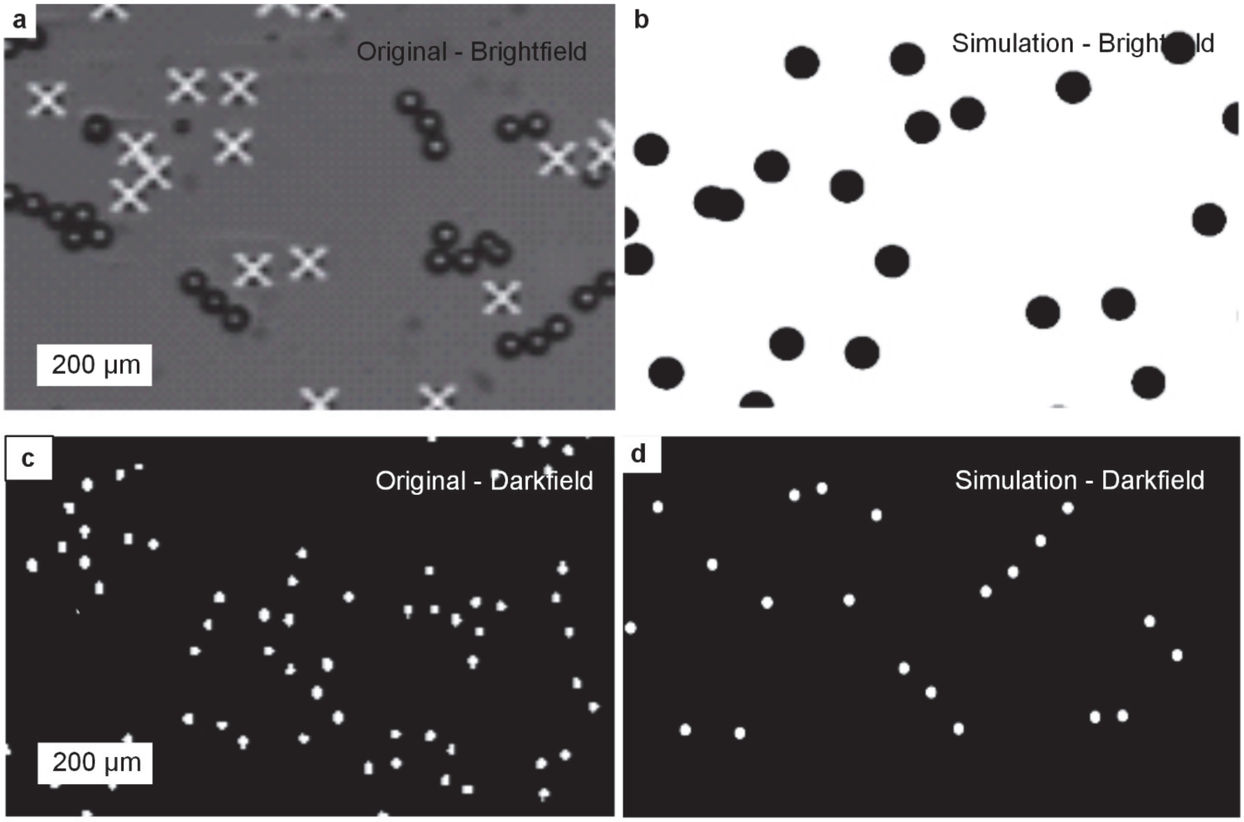


Figure 92. Comparison of brightfield and darkfield lighting to illustrate how simple simulations can be sufficient to describe real images.

One of the key assumptions regarding the velocity results is that the particles do not blur as they move faster, because our ground truth videos are built up frame-by-frame rather than broken down into frames form a live image. This assumption is less feasible in a real system because the shutter speed of the camera is not sufficient to capture a sharp image of particles moving quickly. For this reason the velocity results were expected but could be very misleading.

Therefore, caution should be used when setting up a real world system as to ensure the shutter speed is properly adjusted to account for faster moving particles. Without having the capabilities to receive ground truth data from a live video feed, it is difficult to empirically determine the relationship of shutter lag. But one could measure the induced motion blur due to a slow shutter and correlate the results reported for blurred particles.

Complexity of the particle is very difficult to control in biological systems. They can be represented by any combination of transparency, color gradients and granular appearance. Any combination of these alterations can easily divide a normally whole particle into two or three apparently different objects. Fluorescence is probably one of the best potential solutions because the detection point can be localized and uniform. Therefore, our recommendation is to consider fluorescence capabilities for any system that has the potential to be confounded by these complexities

Real-time performance is achievable yet should be pursued with some caution, with off-the-shelf workstations performing very well at lowered resolutions, but meaning that they are not yet sufficient to match the resolution of the most capable high resolution cameras (4K 3840x2160). Further, the number of particles that can be processed at full frame is system-dependent. Therefore we recommend looking carefully at the system's resolution and particle volume limitations before expecting real-time performance.

The detection and tracking software is available to the community either for further analysis of new and exciting algorithms against ground truths or as a means of implementing a real-time particle-tracking package into another research area altogether. In particular, the design is well suited to real-time systems such as dynamic lithography or optogenetics.

Conclusion

Based on the present study, we conclude that (when possible) controlling density, blur, and particle complexity are the best ways to maximize performance of detection and tracking methods. We acknowledge that for many applications this may not be possible. For example, in a microfluidic environment it is most readily possible to control the particle density and the illumination. A counter example is tissue culturing where the density is determined by the growth of the tissue and imaging with fluorescence is not suitable during the growth. However, counting cells in a tissue culture is not necessarily a real-time scenario and care can be taken to tune the algorithm parameters.

One takeaway from this study is that the shallow depth of field in microscopy images reduces the image structure to parameters found in Figure 77. This could also be why simple and easily implemented methods such as Thresholding remain interesting research areas because they are adaptable and fast under these conditions. However, continued research into new algorithms will likely yield more effective results when coupled with machine learning that can more robustly adapt to a variety of scenarios, no longer requiring the detailed categorization of images according to common metrics. Users of detection and tracking methods can apply this study to identify a starting point for choosing the correct method in their own application, which becomes more challenging as the number of options continues to grow.

Developers may find both the methods used here to evaluate the algorithms engaging for their own purposes. Moreover, they should take care to consider the selection of input parameters into the algorithm carefully as it has a significant impact on performance. In future work, it would be similarly useful to study the in-depth sensitivity and relationships between the method performance and the chosen parameters. Based on largely anecdotal observations throughout our study whilst tuning the parameters, we noted many instances where there was a large either linear or non-linear, sometimes even binary or threshold, effect.

We hope that other investigators expand on the experiments performed here through the addition of new methods of interest and videos of real world particle systems and their respective ground truths. It will also be useful to compare between CPU and GPU-based methods to increase the resolution at which real-time analysis can be performed.

CHAPTER VII. DYNAMIC PHOTOPRINTING OF CELLS AND MICROBEADS TOWARDS FLEXIBLE TISSUE MANUFACTURING

Abstract

One of the key applications enabled by dynamic lithography is photopatterning micro-scale objects, such as cells, into arbitrary 2D arrangements with single cell resolution (~10-20 µm or better). In this chapter, we use the dynamic lithography platform along with a new algorithm which observes cells flowing across the field of view and anchors them to a substrate by photopolymerization if they pass through the target location(s). We then measure the error involved in printing PS beads into various structures (20 µm). Finally we characterize the biocompatibility of the polymer and substrate for hepatocyte cells and pattern two types of cells onto a substrate. We believe this demonstrates a clear direction towards engineering sophisticated liver like tissue as an organ on a chip demonstration. I'd like to thank Arnout Schepers and Prof. Sangeeta Bhatia for supplying us with hepatocytes and for contributing their expertise and resources to culture samples.

Introduction

Liver disease and loss of liver function is the 12th most frequent cause of death in the United States[97]. We need alternatives to liver transplantation to properly address developing diseases such as non-alcoholic fatty liver disease and steatohepatitis. There are currently no other strategies that directly alter mortality.

Printing of mature hepatocytes and hepatocyte aggregates into polymeric scaffolds and structures is being pursued by engineers and biologist to organize cells and their constituent biomolecules into a structure that, after culturing, functions like that of the original liver. This is being pursued not only to meet the grand challenge of synthetic liver transplantation but also to generate organs-on-chip that exhibit similar function and can be used to test drugs in a controlled environment.

One approach to creating polymeric scaffolds is by cellular encapsulation in a polymeric semi-permeable membrane. This is a tissue engineering strategy that protects the cells from immune cells and antibodies [98]. Macrophages specifically are the first line of defense for the first 8-10 days of wound healing or transplantation [99]. In addition, encapsulation permits bi-directional diffusion of small molecules such as oxygen, nutrients, growth factors and outward diffusion of waste and therapeutic proteins. Common materials for fabricating cell scaffolds include making sheets of collagen and chitosan as they are readily biocompatible and inexpensive to fabricate. More recent work has also applied PEG-DA hydrogels as the encapsulation material with success equal to that of the biologically inspired materials.

Microparticles can be assembled like building blocks into 2D and 3D structures that resemble the organization found in tissue. To this end polymer microparticles have been investigated as a promising strategy for producing conformable cell seeded scaffolds [100]. Sahoo et al. has shown polymer spherical PLA-PVA microparticles being used to culture breast cancer cells. The authors showed that microparticles with properly tuned adhesion and growth factors can produce tissue like environments around the cell after a 5 day incubation period. PEG-DA spherical particles and plates have also been employed by Bhatia et al. and shown to produce liver like function when implanted into rat models. However, spherical microparticles, while readily available, limit the ability of researchers and clinicians to leverage the variety of material characteristics available to a tissue engineer.

Here we seek to establish the capability of dynamic lithography to print multiple cell types in arbitrary 2D patterns. This requires: (1) biocompatibility (2) a printing. One of the remaining challenges in this field is how to best rapidly position cells to mimic the shape of complex liver tissues structures [101]. We present work towards printing cells into tissue like structures by photopatterning cells and microbeads onto a substrate based on an arbitrary pattern. First, an algorithm for printing cells or other particles flowing in a fluidic device is discussed and then Polystyrene (PS) microbeads are printed into arrays and images. We characterize the error and limits of printing beads. Next, we characterize the biocompatibility of the polymer and substrate for hepatocyte cells. Finally, we statically pattern two types of cells into a tissue like material and culture for viability. Challenges are identified and future solutions are analyzed.

Printing algorithm

We developed an algorithm to print microparticles using machine vision to efficiently track particles of interest and anchor them on demand. On top of the foundation built into dynamic lithography so far an additional function matches particles to print locations. As shown in Figure 93 a file is loaded that represented the desired print locations on the screen. The print locations are filled from the right to left using the dynamic lithography process. As particles flow across the screen, the algorithm detects if the particle trajectory at time t+1 will be in the region of interest (ROI) around the print location. Particles that do not meet this requirement are filtered from the candidate list. Further filtering is performed on the particles in the ROI by measuring their trajectories and velocities. If the trajectory will pass within a specified distance from the print location then the candidate is marked. Finally, due to polymerization kinetics and latency in the system a particle with a velocity too fast to anchor accurately is rejected. Rejection occurs based on knowledge of the polymerization time (~0.05 s) of the photopolymer. A particle that passes each of these checks is then exposed to polymerize the photopolymer around it until it is anchored to the substrate. The timing to begin exposing the particle is a function of particle velocity so that it is anchored to the correct location. Using the velocity of the particle and its distance from the print location upon detection an estimate is calculated of the arrival time of the particle to the print location. This is updated every frame and when the value is equal to 0.05 seconds then exposure begins.

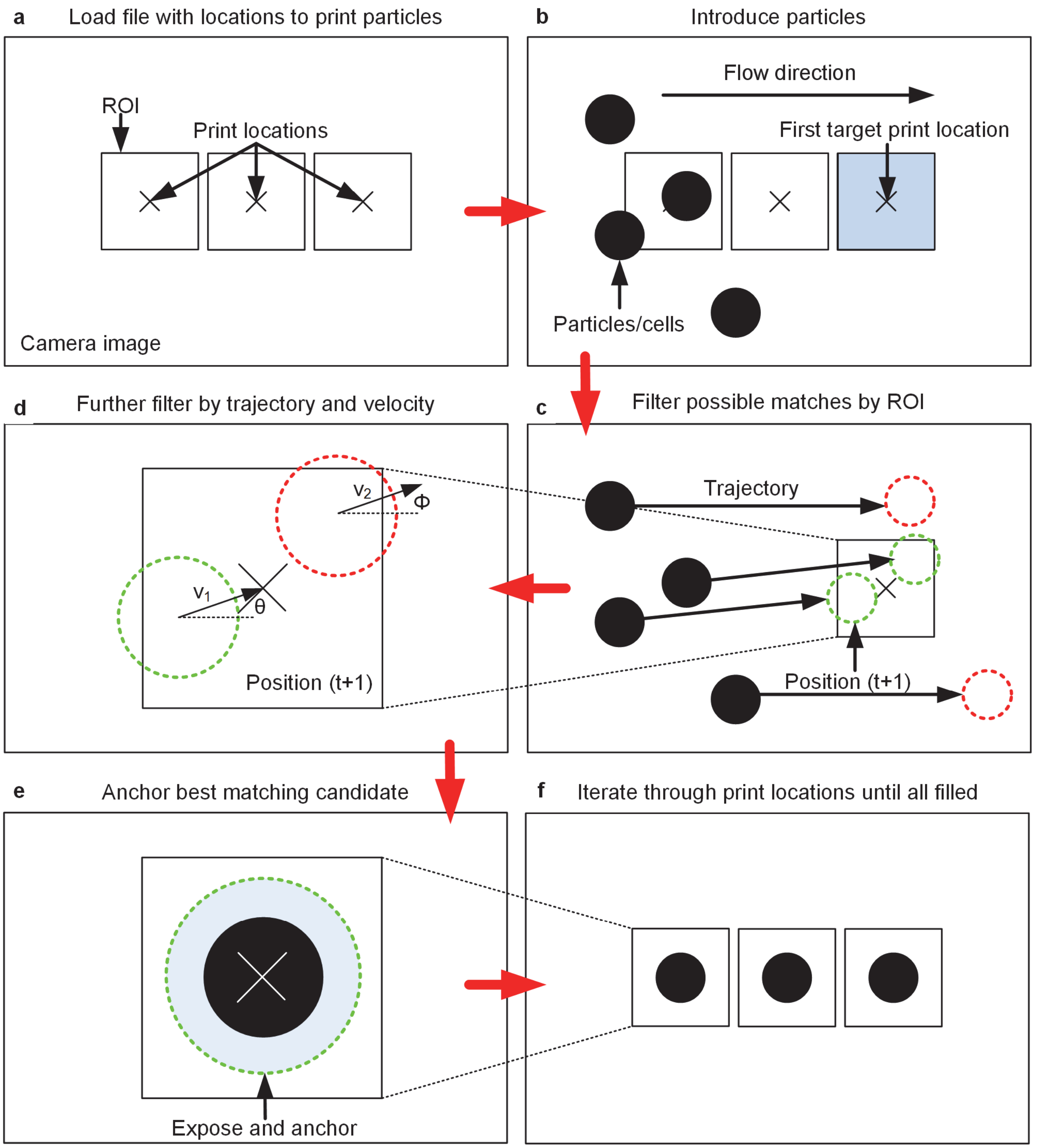

a
Load file with locations to print particles
ROI
Print locations
Camera image
b
Introduce particles
Flow direction
First target print location
Particles/cells
d
Further filter by trajectory and velocity
v2
Φ
v1
θ
Position (t+1)
c
Filter possible matches by ROI
Trajectory
Position (t+1)
e
Anchor best matching candidate
Expose and anchor
f
Iterate through print locations until all filled

Figure 93. Flow chart of printing algorithm. a) Load print locations into memory. b) Flow particles across print locations. c) Filter particles not entering the region near print location. d) Further filter candidates by trajectory and velocity. e) If best candidate is found anchor it. f) Iterate this process over each column of print locations.

Methods

Culturing cells

Rat hepatocytes (Hep G2) were cultured at 37 °C at 5% CO2 using a culture medium composed of Dulbecco's Modified Eagle's Medium (DMEM) (Invitrogen) supplemented with 10% Fetal Bovine Serum (FBS) and 1% antibiotic-antimycotic (Gibco). Cells were cultured in 100 mm culture dishes until reaching ~ 80% confluence and were then passaged using 0.25% Trypsin/EDTA (Gibco) to detach cells from plates prior to diluting and transferring them to devices. Those used for experiments followed standard protocol in a biological hood and had the media drained first. Then they were washed in 8 ml of phosphate buffer solution (PBS) for 30 seconds. The PBS was removed and 2 ml of trypsin was pipetted into the dish. The dish was gently agitated and placed into the incubator for 3 minutes. Once, 90% of the cells were suspended in the trypsin 8 ml of media was added to the dish. The cell solution was moved to a 15 ml centrifuge tube and spun at 200 RCF for 5 minutes. The media and trypsin was drawn leaving the cell pellet and an additional 10 ml of PBS was added to the tube. If the cells were being used in a device counting and partitioning was performed prior to spinning.

After printing the hepatocytes the excess PEG-DA solution and cells were rinsed into a biohazard container and disposed of. Then the device was submerged in media and placed in an incubator.

Testing viability and exposure time

Bio-compatibility of PEG-DA and the Photoinitiator were unknown for this experiment. A full factorial survey of photoinitiators and PEG-DA/PBS concentrations and exposure times was performed to determine what combination met criteria of viability and rapid polymerization. Unlike other studies in which very small levels of photo-initiator can be used and exposed for 15-30 minutes, dynamic lithography requires exposure times on the order of milliseconds to seconds. Three photo-initiators were studied including (a) 2-hydroxy-2methyl-1-phenyl-1-propanone (Darocur 1173), (b) Phenylbis(2,4,6-trimethylbenzoyl)-phosphine oxide and a mixture of (c) Eosin Y and TEA. The first was prepared by mixing Darocur 1174 with PEG-DA at a concentration of 5% v/v. The second prepared by mixing 98 ml of PEG-DA and 2 grams of Phenylbis(2,4,6-trimethylbenzoyl)-phosphine oxide. The last mixture was prepared with 100 ml of PEG-DA and 0.01mM Eosin Y and 0.1% TEA. Each 100 ml PEG-DA solution was then diluted with the cell/PBA solution to PEG-DA Molar concentrations of 50%, 25% and 12.5%.

For testing the effect of concentration and exposure time on viability, 100 µl droplets of cells were dispensed into a line of 8 droplets. Each droplet was cured by the system sequentially at the following exposure times: 0.5 s, 0.75 s, 1 s, 1.25 s, 1.5 s, 2 s, 5 s, 8 s. The droplet rows were ordered on the substrate by concentration.

To study the effect of substrate choice on cell viability and PEG-DA wetting several materials were tested including collagen, cellulose, PFA, nylon, polyester and cellophane. These were in addition to glass slides and PDMS, which have been used in previous studies. In addition, the engineering materials were studied to identify candidate materials to print cells on large areas.

When printing, the cell/PBS solution was mixed with the PEG-DA solution to a concentration of 10% PEG-DA/Photoinitiator. The final composition after biocompatibility assessments was prepared by mixing 100 ml of PEG-DA with 0.01 mM Eosin Y and 0.1% TEA.

Polystyrene microparticle PEG-DA solution

The polystyrene (PS) microparticles (Poly Sciences) were mixed into a pre-prepared solution of PEG-DA. The PEG-DA was prepared as before by mixing 98 ml of PEG-DA and 2 grams of Phenylbis(2,4,6-trimethylbenzoyl)-phosphine oxide. This photo-initiator is known from previous work to polymerize near 405 nm and is compatible with the system. Concentrations of beads were estimated by the manufacturer per droplet from the bottle.

Experiment for printing beads and cells

Beads were printed onto a glass slide covered by a PDMS microfluidic channel. A syringe pump was used to control the flow rate of microparticles across the field of view of the system. This approach was used for two example prints. The first was a grid array of particles while the second was the word MIT. The error in printing position was measured using ImageJ. Finally, aggregates of cells were cultured and suspended in a PEG-DA solution. These cells were cast onto a glass slide and sequentially photo patterned by type. The cells were cultured and a live dead stain was performed after 3 days to confirm growth.

Results and discussion

Substrate selection for large area tissue printing

To determine candidate materials for printing liver like tissue, wetting studies were first performed. For this study Phenylbis(2,4,6-trimethylbenzoyl)-phosphine oxide in PEG-DA and PBS was used at varying concentrations of PBS. Table 11 shows the observations of this experiment. The contact angle was measured to determine wetting angle and polymerized 100 µm pillars patterned on to the material were tested for adhesion by rinsing them twice in water.

Table 11. Substrate tests for qualitative wetting and adhesion.

Material	Wetting (concentrations)	Adhesion
Collagen	Low	Low
Cellulose	High (25/25)	High
PFA	Absorbed water and deformed	Low
Nylon	Medium (25/25)	Medium
Polyester	Medium (25/25)	Medium
Cellophane	Low	Low

From this data we chose cellulose as a base substrate for future experiments in addition to PDMS devices for lab on a chip experiments.

Cell viability on cellulose

Because it was previously known that PEG-DA was compatible with these cells, we examined the photo-initiator that would provide the fastest polymerization time while maintaining viability. Using the methods outlined above, viability was tested for three different photo-initiators on cellulose. The results of this viability test are reported in Table 12.

Table 12. Photo-initiator viability

Photo-initiator	Primary wavelength	Biocompatible (Molarity)	Polymerization time (s) (50, 25, 12.5, 7.5)	Water solubility level (PEG/PI concentration)
2-hydroxy-2methyl-1-phenyl-1-propanone (Darocur 1173)	365	N/A	∞	N/A
Phenylbis(2,4,6-trimethylbenzoyl)-phosphine oxide	450	Yes (25)	0.05, 0.8, 1.2,	< 20 %
0.01 mM Eosin Y and 0.1% TEA	405	Yes (12.5)	0.06, 0.8, 1.5, 3	All

Printing beads

As a first evaluation, we examine the performance of the system when printing PS beads onto a stationary substrate. The beads flow within the channel until anchored. Figure 94a shows an example of printing a 2x2 matrix of beads while Figure 94b shows a more complicated pattern of printing the acronym MIT.

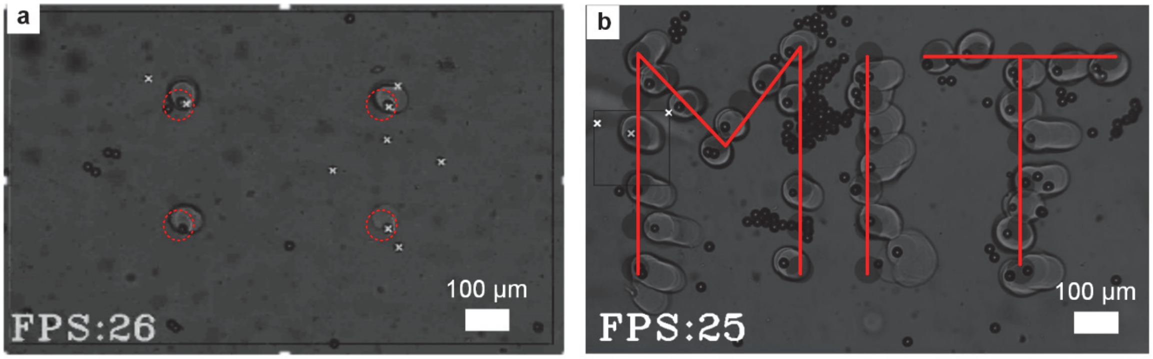


Figure 94. Dynamic lithography based printing of PS beads. a) A simple sparse 2x2 matrix of pillars. b) A more complicated MIT pattern, displaying blocking and clustering problems.

Both examples demonstrate opportunities for improvement. In the first example all print locations are successfully printed with a single object however, accuracy can still be improved. The distance between where the particle is supposed to be anchored and where it is actually anchored is 22 µm on average with a standard deviation of 2.5 µm. The arrival rate of particles that will flow over a print location within 2 pixels is directly proportional to the density of the particles in solution. Because the density of particles (number/volume) can increase as particle diameter decreases, the arrival rate of particles can be increased by reducing the particle size and increasing the density. Moreover, because the particles are not tracked until they are inside the ROI, the ROI size is a function of flow rate. Half of the ROI length must be greater than the particle velocity multiplied by exposure time. So, there is a tradeoff between velocity and computational performance that limits throughput.

The second example shows other challenges that need to be addressed including aggregation of particles before printing and blocking of print locations. Aggregation can be addressed by the use of surfactants to improve dispersion. Blocking can be addressed by integrating a z motion into the fluidic channel that draws trapped particles upward emptying the location and enabling the correct particle to find its position. Implementing improvements to the algorithm such as those discussed in 0 is the path forward to improving the accuracy. But a better understanding of the variation in polymerization time to anchor a particle as it relates to flow rate would help identify the source of variation.

Printing cells

Finally, we printed two cell types onto cellulose using sequential exposure by casting the substrate with cell (hepatocyte aggregates) laden solution and polymerizing the pattern shown in Figure 95. Then the substrate was rinsed and cast with the second cell type (Hep G2). This was also polymerized and excess PEG-DA was rinsed away. The final device was incubated, and live/dead stain after 3 days showed viability of the pillar arrangements. Live/dead staining uses selective staining of intracellular esterase activity and intact plasma membranes to identify cytotoxicity. Using green fluorescent calcein-AM green stain images show intracellular esterase activity in the marked "Live" in the figure. While a red-fluorescent ethidium homodimer-1 stain is used to indicate loss of the plasma membrane. The live/dead figure is a combination of the live stain image and the dead stain image to show the proportion of cells from the population that survived. Note that fluorescent dye is also absorbed into the hydrogel structure.

Two patterns were tested by sequential maskless lithography, not using the photopatterning process to help us understand how to use it in the future. The first was a perfusable arrangement of pillars with a vein created by the pillar arrangement. Future work may take advantage of such an arrangement to then pattern endothelial cells. The second pattern is an arrangement of pillars and rods. The best combination of cell types and their arrangement is still an open question in the tissue engineering field. This method and the addition of dynamic lithography may make it easier to test a variety of patterns to identify trends that improve tissue growth.

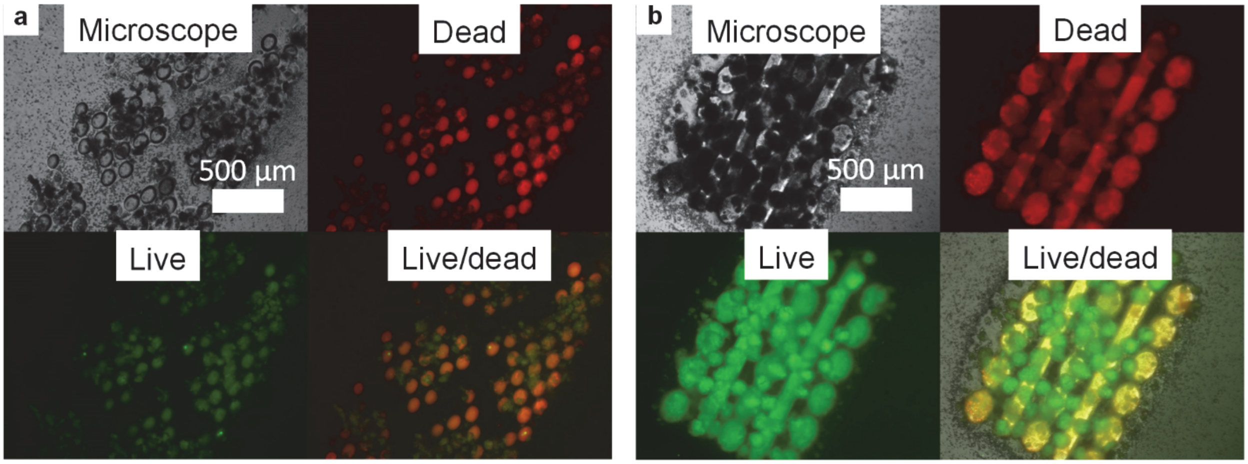


Figure 95. Demonstration patterning cells. a) Live/Dead stain of Hepatocyte cells polymerized into a grid of pillars with a vein structure running through the middle. b) Live/Dead stain of single hepatocytes (bars) and HepG2 clusters (pillars).

We observed that clearing the PEG-DA from in between polymerized regions was difficult and cells that were nearby but not in the structure were also bound to the substrate. This is why we expect printing the cells sequentially via machine vision driven photo-patterning to be more robust method over time.

Having established the cell viability and demonstrated growth of multi-cell structures future work will leverage dynamic lithography to identify the improved combinations of pattern size, spacing, arrangement and biological building blocks. After this has been accomplished the results will be applied to pattern large areas a via roll-to-roll dynamic photopatterning system. The layout of these experiments is shown in Figure 96. In this concept, the print region is translated at the same velocity as the tape.

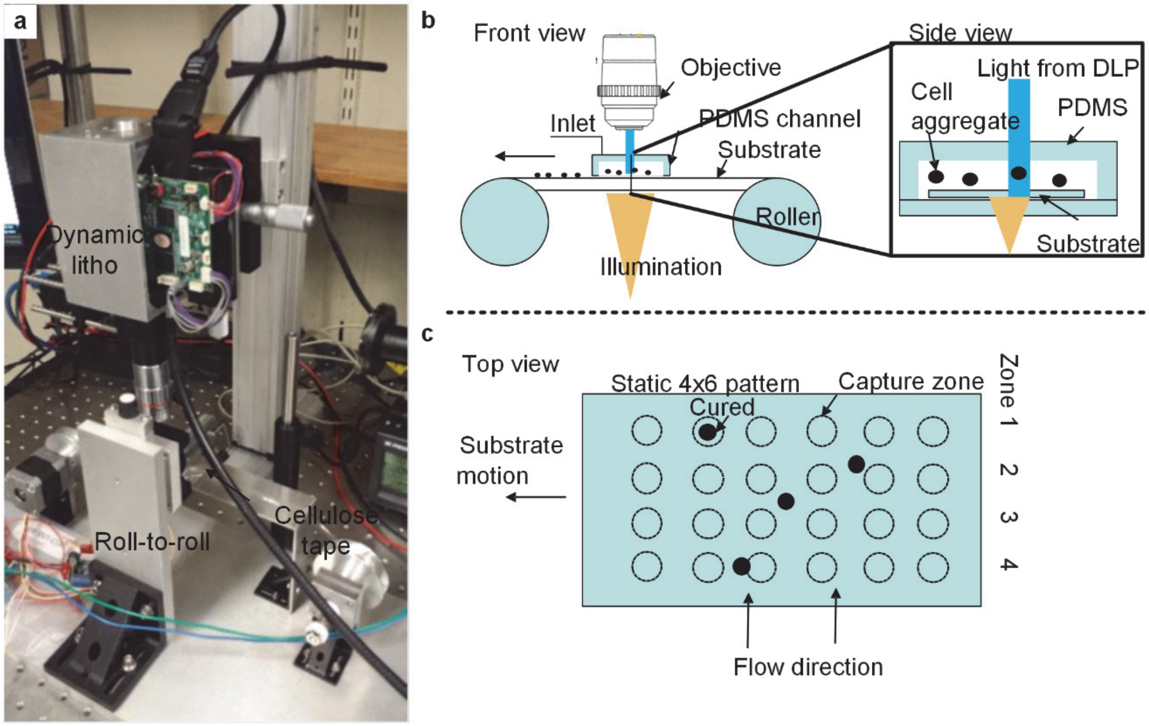


Figure 96. Roll to roll printing concept of large area tissue. a) Arrangement of the roll-to-roll system integrated into the dynamic lithography system. b) Front view schematic illustrating the arrangement of a fluidic device, substrate, objective and cell flow. c) Top view showing particles flowing across zones and being anchored into a pillar array.

Conclusion

The main advantage of dynamic lithography for printing is its flexibility and capability to pattern objects across length scales from nanometers to microns and adapt to particle position in the micro-environment. This is the distinct difference between other methods. For example this method could be adapted to identify particle clusters on the surface and anchor only those that have assembled into dimers and trimmers. A comparison of printing methods is shown in Figure 97. These include contact printing, inkjet, LaserJet, electro hydrodynamic printing and pick-and-place. The advantages and challenges of pick-and-place were discussed in Chapter I but each method is capable of compared in more detail relating throughput, minimum object/feature and the ability to print different types of materials and objects.

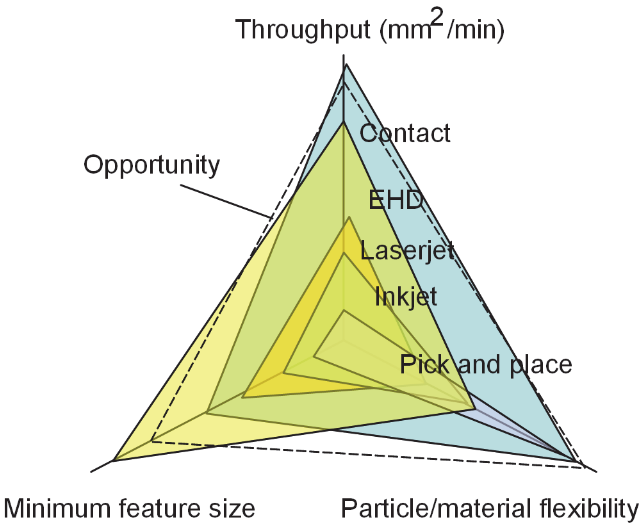


Figure 97. Relative performance of common micro-printing methods.

Experiments shown here demonstrated the ability to print particles with 20 μm accuracy. Continued improvement of the method should enable better localization of the cells onto the substrate. Moreover, we expect that in future work, it will be important to consider new methods of delivering particles to the system to address the scalability of this method, because current implementation may suffer from slow cycle times waiting for particles to arrive at the desired location along with blockage of unfilled sites. However, this is not a challenge for applications that can take advantage of low density patterns.

CHAPTER VIII. A ROBOTIC SYSTEM FOR PHOTOPATTERNING OF LARGE THREE DIMENSIONAL SURFACES

Abstract

The fabrication of organized micro- and nanoscale textures on everyday objects demands new methods for scalable patterning, particularly when the substrate is curved in one or more dimension. We describe the design of a dynamic lithography system mounted to a six axis robot that projects images onto a curved surface and alters the size and shape of the projection based on the local radius of curvature of the substrate. Motion of the robot is coordinated with a rotary stage to access all sides of an object. The resolution, repeatability and accuracy of the system are measured. Finally, we demonstrate the method by patterning North America onto a precision ground steel ball coated in photo-resist. The revolution in customized manufacturing has been led by additive printing methods which build up structures from a polymer. Here, we rather examine a method to pattern pre-existing objects by adding material to what already exists. For example, this work may be used to pattern cells directly onto bone to aid healing. Towards, this ultimate goal Adam Stevens deserves equal credit for this work and I enjoyed working side by side with him to develop and test the final system. The project began in the ME450 course at the University of Michigan, where it was co-mentored with Erik Polsen and included the following team members Chad Archer, Jenna Garber and Casey Boyle.

Introduction

Patterning of photosensitive polymers on arbitrarily shaped surfaces would enable the application of lithography methods to large curved objects, and therefore would be an enabling technology for developing highly integrated electronics in biomedical applications, such as external devices (body armor, helmets) and implants (eye, heart, and bone implants)[102]–[104]. One approach is to fabricate stretchable and/or flexible electronics on PDMS that then contour the device/object. John Roger's group, has applied this method to demonstrate conformal brain implants and a glove-like pericardium[105].

Several other methods have been discussed before including stereolithography as discussed in 0 and laser based methods of rastering across the surface of an object[106], [107]. We aim to overcome limitations in patternable area by using robotic lithography to position a six axis microlithography system.

Here we discuss the system design and process for patterning curved substrates. Design of a multi-axis micro lithography system is subject to mechanical error so we discuss the role of error build up in the system components[108], [109]. We characterize the resolution and aberration performance of the system both optically and lithographically. Using this framework, we finally demonstrate the patterning of a micro-scale map onto a spherical metal ball.

System design

The system is comprised of three major components: 1) Maskless lithography system. 2) Positioning system for the part and lithography system. 3) An object to be patterned and its mount. The coordination of these is controlled via software.

The design of the maskless lithography system is described in detail in 0 and 0. The maskless lithography system is mounted to the robot end effector and can be positioned anywhere within a 20x20x20 cm work envelope to manipulate the projection relative to the part. The back of the housing shown in Figure 60 in Chapter V, has an alignment pin on the back of it which mates with the six axis robot end effector.

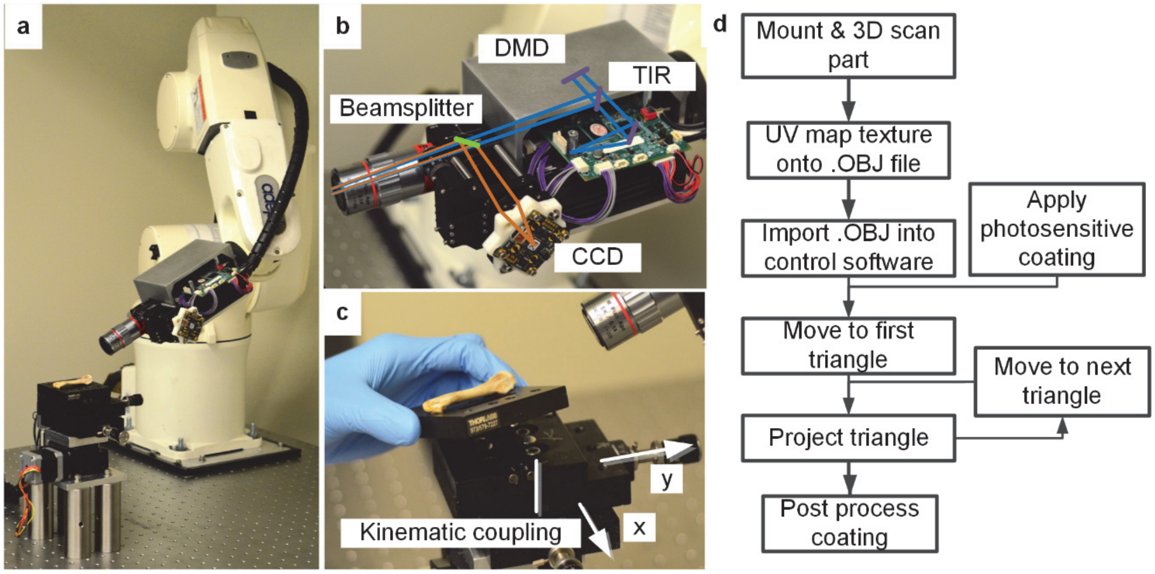


Figure 98. Robotic lithography system and process overview a) Photograph of the robotic lithography system and rotary stage. b) Close up view and schematic of the dynamic lithography system mounted to the robot. c) Researcher showing the kinematic mount for precision alignment between mounted objects. d) The basic process of printing onto a curved surface.

To enable the large work envelope the 6-axis robot (Adept Viper s650) and a rotary stage (Velmex B5990TS) (Figure 98) are positioned relative to one another. Both are mounted onto a precision optical table. The goal of the positioning system is to accurately and repeatably focus the projection from the maskless lithography system normal to the substrate at the correct focal distance. The rotary stage is mounted on a 12.5 mm thick aluminum plate that is spaced from the optical table using four 100 mm optical posts (Thorlabs P4) which have a tolerance of ±0.127 mm.

The part to be patterned is mounted on a kinematic mount (Thorlabs KB3X3) (Figure 97c). The purpose of this is so that the part can be removed to 3D scan and coat in photoresist and then placed back in the system while minimizing error. The kinematic mount is attached to the rotary stage using a custom adapter plate made of Aluminum. Two precision stages (Thorlabs PT1A) are mounted between the kinematic mount and the rotary stage. Because multiple top plates of the kinematic mount are used the stages remove the error between the mount and the rotary stage z-axis.

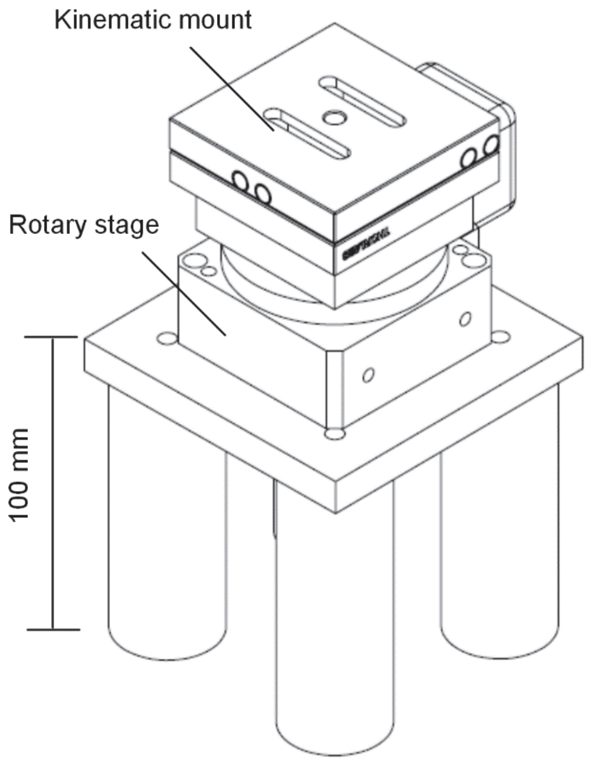


Figure 99. Rotary stage and kinematic mount detail.

The motion and timing of each component necessary to conformally pattern in three dimensions is controlled by a custom C++ algorithm that integrates the drivers from the rotary stage, robot and lithography system. First, a 3D scan of the object is taken using a 3D scanners (NextEngine) from which three dimensional coverage is achieved by triangulating the surface of the object and saving the data in the obj format. The location of each triangle in the digital-space is then associated with that in the real space. Each triangle is then associated with the texture contained within its borders and stored in computer memory for later projection. The software iterates over the list of triangles in saved in the obj list. For each triangle it finds the centroid of the triangle and commands the rotary stage and robot to move so that the robot is normal to the centroid and the correct focal distance. It then displays the texture and triangle on the projector, scaled to match the optical magnification and turns on a blue LED to expose the surface and cure the photoresist for 0.1 s before moving to the next triangle. This process is repeated until all triangles have been patterned.

Methods

Using this system the experimental process for photopatterning onto the surface involves mounting a part, depositing photoresist on its surface, 3D scanning the object and preparing a texture file of it and then exposing sequentially over the entire part. For the demonstrations shown in this study a steel ball was selected as the curved substrate and a glass slide was used for linear patterning. The steel ball has a precision of < 1 µm so it is ideal for this study.

Preparing photoresist

First the mounted substrate (e.g. the sphere) is cleaned with acetone and IPA and dried with a nitrogen gun. The photoresist used is MicroSpray MicroChem positive spray photoresist. Following the manufacturer guidelines we shake the can approximately 10 times and spray once away from the substrate to clear the nozzle. Next, 3 quick overlapping passes are sprayed onto the substrate to deposit ~5 µm of resist. The substrate sits for 5 minutes and is then warmed with a hot air gun set to 116 C° for 5 minutes.

After exposure across the surface of the substrate (1.9s/triangle) the part is immersed in 0.1N NaOH for 5 minutes without agitation. After 5 minutes the part is rinsed with DI water to remove unpolymerized resist. The final part is dried under nitrogen or dry air.

The hydrogel was applied to linear curved substrates by dipping the substrate upside down in a 100 ml beaker. Smaller areas were covered by pipetting 10 ml onto the surface and wetting corners.

Results

Accuracy and repeatability of the motion system

The measured planar repeatability and accuracy of the motion system and thus the projection is 20 microns and 30 microns, respectively. It was measured by moving the robot to and from a desired location and then measuring the shift of a target object as imaged by camera via ImageJ. The accuracy of the robot in 3D is much more difficult to characterize but shows degraded performance. It was measured by patterning triangles along a surface with a radius of curvature of 12.5 mm. The triangles were patterned from the x-axis of the ball to 45 degrees off axis in both the horizontal and vertical directions. It was observed that the system had a maximum positioning error of 325 µm in 3D space.

Moreover, the resolution of the system was investigated. We hypothesized that the photoresist monomer length would affect quality of the pattern. Both the hydrogel and the photoresist were applied to the glass substrates and patterned using the USAF 1951 test target. The results show that using MicroChem resist a resolution of 5µm in a 5 µm thick layer was achieved. This was also seen in the rounding of corners for example in Figure 100c. The same experiment was performed using PEG-DA and this showed worse performance with a 10 µm resolution. This may also be affected by the nature of each resist (positive vs. negative).

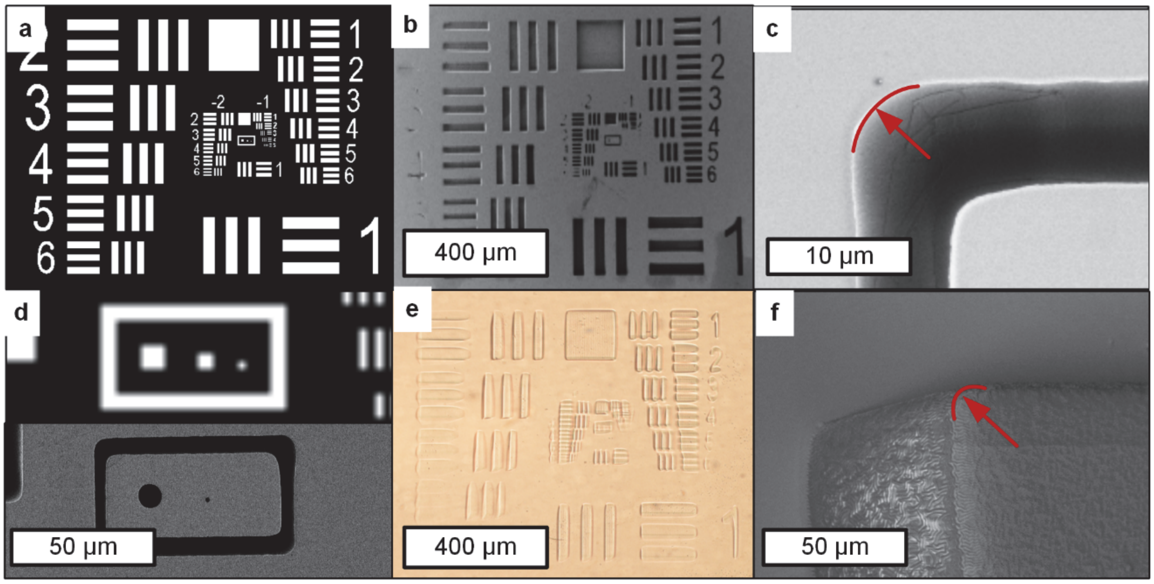


Figure 100. Resolution characterization for robotic lithography. a) USAF 1951 resolution test pattern. b) USAF 1951 test pattern exposed into a 5 µm thick layer of MicroChem spray photoresist. c) Close up view of smallest channel around the pattern shown in d. d) Comparison of 1, 4 and 16 pixels used for exposure. e) USAF 1951 test pattern in PEG-DA on a glass slide. f) SEM of corner of small feature on fabricated from PEG.

Taking into account the depth of field of the system (δ) and the local radius of curvature of the substrate (R) we calculate (Equation 47) the parameter α^2 representing the area of the DMD that can be used for each exposure accounting for the DOF of the objective. This relationship is plotted in Figure 101. However if the radius of curvature is smaller than δ then α^2 is just $(\pi R)^2$.

$$\alpha^2 = 2R\,sin^{-1}(\frac{\sqrt{2R\delta-\delta^2}}{R})^2 \qquad \text{Equation 47}$$

This is used to decide the triangle size when generating the mesh. By measuring the smallest radius of curvature on the scanned object we can calculate α^2 and correlate it to a triangle length. While we expect this to critically determine quality of the projection over the surface this is a tradeoff with processing time as smaller triangles are proportional to number of exposures.

Demonstration

As seen in Figure 101a, the ball mounted on the kinematic mount attached to the stage was coated with resist, exposed by the system, and rinsed. The pattern that was exposed onto the ball is shown in Figure 101b and is represented by a triangular mesh with the pattern of North America textured onto its surface by UV mapping. The system sequentially moved the centroid of each triangle in the mesh and projected the pattern linked to the triangle. The result of this patterning process is shown in Figure 101c. The micrograph shows the state of Michigan patterned onto the ball with feature resolutions approaching 5 µm. An SEM of the surface shows the steel ball surface and the texture of the photoresist after processing for a bay (Figure 101).

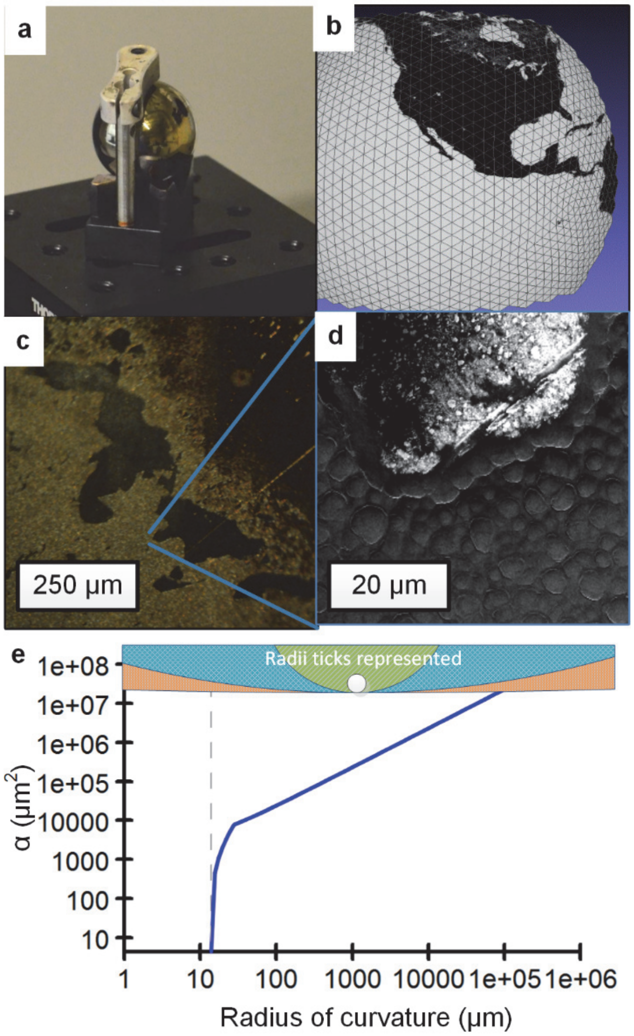


Figure 101. Demonstration of patterning a curved surface. a) Steel ball mounted onto stage. b) .OBJ file with texture mapped onto surface. c) Photograph of the ball after processing showing Michigan. d) SEM image of bay showing metal surface and photoresist structure. e) Area the DMD can pattern as a function of radius of curvature of the substrate and depth of field of the system.

Conclusions

While limitations due to inherent machine geometry exist, this demonstrations shows the potential for patterning arbitrary curved surfaces. The role of the depth of field and radius of curvature play a balance in both throughput and clarity of the projection.

This technique bridges the gap between microlithography and macroscale objects as the size of the eventual pattern is limited only by the combined work envelope of the robot and rotary stage. It is also flexible and able to work in different environments such as manufacturing factories and operating rooms. While the goals differ the methodology presented here is suited for both.

Considering, the potential use in manufacturing we imagine this capability being used in concert with a variety of post processing steps. Industry may have the goal of patterning a surface to protect it during an etching or deposition process. For example the area not covered by resist could be coated with metallic spray to make a conformable PCB. Another use is to cover a surface before applying catalyst that is then used to generate nanostructures selectively.

To meet these challenges further improvement in micropatterning accuracy may be found by the addition of an absolute positioning systems. As discussed, error in absolute robotic position is inherent to the design of six axis robots but corrections can be made if the position of the end effector is known by a secondary measurement. New methods are being commercialized (Creaform) to provide this correction and achieve accuracy down to 15 µm.

The second case may be depositing material directly such as cell laden hydrogels. We imagine that uses within the operating room in which 3D scans of the patient and then subsequent patterning may enable improved healing or the integration of sensors and electronics that interact with cellular features. The ability to rapidly create custom two-dimensional patterns on three-dimensional substrates as demonstrated is expected to unlock new functionality in fields including, but not limited to, experimental semiconductors, personalized medical devices, and optics.

CHAPTER IX. CONTRIBUTIONS AND FUTURE WORK

Key contributions of this work

This thesis was concerned with the development of a machine vision based optofluidic lithography platform for synthesizing and manipulating microparticles, objects and organisms. This was then applied to sort and print microparticles and cells, achieving accuracy and resolution that is suitable to biologically relevant applications. The design of the maskless platform was applied to a robotic end effector to pattern curved large area curved objects. To summarize, the contributions of this thesis include:

- Chapter II I designed and constructed a modular maskless system for 365 nm wavelength lithography. The system has diffraction limited performance which is comparable to commercial microscopes yet the system offers additional flexibility such as the capability to be integrated into manufacturing systems. Traditional systems require that the substrate match the lithography machine which I overcome. The design is also simple enough for any laboratory to fabricate.
- Chapter III Continuous microfluidic synthesis of microparticles with monodisperse sizes (5-15% CoV) was studied. This was done to address limitations in fabrication of particles for self assembly and multiplex diagnostics. Previous methods were not able to fabricate small particles with sufficient throughput for these applications. I hypothesized that microfluidic channel design plays a role in fluid stop time, which is critical to throughput. Using this understanding we designed a new microfluidic device to increase throughput up to 1500X for particles width a 5 μm height enabling synthesis of up to 2.5×10^6 particles per minute.
- Chapter IV I hypothesized that interactive, *in vivo* lithography could be used to study the locomotive, force response and decision making ability of *C. elegans*. I first studied the behavior of *C. elegans* in assays designed by dynamic lithography to test variations in worm locomotion to pillar spacing. I hypothesized that C. elegans would swim faster as pillar spacing decreased. By varying pillared structure spacing we quantified locomotion

of the worms and observed an increase in maximum velocity of up to 89% as 100 μm pillar arrays begin to overlap to form a channel. I also, hypothesized that free floating structures could be fabricated by controlling exposure time and focus position. This capability was verified to be true and patterned movable structures within the microfluidic device which could measure force response from worm locomotion. This is the first to the author's knowledge that lithography has been applied to manipulate whole animal models such as *C. elegans*. Moreover, I studied the diffusion of PEG-DA into agar as I hypothesized that this would enable adhesion to the agar. Unlike in PDMS devices which are inhibited by oxygen diffusion through the material.

- Chapter V I wanted to understand how machine vision might enable the lithography system to dynamically respond to particles in flow and intelligently project masks to encapsulate or capture them for sorting and self assembly applications. To do this I hypothesized that a simple algorithm could detect the particles from a binary image and draw the projection using their moments. I found that this was not true and latency of the system had to be modeled and accounted for to project in the correct location. To do this I wrote software to integrate system components and detect, track and process microparticles at 1140 fps with 4.4 pixels accuracy. I then demonstrate the ability to sort PS beads with a sensitivity of 100%, specificity of 99.9%, accuracy of 99.9% and precision of 98.9%. This is comparable to other methods such as Flow cytometry which can process 60 ml/min, however I processed 300 ml/min of particle laden fluid. In addition, this method has additional flexibility when compared to flow cytometry because metrics from the image can be used to describe particle properties.
- Chapter VI As I developed the dynamic lithography system, I was faced with a question of which detection and tracking algorithm to use. I hypothesized that the particle environment could be simulated by independently testing the following image properties: particle size, particle density, particle dynamics, particle velocity, particle complexity, blur. Using custom software I compared the performance of the detection and tracking algorithms to simulated ground truth and found that by controlling the lighting of my system to match the simpler simulations I could use traditional algorithms to accurately and efficiently detect and track particles. This study when compared to recent publications expanded on the parameters of interest from three to seven. This

additional information was valuable to give context about applying this method to other types of bio-imaging problems.

- Chapter VII I hypothesized that printing cells onto substrates in a template free manner could achieve high density, multi cellular environments that contribute to improved organ-on-a-chip fabrication. Using the dynamic lithography system could I tested the capability by printing PS beads as a surrogate cells. This identified two problems that need to be addressed in future work (blocking and aggregation). However, I did demonstrate sequential patterning of hepatocyte cells into organized structures using multiple cell types. I expected that photo-initiator selection and the PEG-DA dilution would impact viability of the cells. Therefore, I characterized the viability of hepatocytes to hydrogel encapsulation for 3 photo-initiators and 5 engineering substrates. This was the first time this photo-initiator was used with hepatocytes and cells were found to be viable in these structures for the incubation period of three days. Moreover, the speed of synthesis compared to previous template method using PEG-DA for cell encapsulation was an order of magnitude faster (1 second vs 30 seconds).
- Chapter VIII In this chapter I highlighted another potential application in patterning directly onto curved hard surfaces such as bone. I expected that the lithography system I used to print cells could be positioned arbitrarily and take into account the curvature of the substrate. To test this hypothesis I designed and characterized a system to pattern curved large area surfaces by mounting the dynamic lithography system to a six axis robot. This included establishing a standard procedure for scanning, mounting, coating and processing objects. During the demonstration of the technique I quantified the resolution of the system to pattern resist (5 μm) and hydrogels (4 μm). Finally, I patterned a spherical object across a 10 mm area. The result demonstrated that we could pattern curved objects and gave us a theoretical framework for the area that can be patterned in relation to the local radius of curvature. Furthermore, we now understand that mechanical error in the robot is not sufficient to use this method across the entire working envelope and further correction is needed to establish absolute positioning of the projected image relative to the substrate. Compared to other methods, stretchable soft lithographic features do not adhere to the substrate and cannot be used for post processing as this method could. The only other comparable method is a laser raster system that has

been commercialized. The throughput of my method compared to the laser method is more scalable because we can utilize a greater area of the DMD while the laser beam must remain focused to a point.

In addition, several important accomplishments exceed the state of the art and advance the field in a meaningful way:

- I designed and constructed an off the shelf system was designed for maskless microparticle synthesis that is more flexible than off the shelf lithography.
- I confirmed our hypothesis that microfluidic channel structure effects throughput and as a consequence tested a new microfluidic device that may synthesize up to 2.5×10^6 particles per minute is 1500X faster than a comparable system.
- This lithography system was used to fabricate *in vivo* and *ex vivo* assays 5000X faster than soft lithography. Moreover, this is the only system that enables *in vivo* manipulation in an unstructured way. I hypothesized that PEG-DA would diffuse into agar overcoming the oxygen inhibition layer. This was verified by ATR-FTIR measurements. I also discovered that *C. elegans* lifespan was not affected by assays that expose them to PEG-DA on agar. This was the first time that movable objects were fabricate directly inside of an assay device to interact with whole animal organisms.
- Developed the first machine vision based method to manipulate (sort, print) particles and biologically relevant materials via lithography. Flow rates of 10 mm/s enable sorting of 300 μL/min. Hypothesized that because particles move in a flow that the flow could be polymerized around them accurately encapsulating them. Discovered this to be true given accurate projection light.
- Finally, the robotic lithography system is the only system that can pattern arbitrary curves surfaces.

Key questions and future directions

While conventional 2D lithography continues to achieve higher performance for use in the semiconductor industry, the paradigm of designing intelligent lithography unlocks new opportunities. Going forward, we view the dynamic lithography system, comprising both hardware and software, as a general tool for manipulation, interrogation, synthesis, assembly, printing and sorting of particles, biological components and objects (Figure 102).

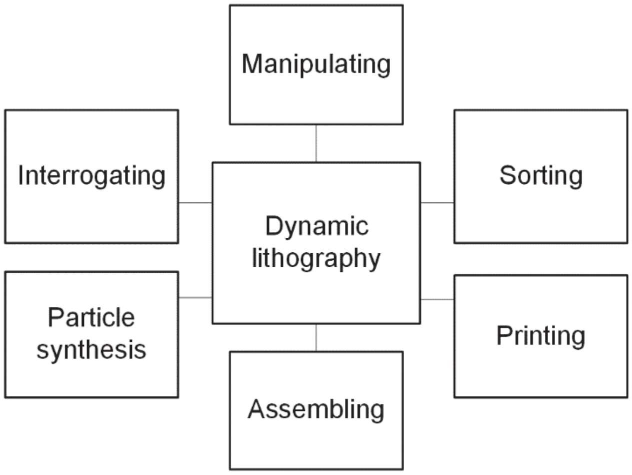


Figure 102. Potential directions for dynamic lithography.

To be able to adapt the system to each problem, key questions must still be answered. Specifically, we must understand, based on fundamentals of the system, how to improve its accuracy, repeatability, resolution and responsiveness. To improve these metrics we want to answer questions such as, “What is the most efficient detection and tracking algorithm?” Potential paths forward involve using patterned light sensitive photo-chemistry to control the microenvironment so that particles can be driven to the right location.

At Michigan and MIT I will continue to work on this problem to improve the capability of the system to print hepatocytes into functioning organs on a chip for drug testing. Primarily, we need to study how increasing particle density will affect patterning performance. This may arise from “blocking” and aggregating of particles as was seen when printing the MIT logo. In experiments so far (Figure 94), the first is seen when an undesired particle arrives at a print location and blocks a desired particle. The second occurs when particles begin to aggregate due to micro and nano scale forces (Van der Waals, depletion, etc). When this happens, anchoring a single particle becomes impossible. Understanding the scaling laws of interactions and developing a particle motion model may provide a clear pathway to understanding the underlying principles of blocking, aggregating and identifying the limitations of processing throughput.

To transfer this work to commercial manufacturing scales the speed of photopatterning tissue into 3D structures with arbitrary placement and multiple cell types needs to be improved. With the combination of the techniques presented in this thesis, it is anticipated that continued interest in the area of dynamic lithography will provide additional improvements in quality and throughput to leading to full scale implementation. This may be accomplished by improving the hardware (workstation, GPU), software (detection, tracking), modeling (photo polymerization, flow) and photochemistry.

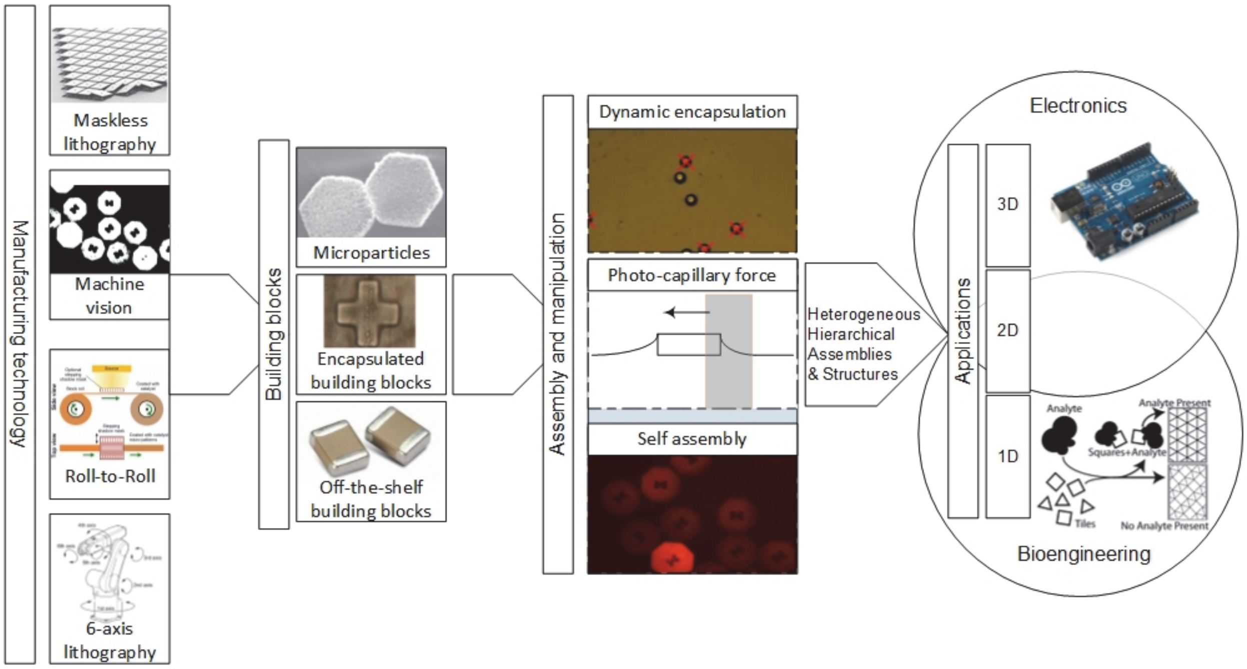


Figure 103. Vision for the future of flexible adaptive microfabrication methods.

To complete the long-range vision, we have identified a number of technologies as part of a vision for flexible adaptive microfabrication that takes advantage of and compliments the work in this thesis. Automated manufacturing technologies such as maskless lithography, machine vision, roll to roll processing and robotic lithography can be used to fabricate or augment building blocks at different length scales in 2D and 3D. We then see dynamic lithography, photo-capillary driven assembly and self-assembly as a means of achieving heterogeneous hierarchical assemblies and structures. The structures and particle designs realized should be driven by the identification of important bioengineering and technology applications that leverage the adaptive nature of the dynamic lithography platform. For example, we envision future possibilities in which electronics and cells are combined/printed into the same structure enabling responsive grafts to monitor the health of customer and respond to changes in their physiology.

APPENDICES

APPENDIX A. WAFER DESIGN

Table 13. Wafer layout and characteristics.

Wafer Pos.	Pillar Cross Section Type (I-III)	Arrang -ement Type (A-H)	Pillar Spacing , x (um)	Channel Width (um)	Channel Length (cm)	Diameter (um)	Major Radius (um)	Major Radius (um)	Width (um)	Length (um)
1	III	H		400	0.5				20	60
2	III	H		400	0.5				20	60
3	III	H		400	0.5				20	60
4	III	H		400	0.5				20	60
5	III	H		400	0.5				20	60
6	I	A		100	0.5	20				
7	I	B		100	0.5	20				
8	I	C		100	0.5	20				
9	I	D		100	0.5	20				
10	II	A		100	0.5		40	10		
11	II	B		100	0.5		40	10		
12	II	C		100	0.5		40	10		
13	II	D		100	0.5		40	10		
14	III	A		100	0.5				20	60
15	III	B		100	0.5				20	60
16	III	C		100	0.5				20	60
17	III	D		100	0.5				20	60
18	I	A		200	0.5	40				
19	I	B		200	0.5	40				
20	I	C		200	0.5	40				
21	I	D		200	0.5	40				
22	II	A		200	0.5		80	20		
23	II	B		200	0.5		80	20		
24	II	C		200	0.5		80	20		
25	II	D		200	0.5		80	20		
26	III	A		200	0.5				40	120
27	III	B		200	0.5				40	120
28	III	C		200	0.5				40	120
29	III	D		200	0.5				40	120
30	I	E	25	500	0.5	20				

31	I	E	50	500	0.5	20				
32	I	E	80	500	0.5	20				
33	I	E	100	500	0.5	20				
34	I	E	130	500	0.5	20				
35	I	E	130	500	0.5	40				
36	I	E	130	500	0.5	60				
37	I	E	130	500	0.5	80				
38	I	E	25	200	0.5	20				
39	I	E	25	300	0.5	20				
40	I	E	25	400	0.5	20				
41		Blank Device		100	0.5					
42		F		10	1					
43		F		25	1					
44		F		50	1					
45		F		100	1					
46		G		10	1					
47		G		25	1					
48		G		50	1					
49		G		100	1					
50	III	H		400	0.5				20	60
51	III	H		400	0.5				20	60
52	III	H		400	0.5				20	60
53	III	H		400	0.5				20	60
54	III	H		400	0.5				20	60

APPENDIX B. MATLAB CODE FOR SIMULATING PARTICLES

```
% Particle Simulator for Objective Detection and Tracking Comparison
% Written by John Lewandowski and Ryan Oliver 10/23/2014

% The operation of this code depends on the motion type, speed, density,
% and size of particles needed in the system

% In this file, change the following:
% Add a new speed using the formula global stepX and stepX = (X/1276)*2 in
line 36
% Change the number of frames in line 48
% Change titles of output in lines 61 and 66
% For higher densities, increase number of particles added per frame in line
91
% Comment and uncomment lines 127 and 128 for ability to have multiple sized
particles

% These lines control Brownian in this file:
% Comment line 54 and uncomment line 56
% Uncomment entire Brownian section lines 93-113
% Comment entire Linear and Sinusoidal section lines 118-138

% These lines control Linear and Sinusoidal in this file:
% Comment line 56 and uncomment line 54, adjust speed


% Only change lines 544-573 in particlesystem.m for speed and motion type
(comment and uncomment)
% and lines 769-779 for density and field of view (comment and uncomment)

% Only change lines 171-186 in particle.m for sizes in pixels


% Clear memory
clear all
close all
clc

% Create the particle system that contains all particles
% ([gravity x, gravity y, gravity z], resistance, axes)
Particle_System = particle_system ([0, 0, 0], 0, 1);
% View is of 2D-motion
view (0, 0);

% Step time of simulation (speed of motion)
% Sets frame rate at 30 frames per second (1/step_time)
```

```
step_time = 0.033;

% Global step delta x = velocity
% This sets the global velocity variable
global step2
step2 = (11/1276)*2;
global step1
step1 = (5.5/1276)*2;
global step05
step05 = (2.75/1276)*2;
global step01
step01 = (0.55/1276)*2;

% Set length of simulation in frames
N = 60;
frames = N;

% Lifespan of particle adjusted for three motion cases
% Comment for Brownian
% Replace step2 with whatever global variable speed you use
lifespan = (2/step2)*step_time;
% Uncomment for Brownian motion
% lifespan = N*step_time;

% Name your video file here
writerObj = VideoWriter('particlevideofile','MPEG-4');
writerObj.FrameRate = 1/step_time;
open(writerObj);

% Name your data text file here
fileID = fopen('particletextfile.txt','w');

% Make headers for the text file
fprintf(fileID,'%10s %10s %7s %10s %10s %10s \n', 'frame', 'ID', 'false',
'Xpos', 'Ypos', 'Size');

% Initialize ID, frame, particle matrices, particle sizes, particle id, and
% position matrices
ID = ones(0);
frame = ones(0);
particlepositions = ones(0);
particlesize1 = 45;
particlesize2 = 90;
id = 1;
fxy = [];

%Get rid of axes and ticks in this simulation for clean background
set(gca, 'XTick', []);
set(gca, 'YTick', []);
set(gca, 'ZTick', []);

% Only for linear and sinusoid
% Determine how many particles need to be created per frame
% Creates a X multiplier for the density of the system
global particlesperframe
```

```
particlesperframe = 1;

% Uncomment section for Brownian motion only
%
% numbrownianparticles = 100;
% for i = 1 : numbrownianparticles
%
%     For Brownian, set the boundary coordinates of where particles can
appear in the FOV
%     a2 = -0.85;
%     b2 = 0.85;
%     r2 = (b2-a2).*rand(10000,1) + a2;
%
%     If more than one size of particle uncomment the denom (denom = number
of types)
%     To change particle sizes, also need to adjust particlesizes in
particle.m file
%     if mod(i,k) == 1
%     if mod(i,2) == 1;
%         Particle = particle (Particle_System, 1, [r2(i), 0, r2(i+1)], [0,
0, 0], false, lifespan, fxy, id, particlesize1);
%     else
%         Particle = particle (Particle_System, 1, [r2(i), 0, r2(i+1)], [0,
0, 0], true, lifespan, fxy, id, particlesize2);
%     end
%
%     id = id + 1;
% end

for k = 1 : frames

    % Uncomment section for linear and sinusoid motion only
    % For linear and sinusoid, set the boundary coordinates of where
particles can appear in the FOV
    a = -0.75;
    b = 0.75;
    r = (b-a).*rand(10000,1) + a;
    start = 0;

    % Uncomment section for linear and sinusoid motion only
    for j = 1 : particlesperframe
%   If more than one size of particle uncomment the denom (denom = number of
types)
%   To change particle sizes, also need to adjust particlesizes in particle.m
file
%   denom = k;
    denom = 2;
        if mod(k,denom) == 1
            Particle.id = particle (Particle_System, 1, [-1, 0, r(j)], [0, 0,
0], false, lifespan, fxy, id, particlesize1);
            id = id + 1;
        else
            Particle.id = particle (Particle_System, 1, [-1, 0, r(j+1)], [0,
0, 0], true, lifespan, fxy, id, particlesize2);
            id = id + 1;
        end
    end

    % For all types of motion, simulate one single time step by advancing
time
    Particle_System.advance_time (step_time);

    % Calculate size of array
    for l = 1 : size(Particle_System,1)
        %Get the particle positions of each time step
        particle_positions = get_particles_positions (Particle_System);
    end

    % Append the frame, x, and y to the particle entity
    Particle_System.appendfxy (k);

    % Saves each frame from iteration k
    particlemovieframe(k) = getframe;
end

%Add frames together into a movie that is saved in your MATLAB folder
writeVideo(writerObj,particlemovieframe);

%Close the video file
close(writerObj);

%Parse data
for i = 1 : k
    for j = 1 : size(Particle_System.particles,2)
        %idmatrix = j*ones(size(Particle_System.particles(j).fxy,1),1);
        I = find(Particle_System.particles(j).fxy(:,1) == i);
        parseddatatemp = [Particle_System.particles(j).fxy(I,1),
Particle_System.particles(j).id,
round(Particle_System.particles(j).fxy(I,2)*638)+638, 794-
(round(Particle_System.particles(j).fxy(I,3)*397)+397),
Particle_System.particles(j).particlesize];
        if size(I) > 0
        Particle_System.parseddata = [Particle_System.parseddata ;
parseddatatemp];
        end
    end
end

% Sort the data so that it can be interleaved
parseddatasorted = sortrows(Particle_System.parseddata,[1 2]);

% Initialize the final vector to be the length of all the sub vectors
final = zeros(1,numel(parseddatasorted));

% Interleave the sub vectors into the final vector
final(1:5:end) = parseddatasorted(:,1);
final(2:5:end) = parseddatasorted(:,2);
final(3:5:end) = parseddatasorted(:,3);
final(4:5:end) = parseddatasorted(:,4);
final(5:5:end) = parseddatasorted(:,5);
fprintf(fileID,'%10.8f, %10.8f, false, %10.8f, %10.8f, %10.8f\n', final)
```

```
% ParticleSystem.m as written by _ with changes implemented

classdef particle_system < handle

    properties

        gravity
        drag
        particles
        springs
        attractions
        time = 0
        graphics_handle
        parseddata

    end

    methods

        function Particle_System = particle_system (gravity, drag, limits)
            %PARTICLE_SYSTEM  Particle system object constructor.
            %
            %   Examples
            %
            %   PS = PARTICLE_SYSTEM creates particle system PS with default
properties:
            %   gravity: [0 0 0] (no gravity)
            %   drag: 0
            %   (axes) limits: 1
            %
            %   PS = PARTICLE_SYSTEM (G, D, L) creates particle system PS
with the following properties:
            %   gravity: G
            %   drag: D
            %   (axes) limits: L


            if nargin == 0

                gravity = [0, 0, 0];

                drag = 0;

%               Change to your specific aspect ratio (x,y,z)
                limits = [1.607,1,1];

            end

            Particle_System.gravity = gravity;

            Particle_System.drag = drag;

            Particle_System.graphics_handle = figure;
            set (Particle_System.graphics_handle, 'renderer', 'opengl')

            Particle_System.parseddata = [];

%              Decide if you want to adjust for screen size or hard code a
specific size
%              screen_size = get (0, 'screensize');
%              set (Particle_System.graphics_handle, 'position', screen_size +
[20 40 -40 -120]);
              set (Particle_System.graphics_handle, 'position', [20 40
(1644.5) (971.75)])

%              Set limits to 1
%              axis ([-limits limits -limits limits -limits limits]);
              axis ([-limits limits -1 1 -1 1]);

%              axis equal

            grid on

            rotate3d

        end

        function advance_time (Particle_System, step_time)
            %ADVANCE_TIME  Advance particle system time.
            %
            %   Example
            %
            %   ADVANCE_TIME (PS, ST) increments the current time property
            %   of the particle system PS for step time ST.


            Particle_System.kill_old_particles;

            time_start = Particle_System.time;
            time_end = time_start + step_time;

            phase_space_state = Particle_System.get_phase_space_state;

            phase_space_states = ode4 (...
                @compute_state_derivative, ...
                [time_start, time_end], ...
                phase_space_state, ...
                Particle_System);

            phase_space_state = phase_space_states(2,:);

            Particle_System.set_phase_space_state (phase_space_state);

            Particle_System.time = time_end;

            Particle_System.advance_particles_ages (step_time);
            Particle_System.aggregate_custom_forces;
            Particle_System.update_graphics_positions;

            function Y = ode4(odefun,tspan,y0,varargin)
                %ODE4  Solve differential equations with a non-adaptive
method of order 4.
                %   Y = ODE4(ODEFUN,TSPAN,Y0) with TSPAN = [T1, T2, T3, ...
TN] integrates
                %   the system of differential equations y' = f(t,y) by
stepping from T0 to
                %   T1 to TN. Function ODEFUN(T,Y) must return f(t,y) in a
column vector.
                %   The vector Y0 is the initial conditions at T0. Each row
in the solution
                %   array Y corresponds to a time specified in TSPAN.
                %
                %   Y = ODE4(ODEFUN,TSPAN,Y0,P1,P2...) passes the additional
parameters
                %   P1,P2... to the derivative function as
ODEFUN(T,Y,P1,P2...).
                %
                %   This is a non-adaptive solver. The step sequence is
determined by TSPAN
                %   but the derivative function ODEFUN is evaluated multiple
times per step.
                %   The solver implements the classical Runge-Kutta method of
order 4.
                %
                %   Example
                %         tspan = 0:0.1:20;
                %         y = ode4(@vdp1,tspan,[2 0]);
                %         plot(tspan,y(:,1));
                %     solves the system y' = vdp1(t,y) with a constant step
size of 0.1,
                %     and plots the first component of the solution.
                %

                if ~isnumeric(tspan)
                    error('TSPAN should be a vector of integration steps.');
                end

                if ~isnumeric(y0)
                    error('Y0 should be a vector of initial conditions.');
                end

                h = diff(tspan);
                if any(sign(h(1))*h <= 0)
                    error('Entries of TSPAN are not in order.')
                end

                try
                    f0 = feval(odefun,tspan(1),y0,varargin{:});
                catch
                    msg = ['Unable to evaluate the ODEFUN at t0,y0.
',lasterr];
                    error(msg);
                end

                y0 = y0(:);   % Make a column vector.
                if ~isequal(size(y0),size(f0))
                    error('Inconsistent sizes of Y0 and f(t0,y0).');
                end

                neq = length(y0);
                N = length(tspan);
                Y = zeros(neq,N);
                F = zeros(neq,4);

                Y(:,1) = y0;
                for i = 2:N
                    ti = tspan(i-1);
                    hi = h(i-1);
                    yi = Y(:,i-1);
                    F(:,1) = feval(odefun,ti,yi,varargin{:});
                    F(:,2) =
feval(odefun,ti+0.5*hi,yi+0.5*hi*F(:,1),varargin{:});
                    F(:,3) =
feval(odefun,ti+0.5*hi,yi+0.5*hi*F(:,2),varargin{:});
                    F(:,4) = feval(odefun,tspan(i),yi+hi*F(:,3),varargin{:});
                    Y(:,i) = yi + (hi/6)*(F(:,1) + 2*F(:,2) + 2*F(:,3) +
F(:,4));
                end
                Y = Y.';

            end

        end

        function appendfxy(Particle_System, k, psize)
             for i_particle = 1 : length (Particle_System.particles)

                Particle = Particle_System.particles(i_particle);
                Particle.fxy = [Particle.fxy;[k,
Particle.position(1,1),Particle.position(1,3)]];
             end

        end

        function setsize(Particle_System, psize)
             Particle = Particle_System.particles(i_particle);
             Particle.particlesize = psize;
        end

         function getsize(Particle_System, currentsize)
             Particle = Particle_System.particles(i_particle);
             currentsize = Particle.particlesize;
         end

        function particles_positions = get_particles_positions
(Particle_System)
```

```
            %GET_PARTICLES_POSITIONS  Retrieve particle system particles
positions.
            %
            %   Example
            %
            %   P = GET_PARTICLES_POSITIONS (PS) returns the vectors of all
            %   particles positions of the particle system PS concatenated
            %   into one long row vector P.
            %
            %   See also get_particles_velocities.


            n_particles = length (Particle_System.particles);

            particles_positions = zeros (1, 3*n_particles);

            for i_particle = 1 : n_particles

                Particle = Particle_System.particles(i_particle);

                particles_positions (3*i_particle - 2 : 3*i_particle) = ...
                    Particle.position;

            end

        end

        function particles_velocities = get_particles_velocities
(Particle_System)
            %GET_PARTICLES_VELOCITIES  Retrieve particle system particles
velocities.
            %
            %   Example
            %
            %   V = GET_PARTICLES_VELOCITIES (PS) returns the vectors of all
particles velocities
            %   of the particle system PS concatenated into one long row
vector V.
            %
            %   See also get_particle_positions.


            n_particles = length (Particle_System.particles);

            particles_velocities = zeros (1, 3*n_particles);

            for i_particle = 1 : n_particles

                Particle = Particle_System.particles(i_particle);

%                 if Particle.fixed
%
%                     velocity = [0, 0, 0];
%
```

```
%                 else
%
                    velocity = Particle.velocity;

%                 end

                particles_velocities (3*i_particle - 2 : 3*i_particle) = ...
                    velocity;

            end

        end

        function kill_attraction (Particle_System, Attraction)
            %KILL_ATTRACTION  Delete attraction from particle system.
            %
            %   Example
            %
            %   KILL_ATTRACTION (PS, A) deletes the attraction A
            %   from the particle system PS.
            %
            %   See also kill_particle, kill_spring.


            index = Particle_System.attractions == Attraction;

            Particle_System.attractions(index) = [];

            Attraction.delete;

        end

        function kill_particle (Particle_System, Particle)
            %KILL_PARTICLE  Delete particle from particle system.
            %
            %   Example
            %
            %   KILL_PARTICLE (PS, P) deletes the particle P
            %   from the particle system PS. Additionally, all
            %   attractions and springs that are connected
            %   to the deleted particle are deleted as well.
            %
            %   See also kill_attraction, kill_spring.


            % Initialize the buffer of the attractions to be killed
            attractions_to_be_killed = [];

            % Loop over all attractions
            for i_attraction = 1 : length (Particle_System.attractions)

                % Extract the current attraction
                Attraction = Particle_System.attractions(i_attraction);
                % Extract both connected particles
                Particle_1 = Attraction.particle_1;
                Particle_2 = Attraction.particle_2;

                % If the particle to be killed is one of the particles
                % at the ends of the current attraction,
                if Particle == Particle_1 || Particle == Particle_2

                    % the current attraction has to be killed too (-> later).
                    % This can't be done immediatedly
                    % (the handle has to be buffered),
                    % because we are still inside a loop over all attractions
                    attractions_to_be_killed = [attractions_to_be_killed,
Attraction];

                end

            end

            % Loop over all attractions to be killed
            for i_attraction = attractions_to_be_killed

                % Kill it
                Particle_System.kill_attraction (i_attraction);

            end

            % Initialize the buffer of the springs to be killed
            springs_to_be_killed = [];

            % Loop over all springs
            for i_spring = 1 : length (Particle_System.springs)

                % Extract the current spring
                Spring = Particle_System.springs(i_spring);

                % Extract the indices of both connected particles
                % Extract both connected particles
                Particle_1 = Spring.particle_1;
                Particle_2 = Spring.particle_2;

                % If the particle to be killed is one of the particles
                % at the ends of the current spring,
                if Particle == Particle_1 || Particle == Particle_2

                    % the current spring has to be killed too (-> later).
                    % This can't be done immediatedly
                    % (the handle has to be buffered),
                    % because we are still inside a loop over all springs
                    springs_to_be_killed = [springs_to_be_killed, Spring];

                end

            end
            % Loop over all springs to be killed
            for i_spring = springs_to_be_killed

                % Kill it
                Particle_System.kill_spring (i_spring);

            end

            % Find the reference of the particle in the particle system
            % particles property
            index = Particle_System.particles == Particle;

            % Delete the particle reference in the particle system
            % particle property
            Particle_System.particles(index) = [];

            % Delete the actual particle
            Particle.delete;

        end

        function kill_spring (Particle_System, Spring)
            %KILL_SPRING  Delete spring from particle system.
            %
            %   Example
            %
            %   KILL_SPRING (PS, S) deletes the spring S
            %   from the particle system PS.
            %
            %   See also kill_attraction, kill_particle.

            index = Particle_System.springs == Spring;

            Particle_System.springs(index) = [];

            Spring.delete;

        end

    end

    methods  (Access = 'private')

        function advance_particles_ages (Particle_System, step_time)
            %ADVANCE_PARTICLES_AGES  Increment the ages of all particles.
            %
            %   Example
            %
            %   ADVANCE_PARTICLES_AGES (PS, ST) increments the age property
            %   of all particles in the particle system PS for step time ST.
            for i_particle = 1 : length (Particle_System.particles)

                Particle = Particle_System.particles(i_particle);

                Particle.age = Particle.age + step_time;

            end

        end

        function aggregate_attractions_forces (Particle_System)
            %AGGREGATE_ATTRACTIONS_FORCES  Aggregate attraction forces.
            %
            %   Example
            %
            %   AGGREGATE_ATTRACTIONS_FORCES (PS) aggregates the forces of 
all
            %   attractions in the particle system PS on all particles they 
are connected to
            %   in the corresponding particle force accumulators.
            %
            %   See also aggregate_forces, aggregate_drag_forces, 
aggregate_gravity_forces,
            %   aggregate_spring_forces.

            for i_attraction = 1 : length (Particle_System.attractions)

                Attraction = Particle_System.attractions(i_attraction);

                Particle_1 = Attraction.particle_1;
                Particle_2 = Attraction.particle_2;

                position_delta = Particle_2.position - Particle_1.position;

                position_delta_norm = norm (position_delta);

                if position_delta_norm < Attraction.minimum_distance

                    position_delta_norm = Attraction.minimum_distance;

                end

                attraction_force = ...
                    Attraction.strength* ...
                    Particle_1.mass* ...
                    Particle_2.mass* ...
                    position_delta/ ...
                    position_delta_norm/ ...
                    position_delta_norm/ ...
                    position_delta_norm;

                Particle_1.add_force (attraction_force);
                Particle_2.add_force (-attraction_force);
            end

        end

        function aggregate_drag_forces (Particle_System)
            %AGGREGATE_DRAG_FORCES  Aggregate drag forces.
            %
            %   Example
            %
            %   AGGREGATE_DRAG_FORCES (PS) aggregates the drag forces
            %   in the particle system PS on all particles
            %   in the corresponding particle force accumulators.
            %
            %   See also aggregate_forces, aggregate_attraction_forces,
aggregate_gravity_forces,
            %   aggregate_spring_forces.

            for i_particle = 1 : length (Particle_System.particles)

                Particle = Particle_System.particles(i_particle);

                drag_force = - Particle_System.drag*Particle.velocity;

                Particle.add_force (drag_force);

            end

        end

        function aggregate_forces (Particle_System)
            %AGGREGATE_FORCES  Aggregate forces on all particles.
            %
            %   Example
            %
            %   AGGREGATE_FORCES (PS) aggregates the spring, attraction,
drag, and
            %   gravity forces in the particle system PS on all particles
            %   in the corresponding particle force accumulators.
            %
            %   See also aggregate_attraction_forces, aggregate_drag_forces,
            %   aggregate_gravity_forces, aggregate_spring_forces,
clear_particle_forces.


            Particle_System.clear_particles_forces;

            Particle_System.aggregate_springs_forces;
            Particle_System.aggregate_attractions_forces;
            Particle_System.aggregate_drag_forces;
            Particle_System.aggregate_gravity_forces;
            Particle_System.aggregate_custom_forces;

        end
        function aggregate_gravity_forces (Particle_System)
            %AGGREGATE_GRAVITY_FORCES  Aggregate gravity forces.
            %
            %   Example
            %
            %   AGGREGATE_GRAVITY_FORCES (PS) aggregates the gravity forces
            %   in the particle system PS on all particles
            %   in the corresponding particle force accumulators.
            %
            %   See also aggregate_forces, aggregate_attraction_forces,
aggregate_drag_forces,
            %   aggregate_spring_forces.

            %   Copyright 2008-2008 buchholz.hs-bremen.de

            for i_particle = 1 : length (Particle_System.particles)

                Particle = Particle_System.particles(i_particle);

                gravity_force = Particle.mass*Particle_System.gravity;

                Particle.add_force (gravity_force);

            end

        end

        function aggregate_custom_forces (Particle_System)
            %AGGREGATE_GRAVITY_FORCES  Aggregate gravity forces.
            %
            %   Example
            %
            %   AGGREGATE_GRAVITY_FORCES (PS) aggregates the gravity forces
            %   in the particle system PS on all particles
            %   in the corresponding particle force accumulators.
            %
            %   See also aggregate_forces, aggregate_attraction_forces,
aggregate_drag_forces,
            %   aggregate_spring_forces.


            %Global step delta x
            global step2
            step2 = (11/1276)/6*(1/0.033);
            global step1
            step1 = (5.5/1276)/6*(1/0.033);
            global step05
            step05 = (2.75/1276)/6*(1/0.033);
            global step01
            step01 = (0.55/1276)/6*(1/0.033);

            for i_particle = 1 : length (Particle_System.particles)

                Particle = Particle_System.particles(i_particle);
```

```
%                Comment for non-Linear motion
%                Make sure to change step value to the speed you are using
                 newposition =
[Particle.position(1,1)+step2,0,Particle.position(1,3)];

%                Uncomment for Sinusoidal motion
%                Make sure to change step value to the speed you are using
%                newposition =
[Particle.position(1,1)+step2,0,Particle.position(1,3)+0.03*sin((i_particle+1
0)*pi/10)];

%                Uncomment for Brownian motion
%                Make sure to change magnitude value to adjust diffusion rate
%                  magnitude = 25
%                  a2 = -1;
%                  b2 = 1;
%                  global r2
%                  r2 = (b2-a2).*rand(10000,1) + a2;
%                newposition =
[Particle.position(1,1)+r2(Particle.id)/magnitude,0,Particle.position(1,3)+r2
(Particle.id+1)/magnitude];


                 Particle.position = newposition;

            end

        end

        function aggregate_springs_forces (Particle_System)
            %AGGREGATE_SPRINGS_FORCES  Aggregate spring forces.
            %
            %   Example
            %
            %   AGGREGATE_SPRINGS_FORCES (PS) aggregates the forces of all
            %   springs in the particle system PS on all particles they are
connected to
            %   in the corresponding particle force accumulators.
            %
            %   See also aggregate_forces, aggregate_attraction_forces,
aggregate_drag_forces,
            %   aggregate_gravity_forces.

            for i_spring = 1 : length (Particle_System.springs)

                Spring = Particle_System.springs(i_spring);

                Particle_1 = Spring.particle_1;
                Particle_2 = Spring.particle_2;

                position_delta = Particle_2.position - Particle_1.position;

                position_delta_norm = norm (position_delta);
                % If the user makes the initial positions of two particles
identical
                % we have to avoid a "divide by zero" exception
                if position_delta_norm < eps

                    position_delta_norm = eps;

                end

                position_delta_unit = position_delta/position_delta_norm;

                spring_force =
Spring.strength*position_delta_unit*(position_delta_norm - Spring.rest);

                Particle_1.add_force (spring_force);
                Particle_2.add_force (-spring_force);

                velocity_delta = Particle_2.velocity - Particle_1.velocity;

                projection_velocity_delta_on_position_delta = ...
                    dot (position_delta_unit,
velocity_delta)*position_delta_unit;

                damping_force =
Spring.damping*projection_velocity_delta_on_position_delta;

                Particle_1.add_force (damping_force);
                Particle_2.add_force (-damping_force);

            end

        end

        function clear_particles_forces (Particle_System)
            %CLEAR_PARTICLES_FORCES  Clear all particle forces.
            %
            %   Example
            %
            %   CLEAR_PARTICLES_FORCES (PS) clears the force accumulators of
all particles
            %   in the particle system PS.
            %
            %   See also aggregate_forces.

            for i_particle = 1 : length (Particle_System.particles)

                Particle = Particle_System.particles(i_particle);

                Particle.clear_force;

            end
        end

        function state_derivative = compute_state_derivative ...
                (time, phase_space_state, Particle_System)
            %COMPUTE_STATE_DERIVATIVE  Compute state derivative vector.
            %
            %   Example
            %
            %   SD = COMPUTE_STATE_DERIVATIVE (T, SS, PS) returns the 6*N-by-
1
            %   phase space state derivative vector SD of the particle system
PS at time T.
            %   SS is the current 6*N-by-1 phase space state vector.
            %   N is the number of particles.
            %
            %   See also ode4, set_phase_space_state, aggregate_forces,
            %   get_particles_velocities, get_particles_accelerations.
            %

            phase_space_state = phase_space_state(:)';

            Particle_System.set_phase_space_state (phase_space_state);

            Particle_System.aggregate_forces;

            velocities = Particle_System.get_particles_velocities;

            accelerations = Particle_System.get_particles_accelerations;

            state_derivative = [velocities, accelerations]';

        end

        function accelerations = get_particles_accelerations
(Particle_System)
            %GET_PARTICLES_ACCELERATIONS  Retrieve particle accelerations
from particle system.
            %
            %   Example
            %
            %   A = GET_PARTICLES_ACCELERATIONS (PS) returns the 1-by-3*N
acceleration vector A
            %   of all particles from the particle system PS, where N is the
number of
            %   particles.

            n_particles = length (Particle_System.particles);

            accelerations = zeros (1, 3*n_particles);

            for i_particle = 1 : n_particles

                Particle = Particle_System.particles(i_particle);
```

```
%                 if Particle.fixed
%
%                     force = [0 0 0];
%
%                 else
%
                    force = Particle.force;

%                 end

                accelerations (3*i_particle - 2 : 3*i_particle) = ...
                    force/Particle.mass;

            end

        end

        function phase_space_state = get_phase_space_state (Particle_System)
            %GET_PHASE_SPACE_STATE  Retrieve phase space state vector.
            %
            %   Example
            %
            %   SS = GET_PHASE_SPACE_STATE (PS) returns the current 1-by-6*N
            %   phase space state vector SS from the particle system PS.
            %   N is the number of particles.
            %
            %   See also set_phase_space_state.

            positions = get_particles_positions (Particle_System);
            velocities = get_particles_velocities (Particle_System);

            phase_space_state = [positions, velocities];

        end

        function kill_old_particles (Particle_System)
            %KILL_OLD_PARTICLES  Kill old particles.
            %
            %   Example
            %
            %   KILL_OLD_PARTICLES (PS) kills all particles
            %   in the particle system PS
            %   that are older than their life span.

            % Initialize the buffer of the particles to be killed
            particles_to_be_killed = [];
            particledeathrow = [];
            parseddatatemp = [];

            for i_particle = 1 : length (Particle_System.particles)

                Particle = Particle_System.particles(i_particle);

                if Particle.age > Particle.lifespan
```

```
                    % the current particle has to be killed too (-> later).
                    % This can't be done immediatedly (the handle has to be
buffered),
                    % because we are still inside a loop over all particles
                    particles_to_be_killed = [particles_to_be_killed,
Particle];
                end

            end

            % Loop over all particles to be killed
            for i_particle = particles_to_be_killed
                tempparticledeathrow  = i_particle;
                particledeathrow = [particledeathrow;tempparticledeathrow];
            end

            % Ensure that you store the data for every particle in each frame
            global particlesperframe
            particlesperframe = 1;

            for i_particle = 1:size(particles_to_be_killed)
                for j = 1:particlesperframe
                      parseddatatemptemp =
[Particle_System.particles(i_particle+j-1).fxy(:,1),
Particle_System.particles(i_particle+j-
1).id*ones(size(Particle_System.particles(i_particle+j-1).fxy,1),1),
round(Particle_System.particles(i_particle+j-1).fxy(:,2)*638)+638, 794-
(round(Particle_System.particles(i_particle+j-1).fxy(:,3)*397)+397),
Particle_System.particles(i_particle+j-
1).particlesize(:,1)*ones(size(Particle_System.particles(i_particle+j-
1).fxy,1),1)];
                      parseddatatemp = [parseddatatemp ; parseddatatemptemp];
                end
                      Particle_System.parseddata =
[Particle_System.parseddata ; parseddatatemp];
            end

            % Kill the particle and save the data
            for i_particle = particles_to_be_killed
                Particle_System.kill_particle (i_particle);
            end

        end

        function set_phase_space_state (Particle_System, phase_space_state)
            %SET_PHASE_SPACE_STATE  Set phase space state vector.
            %
            %   Example
            %
            %   SET_PHASE_SPACE_STATE (PS, SS) sets the
            %   phase space state vector of the particle system PS to SS.
            %   SS must be a 1-by-6*N vector. N is the number of particles.
            %
            %   See also get_phase_space_state.
            n_particles = length (Particle_System.particles);

            for i_particle = 1 : n_particles

                Particle = Particle_System.particles(i_particle);

                Particle.position = ...
                    phase_space_state(3*i_particle - 2 : 3*i_particle);

                Particle.velocity = ...
                    phase_space_state(3*(i_particle + n_particles) - 2 :
3*(i_particle + n_particles));

            end

        end

        function update_graphics_positions (Particle_System)
            %UPDATE_GRAPHICS_POSITIONS  Update graphics positions.
            %
            %   Example
            %
            %   UPDATE_GRAPHICS_POSITIONS (PS) updates the graphics positions
of all
            %   particles, springs, and attractions of the particle system
PS.
            %
            %   See also get_particles_positions.

            for i_particle = 1 : length (Particle_System.particles)


Particle_System.particles(i_particle).update_graphics_position;

            end

            for i_spring = 1 : length (Particle_System.springs)

                Particle_System.springs(i_spring).update_graphics_position;

            end

            for i_attraction = 1 : length (Particle_System.attractions)


Particle_System.attractions(i_attraction).update_graphics_position;
            end
            drawnow
        end
    end
end
```

```
% Particle.m as written by _ with changes to particle property list
classdef particle < handle

    % Declare all of the properties of the particle
    properties

        mass
        position
        velocity
        newsize
        lifespan
        age = 0
        force = [0 0 0]
        graphics_handle
        id
        fxy
        particlesize

    end

    methods

        function Particle = particle (Particle_System, mass, position,
velocity, newsize, lifespan, fxy, id, size)
            %PARTICLE  Particle object constructor.
            %
            %   Examples
            %
            %   P = PARTICLE (PS) creates particle P with default
            %   properties:
            %   mass: 1
            %   position: [0 0 0]
            %   velocity: [0 0 0]
            %   newsize: false
            %   life span: inf
            %   and appends it to the particle system PS
            %
            %   P = PARTICLE (PS, M, PP, V, F, L) creates particle P with
            %   the following properties:
            %   mass: M
            %   position: PP (PP must be a 1-by-3 vector)
            %   velocity: V  (V must be a 1-by-3 vector)
            %   fixed: F (F must be false (or 0) for a free or true (or 1)
            %   for a fixed particle)
            %   life span: L
            %   and appends it to the particle system PS

            if nargin == 1

                mass = 1;

                position = [0, 0, 0];

                velocity = [0, 0, 0];
                newsize = false;

                lifespan = inf;

                fxy = [];

                id = 1;

                particlesize = 0;


            end

            Particle.mass = mass;

            Particle.position = position;

            Particle.velocity = velocity;

            Particle.newsize = newsize;

            Particle.lifespan = lifespan;

            Particle.id = id;

            Particle.fxy = [Particle.fxy;fxy];

            Particle.append (Particle_System);

            Particle.particlesize = size;

        end

        function clear_force (Particle)
            %CLEAR_FORCE  Clear particle force accumulator.
            %   Example
            %   CLEAR_FORCE (P) clears the force accumulator of the particle
P.
            %   See also add_force.
            Particle.force = [0 0 0];

        end

        function add_force (Particle, force)
            %ADD_FORCE  Add force to particle force accumulator.
            %   Example
            %   ADD_FORCE (P, F) adds the force F to the force accumulator of
the
            %   particle P.
            %   See also clear_force.
            Particle.force = Particle.force + force;

        end
        function delete (Particle)
            %DELETE  Delete particle.
            %   Example
            %   DELETE (P) deletes the particle P.
            if ishandle (Particle.graphics_handle)
                delete (Particle.graphics_handle)
            end
        end

        function update_graphics_position (Particle)
            %UPDATE_GRAPHICS_POSITION  Update the graphical particle
representation.
            %   Example
            %   UPDATE_GRAPHICS_POSITION (P) updates the position of the
            %   graphical representation of the particle P.

            set (Particle.graphics_handle, ...
                'xdata', Particle.position(1), ...
                'ydata', Particle.position(2), ...
                'zdata', Particle.position(3));
        end

        function set.newsize (Particle, newsize)
            %SET.NEWSIZE  Make a new sized particle
            %   Examples
            %   P.FIXED = TRUE pins the new sized particle
            %   P.FIXED = FALSE keeps the old sized particle

            Particle.newsize = newsize;

            if Particle.newsize
                % Set the size of each particle in diameter
                set (Particle.graphics_handle, 'markersize', 120, 'marker',
'.')
            else
                set (Particle.graphics_handle, 'markersize', 60, 'marker',
'.')
            end
        end

        function set.position (Particle, position)
            %SET.POSITION  Set the position of a particle.
            %
            %   Examples
            %
            %   P.POSITION = NEW_POSITION sets the current position of the
            %   particle P to NEW_POSITION.

            Particle.position = position;

            Particle.update_graphics_position;

        end
```

```
    end

    methods (Access = 'private')

        function append (Particle, Particle_System)
            %APPEND  Append particle to particle system.
            %
            %   Example
            %
            %   APPEND (P, PS) appends the particle P to the
            %   particle system PS.

            figure (Particle_System.graphics_handle)

%           Change particle size in pixels for the diameter of single sized
particles
            Particle.graphics_handle = ...
                line ( ...
                Particle.position(1), ...
                Particle.position(2), ...
                Particle.position(3), ...
                'color', [1 0 0], ...
                'markersize', 120, ...
                'marker', '.');

%           Change particle size in pixels for the diameter for two sizes of
particles
            if Particle.newsize
                set (Particle.graphics_handle, 'markersize', 120, 'marker',
'.')
            else
                set (Particle.graphics_handle, 'markersize', 60, 'marker',
'.')
            end

            Particle_System.particles = [Particle_System.particles,
Particle];
        end

    end

end
```

BIBLIOGRAPHY

BIBLIOGRAPHY

[1] C. Y. Li, K. R. Stevens, R. E. Schwartz, B. S. Alejandro, J. H. Huang, and S. N. Bhatia, "Micropatterned cell-cell interactions enable functional encapsulation of primary hepatocytes in hydrogel microtissues.," *Tissue Eng. Part A*, vol. 20, no. 15–16, pp. 2200–12, Aug. 2014.

[2] Y. Xia and G. M. Whitesides, "Soft Lithography," *Angew. Chemie Int. Ed.*, vol. 37, no. 5, pp. 550–575, Mar. 1998.

[3] Y. Xia, "Soft Lithography and the Art of Patterning— A Tribute to Professor George M. Whitesides," *Adv. Mater.*, vol. 16, no. 15, pp. 1245–1246, Aug. 2004.

[4] S. N. Bhatia and D. E. Ingber, "Microfluidic organs-on-chips.," *Nat. Biotechnol.*, vol. 32, no. 8, pp. 760–772, Aug. 2014.

[5] B. K. Chen, "Autonomous Robotic Pick-and-Place of Microobjects," *IEEE Trans. Robot.*, vol. 26, no. 1, pp. 200–207, Feb. 2010.

[6] P. Weaver, "The technique of lithography." Batsford, [London], 1964.

[7] A. K.-K. Wong, *Optical Imaging in Projection Microlithography*. SPIE Press, 2005, p. 254.

[8] J. C. Love, D. B. Wolfe, H. O. Jacobs, and G. M. Whitesides, "Microscope Projection Photolithography for Rapid Prototyping of Masters with Micron-Scale Features for Use in Soft Lithography," *Langmuir*, vol. 17, no. 19, pp. 6005–6012, Sep. 2001.

[9] L. J. Hornbeck and W. E. Nelson, "Bistable Deformable Mirror Device," *OSA Tech. Dig. Ser.*, vol. 8, no. Spatial Light Modulators and Applications, p. 107, 1988.

[10] I. W. Mei, "(12) United States Patent," vol. 1, no. 12, 2002.

[11] C. J. Murphy, T. K. Sau, A. M. Gole, C. J. Orendorff, J. Gao, L. Gou, S. E. Hunyadi, and T. Li, "Anisotropic metal nanoparticles: Synthesis, assembly, and optical applications.," *J. Phys. Chem. B*, vol. 109, no. 29, pp. 13857–70, Jul. 2005.

[12] G. M. Schmid, M. Miller, C. Brooks, N. Khusnatdinov, D. LaBrake, D. J. Resnick, S. V. Sreenivasan, G. Gauzner, K. Lee, D. Kuo, D. Weller, and X. Yang, "Step and flash imprint lithography for manufacturing patterned media," *J. Vac. Sci. Technol. B Microelectron. Nanom. Struct.*, vol. 27, no. 2, p. 573, 2009.

[13] J. A. Liddle and G. M. Gallatin, "Lithography, metrology and nanomanufacturing.," *Nanoscale*, vol. 3, no. 7, pp. 2679–88, Jun. 2011.

[14] M. Han, "3D microfabrication with inclined/rotated UV lithography," *Sensors Actuators A Phys.*, vol. 111, no. 1, pp. 14–20, Mar. 2004.

[15] C. N. LaFratta, J. T. Fourkas, T. Baldacchini, and R. a Farrer, "Multiphoton fabrication.," *Angew. Chem. Int. Ed. Engl.*, vol. 46, no. 33, pp. 6238–58, Jan. 2007.

[16] H. Murat, A. Avci, R. Beernaert, K. Dhaenens, H. De Smet, L. Bogaert, Y. Meuret, and H. Thienpont, "Two LCOS full color projector with efficient LED illumination engine," *Displays*, vol. 30, no. 4–5, pp. 155–163, Dec. 2009.

[17] H. Schmid, "Light-coupling masks: An alternative, lensless approach to high-resolution optical contact lithography," *J. Vac. Sci. Technol. B Microelectron. Nanom. Struct.*, vol. 16, no. 6, p. 3422, Nov. 1998.

[18] S. E. Chung, W. Park, H. Park, K. Yu, N. Park, and S. Kwon, "Optofluidic maskless lithography system for real-time synthesis of photopolymerized microstructures in microfluidic channels," *Appl. Phys. Lett.*, vol. 91, no. 4, p. 041106, 2007.

[19] J. Geary, *Introduction to lens design : with practical ZEMAX examples*. Richmond Va.: Willmann-Bell, 2002.

[20] Newport, "Focusing and collimating," 2014. [Online]. Available: http://www.newport.com/Focusing-and-Collimating/141191/1033/content.aspx.

[21] Y. K. Rabinovitz, "Digital Light Processing Technology (DLP) Beyond any conventional projection," 2014. [Online]. Available: http://www.opli.net/magazine/eo/2011/news/dlp_tech.aspx. [Accessed: 11-Aug-2014].

[22] E. S. Polsen, "Robust synthesis and contiuous manufacturing of carbon nanotube forests and graphene films," 2013.

[23] J. G. Ok, M. K. Kwak, C. M. Huard, H. S. Youn, and L. J. Guo, "Photo-roll lithography (PRL) for continuous and scalable patterning with application in flexible electronics.," *Adv. Mater.*, vol. 25, no. 45, pp. 6554–61, Dec. 2013.

[24] M. E. Helgeson, S. C. Chapin, and P. S. Doyle, "Hydrogel microparticles from lithographic processes: novel materials for fundamental and applied colloid science.," *Curr. Opin. Colloid Interface Sci.*, vol. 16, no. 2, pp. 106–117, Apr. 2011.

[25] J. a Champion, Y. K. Katare, and S. Mitragotri, "Particle shape: a new design parameter for micro- and nanoscale drug delivery carriers.," *J. Control. Release*, vol. 121, no. 1–2, pp. 3–9, Aug. 2007.

[26] S. C. Glotzer, M. J. Solomon, and N. a. Kotov, "Self-assembly: From nanoscale to microscale colloids," *AIChE J.*, vol. 50, no. 12, pp. 2978–2985, Dec. 2004.

[27] C. J. Hernandez and T. G. Mason, “Colloidal Alphabet Soup: Monodisperse Dispersions of Shape-Designed LithoParticles,” *J. Phys. Chem. C*, vol. 111, no. 12, pp. 4477–4480, Mar. 2007.

[28] C. a. Serra and Z. Chang, “Microfluidic-Assisted Synthesis of Polymer Particles,” *Chem. Eng. Technol.*, vol. 31, no. 8, pp. 1099–1115, Aug. 2008.

[29] K. Roh, D. C. Martin, and J. Lahann, “Biphasic Janus particles with nanoscale anisotropy.,” *Nat. Mater.*, vol. 4, no. 10, pp. 759–63, Oct. 2005.

[30] S. Bhaskar, J. Hitt, S.-W. L. Chang, and J. Lahann, “Multicompartmental microcylinders.,” *Angew. Chem. Int. Ed. Engl.*, vol. 48, no. 25, pp. 4589–93, Jan. 2009.

[31] R. F. Shepherd, P. Panda, Z. Bao, K. H. Sandhage, T. A. Hatton, J. a. Lewis, and P. S. Doyle, “Stop-Flow Lithography of Colloidal, Glass, and Silicon Microcomponents,” *Adv. Mater.*, vol. 20, no. 24, pp. 4734–4739, Dec. 2008.

[32] D. Dendukuri, S. S. Gu, D. C. Pregibon, T. A. Hatton, and P. S. Doyle, “Stop-flow lithography in a microfluidic device.,” *Lab Chip*, vol. 7, no. 7, pp. 818–28, Jul. 2007.

[33] P. Panda, S. Ali, E. Lo, B. G. Chung, T. A. Hatton, A. Khademhosseini, and P. S. Doyle, “Stop-flow lithography to generate cell-laden microgel particles.,” *Lab Chip*, vol. 8, no. 7, pp. 1056–61, Jul. 2008.

[34] S. C. Chapin, D. C. Pregibon, and P. S. Doyle, “High-throughput flow alignment of barcoded hydrogel microparticles.,” *Lab Chip*, vol. 9, no. 21, pp. 3100–9, Nov. 2009.

[35] D. Dendukuri, P. Panda, R. Haghgooie, J. M. Kim, T. A. Hatton, and P. S. Doyle, “Modeling of Oxygen-Inhibited Free Radical Photopolymerization in a PDMS Microfluidic Device,” *Macromolecules*, vol. 41, no. 22, pp. 8547–8556, Nov. 2008.

[36] K. W. Bong, K. T. Bong, D. C. Pregibon, and P. S. Doyle, “Hydrodynamic focusing lithography.,” *Angew. Chem. Int. Ed. Engl.*, vol. 49, no. 1, pp. 87–90, Jan. 2010.

[37] S. K. Suh, K. W. Bong, T. A. Hatton, and P. S. Doyle, “Using stop-flow lithography to produce opaque microparticles: synthesis and modeling.,” *Langmuir*, vol. 27, no. 22, pp. 13813–9, Nov. 2011.

[38] S. A. Lee, S. E. Chung, W. Park, S. H. Lee, and S. Kwon, “Three-dimensional fabrication of heterogeneous microstructures using soft membrane deformation and optofluidic maskless lithography.,” *Lab Chip*, vol. 9, no. 12, pp. 1670–5, Jun. 2009.

[39] J. Kim, H. Lee, and S. Kwon, “Micro-pixellated Blue / Brown Color Changing Surface using Magnetically Actuated Nanocomposite Actuators,” *Electr. Eng.*, vol. 1, 2011.

[40] H. Lee, J. Kim, H. Kim, J. Kim, and S. Kwon, “Colour-barcoded magnetic microparticles for multiplexed bioassays.,” *Nat. Mater.*, vol. 9, no. 9, pp. 745–9, Sep. 2010.

[41] T. J. Merkel, K. P. Herlihy, J. Nunes, R. M. Orgel, J. P. Rolland, and J. M. Desimone, “Scalable, shape-specific, top-down fabrication methods for the synthesis of engineered colloidal particles.,” *Langmuir*, vol. 26, no. 16, pp. 13086–96, Aug. 2010.

[42] J. P. Rolland, B. W. Maynor, L. E. Euliss, A. E. Exner, G. M. Denison, and J. M. DeSimone, “Direct fabrication and harvesting of monodisperse, shape-specific nanobiomaterials.,” *J. Am. Chem. Soc.*, vol. 127, no. 28, pp. 10096–100, Jul. 2005.

[43] P. W. K. Rothemund, “Folding DNA to create nanoscale shapes and patterns.,” *Nature*, vol. 440, no. 7082, pp. 297–302, Mar. 2006.

[44] P. W. K. Rothemund, N. Papadakis, and E. Winfree, “Algorithmic self-assembly of DNA Sierpinski triangles.,” *PLoS Biol.*, vol. 2, no. 12, pp. 2041–2053, Dec. 2004.

[45] R. J. Kershner, L. D. Bozano, C. M. Micheel, A. M. Hung, A. R. Fornof, J. N. Cha, C. T. Rettner, M. Bersani, J. Frommer, P. W. K. Rothemund, and G. M. Wallraff, “Placement and orientation of individual DNA shapes on lithographically patterned surfaces.,” *Nat. Nanotechnol.*, vol. 4, no. 9, pp. 557–61, Sep. 2009.

[46] C. Lin, Y. Liu, S. Rinker, and H. Yan, “DNA tile based self-assembly: building complex nanoarchitectures.,” *Chemphyschem*, vol. 7, no. 8, pp. 1641–7, Aug. 2006.

[47] A. Perro, S. Reculusa, E. Bourgeat-Lami, E. Duguet, and S. Ravaine, “Synthesis of hybrid colloidal particles: From snowman-like to raspberry-like morphologies,” *Colloids Surfaces A Physicochem. Eng. Asp.*, vol. 284–285, pp. 78–83, Aug. 2006.

[48] W. Engl, R. Backov, and P. Panizza, “Controlled production of emulsions and particles by milli- and microfluidic techniques,” *Curr. Opin. Colloid Interface Sci.*, vol. 13, no. 4, pp. 206–216, Aug. 2008.

[49] W. Wang, R. Xie, X.-J. Ju, T. Luo, L. Liu, D. a Weitz, and L.-Y. Chu, “Controllable microfluidic production of multicomponent multiple emulsions.,” *Lab Chip*, vol. 11, no. 9, pp. 1587–92, May 2011.

[50] E. B. Mock, H. De Bruyn, B. S. Hawkett, R. G. Gilbert, and C. F. Zukoski, “Synthesis of anisotropic nanoparticles by seeded emulsion polymerization.,” *Langmuir*, vol. 22, no. 9, pp. 4037–43, Apr. 2006.

[51] D. Dendukuri, T. A. Hatton, and P. S. Doyle, “Synthesis and self-assembly of amphiphilic polymeric microparticles.,” *Langmuir*, vol. 23, no. 8, pp. 4669–74, Apr. 2007.

[52] Y. Xia and G. M. Whitesides, “Soft Lithography,” *Angew. Chemie Int. Ed.*, vol. 37, no. 5, pp. 550–575, Mar. 1998.

[53] D. La Vega, "Uniform polymer microspheres : monodispersity criteria , methods of formation and applications R eview," vol. 8, pp. 265–285, 2013.

[54] D. K. Hwang, D. Dendukuri, and P. S. Doyle, "Microfluidic-based synthesis of non-spherical magnetic hydrogel microparticles.," *Lab Chip*, vol. 8, no. 10, pp. 1640–7, Oct. 2008.

[55] R. R. a. Syms, E. M. Yeatman, V. M. Bright, and G. M. Whitesides, "Surface tension-powered self-assembly of microstructures - The state-of-the-art," *J. Microelectromechanical Syst.*, vol. 12, no. 4, pp. 387–417, Aug. 2003.

[56] N. Bowden, I. S. Choi, B. A. Grzybowski, and G. M. Whitesides, "Mesoscale Self-Assembly of Hexagonal Plates Using Lateral Capillary Forces : Synthesis Using the ' Capillary Bond ,'" *Society*, no. 18, pp. 5373–5391, 1999.

[57] N. Chronis, "Worm chips: microtools for C. elegans biology.," *Lab Chip*, vol. 10, no. 4, pp. 432–7, Feb. 2010.

[58] S. Park, H. Hwang, S.-W. Nam, F. Martinez, R. H. Austin, and W. S. Ryu, "Enhanced Caenorhabditis elegans locomotion in a structured microfluidic environment.," *PLoS One*, vol. 3, no. 6, p. e2550, Jan. 2008.

[59] J. Qin and A. R. Wheeler, "Maze exploration and learning in C. elegans.," *Lab Chip*, vol. 7, no. 2, pp. 186–92, Feb. 2007.

[60] D. M. Eisenmann, "WormBook," *WormBook*, pp. 1–17, Jan. 2005.

[61] A. K. Au, W. Lee, and A. Folch, "Mail-order microfluidics: evaluation of stereolithography for the production of microfluidic devices.," *Lab Chip*, vol. 14, no. 7, pp. 1294–301, Apr. 2014.

[62] S. Brenner, "The genetics of Caenorhabditis elegans.," *Genetics*, vol. 77, no. 1, pp. 71–94, May 1974.

[63] S. R. Lockery, K. J. Lawton, J. C. Doll, S. Faumont, S. M. Coulthard, T. R. Thiele, N. Chronis, K. E. McCormick, M. B. Goodman, and B. L. Pruitt, "Artificial dirt: microfluidic substrates for nematode neurobiology and behavior.," *J. Neurophysiol.*, vol. 99, no. 6, pp. 3136–43, Jun. 2008.

[64] T. Majmudar, E. E. Keaveny, J. Zhang, and M. J. Shelley, "Experiments and theory of undulatory locomotion in a simple structured medium.," *J. R. Soc. Interface*, vol. 9, no. 73, pp. 1809–23, Aug. 2012.

[65] S. Pandey, A. Joseph, R. Lycke, and A. Parashar, "Decision-making by nematodes in complex microfluidic mazes," *Adv. Biosci. Biotechnol.*, vol. 02, no. 06, pp. 409–415, Dec. 2011.

[66] S. E. Hulme, S. S. Shevkoplyas, A. P. McGuigan, J. Apfeld, W. Fontana, and G. M. Whitesides, "Lifespan-on-a-chip: microfluidic chambers for performing lifelong observation of C. elegans.," *Lab Chip*, vol. 10, no. 5, pp. 589–97, Mar. 2010.

[67] K. E. McCormick, B. E. Gaertner, M. Sottile, P. C. Phillips, and S. R. Lockery, "Microfluidic devices for analysis of spatial orientation behaviors in semi-restrained Caenorhabditis elegans.," *PLoS One*, vol. 6, no. 10, p. e25710, Jan. 2011.

[68] J. Krajniak and H. Lu, "Long-term high-resolution imaging and culture of C. elegans in chip-gel hybrid microfluidic device for developmental studies.," *Lab Chip*, vol. 10, no. 14, pp. 1862–8, Jul. 2010.

[69] J. R. Goldsmith and C. Jobin, "Think small: zebrafish as a model system of human pathology.," *J. Biomed. Biotechnol.*, vol. 2012, p. 817341, Jan. 2012.

[70] A. Stollewerk, "The water flea Daphnia--a 'new' model system for ecology and evolution?," *J. Biol.*, vol. 9, no. 2, p. 21, Jan. 2010.

[71] J. Jang, S. Han, S. E. Chung, Y. J. Choi, and S. Kwon, "Development of conformal phosphor coating technique for light-emitting diodes using image-processing-based maskless lithography," *Microelectron. Eng.*, vol. 118, pp. 11–14, Apr. 2014.

[72] S. E. Chung, S. A. Lee, J. Kim, and S. Kwon, "Optofluidic encapsulation and manipulation of silicon microchips using image processing based optofluidic maskless lithography and railed microfluidics.," *Lab Chip*, vol. 9, no. 19, pp. 2845–50, Oct. 2009.

[73] K. R. Spring and M. W. Davidson, "Nikon University." [Online]. Available: http://www.microscopyu.com/articles/formulas/formulasfielddepth.html.

[74] D. Ballard, *Computer vision*. Englewood Cliffs N.J.: Prentice-Hall, 1982.

[75] F. Chaumette, "Image Moments: A General and Useful Set of Features for Visual Servoing," *IEEE Trans. Robot.*, vol. 20, no. 4, pp. 713–723, Aug. 2004.

[76] M.-K. Hu, "Pattern recognition by moment invariants," *Proc. IRE(Correspondence)*, vol. 49, p. 1428, 1961.

[77] R. Faragher, "Understanding the Basis of the Kalman Filter Via a Simple and Intuitive Derivation IMPORTANT AND COMMON DATA FUSION ALGORITHMS," no. September, pp. 128–132, 2012.

[78] H. Noda, J. Spaulding, M. N. Shirazi, and E. Kawaguchi, "Application of bit-plane decomposition steganography to JPEG2000 encoded images," *IEEE Signal Process. Lett.*, vol. 9, no. 12, pp. 410–413, Dec. 2002.

[79] W. Wolf, B. Ozer, and T. Lv, “Smart cameras as embedded systems,” *Computer (Long. Beach. Calif).*, vol. 35, no. 9, pp. 48–53, Sep. 2002.

[80] H. W. Sorenson, “Least-squares estimation: from Gauss to Kalman,” *IEEE Spectr.*, vol. 7, no. 7, pp. 63–68, Jul. 1970.

[81] G. Welch and G. Bishop, “An Introduction to the Kalman Filter,” pp. 1–16, 2006.

[82] O. Hugues-salas and K. A. Shore, “An Extended Kalman Filtering Approach to Nonlinear Time-Delay Systems : Application to Chaotic Secure Communications,” vol. 57, no. 9, pp. 2520–2530, 2010.

[83] M. Vose, “The simple genetic algorithm foundations and theory.” MIT Press,, Cambridge, Mass. :, 1999.

[84] H. Lorenz, M. Despont, N. Fahrni, N. LaBianca, P. Renaud, and P. Vettiger, “SU-8: a low-cost negative resist for MEMS,” *J. Micromechanics Microengineering*, vol. 7, no. 3, pp. 121–124, Sep. 1997.

[85] C. Lin, G. Lee, B. Chang, and G. Chang, “A new fabrication process for ultra-thick microfluidic microstructures utilizing SU-8 photoresist,” *Methods*, vol. 12, pp. 590–597, 2002.

[86] W. Hopp, *Factory physics : foundations of manufacturing management*. Boston: Irwin/McGraw-Hill, 2001.

[87] D. E. Knuth, “Big Omicron and big Omega and big Theta,” *ACM SIGACT News*, vol. 8, no. 2, pp. 18–24, Apr. 1976.

[88] N. Chenouard, I. Smal, F. de Chaumont, M. Maška, I. F. Sbalzarini, Y. Gong, J. Cardinale, C. Carthel, S. Coraluppi, M. Winter, A. R. Cohen, W. J. Godinez, K. Rohr, Y. Kalaidzidis, L. Liang, J. Duncan, H. Shen, Y. Xu, K. E. G. Magnusson, J. Jaldén, H. M. Blau, P. Paul-Gilloteaux, P. Roudot, C. Kervrann, F. Waharte, J.-Y. Tinevez, S. L. Shorte, J. Willemse, K. Celler, G. P. van Wezel, H.-W. Dan, Y.-S. Tsai, C. Ortiz de Solórzano, J.-C. Olivo-Marin, and E. Meijering, “Objective comparison of particle tracking methods.,” *Nat. Methods*, vol. 11, no. 3, pp. 281–9, Mar. 2014.

[89] M. Particles and J. Buchholz, “Matlab Particles 2.1,” 2009.

[90] H. Bay, A. Ess, T. Tuytelaars, and L. Van Gool, “Speeded-Up Robust Features (SURF),” *Comput. Vis. Image Underst.*, vol. 110, no. 3, pp. 346–359, Jun. 2008.

[91] P. Viola and J. Michael, “Robust Real-time Object Detection,” *Work. Stat. Comput. Theor. Vis.*, vol. 266, 2001.

[92] A. Neubeck and L. Van Gool, “Efficient Non-Maximum Suppression,” in *18th International Conference on Pattern Recognition (ICPR’06)*, 2006, pp. 850–855.

[93] D. H. Ballard, “Generalizing the Hough transform to detect arbitrary shapes,” *Pattern Recognit.*, vol. 13, no. 2, pp. 111–122, Jan. 1981.

[94] G. Bradski and A. Kaehler, *Learning OpenCV*, 1st ed. Cambridge: O’Reilly Media Inc.

[95] A. A. M. Al-kubati, J. A. M. Saif, and M. A. A. Taher, “Evaluation of Canny and Otsu Image Segmentation,” pp. 24–26, 2012.

[96] S. Arya, D. M. Mount, N. S. Netanyahu, and A. Y. Wu, “An Optimal Algorithm for Approximate Nearest Neighbor Searching in Fixed Dimensions,” vol. 1, no. 212, 1994.

[97] S. N. Bhatia, G. H. Underhill, K. S. Zaret, and I. J. Fox, “Cell and tissue engineering for liver disease.,” *Sci. Transl. Med.*, vol. 6, no. 245, p. 245sr2, Jul. 2014.

[98] T. I. Ergenç and S. Kızılel, “Recent Advances in the Modeling of PEG Hydrogel Membranes for Biomedical Applications,” 2007.

[99] D. Sabiston, *Sabiston’s essentials of surgery*. Philadelphia ; Toronto: W. B. Saunders, 1987.

[100] S. K. Sahoo, A. K. Panda, and V. Labhasetwar, “Characterization of porous PLGA/PLA microparticles as a scaffold for three dimensional growth of breast cancer cells.,” *Biomacromolecules*, vol. 6, no. 2, pp. 1132–9, 2005.

[101] K. R. Stevens, M. D. Ungrin, R. E. Schwartz, S. Ng, B. Carvalho, K. S. Christine, R. R. Chaturvedi, C. Y. Li, P. W. Zandstra, C. S. Chen, and S. N. Bhatia, “InVERT molding for scalable control of tissue microarchitecture.,” *Nat. Commun.*, vol. 4, no. May, p. 1847, Jan. 2013.

[102] J. Kim, M. Dadsetan, S. Ameenuddin, A. J. Windebank, M. J. Yaszemski, and L. Lu, “In vivo biodegradation and biocompatibility of PEG/sebacic acid-based hydrogels using a cage implant system.,” *J. Biomed. Mater. Res. A*, vol. 95, no. 1, pp. 191–7, Oct. 2010.

[103] R. Section, “Adhesive Bonding of Medical and Implantable Devices – A Review of Materials , Processes and Applications Adhesive Bonding of Medical and Implantable Devices,” vol. 3, no. April 1995, pp. 7–11, 2002.

[104] G. Moras, L. Ciacchi, C. Elsässer, P. Gumbsch, and A. De Vita, “Atomically Smooth Stress-Corrosion Cleavage of a Hydrogen-Implanted Crystal,” *Phys. Rev. Lett.*, vol. 105, no. 7, pp. 1–4, Aug. 2010.

[105] H.-J. Chung, M. S. Sulkin, J.-S. Kim, C. Goudeseune, H.-Y. Chao, J. W. Song, S. Y. Yang, Y.-Y. Hsu, R. Ghaffari, I. R. Efimov, and J. a Rogers, “Stretchable, multiplexed pH

sensors with demonstrations on rabbit and human hearts undergoing ischemia.," *Adv. Healthc. Mater.*, vol. 3, no. 1, pp. 59–68, Jan. 2014.

[106] Y. L. Kong, I. a Tamargo, H. Kim, B. Johnson, M. K. Gupta, T.-W. Koh, H.-A. Chin, D. a Steingart, B. P. Rand, and M. C. McAlpine, "3D Printed Quantum Dot Light-Emitting Diodes.," *Nano Lett.*, Oct. 2014.

[107] A. Technologies, "Spherical Surface Lithography Solutions," 2012. [Online]. Available: http://www.aiscenttech.com/en/solutions/sphericallithography/.

[108] T. Treib and E. Matthias, "Error Budgeting — Applied to the Calculation and Optimization of the Volumetric Error Field of Multiaxis Systems," *CIRP Ann. - Manuf. Technol.*, vol. 36, no. 1, pp. 365–368, Jan. 1987.

[109] A. Slocum, "FUN da MENTALS of Design Error Budgets," 2012.