Optimal Control of Acoustic Levitation Displays

Bridging the Gap Between Physical and Virtual Worlds

Von der Universität Bayreuth
zur Erlangung des Grades eines
Doktors der Naturwissenschaften (Dr. rer. nat.)
genehmigte Abhandlung

von
Viktorija Paneva
aus Štip

1. Gutachter: Prof. Dr. Jörg Müller
2. Gutachter: Prof. Dr. Sriram Subramanian
3. Gutachter: Prof. Dr. Stephen Brewster

Tag der Einreichung: 10. August 2023
Tag des Kolloquiums: 11. Dezember 2023

To my parents.

Посветено на моите родители.

Abstract

Acoustic levitation displays use focused ultrasound waves to trap small particles in mid-air. This allows for the creation of versatile physical interfaces, that present visual content, made of levitated matter, in real physical space. The content can be seen from many angles, and contrast, illumination, and depth cues are all correct. In this manner, users are able to employ their innate skills for perceiving and interacting with physical objects in the real world, potentially making the interaction with the display natural and intuitive. Additionally, wave modulation techniques enable the generation of perceivable haptic points in mid-air, further enhancing the multimodal output. However, to develop the full potential of this novel interface, we need algorithms that enable optimal control of the dynamics of the physical trap-particle system, such that the levitated matter moves smoothly, accurately traces the predefined paths, and responds to user input in an agile and robust manner.

In this dissertation, I present several concepts and their implementations aimed at increasing the performance, versatility, and usability of acoustic levitation displays. I demonstrate an interactive levitated cursor that smoothly moves in physical 3D space. In addition, I investigate the optimal spatial and temporal resolution parameters, that result in accurate perception of multi-point ultrasonic mid-air haptic feedback. I derive a semi-analytical mathematical model of the movement dynamics of a levitated particle in an acoustic field. Using this model, I create an interactive simulation in Virtual Reality that exhibits the dynamic properties of the real interface, i.e., a virtual twin of the physical levitator, for the purpose of rapid prototyping and early-stage user interface evaluations. Finally, I introduce a powerful numerical approach that computes physically feasible trap trajectories that enable the rendering of generic mid-air shapes, given only a reference path with no timing information. This allows for the generation of bigger and more complex levitated graphics than previously possible. The individual implementations have been evaluated and validated in multiple user studies and performance tests.

The contributions of this dissertation in advancing acoustic levitation displays open up possibilities for the development of transformative interfaces that integrate the physical and virtual worlds, offering exciting potential for immersive experiences in the future.

Zusammenfassung

Akustische Levitationsdisplays erzeugen durch fokussierte Ultraschallwellen akustische Fallen, die kleine Partikel in der Luft schweben lassen. Damit lassen sich Inhalte im realen physischen Raum darstellen, die aus vielen Blickwinkeln betrachtet werden können. Darüberhinaus sind Kontrast, Beleuchtung und Tiefenhinweise naturgetreu. Dies eröffnet vielseitige physische Benutzerschnittstellen und ermöglicht den Nutzenden, ihre angeborenen Fähigkeiten zur Wahrnehmung und Interaktion mit realen Objekten einzusetzen, um mit dem Display natürlich und intuitiv zu interagieren. Des Weiteren ermöglichen Wellenmodulationstechniken die Erzeugung wahrnehmbarer haptischer Punkte in der Luft und auf der Haut, was die multimodale Ausgabe verbessert. Um jedoch das volle Potenzial auszuschöpfen, werden Algorithmen benötigt, die es ermöglichen, die Dynamik zwischen den erzeugten akustischen Fallen und dem schwebenden Partikel optimal zu steuern, so dass sich das schwebende Partikel schnell und gleichmäßig bewegt, die vordefinierten Pfade genau abfährt und auf Benutzereingaben agil und robust reagiert.

In dieser Dissertation stelle ich mehrere Konzepte und deren Implementierungen vor, die darauf abzielen, die Leistung, Vielseitigkeit und Bedienbarkeit von akustischen Levitationsdisplays zu verbessern. Ich präsentiere einen hochgradig interaktiven schwebenden Cursor, der sich reibungslos im physischen 3D-Raum bewegt. Darüber hinaus untersuche ich die optimalen räumlichen und zeitlichen Auflösungsparameter, die zu einer präzisen Wahrnehmung von haptischem Mehrpunkt-Ultraschall-Feedback in der Luft führen. Ich leite ein semi-analytisches mathematisches Modell der Bewegungsdynamik eines levitierenden Partikels in einem akustischen Feld her. Anschließend verwende ich dieses Modell, um eine interaktive Simulation in der virtuellen Realität zu erstellen, die die dynamischen Eigenschaften der realen Schnittstelle virtuell widerspiegelt – ein sog. virtueller Zwilling des physischen Levitationsdisplays. Dieser erleichtert Rapid Prototyping und die Evaluierung neuer Benutzerschnittstellen in den frühen Phasen des iterativen Designprozesses. Schließlich stelle ich einen leistungsstarken numerischen Ansatz vor, der auf der optimalen Steuerung der Dynamik zwischen den erzeugten akustischen Fallen und dem schwebenden Partikel beruht. Dieser errechnet eine auf dem Levitationsdisplay realisierbare optimale Folge von akustische Fallen, so dass als Eingangsdaten der Referenzpfad ohne zeitliche Informationen ausreicht,

um beliebige Formen darzustellen. Auf diese Weise können größere und komplexere Grafiken in der Luft erzeugt werden, als dies bisher möglich war. Die einzelnen Implementierungen wurden in mehreren Benutzerstudien und Performancetests evaluiert und validiert.

Die Beiträge dieser Dissertation tragen dazu bei, akustische Levitationsdisplays weiterzuentwickeln und eröffnen somit Möglichkeiten für transformative Schnittstellen, die eine Integration der physischen und virtuellen Welt ermöglichen und spannende Potenziale für immersive Erlebnisse in der Zukunft bieten.

Acknowledgements

First and foremost, I am deeply grateful to my advisor **Prof. Jörg Müller** for his invaluable support and guidance throughout my research journey. Thank you for introducing me to the world of research, for helping me recognise my potential, and for providing me with unwavering guidance and support while also granting me the freedom to unfold at my own pace. Jörg, I am truly grateful to you for teaching me how to do good science.

I would also like to express my sincere appreciation to **Prof. Sriram Subramanian** for their exceptional expertise and wisdom in providing valuable insights and feedback to my work. Their dedication and passion for knowledge have inspired me to push my boundaries and strive for greater heights.

Additionally, I am grateful to **Prof. Stephen Brewster** for their guidance and mentorship, setting an exemplary standard of excellence that I aspire to follow in my future endeavours.

I must also extend my heartfelt gratitude to **Prof. Diego Martinez Plasencia** for their invaluable insight, advice, and mentorship. Diego, your contagious enthusiasm has inspired me to approach my research with renewed vigor and excitement.

I am equally grateful to **Prof. Timm Faulwasser** for their expertise in the field and commitment to academic excellence, which have been truly inspiring.

To each one of you, I am sincerely grateful for the profound impact you have had on both my academic and personal growth.

I consider myself incredibly fortunate to have been part of an exceptional team of colleagues and friends at the Chair of Serious Games, at the University of Bayreuth. Big thanks to **Miroslav** for being there to support me at the beginning of my PhD journey. You were a terrific office mate and I really enjoyed our long discussions. Thank you, **Sofia**, for kindly sharing your expertise and for your readiness to provide valuable feedback to my, at times, 'early-stage' ideas. Thanks **Daniel** for offering excellent academic insight and teaching advice. Many thanks to **Arthur** for helping me add rigour to my work. Thanks to **Markus**, **Florian F.**, **Hai**, and **Florian L.** for the support and team spirit. Our shared journey towards our PhDs allowed us to encourage, uplift, and push each other to achieve our goals. Special thanks to **Tiare**, **Andreas**, and **David** for being a great source of inspiration during the start of my PhD. Your passion for research and innovative thinking motivated me to embark on this path with enthusiasm and determination. Last but not least, I would like to express my gratitude to **Nadine** for her tireless efforts in shielding us from bureaucracy, her unwavering readiness to help, and her generosity and hospitality. Thank you all for the wonderful memories we created together during bowling, board-game nights, barbecues, and many more memorable moments.

To my fellow *Levitate* enthusiasts: **Ryuji**, **Orestis**, **Jens**, **Carl**, **Euan**, **Julie**, **Prax**, **Ben**, and **Rafa**, thank you so much for being a tremendous inspiration. I enjoyed taking part in our vibrant, visionary, and thought-provoking meetings and discussions.

Furthermore, I would like to thank our student assistants at the lab: **Timm**, **Jan**, and **Eric** for tirelessly running numerous user studies. Thanks to the many volunteers who participated in the pilot studies. Also, big thanks to **Eduardo** and **Jan** for producing awesome videos and photos of our research, as well as to all the people who appeared in them.

I owe a special thanks to all the brilliant students I got to co-advise over the years: **Michael**, **Anne**, **Jonas**, **Martin L.**, **Vamsy**, **Anuj**, **Alicja**, **Maximilian**, **Nimisha**, and many more. Thank you for being my reflective mirror and teaching me so much without even knowing.

Ништо од ова не ќе беше возможно без безрезервната поддршка од страна на моите родители и мојот брат, и безусловната љубов и грижа на моите баби и дедовци. Секој збор е малку за да ја искаже огромната благодарност што Ви ја должам.

Колку овој труд е мој, толку е и Ваш.

I would like to thank my friends of almost twenty years: **Natalija**, **Kristina**, and **Marija**, for the awesome get-togethers we had every time I made a visit to my hometown. Seeing you fills my heart with unique warmth and joy. Big thanks to my best friend, **Martin**, for the endless supply of memes that made my PhD days brighter.

Lastly, I would like to acknowledge the tremendous effort of numerous teachers, mentors, tutors, and instructors, I was lucky enough to cross paths with over the years. Among the lines of this dissertation, there is a piece of each one of you. Thank you.

Contents

Nomenclature

Acronyms / Abbreviations

$2.5D$	Two and a half dimensional
$2D$	Two-dimensional
$3D$	Three-dimensional
API	Application Programming Interface
AR	Augmented Reality
ATM	Automated Teller Machine
$BFGS$	Broydon-Fletcher-Goldfarb-Shanno (algorithm)
$CAVE$	Cave Automatic Virtual Environment
CSV	Comma-separated Values
DC	Direct Current
EPS	Expanded Polystyrene
$FPGA$	Field Programmable Gate Array
FPS	Frames per Second
GUI	Graphical User Interface
HCI	Human-Computer Interaction
HMD	Head-mounted Display
ID	Index of Difficulty

LED	Light Emitting Diode
$MATD$	Multimodal Acoustic Trap Display
MR	Mixed Reality
MT	Movement Time
OCP	Optimal Control Problem
OTD	Optical Trap Display
$PATs$	Phased Arrays of Transducers
PCB	Printed Circuit Board
PoV	Persistence of Vision
RGB	Red-Green-Blue
$RGBW$	Red-Green-Blue-White
$RMSE$	Root Mean Square Error
SD	Standard Deviation
SDK	Software Development Kit
SLM	Spatial Light Modulator
SUS	System Usability Scale
TLX	Task Load Index
TP	Throughput
UDP	User Datagram Protocol
UES	User Engagement Scale
UI	User Interface
VR	Virtual Reality

Chapter 1

Introduction

The *Ultimate Display* is one of the longest standing visions in HCI. It envisions a portion of physical space, e.g., a room, where the computer is able to fully control the existence of matter. The so conceptualised display would be able to seamlessly merge the physical and digital world - *'A chair displayed in such a room would be good enough to sit in. Handcuffs displayed in such a room would be confining, and a bullet displayed in such a room would be fatal.'* [138], and allow for direct, natural, and intuitive interaction.

A promising approach to create volumetric displays that support direct manipulation is ultrasonic levitation (Figure 1.1). Levitation interfaces exploit the acoustic radiation force, exerted by focused ultrasonic waves, to trap levitated matter in mid-air. The levitated matter is composed of individual millimetre-sized particles, typically made of expanded polystyrene. The ultrasonic waves used to suspend the levitated matter in mid-air are generated by phased arrays of ultrasonic transducers. The displayed content can be varied by changing the position, speed, and acceleration of the levitated particle, and also by augmenting it with auditory, haptic, or visual feedback provided by for example, an external source of illumination (e.g., projectors, LEDs, lasers etc.).

The long-term vision is the creation of a tangible 3D display in mid-air composed of multiple levitated particles, where the existence, form and appearance of the levitated matter can be fully controlled by the user. Users would interact with the display in a multimodal manner. They could see, feel and manipulate the levitated matter, with the visual and haptic feedback being generated in mid air. The result would be a completely dynamic, optimally controlled, and fully interactive levitated content that responds in a realistic manner to touch interactions in mid-air. Apart from the physical particles floating in mid-air, the display could be enhanced using projections onto the particles and/or the background.

An advantage of the ultrasonic levitation display is that the content is tangible and users can make use of their highly-developed, innate skills for sensing and interacting with

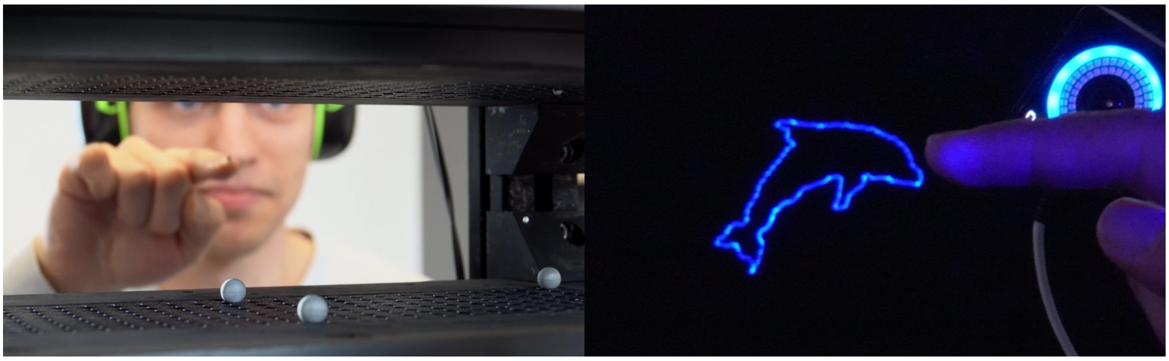


Figure 1.1 Acoustic levitation displays present the content in mid-air in real physical space, hence providing all natural depth cues, contrast, and illumination, and allowing for in situ user interaction.

physical environments, which are often neglected in the interaction with digital content [56]. In addition, the levitated matter is highly dynamic - it can easily morph into different shapes, which is uncharacteristic for most physical media. The hardware of the display is also capable of producing versatile, touchless haptic feedback. By modulating the ultrasonic waves with a frequency detectable by the receptors in the human skin, it is possible to create a perceivable haptic sensation in mid-air [21]. Thus, in addition to seeing, the user has also the possibility of feeling the content. Another advantage of the display is that all users know to be viewing the same content, as opposed to VR applications, for example, where only the user immersed in the virtual world can experience the content. In addition, users can view the content from many angles and see through the display, which facilitates multi-user interaction and collaboration. The display has all natural depth cues, and contrast, focus, and illumination are all correct. Hence, users will be able to employ their spatial understanding to visualise and resolve complex 3D problems. Finally, the user is unadorned when performing the interaction, i.e., they can manipulate the content with their bare hands, without the need for gloves, controllers or other gadgets. In summary, such a levitating physical display can be highly beneficial for most users working with 3D content, whether it is biochemists visualising complex molecules, architects planning building structures or artists designing spatial installations.

With this dissertation, I aim to promote model-based control and simulation of physical user interfaces in the field of Human-Computer Interaction. With the proposed methods, I show the advantages of structured mathematical approaches for improving the control and performance of acoustic levitation interfaces, as opposed to the current practice of trial-and-error. My proposed solution consists of deriving a mathematical model of the governing system dynamics, and formulating and solving an optimal control problem that computes a set of inputs that allow the interface to reveal the desired output, as fast and as accurately as possible. I then use the model of the dynamics of the physical interface to build a digital counterpart in Virtual Reality (VR), that facilitates rapid prototyping and validation of applications for levitation interfaces.

1.1 Challenges

Ultrasonic levitation interfaces, with their unique capabilities of high-resolution spatial and temporal manipulation of physical objects, have the potential to bring about a paradigm shift in the way we view and interact with displays. However, their physical nature poses challenges that the HCI community has not tackled before in a systematic manner. The particles in the levitation display are physical objects interacting with an acoustic field, so they cannot be manipulated in the same way as virtual pixels on a 2D screen. Determining the optimal controls for the trap-particle dynamical system, resulting in the desired display performance and visual/haptic quality, is not a trivial task. We need to develop control methods and design suitable interaction techniques, that take into account the advantages, but also the limitations posed by the physicality of the interface. I list below some of the most important challenges on the road towards a high-performance, agile, interactive, multimodal acoustic levitation display.

Particle Movement

Updating a levitation display means that we need to move the levitated particle, by changing the position of the acoustic trap, i.e., we change the state of a dynamical system, ideally in a controlled manner, avoiding unwanted behaviour. This unwanted behaviour usually manifests in the form of unstable particle movement, and in some cases, it can lead to the particle being completely ejected from the acoustic trap. This poses the question - how do we update the display smoothly, without causing such particle jitters?

Interaction

Reaching into the display with the bare hand and performing direct physical manipulation of the levitated particle would be an ideal and natural mode of interaction, however, it is not possible in practice. The reflected ultrasonic waves from the hand disturb the acoustic field, destabilise the particle, and even cause it to drop. Hence, new forms of interaction need to be designed specifically for this type of displays, that preserve the physicality of the display and offer rich haptic feedback, without disrupting the levitation.

Content Generation

At the current development stage, there are several factors that have a negative impact on the deployability of interactive levitation interfaces for content generation outside of research. First, significant financial and time investment is needed upfront to set up and run the system. Next, building and operating the hardware requires specialised technical skills, precise calibration, and synchronisation of different components. Finally, in case of problems, debugging the system can be a challenging task, as often the only visible effect is the levitated particle shooting out of the acoustic field. Nonetheless, it would be highly beneficial for designers and other creative experts to explore the seemingly *magical* properties of the display and design innovative applications that exploit the unique properties of the interface, e.g., by using a unifying tool that facilitates prototyping for users of different backgrounds and skill sets.

Haptic Resolution

As discussed above, direct physical manipulation of levitated particles is not feasible. However, the possibility of modulating the ultrasonic waves to produce haptic feedback in mid-air seems like a promising approach to enrich the display interaction with directed, meaningful haptic feedback. To achieve this we need to further examine the perceptual properties of this largely unexplored novel type of feedback, in particular, with what fidelity can we represent individual particles of the levitated content using the mid-air haptic feedback?

Particle Control

When dynamically changing the position of the levitated particle in the display, the position where we generate the acoustic trap usually does not match the actual position of the physical particle. In order to accelerate the particle, directional acoustic force has to be generated on it, i.e., the particle has to be displaced from the trap centre where the net acoustic force is (almost) equal to zero [86]. The actual displacement between the trap position and the particle position needed to move the particle in a predefined way is not easy to deduce, because it depends on several factors, such as the velocity the particle already has and the topology of the acoustic trap, i.e., how the acoustic forces are distributed around it. Hence, an algorithm that computes the exact position where we need to generate the acoustic trap in order to obtain the desired particle path is needed. Moreover, the timing of the path needs to be defined, that is, we need information not only about *where* to place the trap, but also *when*. Hence, we need to compute acoustic trap positions, such that the particle does not drop (*feasibility*) and it traverses the prescribed path in optimum time (*minimum feasible time*).

1.2 Approach

In this dissertation, I present four different concepts and their implementations:

1. a levitated 3D cursor, demonstrating smooth particle movement in any direction,
2. a levitation simulator for virtual prototyping, allowing to iteratively develop and prototype ideas for ultrasonic levitation interfaces in VR,
3. methods for rendering haptic stimulation points with ultrasound, investigated in the context of an accessibility application for reading Braille information in mid-air, and
4. an optimisation algorithm that helps render bigger and more complex shapes with levitation, by computing optimal trap trajectories given a reference path.

Together, this amounts to the advancement of multimodal levitated displays, both in terms of performance and usability. Further, performance tests identifying the capability limits of these systems were performed, as well as initial user studies which evaluate and validate the usability and usefulness of the implementations for potential users. These, in turn, allowed me to examine and characterise the potential of ultrasonic levitation interfaces as physical volumetric displays, and generate a set of guidelines for researchers, artists, designers, and engineers who wish to design and create for this innovative type of displays.

1.3 Contributions

In this dissertation, I propose concepts and methods that aim to improve the quality of interaction with ultrasonic levitation displays, allow for a better control of the outputted physical content through modelling and simulation of the underlining dynamics, facilitate the design of interactive applications for this type of displays, and explore its multimodal capabilities. My main contributions are:

- A *method* for achieving smooth, low latency, dexterous interaction with a levitated cursor with high agility, and a *stabilisation mechanism* that enable high-speed particle movement. (Chapter 3)
- An *evaluation* of the pointing performance in ultrasonic levitation displays by conducting the first device-mediated Fitts' Law type pointing study with a levitated cursor in physical 3D space with all natural depth cues. (Chapters 3 and 5)

- An open source *virtual prototyping tool* for developing applications for ultrasonic levitation displays, which employs a *mathematical model* to simulate the dynamical properties of the physical system in the virtual environment. (Chapter 5)
- A *validation* user study of the digital twin of the levitation interface in terms of pointing performance, user engagement, and immersion. (Chapter 5)
- An *investigation* of the perceptual capacity of users to identify individual haptic stimulation points on the palm generated with ultrasonic haptic feedback, in user studies both with sighted and blind participants. (Chapter 6)
- A *semi-analytical model* of the acoustic force around a trap, describing the trap-particle dynamics. (Chapters 4, 5 and 7)
- A *mathematical formulation* of the optimal timing problem for ultrasonic levitation interfaces. (Chapter 7)
- A *numerical optimisation algorithm* that computes physically feasible and nearly time-optimal trap trajectories for levitated particles, given only a reference path, using a path following approach. (Chapter 7)

1.4 Publications

The collaboration with researchers from my home institution as well as external field experts, has played a crucial role in producing the body of work presented in this dissertation. I would like to express my appreciation to my advisor, Jörg Müller, who has been instrumental in guiding this research. I would also like to acknowledge the valuable contributions of Miroslav Bachinski, Sofia Seinfeld, Arthur Fleig, and other members of the Department of Computer Science at the University of Bayreuth. Additionally, I extend my appreciation to Timm Faulwasser and Diego Martinez Plasencia for sharing their vast knowledge and expertise.

It is important to highlight that this collaborative effort is recognised through co-authorship of the publications and will be honoured in this dissertation by using the plural form "we" in the Chapters 3-7. The work presented here is to a large extent based on the following conference and journal publications:

- Chapter 3 was published and presented as *LeviCursor: Dexterous Interaction with a Levitating Object* [9] at ISS 2018, Tokyo, Japan. This is the author's version of the work, posted for personal use. The definitive Version of Record was published in the

Proceedings of the 2018 ACM International Conference on Interactive Surfaces and Spaces, ISS'18, http://doi.acm.org/10.1145/3279778.3279802.

- Chapter 5 was published and presented as *Levitation Simulator: Prototyping Ultrasonic Levitation Interfaces in Virtual Reality* [108] at CHI 2020, virtual. This is the author's version of the work, posted for personal use. The definitive Version of Record was published in the *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*, CHI'20, https://doi.org/10.1145/3313831.3376409.

- Chapter 6 was published and presented as *HaptiRead: Reading Braille as Mid-Air Haptic Information* [110] at DIS 2020, virtual. This is the author's version of the work, posted for personal use. The definitive Version of Record was published in the *Proceedings of the 2020 ACM Designing Interactive Systems Conference*, DIS'20, https://doi.org/10.1145/3357236.3395515.

- Chapter 7 was published as *OptiTrap: Optimal Trap Trajectories for Acoustic Levitation Displays* [109] in ACM Trans. Graph. 2022. Presented at SIGGRAPH 2022, Vancouver, Canada. The definitive Version of Record was published in the *ACM Transactions on Graphics* 2022, https://doi.org/10.1145/3517746.

Chapter 2

Background - *The Ultimate Display*

The long-term fundamental vision of my work is to develop an acoustic levitation interface that will embody **a seamless bridge between the physical and the digital world**. The prospect of one day having an interface that perfectly merges these two seemingly disjoint worlds, first came into light in the 1960s. In the famous essay from 1965, *The Ultimate Display*, one of the pioneers of HCI, Ivan Sutherland, vividly illustrates the capability of computer displays to become our *looking glass into mathematical wonderland* [138]. He argues that computers can not only help us visualise a phenomenon that otherwise might seem abstract or distant, but they will also be able to present it in a palpable way, so that we can interact with it, and get to know it, the way we know the physical world. He envisions the *ultimate display* as a room where the computer can fully control the existence of matter, e.g., the computer can create chairs to sit on, or handcuffs that are confining.

Over the years, this vision has been interpreted in several ways, and different potential enabling technologies have been developed to bring it to life. One interpretation of the *ultimate display* is a fully immersive virtual environment that triggers physiological responses, giving rise to the growing field of VR [23, 135]. It is important to clarify, however, that this approach is not within the scope of this dissertation, hence the VR implementations and the different available systems (e.g., HMD, CAVE, etc.) will not be further addressed here. The reason for the exclusion is that VR focuses on replacing reality with a simulated digital environment, whereas the focus of this dissertation is a more radical Mixed Reality (MR) interpretation of the *ultimate display*, where the computer can manipulate physical matter in a smooth and interactive way. Inspiration for this approach can be found in the work of Ishii et al. [57], who introduced the concept of *radical atoms*. These are physical materials that are dynamic and computationally transformable, taking on the role of physical manifestations of digital bits.

In this chapter, I provide an overview of existing display prototypes that have contributed to advancing the physical-digital integration, taking the HCI field a step closer towards the *ultimate display*. These prototypes present digitally reconfigurable visual content, that can be viewed in physical space from many angles, and by multiple users simultaneously. The list is not exhaustive, still, it provides a detailed account of the most prominent examples and the underlying technologies. Furthermore, I discuss the capabilities and limitations of each prototype, particularly in relation to acoustic levitation, and its potential to drive significant advancements in our pursuit of the *ultimate display* vision in the future.

Consistent with the *radical atoms* terminology [57], I use the term *atom* to denote the individual units that serve as 'building blocks' for rendering the physical manifestation of the digital content. Depending on whether the intelligence, actuation, and power are contained in the atoms themselves or in their immediate environment, I organise the systems into *active* and *passive atom* systems, respectively.

2.1 Active Atoms

The idea of the active atom approach is to use an ensemble of small tangible active units to manifest digital information in physical space. These active physical artefacts embody the concept of *programmable matter*, which Toffoli and Margolus [143] illustrate in 1991 as follows: *'the same cubic meter of machinery can become a wind tunnel at one moment, a polymer soup at the next; it can model a sea of fermions, a genetic pool, or an epidemiology experiment at the flick of a console key'*. The goal is to create an interactive and dynamic rendering system in physical space that can reproduce any physical object or phenomenon in terms of its visual appearance, movement, auditory, and haptic characteristics.

Below, I present some prominent examples of interface prototypes that aim to realise the concept of programmable matter, using one of the following active atom implementations: miniature robots, actuated pins or strings, and drones.

2.1.1 Miniature Robots

Advancements in the field of swarm robotics have resulted in the development of several systems that use small robot units to create dynamic physical representations of digital content. An early example of a modular robotic system that aims to scale up to millions of units is *Claytronics*, introduced by Goldstein et al. in 2004 [45]. Each catom (short for claytronic atom) is envisioned to have computing, sensing, actuation, and locomotion capabilities, allowing it to move in any direction, maintain a 3D shape, communicate with other catoms,

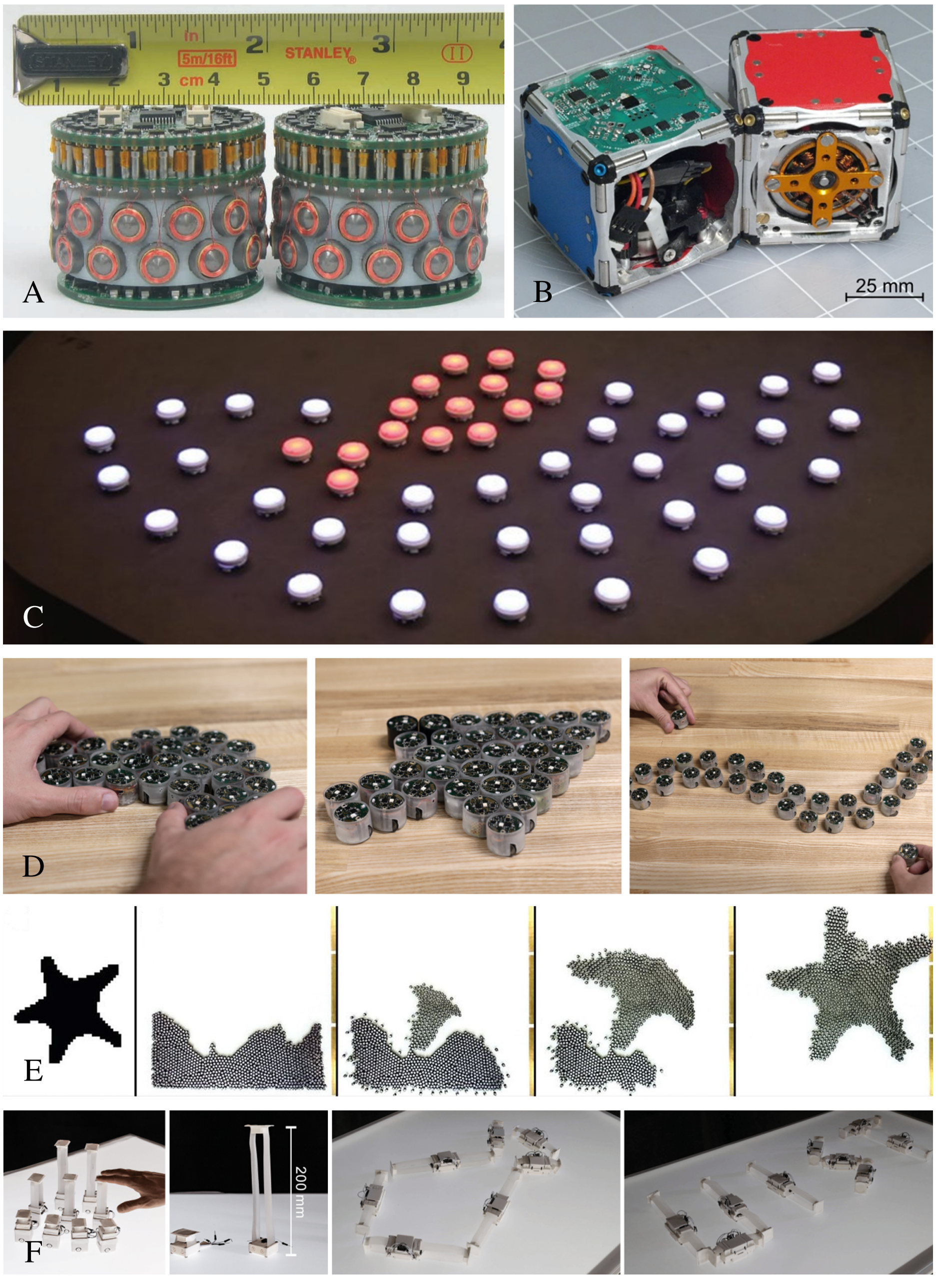


Figure 2.1 Miniature robots as active atoms. (A) *Claytronic* atom prototypes, 44mm in diameter, equipped with 24 electromagnets arranged in pairs [44]. (B) *M-blocks* [123] robots that use an internal flywheel to generate torque to perform a pivoting motion. (C) A 50 robot swarm of 5cm mobile pixel robots with an RGB LED for colour control, moving at speeds up to 25cm/s [4]. (D) *Zooids* swarm, consisting of 30-40 wheeled robots of 26mm diameter, moving at a speed of 44cm/s, used as physical pixels, or as handles to manipulate the rest of the swarm [72]. (E) Self-assembly formation steps for a star shape by a swarm of 1000 *Kilobots*, 30mm in diameter, with a movement speed of 1cm/s [125]. (F) *ShapeBots* [141] swarm robots, moving at speeds up to 17cm/s, using reel-based linear actuators to render large-scale, low-resolution shapes. Figures reprinted from [44, 123, 4, 72, 125, 141].

and perform state computations [44]. Figure 2.1(A) shows a prototype consisting of two Claytronics robotic units. Despite the potential for scaling, the catoms are not able to move independently due to the joint implementation of locomotion and connection using electromagnets along the circumference, which limits their ability to adapt to changing environments or reconfigure themselves in real-time.

The *M-blocks* [123] are cube-shaped momentum-driven miniature robots that are capable of independent and lattice-based locomotion, facilitated by diametrically polarised rotatable magnets along the edges of the cubes that bond neighbouring blocks, as shown in Figure 2.1(B). These magnets also act as hinges, supporting the moving block when performing pivoting motion. Although the magnetic bonds provide a strong and flexible attachment mechanism, a major drawback of M-blocks is their limited reliability in executing motion primitives, due to factors such as not applying enough torque, too much torque, disconnection from the lattice, or overshooting, as well as the need for a sturdy surface to operate on.

Figure 2.1(C) shows a multi-robot display, where conceptually each robot represents a mobile pixel of controllable colour [4]. The system performs a goal generation step offline to determine an optimal representation of the target image or animation, given the available number of robots. Then at run-time, goal assignment, path planning, and local collision avoidance are performed. The mobile pixel robot swarm has been demonstrated with up to 50 differentially-driven robots, but the resulting image is relatively sparse and tangible user interaction is possible only with individual robots. *Zooids* [72] is another platform for developing tabletop swarm interfaces, but with a focus on direct tangible interaction. The robots can be manipulated both collectively or individually, via the computer or via graspable user input (see Figure 1.1(D)). The system has limited scalability due to the projector-based optical tracking and centralised control mechanisms. A further limitation is that touch sensing is implemented using capacitive sensors around the circumference of the robots, causing only the sensors in direct contact with the hand to activate during multi-Zooid interactions.

Rubenstein et al. [125] developed a programmable self-assembly of 2D shapes using a thousand-robot swarm, illustrated in Figure 2.1(E). The system operates in a decentralised and asynchronous way, where the group behaviour is achieved through local interactions. Each robot consists of a microcontroller, two vibration motors to allow movement, and an infrared transmitter and receiver for communication. However, due to the limited communication range, and the use of vibrational motors instead of wheels, the robots are only capable of relatively slow 2D movement on flat surfaces. Additionally, the robots cannot sense their position within the swarm, which may potentially lead to disorganised behaviour or failure in achieving the desired shape.

ShapeBots [141] is a swarm robot system that offers shape-changing capabilities. Each robot can change its shape by employing a 2.5cm wide linear actuator that can extend up to 20cm both in the horizontal and the vertical direction, shown in Figure 2.1(F). While the system has demonstrated simple geometries and physicalizations with 12 robots on a tabletop, rendering more complex geometries can be challenging due to the size and resolution constraints of the shape-changing robots.

In this section, I provided an overview of selected system implementations that use miniature robots on a tabletop to produce dynamic physical representations. While *M-blocks* could be stacked, and *ShapeBots* could extend using linear actuators, the rest of the robot-based implementations were only capable of rendering 2D physical content. A major challenge in increasing the dimensionality of the content is overcoming gravity while maintaining structural stability. An in-depth analysis of the challenges and opportunities of modular robotic systems is provided by Yim et al. [154]. In the next section, I explore shape displays, which use actuated pins and strings as active atoms, providing mechanical stability in the vertical dimension.

2.1.2 Actuated Pins and Strings

Shape displays that use actuated pins consist of 2D pin arrays that can be extended vertically to create a physical representation of digital images or objects. These displays are limited to 2.5D, as they can only approximate the shape and contours of arbitrary objects, and cannot produce a fully three-dimensional representation. Despite being less mobile and having a fixed layout, these displays offer better structural stability and allow for tangible interaction through touching or pressing the actuated pins.

The *inFORM* [33] interface is capable of producing a dynamic physical approximation of digital images or objects using a 30×30 array of actuated polystyrene pins that can extend up to 10cm in height. It also features a depth camera for object and user detection and a projector for additional visual information. The system can create dynamic affordances, constraints, and actuation of passive objects, as shown in Figure 2.2(A). However, the full system has a large footprint, comes with high manufacturing costs, and is difficult to scale due to the 900 mechanical actuators that control the movement of the pins. *shapeShift* [131] is a pin-based shape-changing display that was introduced as a more compact alternative to the *inFORM* display. See Figure 1.2(B) for a visual representation. It consists of 288 hollow aluminium pins that can extend up to 50mm in height. The system can be used for rendering and manipulation of static physical content, as well as tangible exploration of virtual information spaces. Lateral mobility of the system is achieved using active and passive platforms, allowing for a larger workspace. While the system is more compact and

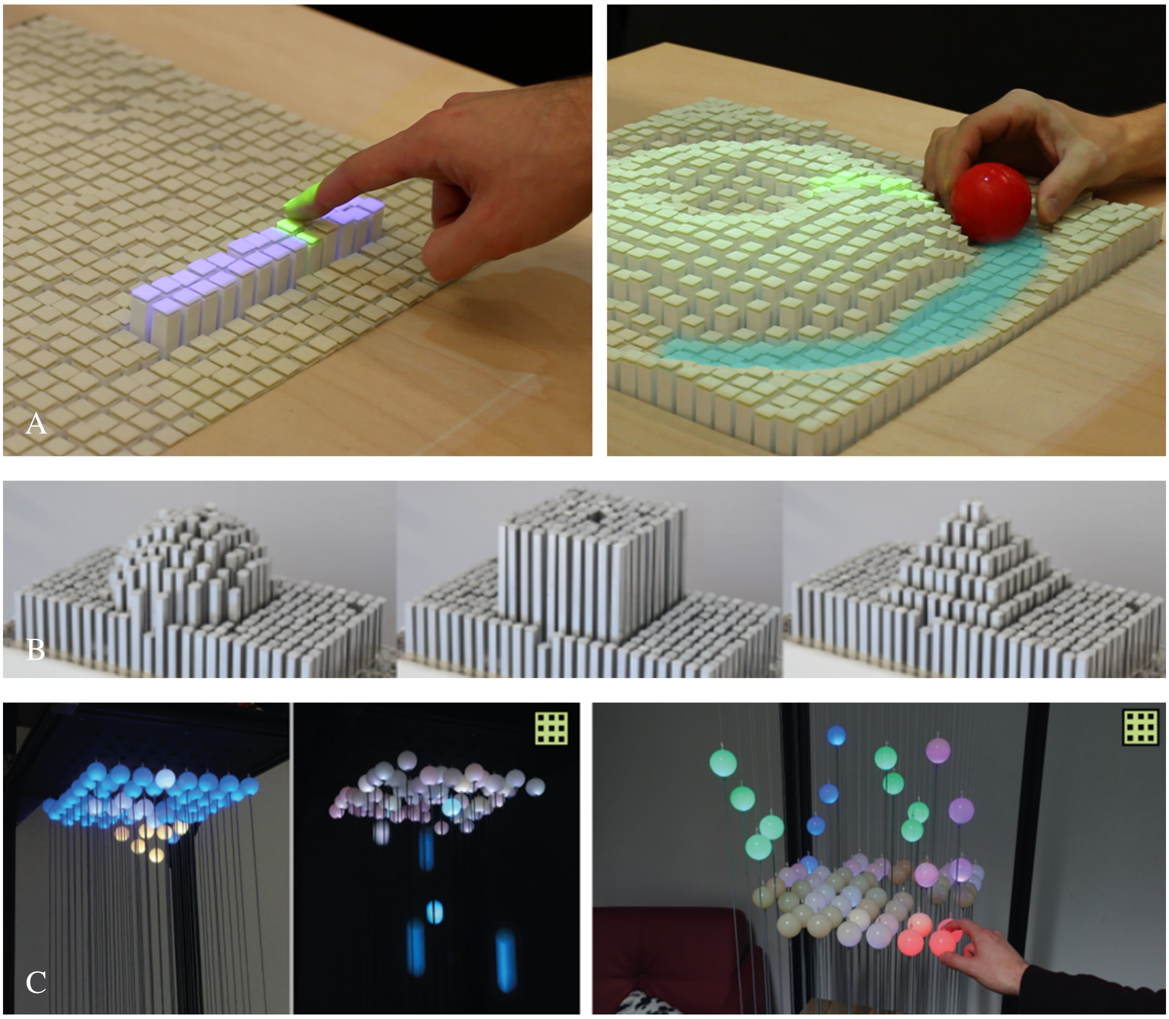


Figure 2.2 Actuated pins and strings as active atoms. (A) The *inFORM* [33] interface offers an interactive tangible area of 381mm×381mm. The shape change can both embody UI (left), and represent content, including enacting dynamic affordances to passive objects (right). (B) Physical shapes rendered by *shapeShift* [131], a modular tabletop shape display consisting of six 2×24 pin modules, covering a total active rendering volume of 178mm×89mm×50mm. (C) An 8×8 matrix arrangement of the *STRAIDE* [26] display, which uses actuated strings and features 64 illuminated spheres made of epoxy resin with embedded RGBW LEDs. The interactive spheres are actuated using towers mounted on top of the display, with each tower capable of moving up to 4 spheres with a spacing of 5cm×5cm. The display is shown in two different applications: a weather application in the two left images and a music visualisation in the image on the right. Figures reprinted from [33, 131, 26].

lighter than the *inFORM* display, it still has a fixed layout and limited mobility. Additionally, the use of a smaller pin array may limit the level of detail that can be achieved in the physical representations.

The *STRAIDE* [26] spatial display is based on a similar concept as the pin-based displays but uses actuated strings with spherical interactive elements instead. This enables strokes longer than 1m compared to pin-based displays, as shown in Figure 2.2(C). However, the actuated string displays have their own set of limitations such as unwanted oscillations due to resonances and limited interaction with the inner display elements because of restricted access.

Displays using actuated pins and strings offer more flexibility in the type of content that can be rendered, but they have notable limitations. They can render only 2.5D content, and they do not support overhangs, limiting the representation of some objects to only their top contours. Similarly, sideways manipulation of the content is not possible. The size and complexity of the linear actuators also make it challenging to make the pins or string elements small and densely packed enough for users to perceive an almost continuous shape. Despite these limitations, this particular subset of shape-changing displays can still present physical approximations of generic content and be utilised more universally, thanks to their many degrees of freedom. In contrast, most other types of shape-changing interfaces focus on a particular transformation, usually linked to a specific use case, and are outside of the scope of this work. For further information on this topic, please refer to the literature [120, 117, 3, 137]. In the next section, I discuss possible approaches to render physical 3D content.

2.1.3 Drones

To create physical interfaces that can operate fully in 3D, researchers have been experimenting with using drones as active atoms. *BitDrones* [46] is a toolbox that employs quadcopters as tangible building blocks for interactive 3D displays, as seen in Figure 2.3(A)). The quadcopters can be equipped with various add-ons, including LED displays and touch sensors, offering new possibilities for interaction and content creation, including 3D drawing, telepresence, and gaming. *Flyables* [67] is a system building upon the *BitDrones* concept, that regulates the quadcopter tangibles as to allow for user interactions. The quadcopters are made graspable by adding a lightweight cage, that provides protection from the rotors, shown in Figure 2.3(B). The *GridDrones* [16] platform tackles some of the limitations of the previous approaches by further reducing the size of the individual units, optimising the communication, and creating a $1 \times n \times n$ grid of quadcopters, which allows for the representation of 2.5D grid relief maps and 3D spatial transformations. A user performing shape deformations to a grid of 15 quadcopters using touch-based interaction is shown in Figure 2.3(C).

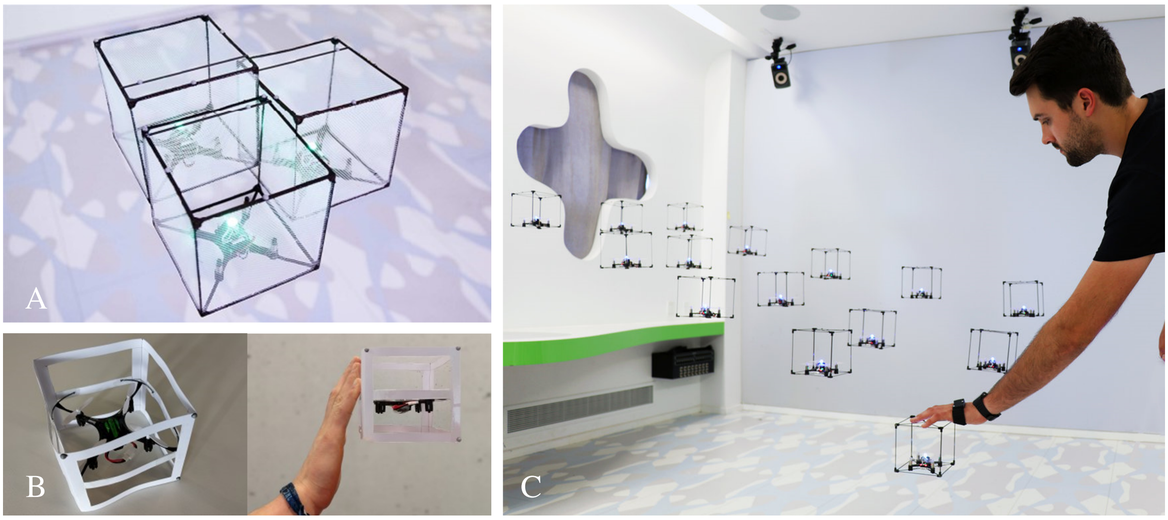


Figure 2.3 Drones as active atoms. (A) Three *BitDrones* flying next to each other in a tight formation. The drones are of the type called ShapeDrones, and have a lightweight acrylic mesh spun over a 3D printed frame with an edge width of 13cm. The system can support up to 12 simultaneously flying quadcopters, in a 4m×4m×3m volume, with an average accuracy of 5cm per axis. [46]. (B) The *Flyable* system consists of interactive drones of size 11cm×11.5cm×11cm, position controlled using a PID controller with an accuracy of ±5cm. The drones are equipped with a cardboard cage to enable safe user interaction [67]. (C) A user interacting with the *GridDrones* system, consisting of 15 cube-shaped drones of 10cm width, positioned in a 1×5×3 grid [16]. Figures reprinted from [46, 67, 16].

While the technology allows for new interaction techniques, it still has several limitations, such as limited battery life and flight time, restricted range, and noise. Another limiting factor is the downdraft, which limits the ability of drones to fly below or on top of each other. In addition, the drones may drift due to turbulence caused by the rotors, especially in closed indoor settings. As a result, the generated visual representations are fairly sparse.

2.1.4 Summary

This section targeted interfaces that can create generic physical representations in multidimensional space using active atoms, i.e., a set of individual physical units with self-contained actuation, locomotion, and power supply. The implementations discussed here involved swarms of small tabletop robots, arrays of linearly actuated pins and strings, and drone grids. These examples demonstrate that the vision of programmable matter bears indeed real-world potential. However, some fundamental challenges, such as scalability, still remain to be solved.

The limitations of the various displays in the active atom approach can be summarised as follows:

- high power consumption and high manufacturing costs of the active units, due to their engineering complexity, stemming from the many functionalities that each individual active atom should be able to perform;
- physical limitations on how much the size of an active atom can be reduced, while still containing all the necessary components (e.g., for motion, sensing, bonding, communication, etc.), hence transition from *individual volumetric pixels to programmable matter* might not be achievable;
- significant high-level planning and low-level control efforts required to produce arbitrary reconfigurable physical shapes, including efficient and scalable (decentralised) communication across multiple units, robust failure recovery mechanisms, and agile collision and obstacle avoidance.

2.2 Passive Atoms

The passive atom approach offers a solution to some of the challenges faced by the active atom approach, such as high manufacturing costs and power consumption, by transferring the computing, locomotion, and power supply from the atoms to their immediate environment. In this approach, passive atoms are simple physical objects made of low-cost materials, which can be turned into dynamic, interactive information carriers by carefully controlling and tuning their environment. This section covers various implementations that use external forces generated by airflow, water, magnets, lasers, and ultrasound to capture and manipulate the passive atoms.

2.2.1 Airflow and Water

The following implementations use directed streams of fluid to counteract gravity and enable passive atoms to move along the vertical dimension, generating dynamic and reconfigurable physical interfaces. The passive atoms are typically floating lightweight balls made of synthetic materials, whose programmable mid-air movements are achieved through carefully controlled fluid flow.

The *Aerial Tunes* [5] system utilises directed airflow to move and manipulate spherical objects in mid-air, as shown in Figure 2.4(A). Users can modify aspects of the ambient sound by grabbing and repositioning these passive atoms within the air stream. However, several

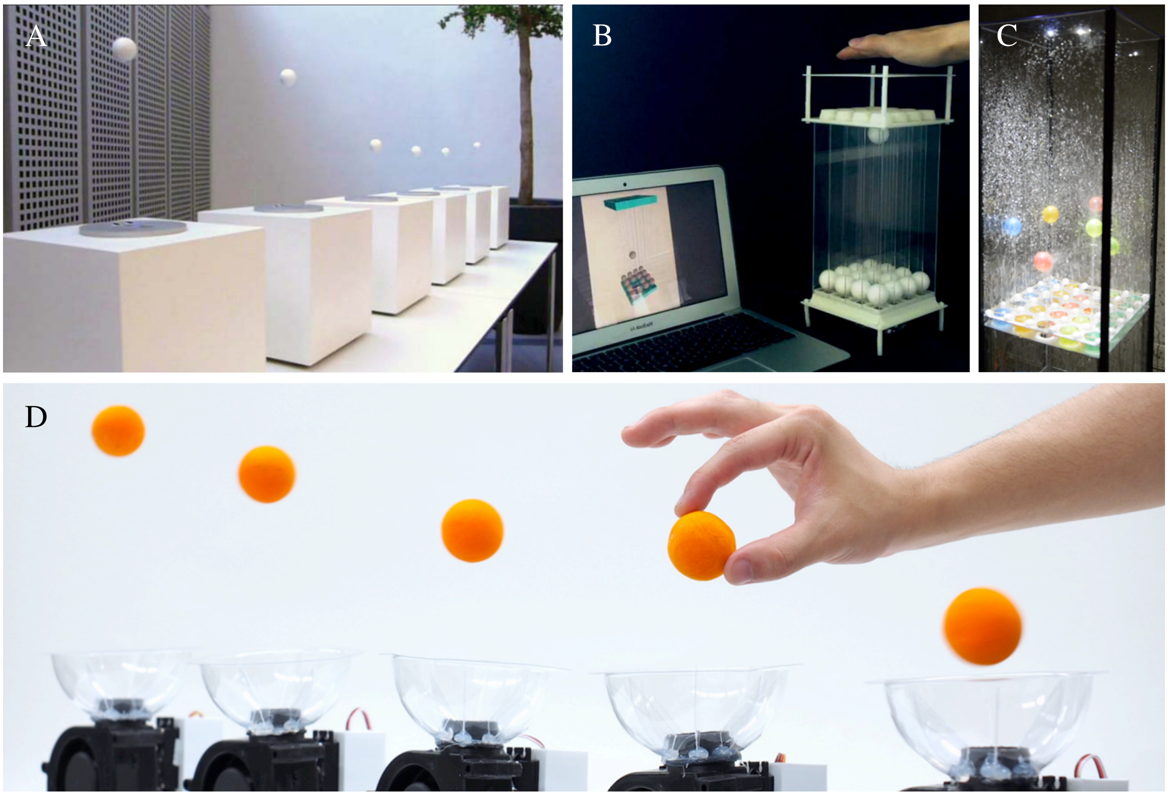


Figure 2.4 Air and water-based levitation of passive atoms. (A) The *Aerial Tunes* [5] interface consists of six 30cm×30cm×30cm wooden cubes, equipped with sensors and an air jet to adjust the balls' height from 2cm to 30cm with a 1cm resolution. Using these physical representations, users can visualise, explore, and manipulate the soundscape, either individually or collaboratively. (B) *AirVis* [116] system comprises a transparent box with a 4x4 array of fan motors, capable of moving 2cm diameter balls in mid-air. In addition to the physical visual information, the display also generates tactile sensations on the user's skin using the airflow pressure. (C) *Garden Aqua* [148] is a water-based levitation interface for physicalisation of the music creation process. It features a lighting system, a motion tracking unit that enables gesture-based input, and a 6x6 cell array, where each cell contains a motor water pump, a nozzle, and a ball. The height of the balls is regulated using the water pressure. The balls can be moved between 30cm and 80cm within the water stream with a spatial resolution of approximately 5cm. (D) A user interacts with *floatio* [155], a floating tangible user interface, where polystyrene balls are suspended in a stream of air. The interface has 5 actuated fans that can move and rotate one or multiple 3.5cm diameter polystyrene balls. The user's hand movement is tracked using a Kinect device, enabling them to give input using the interface as a physical slider. Figures reprinted from [5, 116, 148, 155].

technical challenges may impact the interaction, including turbulence and irregularities in the airflow, and lack of a stabilisation mechanism for the floating objects. Additionally, the constant background noise generated by the air blowers may hinder engagement with the auditory modality. Similarly to *Aerial Tunes* [5], the *floatio* [155] interface also uses airflow to manipulate passive atoms, however, with the added capability of rotational movement. The system features rotatable fans with a saucer around the air nozzle, enabling the levitated balls to move between neighbouring fans. The tangible elements can be positioned both passively by the system, and actively by the user, as illustrated in Figure 2.4(D). *AirVis* [116] is an air-based display that incorporates both visual and tactile feedback. As shown in Figure 2.4(B), it uses 16 light-weight balls to express visual information. The upward airflow, which is used to push and levitate the balls, generates a soft tactile sensation on the user's hand, enabling the interface to create low-resolution patterns and basic tactile surfaces. *Garden Aqua* [148] on the other hand, is a fluid flow interface that uses water as the levitating medium (Figure 2.4(C)). Much like *Aerial Tunes* [5], the *Garden Aqua* system was designed for music creation, allowing users to visualise and manipulate a soundscape. The system provides physicalisation of sound using a tone matrix, where each matrix dimension represents the time, instrument type, and pitch. By manipulating each of the 36 balls within the tone matrix using simple hand gestures, the system enables the users to create harmonious music, that has visual representation in physical space.

While air and water stream based physical interfaces appear lively and engaging, they offer limited control over the passive atoms. Specifically, the physical objects can be displaced only along the direction of the fluid stream, with relatively low precision (e.g., 1cm for *Aerial Tunes* [5], and 5cm for *Garden Aqua* [148]). As each floating object is actuated by a separate jet and has a diameter of several centimetres, the overall display resolution is very limited. Furthermore, in interfaces such as *Aerial Tunes* [5] and *floatio* [155], where direct interaction with the display elements is possible, users need to be cautious not to block any of the air jets with their hand as it may disturb the floating object. With the remaining two interfaces, user interaction is possible only at a distance, as the *AirVis* [116] interface has to be enclosed due to the proximity of the air jets, and *Garden Aqua* [148] due to the nature of the medium.

2.2.2 Magnetic Force

Magnetic forces can operate at a distance, and they are robust enough to support direct interaction. These properties have inspired HCI researchers to explore this phenomena as the actuating force behind the passive atoms of the potential *ultimate display*. Physical magnetic interfaces come in various implementations, ranging from a single electromagnet to multiple arrays and from magnetic levitation of a single permanent magnet as a passive atom to tabletop interaction with multiple magnets. In addition to the systems that employ magnetic force for atom actuation, there are interfaces that utilise the property that magnetically bonded objects can be separated and reused, and use this type of force to hold the passive atoms together in a given structure, while the actuation occurs via another external force. In the continuation of this section, I provide an overview and discuss in more detail the potential as well as the challenges of the different approaches.

ZeroN [73] is a 3D magnetically actuated levitation interface that suspends a permanent magnet of ca. 3cm diameter in mid-air, allowing for dynamic visualisations and interactions (see Figure 2.5(A)). The magnet is held in place by a combination of magnetic and mechanical actuation, with an electromagnet controlling the vertical position and a plotter with linear actuators controlling the horizontal position. However, due to magnetic interference effects, achieving full 3D actuation using only magnetism is challenging. Magnetic levitation of multiple permanent magnets is also difficult due to interference between the magnets, limiting implementations to 2D tabletop surfaces with movement and proximity constraints. One such tabletop surface is *PICO* [113], an interactive 2D interface consisting of an array of electromagnets, magnetic pucks, and a projector, that allows users to set mechanical constraints to a computational optimisation process. Figure 2.5(B) shows an example of using two magnetic pucks as physical representations of cellphone towers, for the purpose of visualising and interacting with a computational network coverage optimisation problem. The interface supports simultaneous control of multiple pucks by creating local magnetic fields in different areas of the array. However, the individual controllability of the pucks is limited by the need to avoid overlapping magnetic fields, which restricts the total number of pucks that can be reliably controlled.

The *Reactile* [139] interface, shown in Figure 2.5(C), uses an array of electromagnetic coils to actuate a swarm of small magnets for visualisation of information in physical space. Unlike *PICO* [113], where all electromagnetic coils are placed in a single grid, the coils of the *Reactile* [139] interface are arranged in layers, and the markers move from one active coil to another adjacent coil in response to current flow. This method allowed for the employment of more than a dozen passive magnetic markers. However, the minimum required marker size and the minimum required distance between pairs of magnets impose interface constraints, that result in sparse spatial resolution. Additionally, to reduce the number of coil switches, the control mechanism of the device turns on only one row at the time, allowing markers to move one at a time. This affects the refresh times of the interface, which increase linearly with the number of markers, i.e., the refresh time of 100ms per marker, becomes 1s for 10 markers.

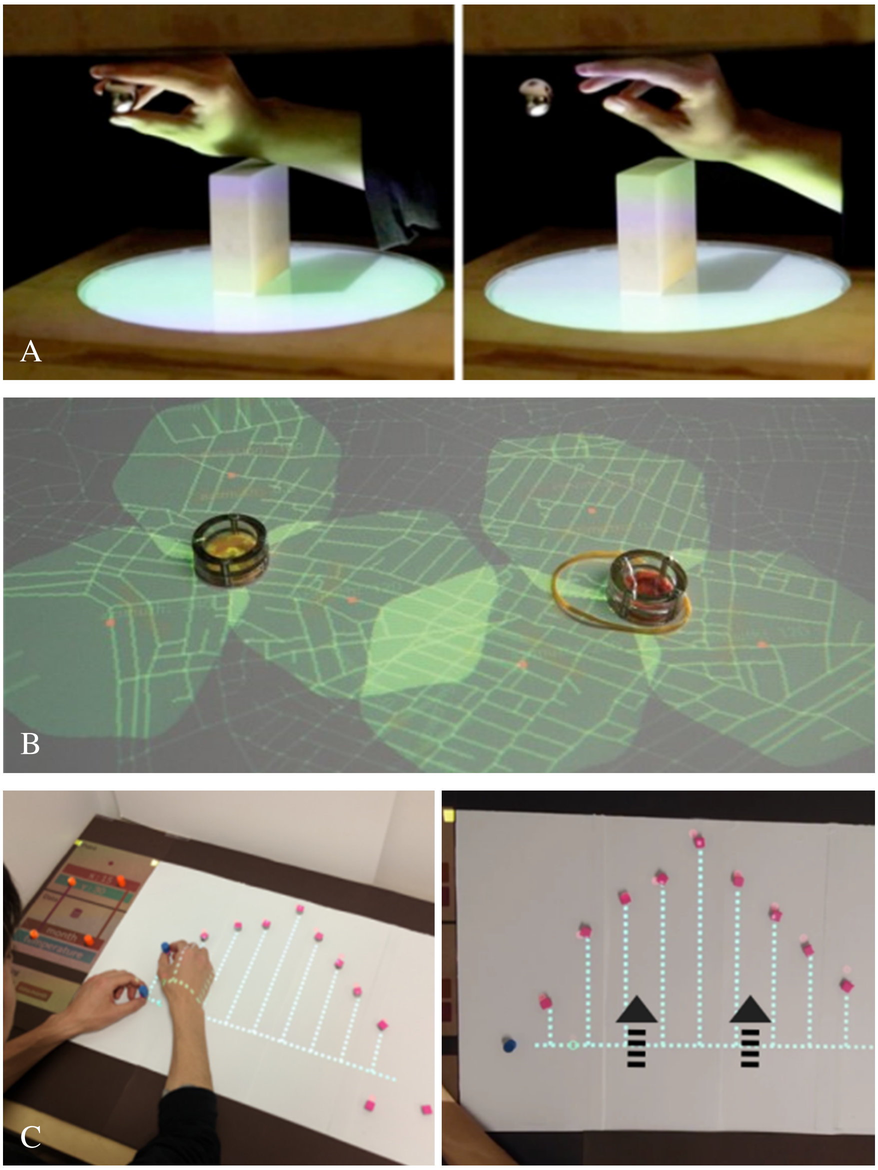


Figure 2.5 Object actuation and manipulation using magnetic force, with permanent magnets as passive atoms. (A) The *ZeroN* [73] interface consists of a levitated ferromagnetic object within a 38cm×38cm×9cm operational volume. It can achieve a maximum velocity of 30.5cm/s, with lateral oscillations of ca. 1.4cm and vertical oscillations of around 0.2cm. In the photos, the levitated object represents the Sun, and a user is exploring the digital shadow cast by the physical object (e.g., a model building), while physically manipulating the position of the levitated ball, i.e., the light source. (B) The *PICO* [113] tabletop interface features 512 electromagnets, positioning magnetic pucks with precision of 1mm, and accuracy of 2mm. The image shows an application for optimal placement of cellphone towers, given a set of constraints. (C) *Reactile* [139] uses direct physical manipulation of a swarm of small magnets (10mm in diameter) and spatial visualisation of program states to facilitate programming of swarm UI applications. Figures reprinted from [73, 113, 139].

The use of magnetic forces for the actuation of passive atoms presents several challenges. Electromagnets require a lot of power, generate heat, and do not scale well to large interactive spaces. Additionally, this method is limited to objects made of materials with ferromagnetic properties. Another constraint arises from the presence of other magnetic objects in the vicinity, which can interfere with the display. Additionally, the force exerted by magnetic fields diminishes inversely with the square of the distance, requiring exceedingly high forces for controlling objects at larger distances. In the context of 3D manipulation, magnetic forces encounter limitations due to the complexities involved in achieving precise control and inherent instabilities of the system.

In the continuation of this section, I turn the focus to interfaces that leverage magnetic forces to create inter-atom bonds, instead, while the actuation is performed by an external (deterministic or stochastic) direct force.

Dynablock [140] uses a pin-based display to assemble arbitrary 3D objects out of magnetic cubes that use permanent magnets to connect both horizontally and vertically (see Figure 2.6(A)). The system can create objects with up to 8 layers, and in contrast to the actuated pin-based displays discussed in Section 2.1.2, the *Dynablock* [140] interface is able to reconstruct overhangs. It lacks, however, the ability to support dynamic self-reconfiguration. Reprogrammable magnetic pixels [98], on the other hand, use stochastic self-assembly of cubic modules covered in a reprogrammable magnetic material. The cubic modules are pre-programmed using a magnetic plotter, such that they are maximally attractive to their intended counterpart and agnostic to all others. The modules can be later reprogrammed to form a different target arrangement. A self-assembly demonstration with 8 modules is shown in Figure 2.6(B). Compared to deterministic methods, stochastic self-assembly sacrifices efficiency and predictability for better scalability and cost-effectiveness. Nonetheless, the process will have to be optimised in the future to reduce the current extremely long assembly times.

In summary, due to the additional mechanical parts required for each added column, an implementation using mechanically-actuated assembly of the magnetic components, such as *Dynablock* [140], is difficult to scale. In comparison, stochastic assembly with reprogrammable magnetic pixels [98] is easier to scale to a large number of passive atoms. Nonetheless, the technology would need to get several orders of magnitude faster in the future, i.e., reduce assembly time from hours to ms, to be a viable option for constructing the *ultimate display*. Additionally, automating the separation of the passive magnets without the need for an external intervention as well as implementing rapid reconfiguration are challenges remaining to be solved for magnetically connected passive atoms, regardless of the type of assembly.

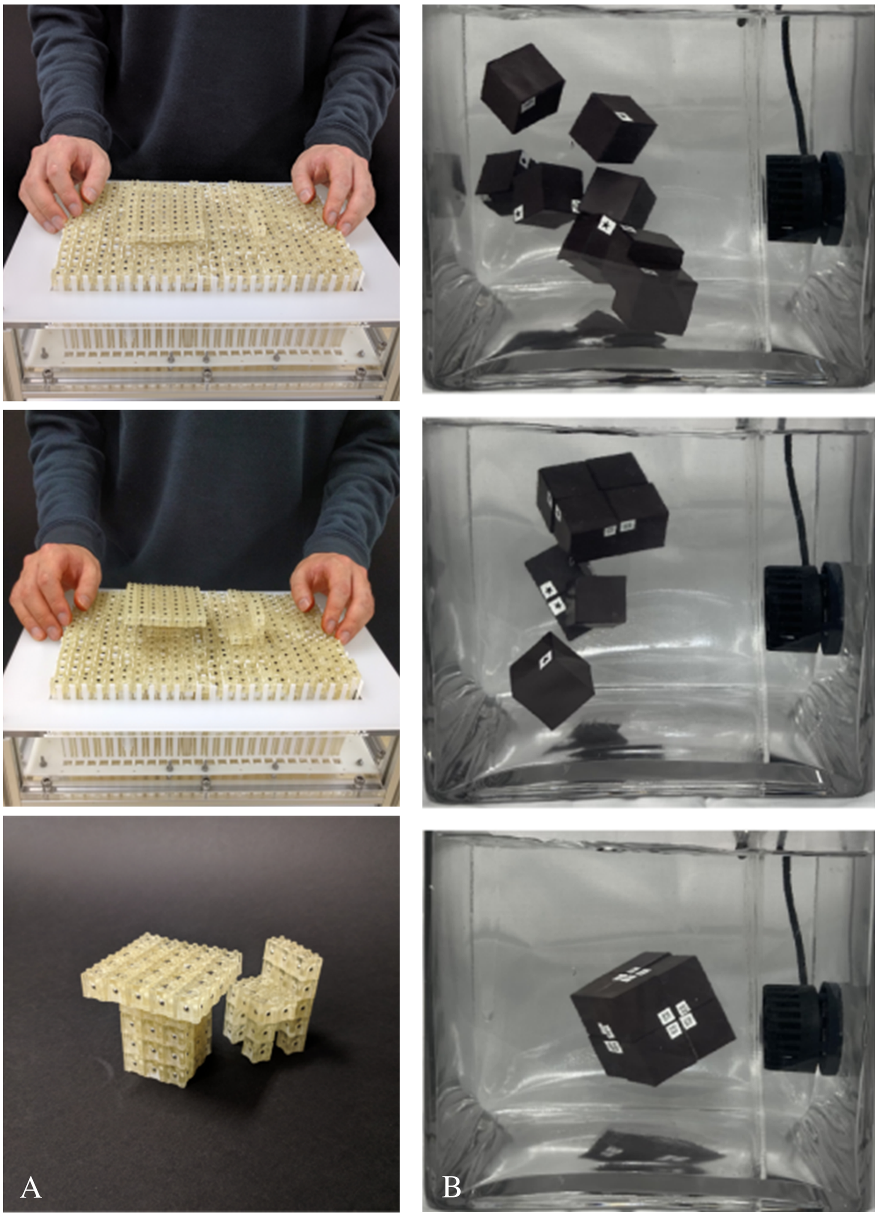


Figure 2.6 Using magnets to create bonds between the passive atoms. (A) *Dynablock* [140] uses a pin-based actuator to assemble an arbitrary physicalisation using 9.4mm wide blocks with embedded omnidirectional permanent magnets. (B) Self-assembly process of 8 magnetically pre-programmed 25mm wide cubes in a tap water container, equipped with a hydraulic pump that stimulates Brownian motion. The cubes self-assembled in the target shape within 32 hours [98]. Figures reprinted from [140, 98].

2.2.3 Laser-based Displays

Lasers are a powerful source of coherent light that has been utilised in HCI research to develop innovative displays suited for diverse environmental and situational conditions, and presented on a range of different surfaces and spaces. Some notable examples include: the large-area laser-projected interactive floor display *BaseLase* [91], a wristwatch with on-arm projected graphics and skin-based input called *LumiWatch* [152], a drone-assisted light field display facilitating telepresence [156], and various HMD and stationary projection-based AR systems for industrial training tasks [128, 20]. In this section, in line with the *ultimate display* vision, the focus lies on laser-based displays that are able to present volumetric content. Blundell [12] defines a volumetric display as a device that *permits the generation, absorption, or scattering of visible radiation from a set of localised and specified regions within a physical volume*. Accordingly, the displays discussed below use laser light to generate volumetric images in transparent physical space that can be viewed by multiple unadorned users simultaneously, from many different angles. Depending on whether the system actively emits light to generate the visual content, or reflects incident light, I categorise the displays as light-emitting or light-scattering type, respectively.

Light-emitting Type

The light-emitting type of laser-based volumetric displays use high-intensity lasers to excite air particles to emit light in an arbitrary position in 3D space. When a high power laser beam is focused in a gas, e.g., in air, the gas is ionised and air plasma emission occurs. This phenomenon is called *laser-induced breakdown*, and it is accompanied by a short, but intense bluish-white light emission (see Figure 2.7). One of the main advantages of this type of displays is that they eliminate the need for a reflective screen or any other additional material to be suspended in mid-air to produce the visualisation. The volumetric content is created by generating many light emission pulses in close succession on predefined positions in the physical space. Air plasma pulses have a very short duration, however, due to the Persistence of Vision (PoV) effect [14], human eyes are able to integrate visual stimuli, e.g., multiple air plasma pulses, that occur within an interval of around 100ms into a single image. In addition, since the air plasma pulses emit high-intensity light, an afterimage occurs, and the total interval where the image is persistent is roughly estimated to be 200ms [58]. Thus the displays operating on this or similar principle should render all the points comprising the volumetric content within this interval, so that the user can experience the full content at once.

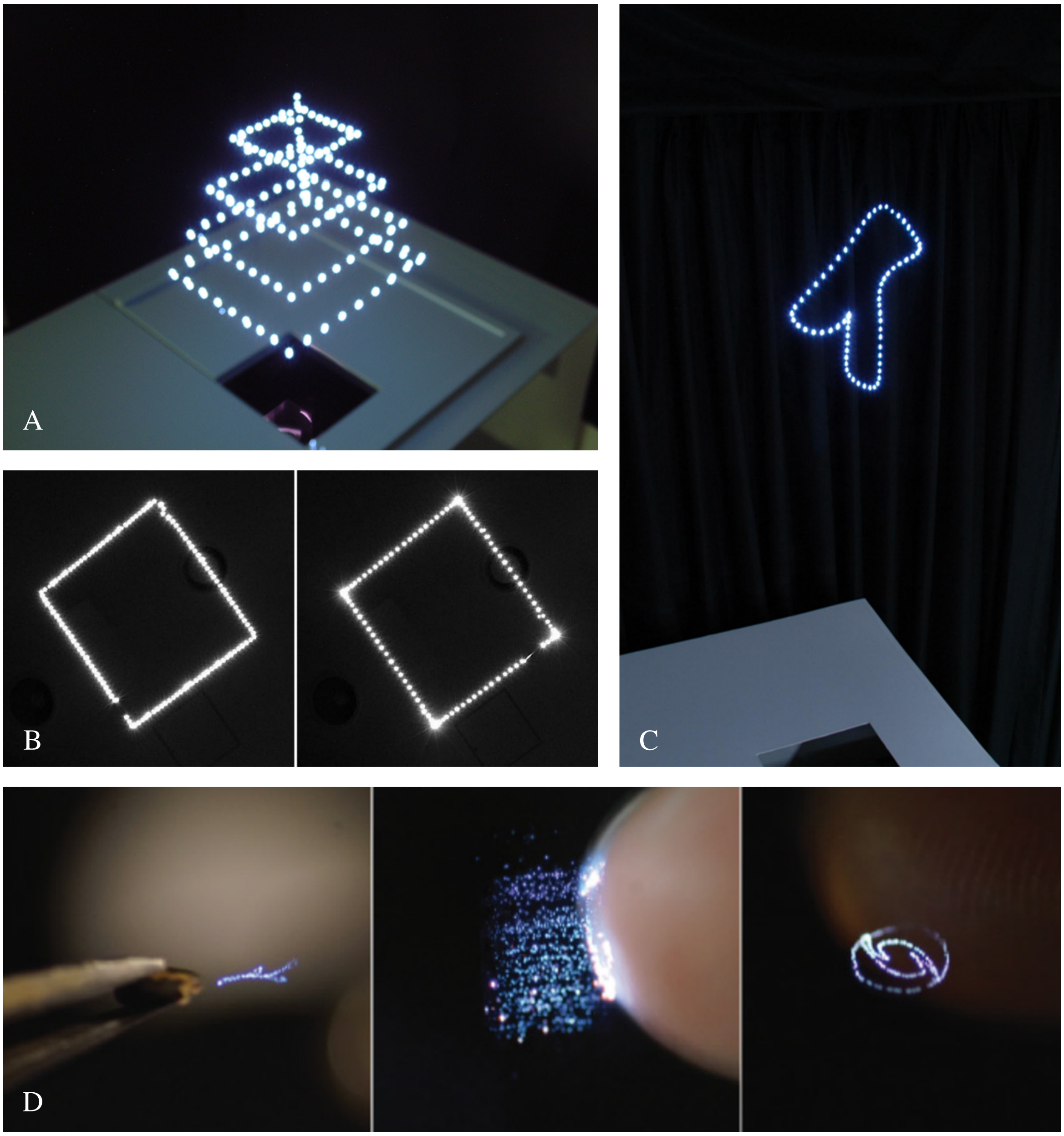


Figure 2.7 Light-emitting laser-based volumetric displays with dots of air plasma as passive atoms. (A) A 3D graphic produced by an air plasma scanning display using a nanosecond infrared laser [65]. (B) Comparison of rendering a square with a femtosecond laser with 128 pulses, using equally spaced intervals (left) vs. intervals adjusted such that the acceleration profile of the laser scanner is smoothed (right) [58]. (C) A wireframe of a Japanese syllabary generated by a femtosecond laser [127]. (D) Example of different graphics rendered using femtosecond lasers: (left) sprout coming out of a seed, (middle) user interaction with a point cloud of air-plasma, and (right) a logo [102]. Figures reprinted from [65, 58, 127, 102].

An early laser-based volumetric display prototype from 2006 [65] used an infrared pulsed laser to generate 100 dots of air plasma per second, where the emission time of each light pulse was in the nanosecond order of magnitude. The system produced 2.5D visual content, shown in Figure 2.7(A), by incorporating galvanometric mirrors to scan the two lateral dimensions, and a linear motor system that allowed manipulation along the vertical dimension. Out of the 100 air plasma dots generated per second, only around 20 can be simultaneously observed, resulting in a relatively sparse image. The number of possible dots is restricted by the speed of the scanning system, and the laser frequency, as one laser pulse generates a single dot. In addition, the bluish-white colour of the air plasma dots is fixed and determined by the display medium, which can limit the expressivity and variability of the display. The main concern with the display, however, is the safety aspect. Due to the high laser power and the relatively long pulse duration, eye protection with infrared filter should be worn, and direct interaction with the air plasma content should be avoided to minimise skin exposure. In order to produce more accurate and easy-to-execute graphics, Ishikawa et al. [58] proposed smoothing the acceleration profile of the laser scanner using moving averages method. A comparison of rendering a square before and after the acceleration adjustment is shown in Figure 2.7(B).

A later prototype used a femtosecond instead of the nanosecond laser, which allowed for the generation of 1000 dots of air plasma per second [127]. Due to the increased resolution and the improvements in the 3D scanning system, the updated display was able to produce smoother graphics than the previous prototype, illustrated in Figure 2.7(C). Ochiai et al. [102] integrated a spatial light modulator (SLM) in the setup of a femtosecond laser (repetition frequency 1000 dots/s, pulse duration 30fs) with the aim to increase the spatiotemporal resolution of the display. The SLM modifies the phases of the light rays in order to generate multiple light distributions using interference. Using this method, up to four visible air plasma dots can be generated in the lateral dimension simultaneously, resulting in a spatiotemporal resolution of up to 4000 dots/s. As it can be observed in Figure 2.7(D), when multiple air plasma dots are simultaneously produced, they are less bright compared to when a single dot is produced (Figure 2.7(A-C)), since the energy is split between the multiple light distributions. To probe the feasibility of direct user interaction with this type of air plasma visualisations, safety tests were conducted. In the safety experiments, exposure durations from 50ms to 6000ms were tested on leather using the femtosecond laser. The results showed that heat damage occurred for exposure times grater that 2000ms (2000 pulses). The same experiments were repeated with a nanosecond laser, where heat damage appeared on the leather only after 100ms of exposure. Hence, the technology should be handled with caution, and in the presence of trained individuals, as it is not yet mature for widespread use.

Light-scattering Type

Light-scattering laser-based volumetric displays direct laser light through a particle-filled space in a controlled manner, and the particles scatter the light to form the image in physical space. Below, I discuss systems that use a variety of methods, and different types of light-scattering particles to generate the volumetric image. These include dust, or fog particles suspended in mid-air as well as microbubbles generated in a high-viscosity liquid.

The volumetric display by Perlin et. al [114] generates the image by scanning an optically transparent medium such as air, and illuminating a suspension of light-scattering dust particles using a collimated light source. An infrared laser is used to scan through the display volume and detect dust particles, which are then illuminated by an RGB laser. The brightness of the visible beam is modulated in time based on the direction and distance to the dust particle. The display volume is scanned 50 times per second, and the dust particle concentration is maintained with an air curtain boundary. The quality of the image depends on the particle concentration, as higher concentrations produce more recognisable images. However, issues with image rendering such as undefined edges, non-homogeneous surface, and noise from illuminated neighbouring particles persist even with high particle concentrations. A teddy bear visualisation created using lint particles is shown in Figure 2.8(A).

The *RayGraphy* [153] display uses weak laser beams to generate volumetric graphics in physical spaces filled with fog. The interface consists of 32 projectors arranged in a circular formation, with each projector equipped with a single-board computing unit and mounted on a rotatable bi-axial head for adjusting the position of the rendered graphics. The display relies on the presence of fog, with smaller particles that remain suspended in the air for longer durations being preferred. In the prototype, a water-based liquid was used to generate fog particles. Figure 1.8(B) showcases various visualisations produced using *RayGraphy*. The system faces limitations in terms of low contrast caused by ambient light that is not part of the target image. The resolution of the images is also low, and the system's ability to render fine details is limited, making some images difficult to recognise.

The volumetric bubble display [70], depicted in Figure 2.8(C), utilises light-scattering microbubbles produced by a focused femtosecond laser in a high-viscosity liquid medium to create 2D and 3D graphics. However, there are some limitations to this approach. The content is created within a sealed container, making it inaccessible for direct interaction by users. Next, the shape retention of the content is of around 30s before the microbubbles start to burst, and aberrations are possible due to differences in the refractive index. Finally, the exact retention period is difficult to control and to allow for faster updates some form of a precise bubble bursting mechanism would have to be engineered in the future.

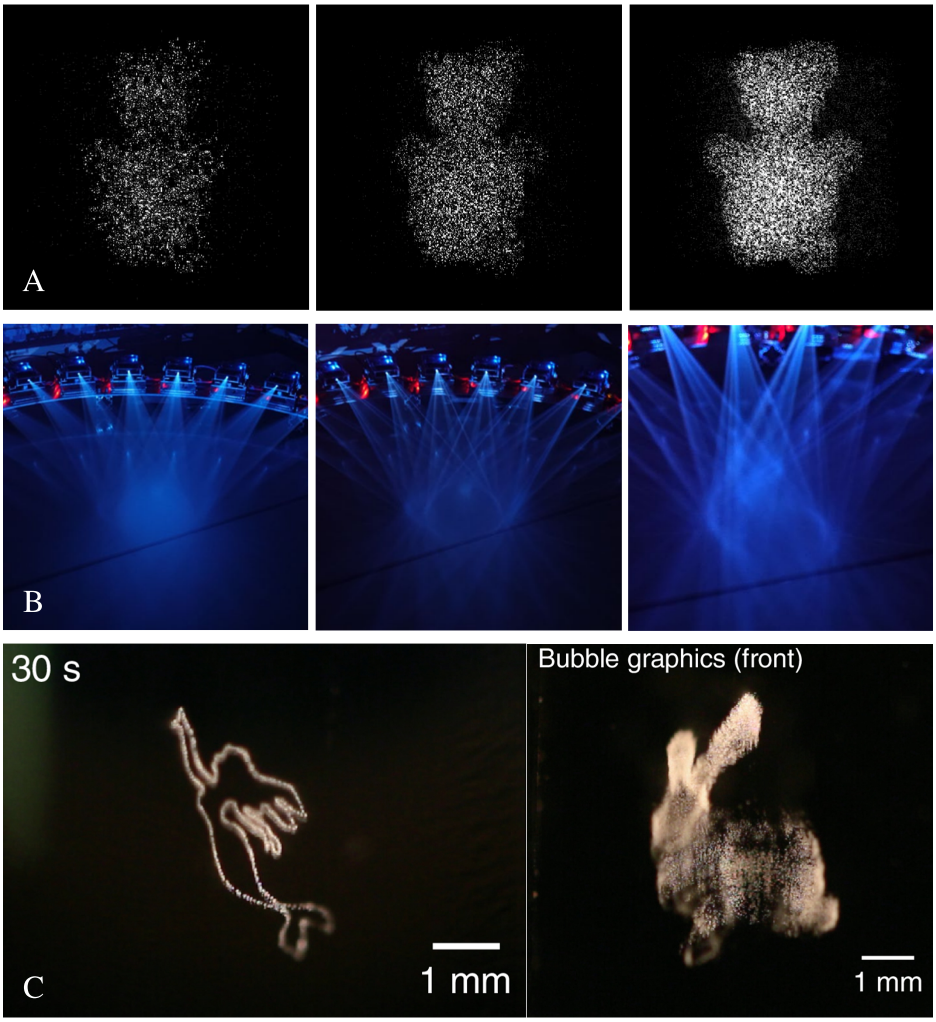


Figure 2.8 Light-scattering laser-based volumetric displays with dust, fog, and microbubble particles as passive atoms. (A) Volumetric image generated by illuminating 1070, 3400, and 10600 dust particles out of a 10000, 32000 and 100000 particle cloud, respectively [114]. (B) Each projector of the *RayGraphy* [153] interface projects a 2D perspective image of the target graphic according to its angle of view, and position in space. The final visualisation is rendered by the superimposing beams of the individual 2D images. The two visualisations of the Utah teapot (left and centre) illustrate that the system produces better rendering quality when projecting only the contours of the target image. (C) The volumetric microbubble display uses a femtolaser that emits 1000 pulses/s of 100fs, in a high-viscosity liquid (glycerine) sealed in a glass cell of size 10mm×10mm×10mm. (left) 2D bubble graphic of a mermaid generated at a pulse energy of 4.3μ J showing shape retention of 30s. When the graphics are generated with a higher energy, the shape retention interval shortens due to the increased diameter of the microbubbles, causing them to burst sooner. (right) 3D bubble graphic of a bunny, generated by scanning 100 2D outlines of cross-sectional images. The non-uniformities, most noticeable towards the centre of the bunny, are cased by aberration due to the different refractive index of glycerine and air [70]. Figures reprinted from [114, 153, 70].

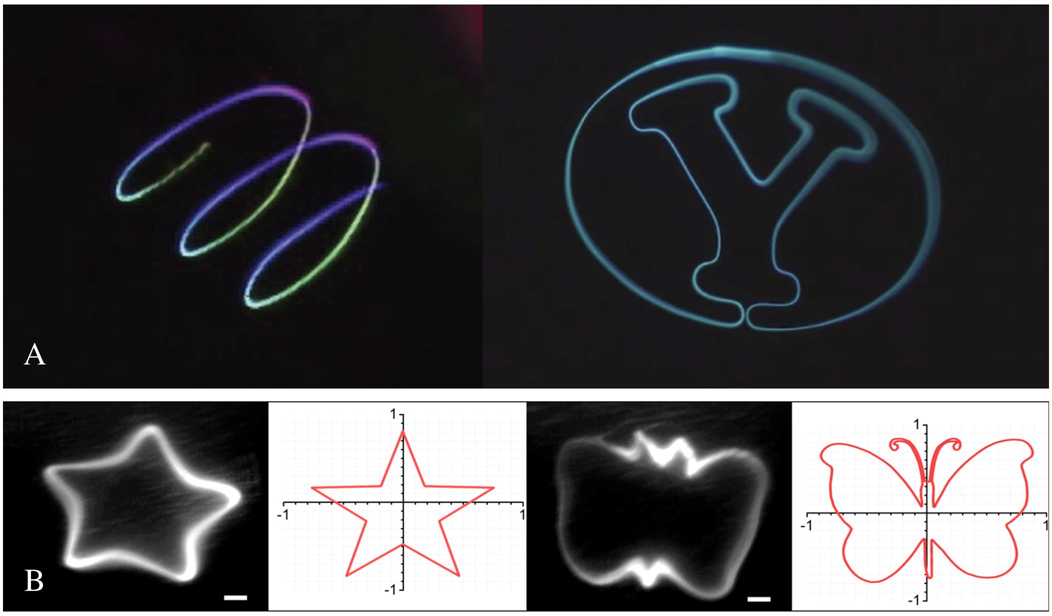


Figure 2.9 Light-scattering volumetric PoV displays using optical and electric levitation. (A) Centimetre-sized graphics generated by scanning a cellulose particle trapped in a photophoretic trap at PoV rates [132]. (B) A time-varying electrical field, combined with a specific electrode arrangement, is employed to trap an electrically charged 100nm-wide gold particle colloid and trace submillimetre-sized images at a frequency of 25Hz [11]. The respective reference trajectories consisting of 4000 points are shown in red. The white scale bar corresponds to a length of 10μm. Figures reprinted from [132, 11].

The last two interfaces discussed in this section, similar to the light-emitting displays shown in Figure 2.7, utilise the PoV effect to generate visual content in 3D space. However, instead of producing individual light points, they employ a single levitating particle that scatters incident light to form the desired shape. One such interface is the Optical Trap Display (OTD) developed by Smalley et al. [132]. The OTD generates content in mid-air by trapping a micrometer-sized opaque particle in a photophoretic trap using a nanosecond laser. The optical trap is formed by a combination of oblique astigmatism and spherical aberration. The trapped particle, which is less than 10μm in size is made of black liquor (a cellulose solution). By scanning the optical trap at a rate greater than 10 times per second, simple vector graphics of approximately 1cm in width can be generated (see Figure 2.9(A)). To achieve full-colour 3D images, the particle is illuminated by a system of collinear RGB lasers. The rendering capabilities of the OTD are limited by the quality of the optical trap, the range of materials suitable for optical levitation, and the speed of the scanning system.

In the previous systems discussed in this section, the scanning velocity was limited due to the (partial or complete) reliance on mechanical displacement of optical components, such as galvanometric mirrors. Figure 2.9(B) presents a system where the trajectory of the light-scattering particle is fully driven electrically [11]. This is achieved through a specific electrode arrangement comprising inner and outer ring electrodes for controlling vertical displacement, and four compensation DC electrodes for controlling lateral displacement of the charged levitated particle. The electric field is then varied is such a way that the electrostatic potential features a saddle shape where the particle can be trapped. The image is then formed by displacing the particle within the electric field while illuminating it with collimated laser light. Although the high update frequencies of the electric field (150kHz) show promise for traversing longer particle paths in PoV time, only planar submillimetre-range images of limited accuracy have been demonstrated thus far (examples in Figure 2.9(B)). To render larger and more detailed images, the use of high voltage amplifiers and larger electrodes would be necessary.

Summary

In this section, I discussed different laser-based passive atom approaches to generate volumetric images in real physical space. Based on their operating principle, the approaches were categorised into two types: light-emitting and light-scattering. The light-emitting type involves scanning the display volume with a high-intensity laser to sequentially produce air plasma emissions. The number of dots generated in this manner is limited by the repetition frequency of the laser and the speed of the 3D scanning system. The colour of the dots is determined by the properties of the air plasma, and it is limited to bluish-white [65, 127, 102]. The light-scattering type requires a single ([132, 11]) or multiple particles ([114, 153, 70]) to be suspended in a fluid within the display volume. The image is created as the particles scatter laser light directed at them. These type of displays can be more complex and costly to build, and they require particular environmental conditions to operate, e.g., a dark indoor space filled with fog [153]. Factors like room temperature, humidity, and air particle settling can potentially impact the stability and robustness of these displays.

A significant constraint shared by both light-emitting and light-scattering approaches is the safety risks associated with laser usage. The concentrated high-energy laser beam poses a potential threat to the retina and skin if direct contact occurs. It is crucial to prevent unscattered laser light from entering the eye, as even scattered light, although less hazardous, should still be handled with caution. The danger is heightened when the irradiation beam is invisible, as individuals may not be aware of its presence, and the instinct to blink or avert gaze is not triggered. Therefore, it is imperative to ensure that users interacting with the optical volumetric display do not align their line of sight with the laser beam. Consequently, while the volumetric images are visible from a complete 360° range, a slight reduction in the viewing angle may be necessary for safety purposes. At present, due to their complexity and safety implications, the discussed laser-based volumetric displays cannot be operated independently by an average user, highlighting the need for further development before they can be deployed.

2.2.4 Acoustic Levitation

One promising technology that could one day potentially lead to the *ultimate display* is acoustic levitation. Acoustic levitation interfaces use the mechanical energy of ultrasound to suspend small objects in mid-air. The hardware typically consist of Phased Arrays of Transducers (PATs) that emit ultrasonic waves of a 40kHz frequency (inaudible for humans). By appropriately setting the phases of the individual transducers, *acoustic traps* can be generated, i.e., regions where the acoustic forces converge and objects can be trapped. As opposed to other types of levitation, such as, magnetic (2.2.2) or electric (2.2.3), with acoustic levitation there are no special requirements on the type of objects that can be levitated, e.g., in terms of their state, material, or with it associated properties (magnetic, electrostatic, etc.). Thus far acoustic levitation of solids, liquids, crystals, and gels has been demonstrated, ranging from metals and synthetic materials, to food and beverages [82, 146, 101]. Additionally, acoustic levitation allows for more accurate, fine-grained position control of the levitated object compared to fluid (2.2.1) and magnetic (2.2.2) levitation, and it is safer and more affordable compared to the optical methods (2.2.3). This section discusses notable acoustic levitation systems. Please note that this overview covers the research landscape up to 2019, with subsequent literature discussed in Chapter 8.

The *TastyFloats* [146] system showcased the use of acoustic levitation in a food delivery interface, enabling new gustatory experiences by levitating liquid and solid food morsels. The acoustic trapping of the food morsels is implemented only using the spatial arrangement of the transducers, shown in Figure 2.10(A). This means that all the transducers are driven by a single signal, i.e., all waves have the same phase. Although this significantly simplifies the control effort, it also limits the possibilities for particle movement control. *Floating Charts* [105], on the other hand, is a modular display consisting of multiple arrays of transducers, arranged in a hexagonal flower pattern. Each modular unit, as shown in Figure 2.10(B), independently moves a levitated particle, that represents one data point in the floating chart. The units are positioned at a distance from each other to allow for independent vertical movement of each particle without mutual interference, providing unobstructed views and natural depth perception. The *JOLED* [126] levitation display employs electrostatic rotation to change the appearance of the levitated passive atoms. By applying thin dielectric coating on the particles and utilising transparent electrodes surrounding the display volume (Figure 2.10(C)), a tailored electric field is created that can control the orientation

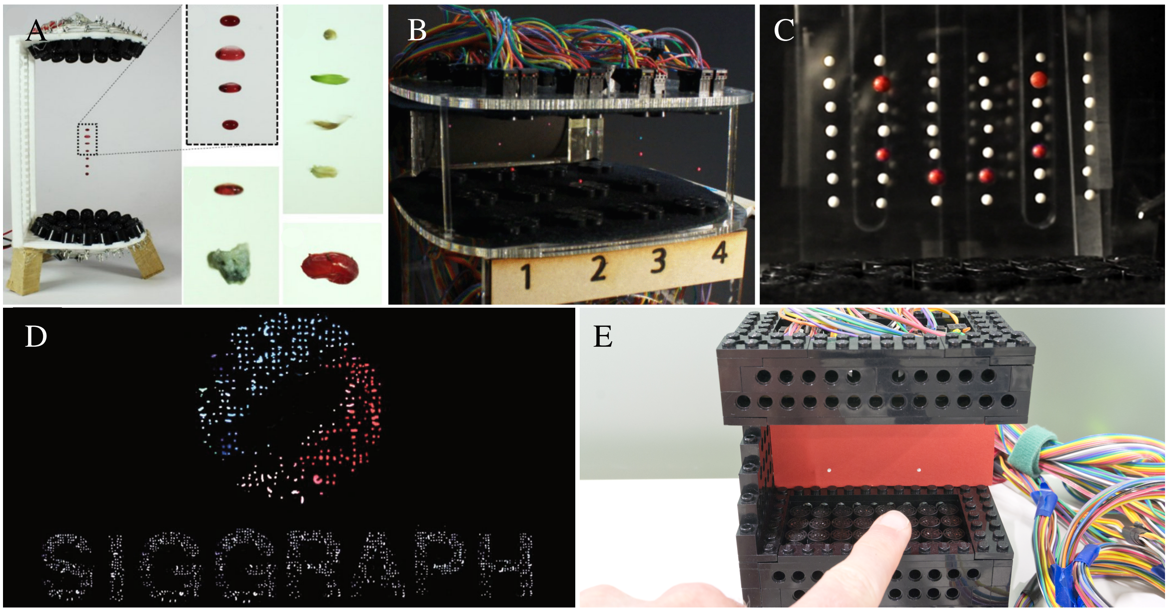


Figure 2.10 Acoustic levitation displays using individual levitated particles as passive atoms. (A) The *TastyFloats* [146] contactless food delivery system utilises a single axis levitator comprising a total of 72 transducers. At the centre, a column of acoustic traps is created, enabling the levitation of various food morsels such as wine, cheese, and raspberry grains. (B) A *Floating Charts* [105] display consisting of 3 rows and 4 columns of opposing flower pattern arrays, resulting in a display volume of 14cm$\times$17cm$\times$8cm. Each levitated EPS particle, with a diameter of 2mm, is positioned vertically in proportion to its corresponding numerical value derived from tabular data. (C) The *JOLED* [126] display utilises electrostatic force to independently flip individual levitated particles between their white and red sides. The display configuration includes two 3$\times$8 arrays of transducers, which levitate a 2D grid comprising 7 rows and 6 columns of particles. The display area spans approximately 55mm$\times$29mm. (D) A logo generated by the *Pixie Dust* [101] graphics system, featuring a 2D grid of acoustically levitated particles. The grid consists of up to 85$\times$85 particles, spaced at a distance of 4.25mm (approximately half the wavelength of 40kHz ultrasound), and a positioning accuracy of ca. 0.5mm. (E) A *Point-and-Shake* [37] interaction technique that allows the user to select a levitating particle by pointing at it. Successful selection prompts dynamic feedback in the form of shaking. The levitation display consists of two 8$\times$4 PATs, positioned at a vertical distance of 65mm, along with a Leap Motion sensor that registers the pointing hand gestures, i.e., finger position and extension. Figures reprinted from [146, 105, 126, 101, 37].

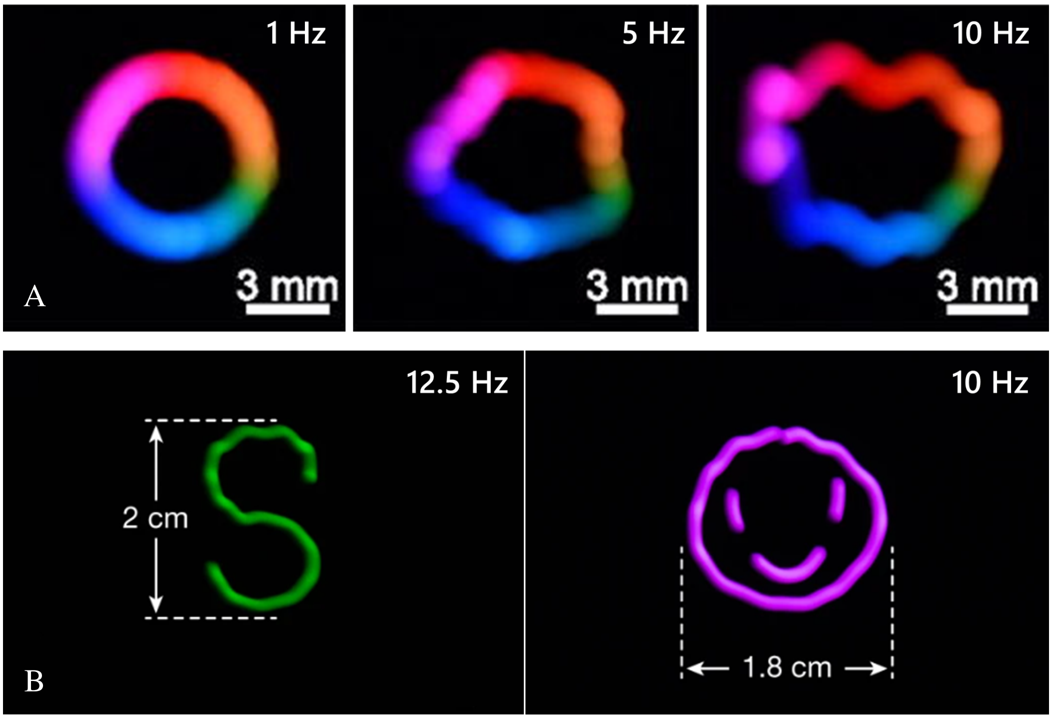


Figure 2.11 Acoustic levitation displays using the PoV effect. (A) A circle of 6mm diameter generated with a rendering frequencies of 1, 5, and 10Hz. This was achieved using a levitator consisting of two PATs of 30 transducers each, with a working volume of 4cm×5cm×8cm [40]. (B) The MATD is composed of two 16×16 PATs at a distance of 23.4cm, surrounding a working volume of 10cm×10cm×10cm. (left) A 1cm wide and 2cm tall letter, generated with a rendering frequency of 12.5Hz and particle speeds of up to 0.8m/s. (right) A smiling face of 1.8cm diameter, rendered at 10Hz, with particle speeds of up to 1.3m/s [53]. Figures adapted from [40, 53].

of individual particles. Figure 2.10(D) shows a mid-air raster graphic created by the *Pixie Dust* [101] graphics system. By employing four PATs positioned pairwise to face each other, a 2D grid of acoustic traps is generated. The particles that are not part of the final graphic are subsequently removed (using an additional PAT or an air jet), and the rest are illuminated with a projector. Freeman et al. proposed the *Point-and-Shake* [37] interaction techniques for selecting individual levitating particles in an acoustic levitation display. This approach combines mid-air pointing using ray-casting, as depicted in Figure 2.10(E), with visual feedback in the form of side-to-side particle movement.

By pushing the boundaries of acoustic levitation, higher update rates of the transducer boards have made it possible to achieve acoustically levitated graphics in PoV time. Similar to the optical and electrostatic levitation systems discussed earlier [132, 11], PoV graphics using acoustic levitation are generated by rapidly moving a levitated particle along a periodic path within the display volume. Figure 2.11(A) shows a system that is able to render simple shapes of 5-6mm size in PoV time using a particle of 1.5mm diameter, and maximal recorded particle velocity of 0.67m/s [40]. The Multimodal Acoustic Trap Display (MATD) [53] has further enhanced performance by offloading the computation of acoustic traps on hardware level using an FPGA. This advancement enables the rendering of simple graphics with a width of approximately 2cm in PoV time, as shown in Figure 2.11(B). Moreover, the MATD incorporates techniques such as time multiplexing with a secondary trap, amplitude modulation, and phase-difference minimisation to deliver haptic and auditory feedback without the need for additional hardware, leveraging only the acoustophoretic working principle.

2.2.5 Summary

In this section, I delved into a diverse range of systems that utilise passive atoms to create visual content in physical space. From the manipulation of passive atoms using air and water flow to magnetic assemblies, laser-based volumetric displays, and acoustic levitation, these technologies demonstrate the potential for generating captivating visual experiences. However, the different technologies come with their own set of limitations, which can be summarised as follows:

- Safety concerns: Laser-based systems pose risks to users due to the concentrated high-energy laser beams, which can potentially cause damage to the retina or skin. Proper precautions, such as wearing protective gear and avoiding direct exposure to the laser beam, are necessary.

- Environmental requirements: Some systems require specific environmental conditions, such as a dark indoor space filled with fog ([153]) or the use of a high-viscosity liquid medium ([70]). Meeting these conditions may limit the practicality and accessibility of the displays.

- Control and manipulation limitations: Systems utilising air or water flow ([5, 155, 116, 148]) have inherent challenges in precisely controlling particle movement, which can greatly restrict the ability to achieve higher display resolutions. Similarly, displays relying on magnetic force manipulation only allow for very close-range levitation ([73]), while magnetic tabletop systems ([113, 139]) have proximity constraints, which limit the range of possible visualisations and interactions.

Among the various approaches discussed, acoustic levitation stands out as a highly promising approach to pursue the *ultimate display* vision in the future. Unlike other systems that have to (at least partially) rely on mechanical actuators for displacement, acoustic levitation achieves movement in all spatial dimensions solely through the acoustic force exerted by ultrasonic waves. This unique characteristic provides a firm basis for developing a high-performance versatile physical interface. By carefully designing sound waves, acoustic levitation can suspend and precisely manipulate particles in mid-air, enabling intricate visualisations with a wide range of particle arrangements and movement patterns. While there are still challenges to overcome, such as controlability, stability, and robustness of the complex dynamical system behind the interface, acoustic levitation shows great potential for creating dynamic, interactive, and multimodal interfaces that can enhance user experiences in physical spaces. Therefore, in this dissertation, I propose methods and algorithms for achieving smooth and accurate movement, computing time-optimal particle trajectories, and conducting dynamics simulations, among other advancements, to further advance the field of acoustic levitation and drive innovation towards achieving the *ultimate display*.

Chapter 3

Dexterous Interaction with a Levitating Object

3.1 Introduction

One of the longest-standing visions in Human-Computer Interaction is that of the "Ultimate Display" [138]. This entails a room in which the computer can control the existence of matter. The computer could create chairs, or bullets in such a room, and the virtual and physical worlds would truly be merged. Undoubtedly, there are several challenges towards actually creating such an "Ultimate Display".

One approach to creating an "Ultimate Display" envisions the existence of Programmable Matter [44]. The programmable matter would consist of millions of miniature robots, called Claytronic Atoms. The main difficulties associated with this "active atom" approach are ensuring sufficient and reliable power supply to the individual units, costs per unit, and miniaturisation of the units. Ishii et al. [57] explore the possibilities, opportunities, and implications for HCI of developing interfaces that can generate physical manifestations of computational information. In their concept of "Radical Atoms", they propose that such interfaces should *transform* their shape to reflect computational state and user input, *conform* to constraints imposed by environment and user input, and *inform* users of their transformational capabilities.

To overcome the limitations of the "active atom" approach, an alternative is to use "passive atoms". In this approach, the "atoms" themselves are passive, but actuation, power supply and intelligence are provided by the environment. This solves the three problems referred to above. One solution to provide actuation from the environment is to use sound

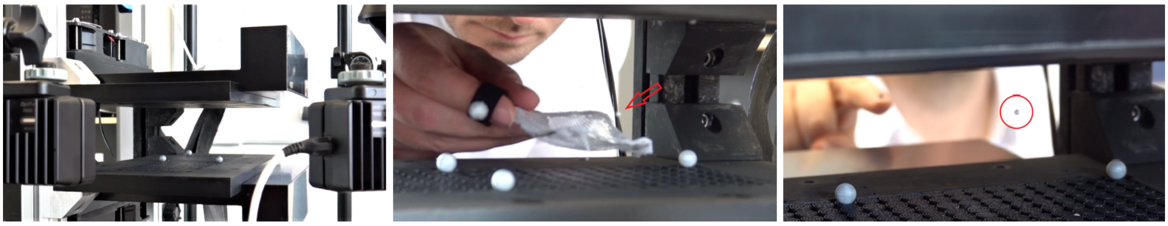


Figure 3.1 LeviCursor enables dexterous interactive control of levitated particles. Users can control particle motion using an optical marker attached to a fingernail. Because we use the optimisations-based approach to ultrasonic levitation, particle motion is smooth in any direction. We achieve round-trip latencies of 15 ms, sub-millimetre accuracy, and stability in levitation.

waves that exert forces on the “atoms”. This concept was for example employed by Pixie Dust [101].

Pixie Dust uses phased arrays of ultrasonic transducers to generate acoustic standing waves and create a grid of nodes, where small objects can be levitated. Pixie Dust uses this method to introduce a novel graphics system, where physical visualisations are created using levitated particles and visual projection.

From a Human-Computer Interaction perspective, an important problem of the “Ultimate Display” approach is how to interact, for example, how to move particles interactively. Pixie Dust [101] explores interactive particle control methods, but the use of the classical standing-wave approach for trap generation, introduces limitation in the movement of the particle. While the traps are quite stable, smooth movement is only possible in one dimension between two opposing arrays or array and reflector. Thus smooth movement in 3D would require six opposing arrays. This would significantly impair the visibility of the levitating display. The Pixie Dust setup consists of four transducer array, allowing for smooth particle movement in 2D. In addition, levitation is only possible within the boundaries of the arrays and is limited to parallel array arrangements.

LeviPath [103] provides an algorithm for moving levitated particles with two opposing arrays, on a 3D grid. The phase values at each step, of approximately 0.2 mm, are precomputed and stored in a table. However, as shown in the video [104], the particles move at a relatively low speed and experience some jittering.

JOLED [126] uses optimisation for phase computation, which enables smoother particle movement than the pure standing-wave approach. The JOLED setup is composed of 60 transducers in total. Due to the low number of transducers, real-time particle control using mouse or keyboard is possible. As a consequence, however, the display volume is relatively small.

In general, particle interaction has been implemented at relatively low speeds, due to the risk of particles being dropped during high speeds or high accelerations. In summary, although interaction with levitating particles has been explored by Ochiai et al. [101], Omirou et al. [103] and Sahoo et al. [126], real time gesture interaction with a particle moving along a smooth 3D path with high speed has not been yet achieved.

In this chapter, we address the problem of dexterous interactive movement of levitated particles. The main difficulties in achieving this with homogeneous movement along all three dimensions are ensuring: (1) low latency, (2) continuous movement without steps, and (3) stable movement enabling high velocities and accelerations.

We use the optimisation-based approach of Marzo et al. [86], avoiding the inhomogeneties of the classical standing-wave levitation. The main limitation of applying the optimisation approach in [86] to our setup is that, due to the larger number of transducers, optimisation takes about 1 s for each levitation point, thus preventing interactive rates. We solve this problem and achieve (1) low latency by precomputing optimal phases for *all possible levitation points within the entire array at 0.5 mm resolution*, resulting in a round-trip latency of 15 ms. Jumps of the trap, even if just by 0.5 mm, result in noticeable jumps followed by oscillations of the particle. We achieve (2) continuous movement by interpolating between the precomputed levitation points at 1 kHz, achieving arbitrarily small step sizes. Optimisation creates numerous weaker traps in the vicinity of the main trap. Previously, when placing the particle, one could not be sure that it is actually located in the main trap. Furthermore, over time, the particle might jump to weaker secondary traps, resulting in offset and reduced stability. We achieve (3) stable movement by providing a mechanism to stabilise levitation and ensure the particle is always in the main trap.

The *LeviCursor* method can be beneficial for studies and applications involving 3D selection with a physical object as cursor, where the correct perception of the 3D targets and the 3D cursor is crucial. It provides a novel method of interacting with tangible interfaces, while opening up new research questions in the HCI community concerning perception, motor control and transfer function of physical cursors which are detached from the user's body. In addition to pointing and selection, precise and accurate manipulation of levitating particles can be used to improve graphical visualisations and animations in mid-air [101], provide better gaming experience in levitation-based games [126] as well as facilitate containerless handling and mixture of sensitive materials, i.e. lab-in-a-drop [22], in favour of preventing contamination.

3.2 Related work

3.2.1 Acoustic Levitation

Acoustic radiation force can be used to counteract gravity and trap millimetre-sized objects in mid-air. This effect is most often achieved by using phased arrays of ultrasonic sound emitters of the appropriate phase and amplitude to create acoustic nodes in mid-air, where particles can be trapped. Acoustic levitation does not require any special (e.g. optical, magnetic, electric etc.) properties of the levitating object. Therefore a variety of objects can be levitated, including solids, liquids and insects [82]. Furthermore, particles of smaller (i.e. Rayleigh particles) [101] and larger (i.e. Mie particles) [83] radius than the wavelength have been levitated.

3.2.2 Moving Levitated Particles

A few methods for achieving controlled movement of levitating particles in the acoustic field have already been developed. *LeviPath* [103] employs an algorithm which combines basic patterns of movement to levitate objects across 3D paths, in a setup consisting of two opposed arrays of transducers. The input path is decomposed into a height variation, controlled by the phase difference between the top and bottom transducer array and a 2D path. The 2D path is then adapted to a possible pattern, obtained by interpolation between adjacent pairs of levitating points. In addition to controlled translational movement in the field, controlled rotations have also been achieved, but with the help of electrostatic forces. In *JOLED* [126] levitating particles of different physical properties are coated with titanium dioxide in order to induce electrostatic charge. This allows for the control of the angular position of the particles by the means of electrostatic rotation. The 3D position of the particles is determined by optimising the phases of the acoustic arrays.

3.2.3 Interaction with Levitated Particles

For the purpose of contactless manipulation of particles using acoustic levitation, the wearable glove *GauntLev* [80], with integrated ultrasonic transducers, has been designed. The *GauntLev* gloves trap particles either in front of the palm or between a pair of fingers, enabling a set of basic manoeuvres such as capturing, transferring and combining levitating particles, a process performed manually or computer assisted. Alternative devices that can be used to manipulate levitated particles which are not attached to the hand are the *Sonic Screwdriver*, a parabolic head with a handle that can generate twin traps and *UltraTongs*, tweezers that generate standing waves [80]. Some of the configurations in [80] and [86] support one-sided levitation, which provides very good display visibility, but achieving fast and stable levitation is more challenging.

Concerning levitation with static acoustic elements, thus far, only interaction techniques for selection and step translation of particles have been developed. With *Point-and-Shake* [37], users can point a finger to select levitating objects and receive visual feedback in the form of a continuous side-to-side (*shake*) movement. The hand gestures are tracked using a Leap Motion sensor. Interactive control of a single levitated particle using a keyboard, mouse, GUI buttons and a Leap Motion sensor was presented in *LeviPath* [103]. The particle was moved by small steps on a 3D grid. The *Pixie Dust* [101] setup comprises four vertical transducer arrays, facing inwards, which generate a 2D grid of acoustic nodes. Interactive techniques were tested either by using Kinect to detect users' hand gestures, which were then mapped to a particular particle path in the acoustic field (e.g. translating a cluster of particles along one horizontal axis) or by using a pointing touch screen device to assign the trajectories.

3.2.4 Summary

A variety of approaches to ultrasonic levitation have been developed. However, dexterous interaction with levitated objects has not yet been demonstrated. For approaches using only standing waves (in the form of focal lines), the main limitation is that different techniques must be used to move particles in different dimensions. Up to now, this has resulted in less smooth, less agile and often jumpy object motion. Marzo's [86] optimisation approach allows for continuous placement of traps at arbitrary locations within the working volume. By displacing these traps with small amounts (approx. 0.1 mm), continuous particle motion can be achieved. In [86] real-time interaction with the system was possible using a keyboard or GUI buttons, however the rates are still too slow for continuous interaction. On larger setups, the optimisation would take several seconds for each location. Up to now, this prevented a smooth interactive use of this technique.

Our work presented here contributes the first implementation of a low-latency, high frame-rate, smooth interactive control of a levitated particle in 3D space, as well as a method that ensures sustained particle positioning in the main trap. In addition, we conduct the first device-mediated Fitts' law study in 3D with a levitated particle as cursor, providing all natural depth cues.

3.3 Methods

We use the same method proposed by Marzo et al. [86] to model acoustic levitation. In this section, we provide an overview of this method for the reader.

The acoustic radiation force F on a small (radius $<<$ wavelength), spherical particle in an invicid medium is given by the gradient of the Gor'kov potential U [48] [18]

$$F = -\nabla U. \tag{3.1}$$

The equation

$$U = k_1(\mid p \mid^2) - k_2(\mid p_x \mid^2 + \mid p_y \mid^2 + \mid p_z \mid^2), \tag{3.2}$$

describes the Gor'kov potential [86]. The complex modulus denoted by $\mid \cdot \mid$ is defined as $\mid \beta + i\gamma \mid = \sqrt{\beta^2 + \gamma^2}$. Equation (3.2) consists of two parts; the first is a complex pressure part, which shows that particles move from areas of high modulus of the complex pressure p, to areas of low pressure modulus and the second - a velocity part, written in terms of the spatial pressure derivatives, describing how particles are drawn to areas of large modulus of the velocity gradient. The two constants are given by:

$$k_1 = \frac{1}{4}V\left(\frac{1}{c_m^2 \rho_m} - \frac{1}{c_p^2 \rho_p}\right),$$
$$k_2 = \frac{3}{4}V\left(\frac{\rho_m - \rho_p}{\omega^2 \rho_m(\rho_m + 2\rho_p)}\right),$$

where V is the volume of the (spherical) levitating particle and ω the wave frequency. c_m and c_p denote the speed of sound through the medium and the particle respectively. The density of the medium is given by ρ_m, and of the particle by ρ_p. In our case, the medium is air and the particle is an expanded polystyrene bead, so we have: $c_m = 343\frac{m}{s}$, $c_p = 2400\frac{m}{s}$, $\rho_m = 1.2\frac{kg}{m^3}$ and $\rho_p = 25\frac{kg}{m^3}$ at a room temperature of $20°C$. From (3.2), it is clear that for determining the Gor'kov potential, it is necessary to know the pressure field.

Acoustic levitation traps are regions in the field where the acoustic radiation forces converge. Consequently, objects placed in the levitation traps remain suspended in mid-air. Strong and stable acoustic traps can be created using optimisation [86]. Maximising the converging radiation forces is equivalent to maximising the Laplacian of the Gor'kov potential, given by:

$$\nabla^2 U = U_{xx} + U_{yy} + U_{zz}, \tag{3.3}$$

with the notation $U_a = \frac{\partial U}{\partial a}$ and $U_{aa} = \frac{\partial^2 U}{\partial a^2}$.

The pressure inside the traps tends to be very high, which can create disturbances for the levitating object. In order to avoid such disturbances, in addition to maximising the Laplacian, following [86], we minimise the pressure as well.

The two arrays of transducers we employ in this study emit acoustic waves with constant amplitude and frequency. For simplicity, we assume the transducer to be a piston source, and use a model that neglects reflections and nonlinear effects, which is also in the interest of fast computation.

The complex acoustic pressure of the j^{th} transducer in the array can be written as [86]

$$p^j = e^{i\phi^j} M^j, \tag{3.4}$$

where ϕ is the phase shift and M a complex number, specific to a transducer and a given point in space. Due to the rule of linearity of differentiation, $p_x^j = e^{i\phi^j} M_x^j$ also holds. We calculate M^j by the means of the example of Marzo et al. [86], assuming a circular piston source and using a single frequency far-field model.

$$M^j = P_0 J_0(krsin\theta_j) \frac{1}{d_j} e^{ikd_j}, \tag{3.5}$$

where P_0 is a constant determined by the transducer power, J_0 is a zeroth-order Bessel function of the first kind, k is the wave number, r is the radius of the piston source, θ_j is the angle between the transducer normal and the focus point and d_j the distance between the j^{th} transducer and the focus point.

The total acoustic pressure field generated by N transducers, assuming linear superposition of waves, is given by the sum of the pressures $p = \sum_{j=1}^{N} p^j$, generated by individual transducers. As before, by linearity of differentiation, it holds that $p_x = \sum_{j=1}^{N} p_x^j$. Going back to (3.3), it is clear that now, the Laplacian of the Gor'kov potential can be expressed as a function of only one variable - the phase shift: $\nabla^2 U = f(\phi_1, ..., \phi_N)$. Hence, to produce a specific pattern in the acoustic field, the phase shift for each individual transducer needs to be calculated.

Objective Function

Numerical optimisation methods make it possible to choose phase shifts for the individual transducers that best fulfil our predefined requirements. To this end, following [86], we define a function which represents our problem objective - to minimise the pressure and maximise the Laplacian of the Gor'kov potential at a given point in space. This function is used as a criterion by the optimisation procedure to select better rather than poorer solutions. To obtain a levitation trap at the point $\vec{q}$, we minimise the objective function

$$O(\phi_1,...,\phi_N;\vec{q}) = \mid p(\vec{q}) \mid^2 - \nabla^2 U(\vec{q}). \tag{3.6}$$

Adding weights to control the relative strength of the trap in a particular direction and to balance the contributions of different terms, results in

$$O(\phi_1,...,\phi_N;\vec{q}) = w_p \mid p(\vec{q}) \mid^2 - w_x U_{xx}(\vec{q}) + w_y U_{yy}(\vec{q}) + w_z U_{zz}(\vec{q}). \tag{3.7}$$

We apply equal weights to each direction of propagation, hence generating a vortex trap. An overview of different types of levitation traps is found in [86].

BFGS Optimisation

The Broyden-Fletcher-Goldfarb-Shanno algorithm is an iterative method for solving nonlinear optimisation problems. It belongs to the group of quasi-Newton methods, which have the advantage that the Hessian matrix is not evaluated at each step. Instead, an approximation generated by analysing the successive gradient vectors is used, making the process more time-effective. This is favourable for solving our optimisation problem, as we have to deal with a very large state space, consisting of 252 phase values. Thus, similarly to [86], we employ BFGS optimisation to minimise (3.7).

3.4 System

Our main challenges are to achieve homogeneous movement along all three dimensions with: (1) low latency, (2) continuous movement without steps, and (3) stable movement enabling high velocities and accelerations.

We overcome these challenges by: (1) Precomputation of optimal transducer phases for *all possible levitation points within the entire array at 0.5 mm resolution*. (2) Phase interpolation. (3) A particle stabilisation mechanism to ensure that the particle is always in the main trap.

3.4.1 Precomputation of Optimal Transducer Phases

The main limitation of using the optimisation-based approach to render interactive levitating interfaces, is that optimisation can take several seconds for each new point. We update the levitation points at 1 kHz, rendering this approach to be unfeasible. [86] presents an approach to precomputing discrete animation paths which can then be played back. We extend this approach to precompute *all levitation points in the entire levitation volume at* 0.5 *mm discretization*. Our levitation volume measures 140 mm width * 80 mm height * 90 mm depth. At 0.5 mm resolution, this results in approximately 8 mio. levitation points. For each of these points, the 252 transducer phases have to be optimised. We optimise each point using 20000 iterations of BFGS. We use Armijo line search with coefficient $\alpha = 0.8$ to determine the step size. This takes about 20 seconds per point. The entire calculation takes approx. 44800 hours (> 5 years) of computation time. Since calculation on a workstation is not feasible, we resort to using a computer cluster. We stored the result in a lookup table with a size of 8 GB in RAM.

Because we interpolate the phases between levitation points, it is very important that the phases for neighbouring points are smooth. Unfortunately, the optimisation problem inherently contains many local optima. Ideally, neighbouring points should use the “same” local optimum, and avoid jumping to a distant one, as such a transition would render the interpolated data inconsistent and lead to unpredictable behaviour of the levitated particle. After evaluating diverse approaches to achieving this, we propose the following strategy. First, the centre of the levitation volume is optimised from random starting phases. Any subsequent point is optimised using the phase values of a neighbouring point for starting phases. After the centre point, we optimise progressively in the height dimension (up and down). To ensure smoothness, we optimise with 0.1 mm resolution. From this line, we optimise the entire width of the array with 0.1 mm resolution. This results in an optimised plane at depth 0. From this plane, we optimise in the depth dimension at 0.5 mm resolution. This procedure results in very smooth transducer phases between neighbouring points (see Figure 3.2). Any remaining non-smoothness is mostly in the height dimension.

3.4.2 Phase Interpolation

In order to achieve sub-millimetre precision in manipulation of the sound field, we use trilinear interpolation between the eight neighbouring points from the lookup table. We first evaluate the acceptability of such interpolation by numerically computing smoothness within the whole sound-field volume. We consider the transition between two neighbouring points as smooth, if the differences between the phase values of each transducer are not larger than π radians. The majority (96.2%) of the phase transitions within the sound field volume are smooth and far smaller than π. However, there is still a small fraction of non-smooth transitions, which needs to be investigated. We inspect spatial properties of the transition smoothness, and in particular those of non-smooth transitions, using visualisations of the transducer phases over multiple slice surfaces within the volume (Figure 3.2). As

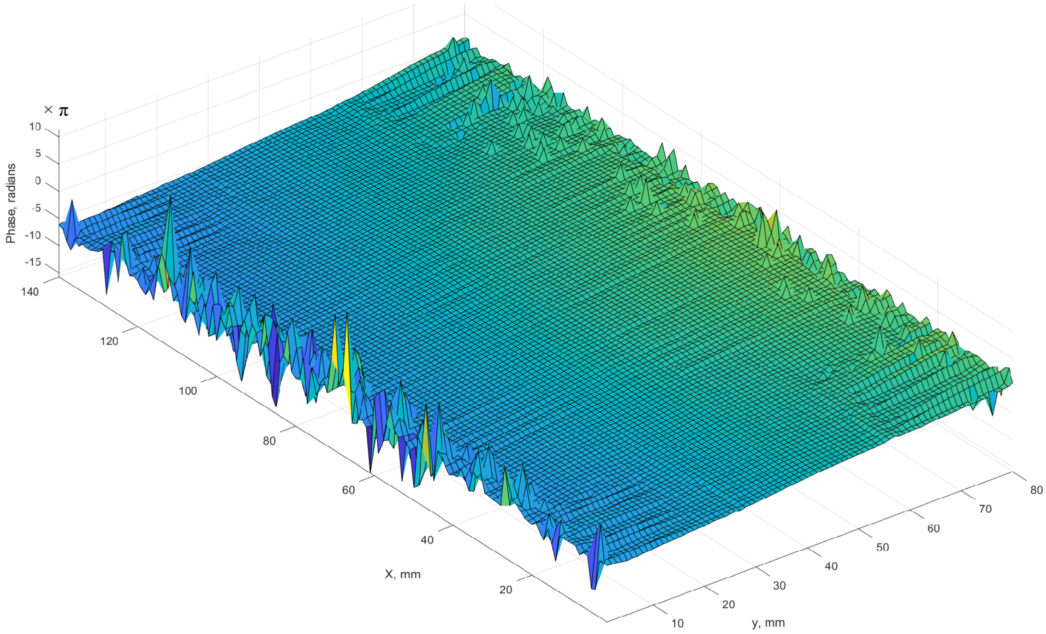


Figure 3.2 Evaluation of phase smoothness within the sound-field volume. The surface represents phases of the transducer 197 at the plane $Z = 9$ mm.

can be observed, the phases are smooth close to the centre of the volume, and become non-smooth closer to the boundaries, in particular in the proximity of the transducers. Based on our observations, we configured the trilinear interpolation so that it is applied if the neighbourhood of the point is smooth, and it is replaced by the nearest neighbour values if the neighbourhood is non-smooth. The particle movement is less smooth (0.5 mm steps) when entering a non-smooth region, but the general stability of the particle movement is increased.

3.4.3 Particle Stabilisation

One major problem with ultrasonic levitation is placing the particles. When a focus point is generated by the optimiser, weaker secondary traps also appear in the acoustic field. These secondary traps can levitate particles, but are prone to disappear and drop them once the primary trap is moved. Since the acoustic field cannot be seen with the naked eye, one can not distinguish between different traps. Consequently, placing the particle in the main trap is not a trivial task. Furthermore, after some time, the particle may jump to a secondary trap.

We stabilise the particle, i.e. reassure it is located in the primary trap, both when the particle is first placed into the acoustic field as well as during direct interaction. When placing the particle, we optimise the field for a levitation point at the origin. We place the particle in the acoustic field, using a piece of an acoustically-transparent fabric. Then we turn on the

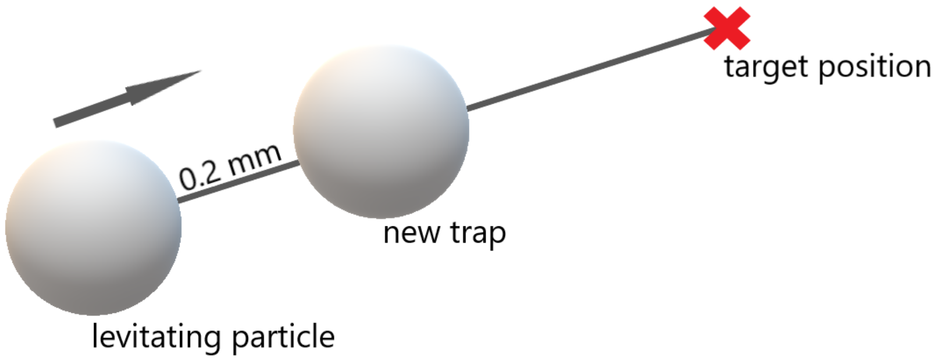


Figure 3.3 When the particle is moving towards a new target position, it never takes steps larger than 0.2 mm per frame, to ensure that it stays in the primary trap.

transducers, which causes levitation of the particle in some secondary trap. We determine the actual particle position using the motion capture system and generate a primary trap at the actual particle position. During interactive control of the particle, excessively large jumps of the primary trap can cause the particle to jump into a secondary trap. Therefore, we interpolate the primary trap position towards the target indicated by the user, while ensuring that the primary trap never moves more than 0.2 mm between frames in the regions with interpolation. In Figure 3.3, a levitating particle moving towards a new target position is shown. In the subsequent frame, a new primary trap is generated in the direction of the target, at a distance of 0.2 mm. This procedure contributes substantially to the stability of the levitated particle.

3.5 Hardware

Our acoustic levitator comprises two 9×14 arrays of *muRata MA40S4S* transducers. The transducers are cylindrical and have a 10 mm diameter and a 7 mm height. The ultrasonic transducers are equally spaced at a distance of 0.3 mm from each other and have maximum input voltage of 20 Vpp. Each emits a sound wave of frequency $f = 40$ kHz (wavelength $\lambda = 8.6$ mm), which is inaudible to humans. The two arrays are mounted horizontally, facing each other, at a distance of 80 mm. We developed an aluminium rail system, which allows for easy adjustment of the distance between the arrays.

A major problem when using transducer arrays for levitation is that the arrays heat up fast, leading to destruction within a few minutes. We solved this problem with a cooling system that generates an air stream on the back of the array PCBs, without leaking an air stream into the levitation volume. This allows us to operate the arrays continuously. We use expanded polystyrene beads of small diameter (approx. 2 mm) as levitating particles, due to their low density.

For driving the transducer arrays, we use the logic board of the *Ultrahaptics* [1] Evaluation Kit. We connected the board to both transducer arrays, leading to on-board synchronisation of both arrays. The logic board is connected to a driving PC using USB.

We track the particle position and index finger of the user using optical motion capture (*OptiTrack*). We use a small velcro-attached retro-reflective marker with a diameter of 9 mm, placed directly on top of the user's fingertip. We use six *Prime 13* infrared cameras capturing 240 FPS. Three cameras observe the levitation volume from the side, while three additional cameras track the user's finger from above. The cameras are connected via Ethernet to a second PC that drives the motion capture system and our levitation software.

3.6 Software

Our precomputation software is based on the system implemented by Marzo et al, which is generously shared in [81]. Based on this, we developed a program for phase optimisation that is suited for execution on a computer cluster. We slice the workload into 88000 task description files using a script. Worker nodes read these files and generate a result file.

The interactive hardware and software has to operate in real-time. We use two workstations to operate the system, so as to reduce latencies. The first workstation operates in high-performance mode and runs the *OptiTrack Motive* motion-capture system. Particle and fingertip are tracked and streamed via NatNet to a custom Java program running on the same machine. The Java program performs particle stabilisation and computes the particle motion. This program reads in the results files from the cluster computation at startup, so as to generate the lookup table. It looks up the necessary transducer phases in this table. Finally, it performs phase interpolation and sends the resulting transducer phases to a C++ program on a second workstation.

The second workstation is tuned to run the C++ application which receives the transducers' states through a UDP socket. The C++ program caches the phases locally and uses the Ultrahaptics Low-level SDK to stream the phases to the Ultrahaptics logic board. To ensure smooth levitation, the C++ software needs to respond to a callback from the Ultrahaptics driver at 1 kHz with a latency of a few milliseconds at maximum. This workstation runs only the critical operating system processes with low priority on one half of the CPU cores, as defined by an affinity mask. The real-time priority and the other half of the cores (non-

[1] http://ultrahaptics.com

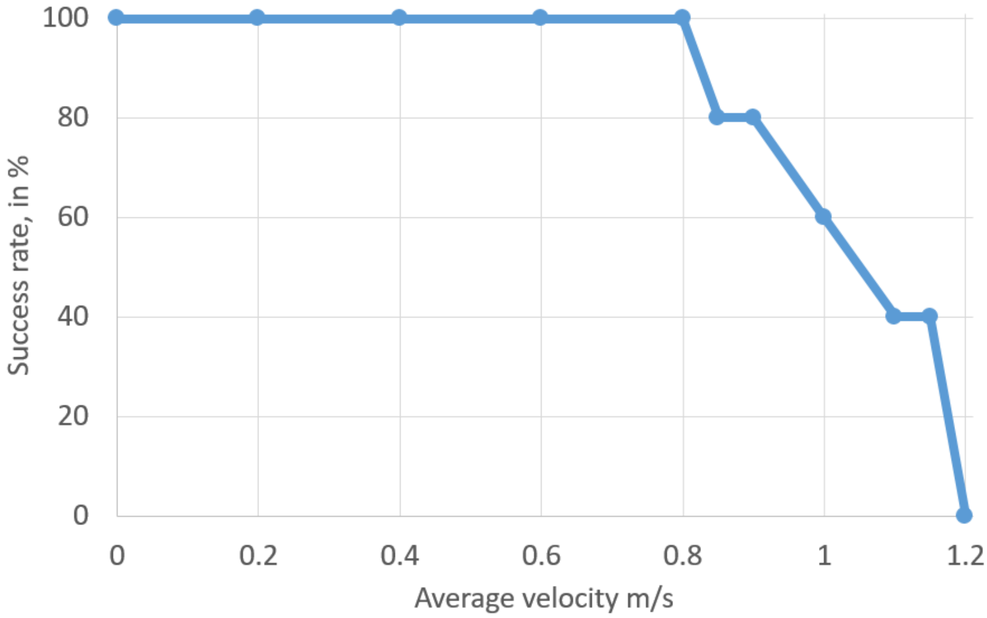


Figure 3.4 Success rate with respect to average particle velocity.

hyperthreaded) are dedicated to the C++ application. The machine runs in high-performance mode with CPU sleep states and SpeedStep disabled. Both workstations are connected via Ethernet using a local Gigabit switch. The experiment is controlled and logged using the Java program on the first workstation, which also computes particle motion.

3.7 Technical evaluation

To evaluate velocities and stability, similarly to LeviPath [103], we performed an experiment in which we moved a particle back and forth within the levitation volume along a 7 cm straight path. We repeated the movement five times at each velocity and recorded the number of successes and failures. When the particle correctly completed the full movement along the given path, success was registered. A failure was noted when the particle fell off or switched to a secondary trap during the movement. We started with a velocity of 0.2 m/s, gradually increasing it by steps of 0.2 m/s up to 1.2 m/s, where failure was observed in all five trials.

As can be seen in Figure 3.4, our system achieved particle velocity of 0.8 m/s with a 100% success rate, thereafter the success rate decreased almost linearly and eventually reached 0 at a velocity of 1.2 m/s. From this experiment we can conclude a lower bound on the maximum velocity of 0.8 m/s. We observed, however, that most of the failure cases consisted of the particle dropping either at the beginning or at the end of the movement. This indicates that the limiting factor is not the velocity, but the acceleration. In fact, we believe that by providing more dynamically consistent control it should be possible to achieve even higher particle velocities. For example, our system was able to achieve velocities close to 1.5 m/s, however in this case the particle was shooting out of the end-trap. In the future, we want to conduct experiments where the maximum reachable velocity and acceleration in the right-most part of Figure 3.4 (0.8 to 1.2 m/s), are explored separately.

We investigated the movement of the particle using a high-frame-rate camera at the highest stable velocity i.e. 0.8 m/s and at the lowest velocity in the experiment i.e. 0.2 m/s. While in both cases, the movement looked smooth to the unequipped eye, in the video we observed that during the fast movement, the particle performed a few "jumps" of approximately 15 mm at an average velocity of 1.2 m/s (computed using length of the "jump" and the number of frames where the particle performs the "jump"), but during the slower movement there were two 3 mm "jumps" at the beginning, and then the movement stabilised and continued smoothly with minor disturbances only. This is another observation demonstrating that dynamically-consistent control is necessary to reach very high velocities.

We also evaluated the total latency of the system using a high frame-rate camera. We setup a motion capture marker-based event (marker crossing a plane) and a response of the levitation system (dropping the currently levitated particle). The camera observed the space where both event and response were generated and recorded the corresponding segments. We repeated the experiment three times and tallied the number of frames between the marker event and the system response. For a system to be perceived as real-time in pointing tasks, the total latency has to be below 20 ms [63]. In our experiment, in all three cases the latency between the event and the response was less than 17 ms, with the average value being 15 ms, which is below the threshold value perceptible for users.

3.8 User Study

As suggested in the introduction, key application areas allowed by *LeviCursor* are physical 3D pointing, including 3D pointing with tangibles, and aimed movement user studies providing all natural depth cues of the cursor and the targets. *LeviCursor* allows user studies of mediated 3D pointing to investigate effects of latency, control-to-display ratio or the transfer function on the pointing process, accuracy, speed, physical ergonomics, cognitive load, movement dynamics, velocity and acceleration profiles etc. There are multiple user studies investigating pointing movements in 3D space, however in contrast to *LeviCursor* they provide either limited cues for depth perception e.g. using volumetric display [49] or virtual reality [142], or they do not allow for any transfer function, for example non-mediated 3D pointing [8].

We demonstrate applicability of *LeviCursor* to pointing tasks by running a short user study of 3D aimed movements.

The task was a variation of Fitts' *serial pointing task* adapted to 3D. It is very difficult to place physical targets for levitating particles. The targets should disturb physical particle motion, the sound field, and the motion capture system as little as possible. We decided to use needles painted with black matte colour to show the centre of the targets. The actual targets were internally represented as spheres around the needle tips and were registered using the motion capture system. We used three target sizes of: 2 mm, 4 mm and 8 mm radius. The distance between the targets was 68 mm. The target size conditions for each user were randomised. The task of the user was to move the particle between the two targets as quickly as possible, reversing direction inside the target. The motion capture system was tracking the position of the particle with respect to both targets. When the particle entered the target, a confirmation tone sounded and a success was registered. When they reversed direction outside the target, a failure tone sounded, and a failed trial was registered. Users were asked to maintain an error rate below 5 %. We used a distance of 12 cm and target widths of 6 cm, 1.5 cm, and 3.7 mm. This resulted in Index of Difficulty of 2, 4 and 6, respectively.

We recruited 8 participants (mean age 30.5 years, std. dev. 5.6, 4 male, all normal or corrected to normal eyesight, all right-handed). Participants sat on a chair in front of the apparatus (see Figure 3.5). A retroreflective marker of 9 mm diameter was attached to the index finger of their right hand. The particle was placed in the levitation volume by the experimenter. Participants could control particle motion in 3D with their fingertip, using a control-to-display ratio of 3. Participants were allowed to explore the particle motion for approx. 30 s. We asked participants to place the particle as accurately at each of the needle tips as they could, in order to calibrate the target location according to their perceptions of the target. Afterwards, participants were asked to move between the targets as quickly as possible. After performing 50 aimed movements, the experiment was shifted to the next target size-condition.

During the experiment, our software was continuously recording the 3D position of the particle, the real-time timestamps and the timestamps when the user reached each target and was notified by the sound. After the experiment, the participants were informally interviewed concerning their experience with *LeviCursor*.

3.8.1 Analysis

We applied Fitts' law analysis, as is typical for the HCI field [77]. While there exist multivariate models of pointing [49], for spherical targets they are equivalent to Fitts' law.

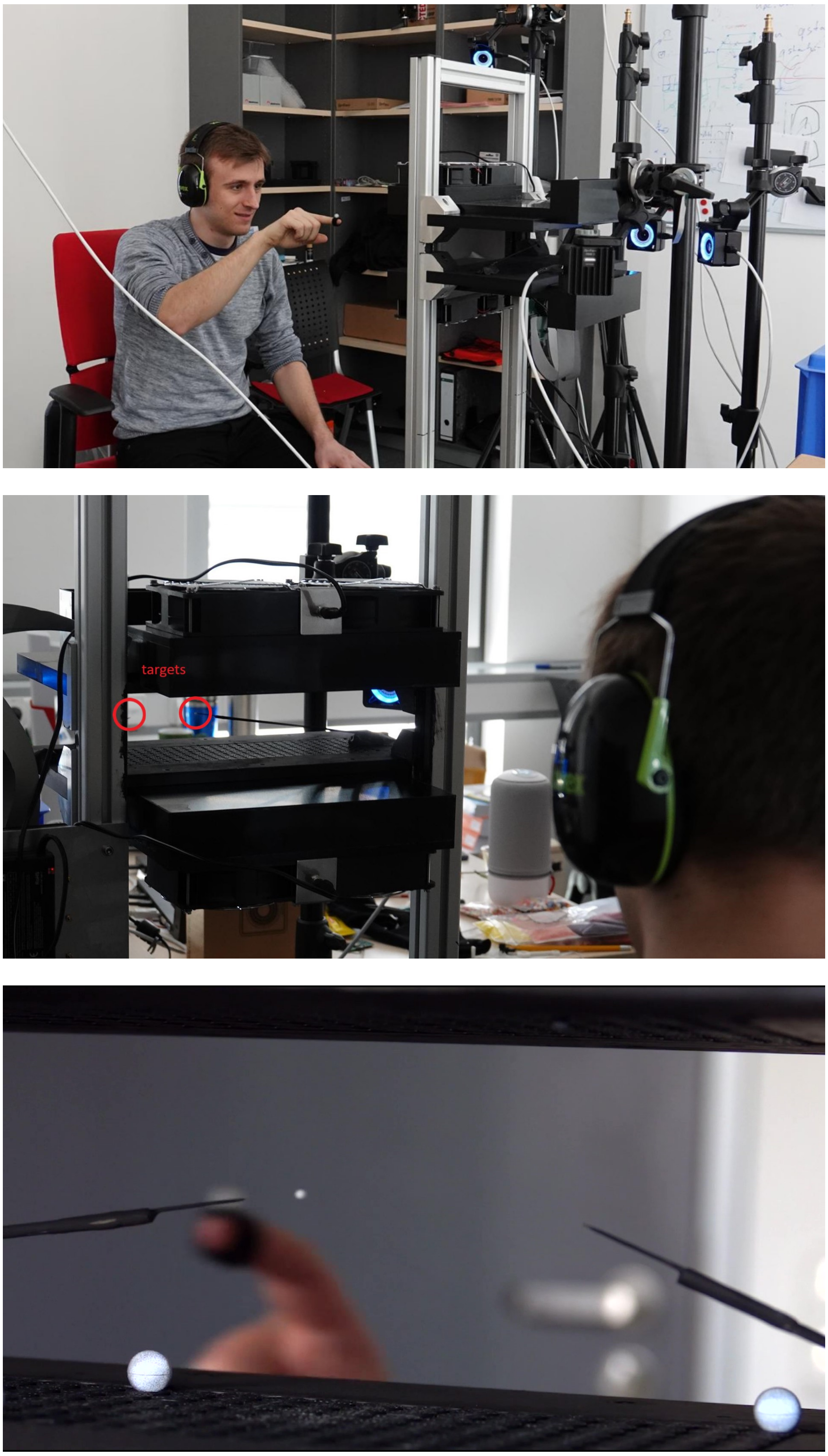


Figure 3.5 Participants had a retroreflective marker attached to their right index finger and were seated on a chair in front of the levitation apparatus. With their fingertip, they were able to control the levitating particle. The two targets, marked with red in the middle photo, where defined as virtual spheres with a centre at the tip of the black needles. The bottom photo shows a user executing the pointing task.

We use Fitts' law in the Shannon formulation

$$MT = a + b \times \log_2\left(\frac{D}{W} + 1\right),$$

where MT is the movement time, D the amplitude, W the target width, a and b are free regression coefficients. Following the recommendations of [77], instead of D and W, we use effective target width W_e, based on the standard deviation of the end-points (σ) as

$$W_e = 4.133\sigma$$

and the effective amplitude D_e as the distance between the corresponding effective target centroids:

$$D_e = \sum_{i=1}^{N} \frac{D_i}{N},$$

where D_i is the amplitude of individual aimed movement and N is the number of movements terminating within the effective target. We group the data into six ranges according to the ID. We average IDs and MTs within each group and then fit a Fitts' law model as a first-degree polynomial optimally representing the data in the least-squares sense. We evaluate goodness-of-fit using the coefficient of determination (R^2). To evaluate the performance of the users using *LeviCursor*, we compute the average effective throughput

$$TP_{ea} = \frac{1}{P}\sum_{i=1}^{P}\left(\frac{1}{C}\sum_{j=1}^{C}\frac{ID_{eij}}{MT_{ij}}\right),$$

where P is the number of participants and C is the number of conditions, as well as maximum effective throughput

$$TP_{emax} = \max_{i=1}^{P}\left(\max_{j=1}^{C}\frac{ID_{eij}}{MT_{ij}}\right).$$

3.8.2 Results

The experimental data can be modelled successfully by Fitts' law with R^2 of 0.92, as can be seen in Figure 3.6. The participants achieved an average throughput of 4.93 $bits/s$ and a maximum throughput of 8.69 $bits/s$. These values are comparable to the throughput of the mouse [134]. Furthermore, they are only slightly below the throughput of uninstrumented mid-air pointing (average $TP = 5.48$ $bits/s$ [8]).

According to the informal interviews, the users experienced the interaction as exciting. It was described, for example, as "Jedi using the force", and they felt "in control of the particle".

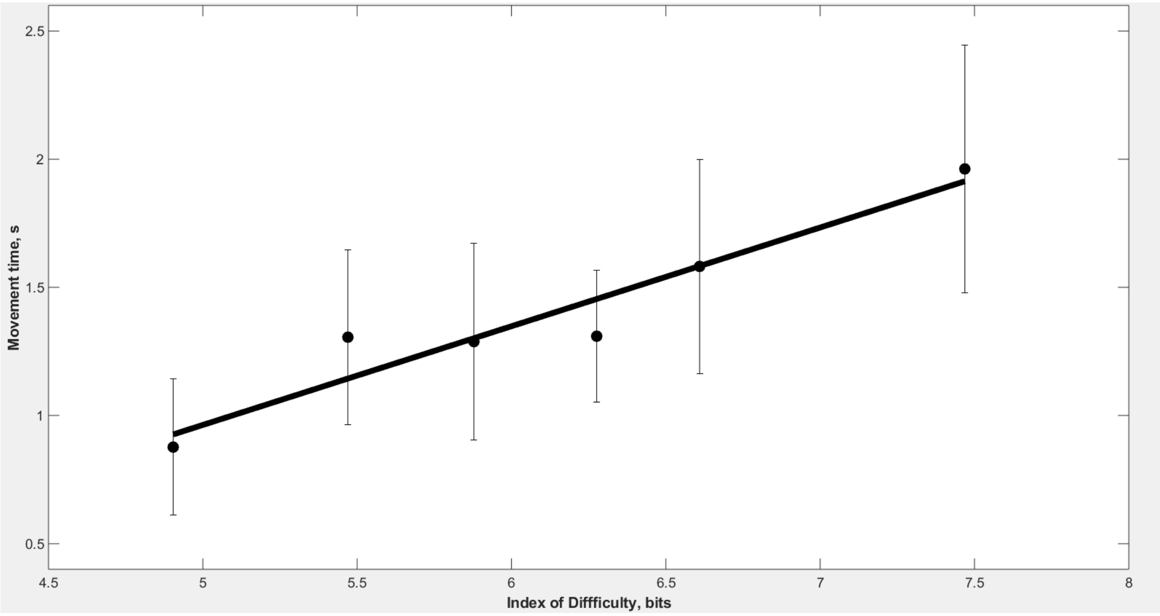


Figure 3.6 Fitts' law model representing the data of all participants.

Some of them mentioned a common problem in mid-air interaction - tension and fatigue in the shoulder, known as the "Gorilla arm".

We find it promising that even though *LeviCursor* has different physical properties than a virtual mouse-controlled cursor on a desktop, it can provide comparable interaction behaviour and performance. This demonstrates that using our method, users can exercise dexterous control over levitated particles.

This was, however, a preliminary study to test a new concept. We plan to conduct bigger studies with more participants in future work.

3.9 Discussion

From the results of both the technical evaluation and the user study, we can clearly see that the proposed method for interactive control of levitated particles is an effective tool for applications which require pointing in the real 3D space. While this is the first body of work which demonstrates such smooth and dexterous control of levitated particles, the method also has multiple limitations and large potential for further improvement. Below we describe the limitations and as future work we plan to explore new approaches to workaround the main limitations.

3.9.1 Limitations

The limitations of *LeviCursor* can be split into two parts - first the limitations inherited from the underlying levitation algorithm [86] and second the limitations of the current algorithm.

The inherited limitations of the method relate to the optimisation approach, namely we can levitate a single particle of size smaller than half of the wave length, preferably spherical (although we have also levitated flat and ellipsoidal particles), made of low-density materials. Levitation of multiple particles should be made possible by changing the objective function for the optimisation or using a method similar to [86]. Although ultrasound technology has passed safety tests and is cleared for commercial use for haptic and parametric audio devices (e.g. *Ultrahaptics*, *Ultrasonic Audio* etc.), there are still concerns about the effects of high-intensity ultrasound on humans. As a cautionary measure, we provided the participants of the user study with earmuffs.

The approach described in this chapter also has multiple limitations, in particular: scalability with respect to the acoustic volume and the computational power necessary for precomputation, flexibility of the ultrasound array setup, extensive hardware both for ultrasound levitation and for motion tracking, and in the current implementation with optical motion tracking - colour of the particle and the surroundings. The scalability is limited by the size of the lookup table and the necessary precomputation time. The required memory and computational time scale linearly in each dimension. While in this chapter we work with a levitating interface of relatively small acoustic volume, the state of the art hardware and software can allow significantly larger setups, for example current supported size of main memory (2TB by Windows) allows a levitation volume of 1.2 m^3 while keeping the entire table in RAM. Considering that the current lookup table is computed by a cluster within few hours, it should be possible to compute the table for the above mentioned movement volume in reasonable time. In regard to flexibility, it is necessary to recompute the lookup table for each ultrasound array setup, which takes significant computation time. Apart from sophisticated ultrasound hardware, the current approach also requires optical motion capture hardware. The optical motion capture cameras need to be positioned in a way, that allows the levitated particle to be visible across the entire volume of the levitating display. As an additional requirement, the particle has to provide high visual contrast in comparison to the surrounding hardware, in the optimal case it should be retroreflective.

3.9.2 Future work

There are multiple potential improvements to the current approach of interactive control of levitated particles, as well as extensions and additional applications.

Figure 3.7 Sketches of potential use cases for LeviCursor in AR, e.g., games where real and virtual objects can interact with each other (top), or navigation in real 3D space with augmented virtual information bubbles (bottom).

As a main direction of our future work we plan to apply other algorithms for levitation which can work in real time instead of the lookup table, namely we plan to work on using holographic acoustic elements (focus point and signature) [86] for computation of the levitation trap in real time. Up to this point we have tried the focus and signature approach, but it was less stable than the optimised phases from the lookup table.

Next, we would like to explore levitation with multiple particles as well as interactive control of them.

Lastly, we would like to identify and test additional realms that can benefit from the *LeviCursor* method. For example, *LeviCursor* could hold great potential for applications in extended reality, where the physical cursor would interact with both real and virtual objects in real 3D space, as illustrated with the examples in Figure 3.7.

3.10 Conclusion

In this chapter, we presented *LeviCursor*, a method for interactively moving a 3D physical pointer in mid-air with high agility. The method allows a levitated particle to move continuously in any direction. We addressed the three problems of low latency, continuous movement without steps, and stable movement enabling high velocities and accelerations. We contribute three solutions for solving these problems. The first is a complete precomputation of all transducer phases, achieving a round-trip latency of 15 ms. The second is a 3D interpolation scheme, allowing the levitated object to be controlled almost instantaneously with sub-millimetre accuracy. Lastly, we presented a particle stabilisation mechanism which ensures that the particle is always in the main levitation trap. This interactive system has been validated by a user study. The results of the study showed that interaction with *LeviCursor* can be successfully modelled by Fitts' law, with throughput that is comparable to interaction with mouse pointers.

Chapter 4

Modelling Dynamics of Interactive Acoustic Levitation

4.1 Introduction

System dynamics describe how interactive systems react to user input in terms of continuous system states changing over time. For example, the mouse pointer (or a virtual spaceship) might change its position and velocity in reaction to a user moving the mouse or joystick. A shape-changing interface might change joint angles in response to a users' distance. A levitating particle might move in response to a user's hand movements.

Interface dynamics are important for interaction for several reasons. First, they determine the "feel" of the interface. Whether the interface feels laggy, responsive, or erratic, are all consequences of interface dynamics. Second, interaction performance is determined by interface dynamics. Lag or noise in the interaction will have direct negative consequences on throughput. Third, physical ergonomics and cognitive load are affected by interaction dynamics: control of higher-order dynamics (e.g., track-point) is cognitively more difficult, than low-order (e.g., mouse); on the other hand, more complex dynamics may require less physical movement and consequently cause less exertion and less fatigue.

Surprisingly, interface dynamics have not received a lot of attention in HCI research. For example, research on pointing has been dominated by summary statistics models such as Fitts's law. These models abstract from continuous system dynamics and model interaction as a series of discrete events (e.g., mouse movements). Such models might have been sufficient when system dynamics were trivial, as with an ideal mouse pointer. However, many areas of human-computer interaction aim at introducing interfaces with non-trivial dynamics. First, when interacting with physics simulations, such as those used in computer

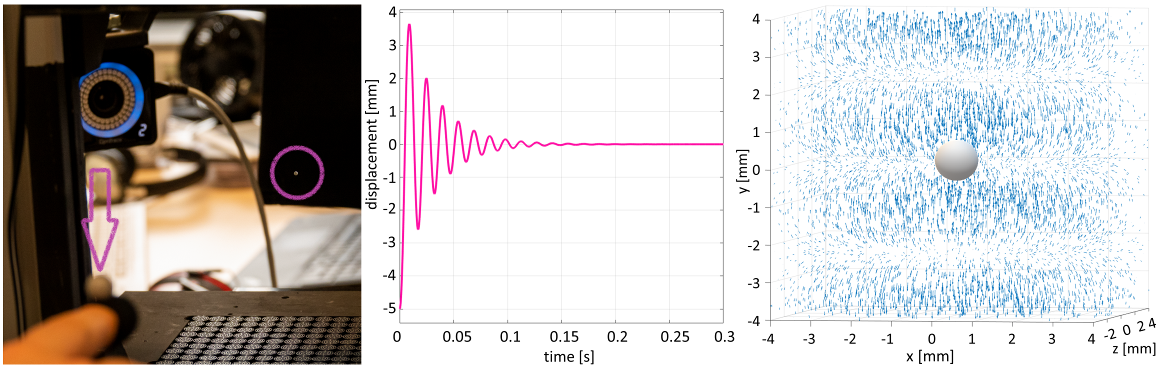


Figure 4.1 With our system users can control the movement of a levitated particle (left, circled) with their finger (left, arrow). Because of the dynamic nature of ultrasonic levitation, interaction involves complex phenomena such as oscillations and instability of the particle. To better understand this, we identify a black-box model of the dynamics of the interactive system. We then analyse the forces acting on the levitated particle (centre). From this, we develop a semi-analytical model of system dynamics (right). This modelling approach allows us to (1) better understand, design, and debug interactive ultrasonic levitation interfaces and (2) to simulate systems before even building a prototype. We hope that this approach to modelling the dynamics of interactive systems can also be useful to other areas of HCI, such as shape-changing or robotic interfaces, interactive physics simulations, and even desktop interfaces.

games, interface dynamics can be complex (e.g., when controlling space ships or aeroplanes). Second, physical interfaces such as shape-changing interfaces or robotic interfaces, move masses in the physical world and therefore have to obey the laws of physics. As one example, levitating interfaces introduce non-trivial dynamics of the levitating particle.

Some research in HCI has aimed at modelling the dynamics of the human part of the human-computer interaction loop. This can be interpreted as the right half of Figure 4.2. Modelling the dynamics of the interactive system itself, that is, the left half of Figure 4.2, has received comparatively less attention.

When designing interfaces with non-trivial dynamics, a simple trial-and-error design process is not sufficient. The number of parameters is often large, and phenomena (such as oscillations and instability) are complex. In this case, it is necessary to model the dynamics of the user interface.

There are two main kinds of models for interface dynamics. The first are *black box* models. For these models, it is not necessary to have an understanding of the internal dynamics of the interactive systems. One can merely feed the system with a number of reference signals and record the system response. Techniques from system identification then allow to build a model of the system behaviour.

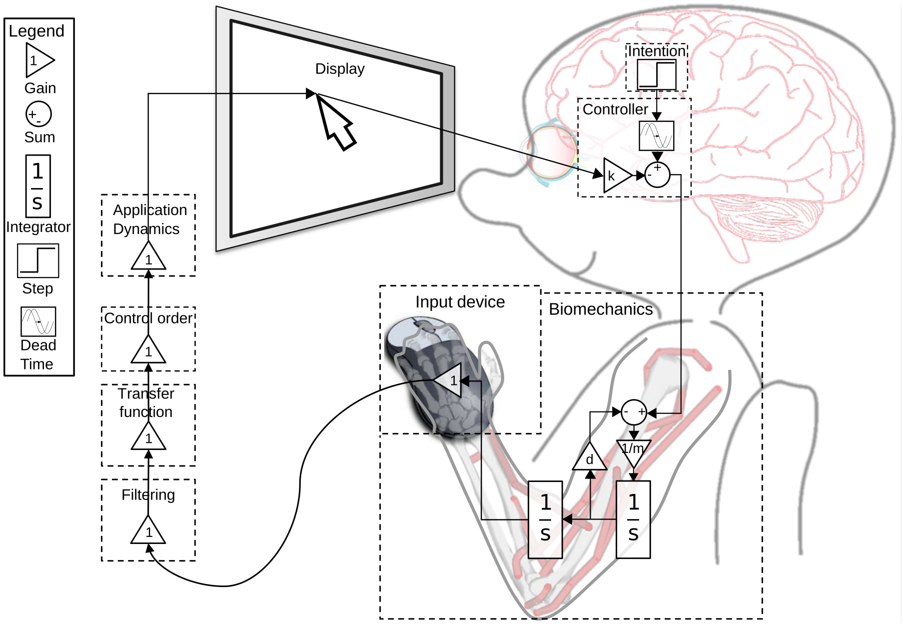


Figure 4.2 The model developed in [92], models interaction between humans and computers as a closed-loop dynamical system. [92] compares different models for the human side of this model (right half), while mostly ignoring the dynamics of the computer side (left half). In this chapter we develop a model for the left half of this image for the special case of interactive ultrasonic levitation.

If the designer has an understanding of the internal dynamics of an interactive system, he can construct an *analytical model*. These models are derived from first principles, e.g., physical laws.

Modelling of interface dynamics has two distinct advantages. Both types of models allow simulation of interface dynamics. This allows to try system changes without actually building (or possibly destroying) the interactive system.

In contrast to black box models, analytical models also serve as mental models for the designer, to develop a deep understanding of the interaction. This guides intuition in the design process, and allows to solve problems (e.g., instability) based on this understanding.

For analytical models, it is usually necessary and useful to make several simplifying assumptions, e.g., approximation and linearisation of nonlinear system parts. Furthermore, often, not all aspects of the system can be described analytically. This is usually the case when some phenomena in the system are very complex. In this case, analytical models can have some parameters which are not derived from first principles, but identified empirically. Such models are called *semi-analytical* or grey box models.

In this chapter, we demonstrate the modelling of interface dynamics in the case of interactive ultrasonic levitation. We show how both black-box and semi-analytical models can be used in HCI. First, we identify a black-box model of the dynamics of interactive ultrasonic levitation. We then analyse the forces acting on an ultrasonically levitated particle. From this, we derive a semi-analytical model for particle behaviour in a static ultrasound field.

The objectives of this chapter are twofold. First, we aim at promoting the modelling of dynamics of interactive systems in HCI. Second, we aim at better understanding interactive ultrasonic levitation.

4.2 Related Work

4.2.1 Modelling Dynamics of Interactive Systems

The modelling of dynamics of systems is at the heart of most engineering disciplines, including automotive, aerospace, electrical, and mechanical engineering. This modelling approach is especially suited for interactive systems, where a human interacts with a machine, resulting in models of the entire human-system loop. This approach has lead to the field of manual control of machines [66, 60, 64, 87]. It might seem surprising that the modelling of dynamics of interactive systems has not received a lot of attention in the field of HCI. [92] presents a control-theoretic perspective of interaction including the model shown in Figure 4.2. The paper compares different models for the human part of the continuous interaction loop, but does not investigate the dynamics of the computer side beyond a unit gain. [90] extends this model to model different functions that map mouse movement to pointer movement, including pointer acceleration functions. There have also been several approaches to model dynamics of scrolling and zooming. For example, [27] present a model of the dynamics of speed-dependent automatic zooming. [69] extends this approach to enable manual control in semi-automatic zooming on touch devices. [118] identifies an empirical model of the dynamics of scrolling in commercial touch interfaces. As a very good overview of continuous interaction, [149] discusses continuous interaction with explicit focus on uncertainty. Thus, a small number of dynamics models of interactive systems exist in particular in the case of scrolling and zooming. However, for physical interfaces that involve moving parts in the real world, we are not aware of any dynamics models. With this chapter, we hope to promote the modelling of interface dynamics in HCI. We showcase the approach on the example of a physical interface that involves levitating particles.

4.2.2 Interaction with Interfaces using Ultrasonic Levitation

There have been several implementations of direct interaction with levitating displays. The *Point-and-Shake* [37] method allows users to select levitated particles by pointing at them. When a particle is selected, visual feedback in the form of a particle side-to-side movement is provided. The *Pixie Dust* [101] setup enabled gesture interaction with levitated particles using a Kinect. Direct control of a single levitated particle using a mouse, keyboard or a Leap Motion sensor was shown in *LeviPath* [103]. The *LeviCursor* [9] method employed precomputation of transducer phases, to achieve smooth, high agility and low latency direct gesture-based interaction with a levitated cursor in any direction.

4.2.3 Summary

In summary, there is a considerable interest in HCI in developing interactive ultrasonic levitation systems. However, developing such systems is challenging. When observing the videos of developed systems, phenomena such as oscillations and instabilities are visible. We believe that a deeper understanding of the dynamics of interactive ultrasonic levitation would greatly help the development and research of such systems. Thus, we hope that our models are useful to researchers and practitioners working with interactive ultrasonic levitation. We also believe that interactive ultrasonic levitation is a good example of a complex interactive system to show the value and utility of models of interaction dynamics in HCI more generally.

4.3 Ultrasonic Levitation System

Our ultrasonic levitation system consists of subsystems for acoustic levitation and optical motion capture. The acoustic subsystem generates the acoustic field that applies forces to a particle which levitate and move it. The motion capture subsystem provides real-time feedback concerning the position of the particle in the levitation volume and the position of the users' fingertip as well as optical markers on the ultrasound transducer boards. The motion capture subsystem allows to record particle position data or stream the data to external applications. We use an OptiTrack system with six Prime 13 cameras.

The acoustic subsystem uses two rectangular arrays of 14x9 ultrasonic transducers each. The arrays oppose each other and generate a sound field in a levitation volume of 24x20x19cm. Both arrays are connected to a single controller board, which ensures accurate synchronisation of the acoustic signals. The controller board receives a stream of signals through a USB 2.0 interface from a high-end workstation optimised for real-time performance. Controller boards and transducer arrays are manufactured by Ultrahaptics Ltd. We use a custom cooling mechanism to prevent overheating of the transducer arrays. To compute the sound field that achieves levitation we use the Focus+Signature (Holographic Acoustic Element) algorithm presented by Marzo et al. [86].

4.4 Calibration

Since the trap is small (approx. 5mm diameter in height, see Figure (4.1, centre)) it is important that the levitation and motion capture subsystems are accurately calibrated. Both systems need to be calibrated *internally*, i.e. the transducer positions and the camera intrinsics and extrinsics. Additionally, both systems need to be *externally* calibrated in relation to each other. That is, the coordinate systems of the sound field and the motion capture need to be related to each other. We calibrate the optical motion capture system *internally* by wanding, using the calibration algorithm of the Motive software achieving up to 0.05mm accuracy. We initially calibrate the levitation subsystem *internally* by measuring the transducer distances as well as the distance between the boards with a ruler. While the boards are accurately produced, we observe small errors between the physical boards and their idealistic representations in software, which can accumulate up to 2mm along the board. In order to provide an initial *external* calibration of both systems in relation to each other, we manually place optical markers on the transducers. This *manual calibration* achieves a calibration accuracy of approx. 2mm.

Beside the main trap, the levitation field contains many secondary traps, which are also able to levitate particles. By visual inspection, it is impossible to see whether the particle levitates in the main trap. Therefore, we use the following procedure to ensure that the particle levitates in the main trap. First, we place the particle in the levitation volume on an acoustically transparent support. We then activate the sound field and manually find a place where the particle levitates. Using the motion capture system, we determine the location of the particle and generate a new sound field with the main trap exactly at the location of the particle. We then move the particle to the starting position using the sound field.

For the purposes of this chapter, a tracking accuracy of 2mm and additionally, board accuracy of 2mm are not sufficient. Therefore, we developed the following *algorithmic calibration* method for accurate *external* calibration between levitation and tracking subsystems and improvement of *internal* calibration of the levitation subsystem. Our method assumes that the tracking subsystem is already *internally* correctly calibrated, the *internal* calibration of the levitation subsystem and the *external* calibration between the two subsystems, however, exhibit errors within the 2mm range. Therefore, the *external* calibration between both subsystems is reduced to a rigid transform (translation and rotation) and the *internal* calibration of the levitation subsystem is reduced to anisotropic scaling.

In order to obtain this calibration, we automatically move the levitating particle through a number of fixed positions, track its observed position, and obtain point correspondences. The set of calibration points has to be symmetric with respect to each of the three orthogonal planes aligned with the coordinate axes to avoid a shift of the centre. After moving the particle to a new location, it oscillates for a fraction of a second before settling. Therefore, we wait for the oscillations to settle and then record the particle positions for one hundred frames. We average these measurements to estimate the observed particle position. We obtain a number of point correspondences at the limits of the levitation volume.

In order to compute the rigid transform from the point correspondences, we use the method described by Sorkine-Hornung and Rabinovich [133]. The method uses the Singular Value Decomposition to obtain a rigid transform that is optimal in the least squares sense. Then we compute ratios between the distances between calibration points specified in the levitation subsystem and the distances between the corresponding points recorded by the motion capture system. We use individual ratios for X and Z dimensions (plane parallel to the transducer arrays) and do not change the Y dimension. The identified transform is applied to all motion capture data to be represented within the acoustic coordinate system.

In order to compare the manual and algorithmic calibration, we performed a short technical evaluation. We performed our algorithmic calibration using seven calibration points (one in the centre and six equidistantly in both directions along each coordinate axis). Then we placed the bead at twenty-seven different measurement positions within the levitation volume. We recorded its tracked position with and without our calibration method. The results show, that with manual calibration, the errors between the motion capture points and the corresponding levitation points are 2.2mm on average and can be up to 4.3mm. The size of error is not correlated with any spatial properties of the field. The algorithmic calibration reduces the mean error to 1mm. The maximum errors are 1.7mm at a distance of 8.6mm from the centre. Additionally, the errors are 0.5mm in the centre of the field and slowly increase towards the boundaries. By improving the anisotropic scaling and taking into account a precision of vertical arrays placement lower than 1mm, it should be possible to reduce the errors in the whole volume further to 0.5mm to match the accuracy in the centre.

4.5 System Identification

As a first step towards understanding the dynamics of ultrasonic levitation, we performed *black box* system identification. This approach can be used when only input and output of the system is known. It is not necessary to understand internal system dynamics.

In this chapter, we only investigate one-dimensional models (as an example, we use the x-axis). This simplifies the situation considerably and many of the important phenomena can already be observed in the one-dimensional case. This model is valid only in the case when the particle is in the centre of the trap along the other axes (y and z). This is the case, for example, when the particle is moved only along the x-axis. The one-dimensional model can easily be adapted to movement exclusively along the y and z-axes. We assume that a decomposition into the individual dimensions might still be a good approximation for three dimensional movement as long as the particle does not deviate much from the trap centre. However, this needs to be validated in future work. We leave modelling of the full three dimensional case, which is considerably more complex, to future work.

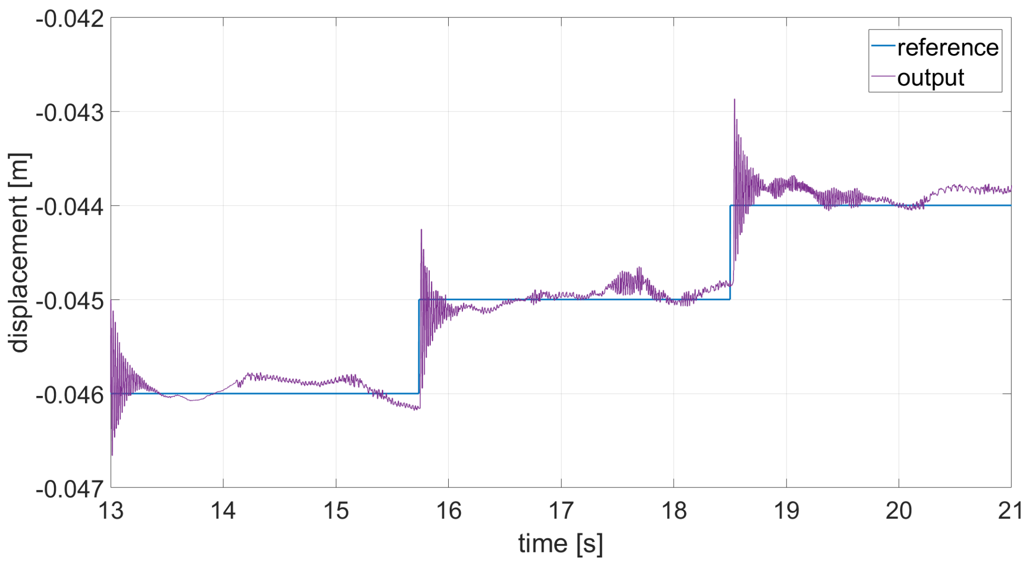


Figure 4.3 Measured position of the levitating particle compared to the reference step function.

In order to identify the black-box model, we record the system response to different input signals. We drive the system with predetermined reference trajectories for the particle (*input*). The input functions we use here are: uniform step, variable step, sine, chirp, sine with 2mm step and sine with 4mm step. We then record the actual particle trajectories, using motion capture (*output*). A sample of the collected data for a step input function is shown in Figure 4.3. The obtained data are preprocessed and partitioned into two sets. The training set comprises $\approx 80\%$ of the total data. The test set comprises $\approx 20\%$ of the total data. We employ the *Matlab System Identification Toolbox*, which uses an iterative numerical algorithm, to produce a model estimation. Thus for our levitation data, we obtained a system with the following transfer function T_f of the Laplace-domain variable s:

$$T_f(s) = e^{-0.01s} \frac{0.001831s + 3.807 \cdot 10^{-6}}{s^3 + 0.02776s^2 + 0.001795s + 3.825 \cdot 10^{-6}}.$$

This model can be used to simulate the movement of the levitated particle for a variety of input functions. On our test set, it achieves 89.7% goodness of fit.

4.6 Forces in Ultrasonic Levitation

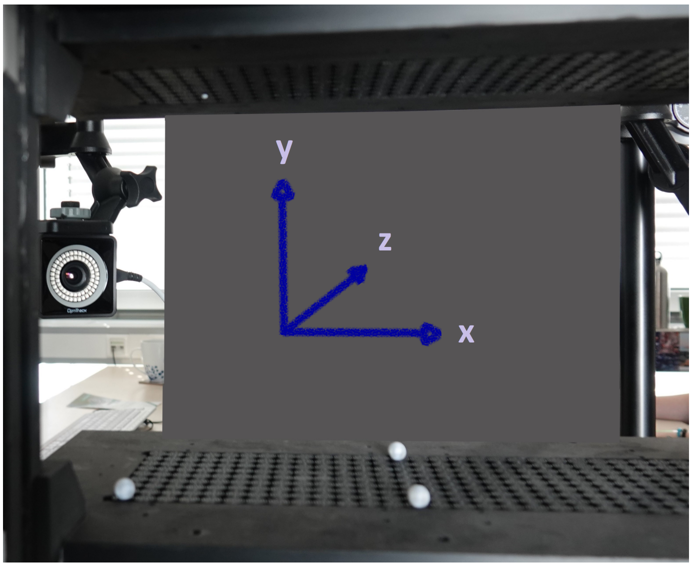


Figure 4.4 The physical levitating interface and the coordinate system we use in our analysis.

Black-box system identification provides a model that allows us to predict particle movement in different situations. However, it does not provide us with a deep understanding of the causes of this particle movement. To achieve this, we now analyse the acoustic forces when a levitation trap is generated. The force data is obtained adopting the implementation by Marzo et al. [86], based on a theory from acoustofluidics, that describes the acoustic force acting on small particles in an ultrasound field using the Gor'kov potential [18].

The levitation volume of the physical display has a cuboid shape with dimensions 24x20x19cm. The coordinate system is defined as shown in Figure 4.4. Figure 4.5 shows the force field within a trap generated at the origin.

We observe strong acoustic force vectors in the y-direction, pushing the particle from above and below towards the centre of the trap. Figure 4.6 shows each force component individually.

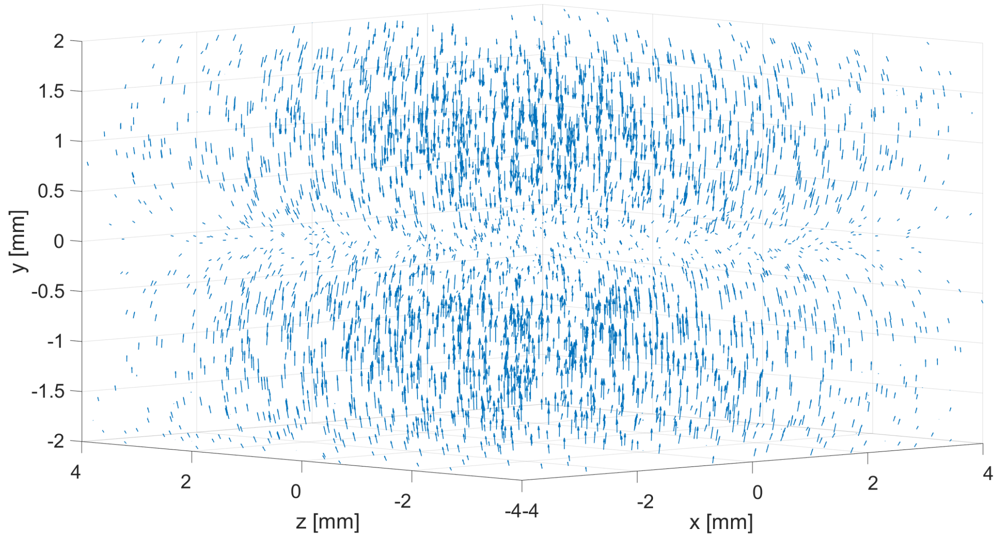


Figure 4.5 Vector representation of the 3D acoustic force field when a levitation trap is generated at (x,y,z)=(0,0,0).

In addition we observe that in the vicinity of the levitation trap, the force profile is of a sinusoidal shape, in all three directions. This information is important for developing the mathematical model in the following section. The fact that we have a system with fairly regular oscillations, motivates us to use a modified version of the *harmonic oscillator* model to model the interface.

A close up look at the force profiles (Figure 4.7) in the trap region, reveals the maximal distance we can push a particle away from the trap centre and it would still be pulled back at its original (equilibrium) position i.e. the trap size. This leads us to a levitation trap with an ellipsoidal shape of diameters $d_x \approx 16$mm, $d_y \approx 5$mm and $d_z \approx 20$mm, as illustrated in Figure 4.8.

The maximal force the levitating particle can experience in the acoustic field sets the physical limit for its velocity and acceleration. The maximal force in the x-direction is $4.4 \cdot 10^5$N, $2.4 \cdot 10^4$N in the y and $3.5 \cdot 10^5$N in the z-direction.

From this analysis, we can already make a number of conclusions. The maximal distance that we can displace the trap instantaneously in the x, z, and y directions are 8mm (x), 10mm (z), and 2.5mm (y). If we displace the trap further than that at a single frame, the particle will leave the main trap. However, the particle is not accelerated maximally if we maximally displace the trap. Instead, maximal acceleration is achieved at displacements of 4mm (x), 5mm (z), and 1.25mm (y), respectively. This means, that the placement of the trap relative to the particle should be carefully computationally controlled.

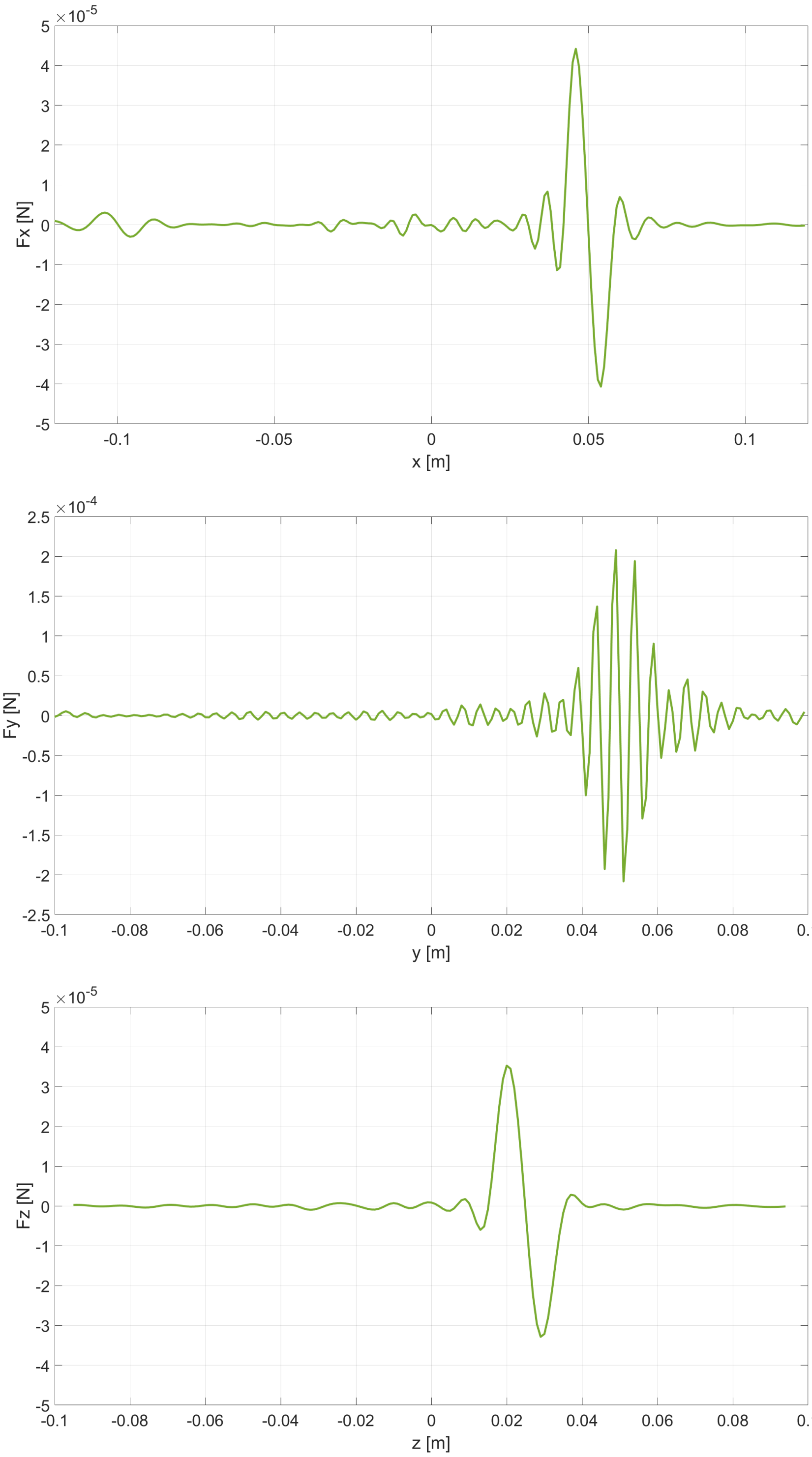


Figure 4.6 Acoustic force profiles in x, y and z-direction, across the entire volume of the levitating interface, when a levitation trap is generated at (x,y,z)=(0.05,0.05,0.025). The trap was generated using a focusing algorithm.

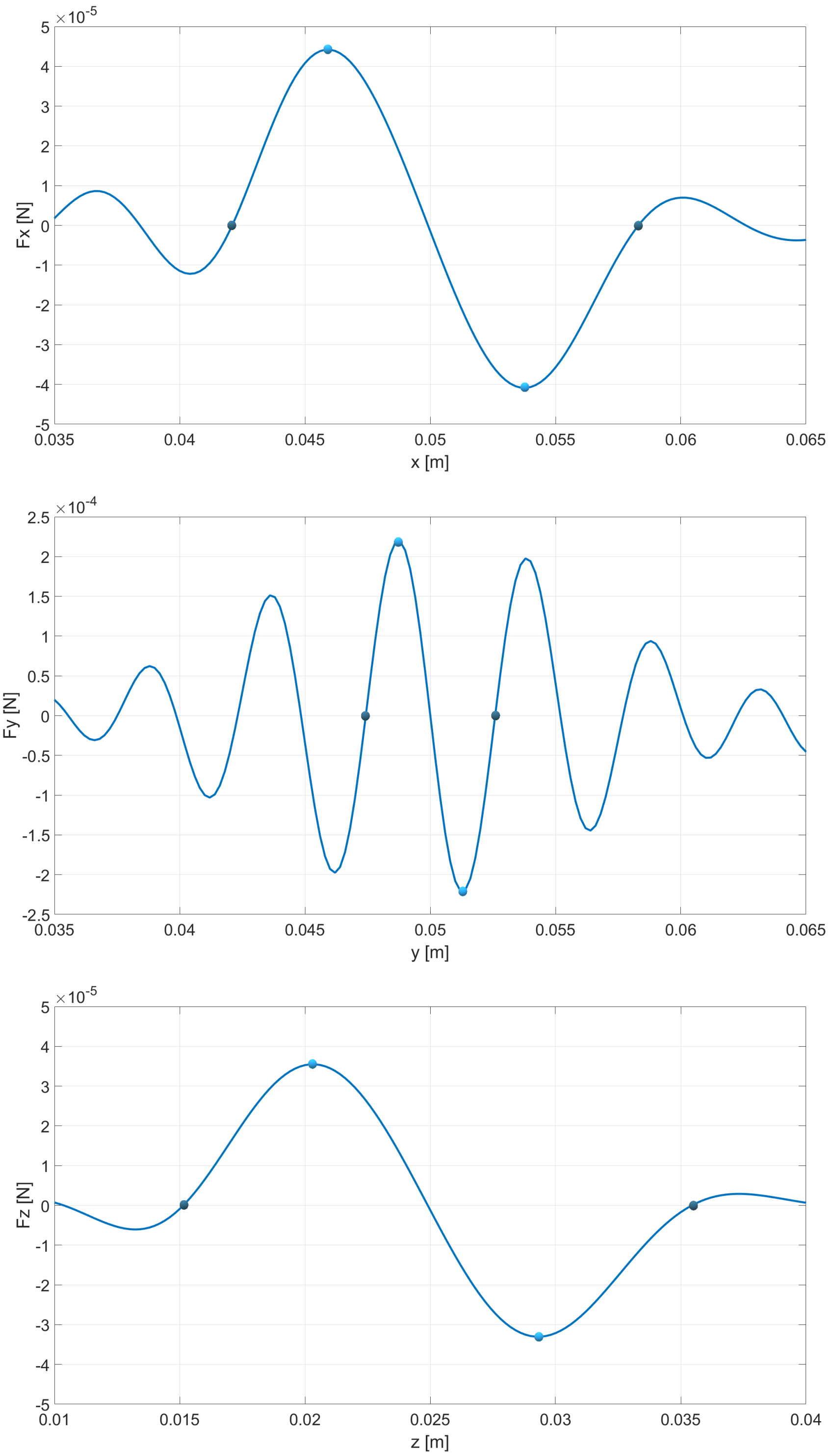


Figure 4.7 Acoustic force profiles in x, y and z-direction, around the levitation trap centre located at the point (x,y,z)=(0.05,0.05,0.025). The dark blue dots denote the trap boundary, whereas the light blue ones - the areas inside the trap where the particle experiences the largest directional force. Note that the force in the y dimension is almost an order of magnitude higher than in the other dimensions.

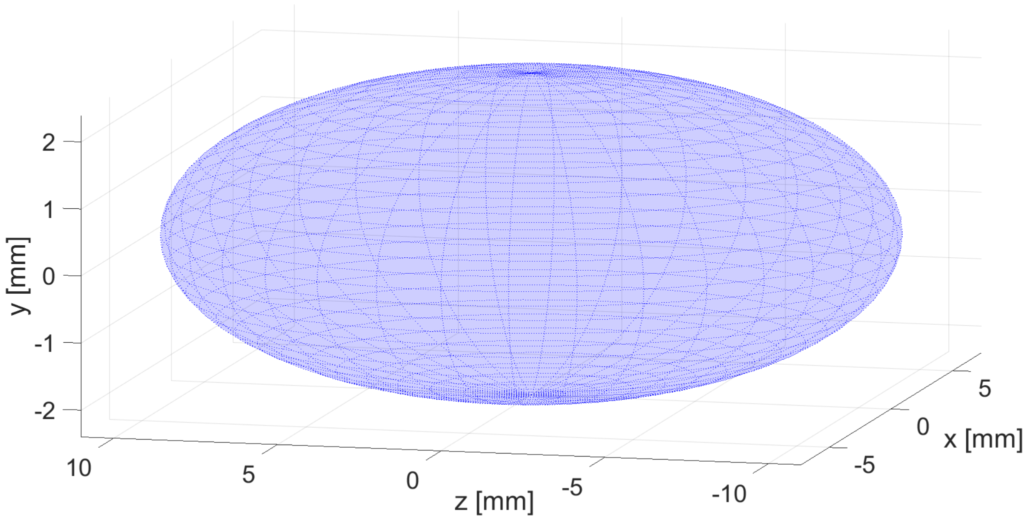


Figure 4.8 3D representation of the shape of a trap generated at (x,y,z)=(0,0,0).

Furthermore, the acoustic force in the y-direction is higher. This implies that the levitating particle can achieve higher velocities and accelerations in this direction compared to the other two directions. This result can have implications for performing pointing tasks on the physical display. While in pointing we often assume homogeneity of movement in all directions, this assumption is clearly violated for ultrasonic levitation.

4.7 A Semi-analytical Model of Ultrasonic Levitation Dynamics

Having obtained a model of the forces within the trap, it is now possible to derive a semi-analytical model of the particle movement. This nonlinear physical model is mainly derived using fundamental physical laws. Our preliminary analysis of the acoustic forces in the levitation volume, shows that the dynamics of the levitating particle in the vicinity of the trap can be approximated by that of a damped harmonic oscillator. The acoustic force acts as a restoring force, pushing the particle towards the centre of the trap, while air friction dampens its motion.

Let us denote the displacement of the levitating particle from the trap by x. The balance of forces using Newton's second law gives

$$F = ma = F_{ext} + F_a - F_{drag}, \tag{4.1}$$

where F is the net force acting on the particle, F_{ext} is the external force, e.g., from gravity or wind; F_a is the acoustic force, produced by the ultrasonic transducers; F_{drag} is the air resistance, m denotes the mass of the levitating particle and a its acceleration. We assume that when the particle moves through air there is no turbulence (i.e. low Reynolds number), so the drag force can be modelled to be proportional to the velocity. We set $F_{drag} = b\frac{dx}{dt}$, where b is the damping coefficient, that describes the decay of oscillations after the particle has been disturbed. The gravitational force acting on the levitating particle is two to three orders of magnitude, depending on the particle size, smaller than the acoustic force, so it is neglected in the model. In addition, we make the assumption that no other external force acts on the system, i.e. $F_{ext} = 0$. Hence, we obtain $F = ma = F_a - b\frac{dx}{dt}$, which by setting $v = \frac{dx}{dt}$, can be rewritten into the system of equations

$$\dot{x} = v \tag{4.2}$$

$$\dot{v} = \frac{F_a - bv}{m}. \tag{4.3}$$

Next we present an analytical expression for the acoustic force, as well as experimentally identify the damping coefficient.

The theory of the acoustic radiation force, stemming from the scattering of acoustic waves on a small particle, has been extensively discussed in [18]. Without going in detail with the derivation, here we only present the fundamental acoustic force expression

$$F_a = -\nabla U = -U_{xx} - U_{yy} - U_{zz}, \tag{4.4}$$

where the force is described as a gradient of the Gor'kov potential U [48], which in turn is dependent on the complex acoustic pressure and its spatial derivatives.

We note that the analytical expression of the acoustic force (second order partial differential equation) is too complex for our modelling purposes, so we would like to approximate it, in the neighbourhood of the trap, by a polynomial function of the form: $p_0 + p_1x + p_2x^2 + \ldots + p_nx^n$. To identify such polynomial, we use the Gor'kov potential. We sample equation (4.4) at a regular grid with 0.2mm resolution within the levitation volume. Then we use the least-squares fitting method to identify a polynomial model from these samples in the vicinity of the trap. We obtain a polynomial model of order $n = 5$ with parameters $p_0 = -1.479 \cdot 10^{-6}$, $p_1 = -1.677 \cdot 10^{-5}$, $p_2 = 2.544 \cdot 10^{-7}$, $p_3 = 4.327 \cdot 10^{-7}$, $p_4 = -4.261 \cdot 10^{-9}$ and $p_5 = -2.675 \cdot 10^{-9}$, shown in Figure 4.9. Optionally, the semi-analytical model can be further simplified, if we linearize the acoustic force at the trap centre,

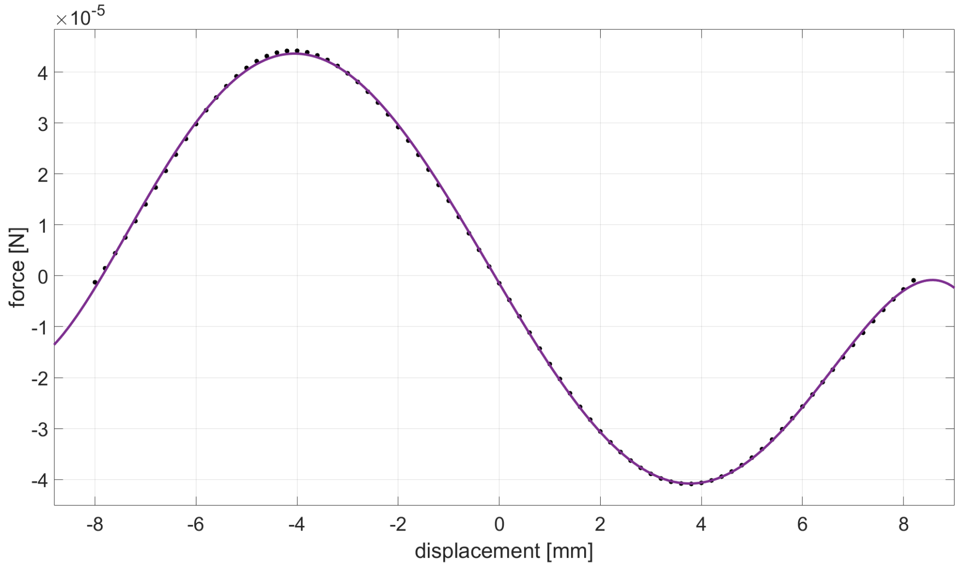


Figure 4.9 Acoustic force F_a in the x-dimension, calculated as the gradient of the Gor'kov potential (black dots) and as a 5^{th} degree polynomial model (purple line) for a trap centred at $x = 0$.

as illustrated in Figure 4.10. This model describes the system dynamics on a more restricted area, but leads to a significantly simpler (linear) overall model.

We measured the diameter of the levitating particle and used an estimate of the material density of expanded polystyrene, to obtain the estimate for the particle mass $m \approx 9 \cdot 10^{-8}$kg. The only model parameter left to complete our gray-box model of acoustic levitation is the damping coefficient b. We conducted a small experiment with a high frame rate camera, where we displaced the levitating particle, a few millimetres (1 to 5mm), away from the trap centre and we recorded the oscillations. The setup is shown in Figure 4.11. We processed the experimental data with the *Matlab Parameter Identification* algorithm and obtained $b = 2.6 \times 10^{-6}\,\mathrm{N\,s\,m^{-1}}$.

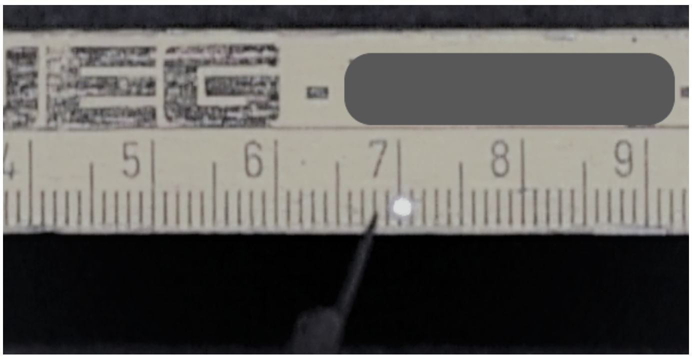


Figure 4.11 Experimental data collection: the levitating particle was displaced with a thin needle, in several trials, 1 to 5mm from the trap centre; the data was recorded with a high frame rate camera and the oscillations were analysed.

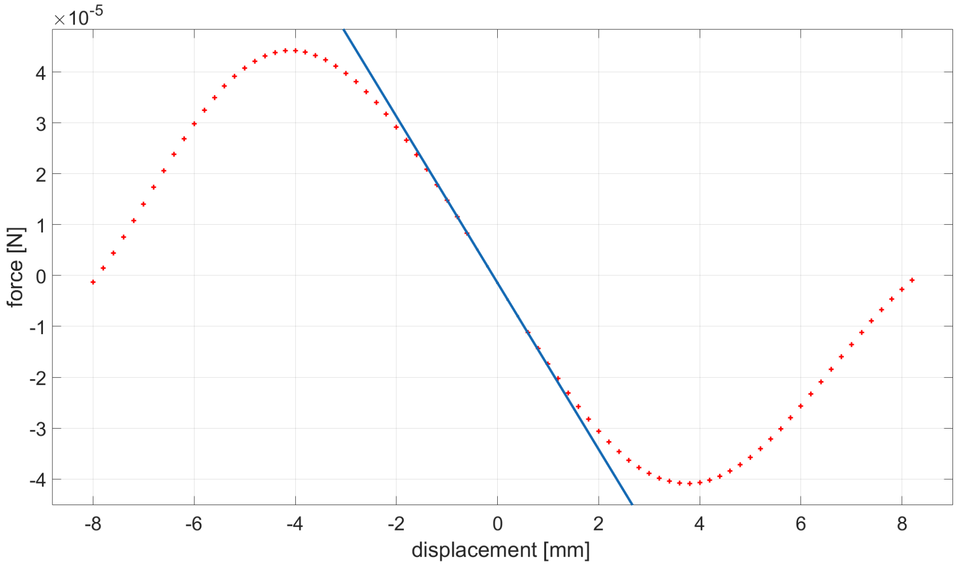


Figure 4.10 Acoustic force F_a in the x-dimension, calculated as the gradient of the Gor'kov potential (red crosses) and as a linear approximation at the trap centre $x = 0$ (blue line).

The full model was implemented in the *Matlab Simulink* environment, as shown in Figure 4.12. Figure (4.13, up) displays the results of the simulation in the case when the levitating particle, resting in a stationary acoustic field, is initially displaced 3.1 mm away from the trap centre. With a minor modification of the model, we are also able to simulate the case when the particle is being moved across the field by the acoustic force. This is done by continuously moving the trap centre along the desired particle trajectory. The simulation results for a moving acoustic field are shown in Figure (4.13, down).

From our semi-analytical model, we can already make a number of conclusions about fundamental limitations of interaction with the system. Given the force of the sound field that we obtained in the previous section, and the damping coefficient b obtained in this section, we can compute the theoretical limit for the velocity of the levitating particle. We can also use force and mass estimates to compute the theoretical limit of acceleration of the particle. We obtain a velocity of 17 $\mathrm{m\,s^{-1}}$ and acceleration of 495 $\mathrm{m\,s^{-2}}$ in the x-direction, 85 $\mathrm{m\,s^{-1}}$ and 2473 $\mathrm{m\,s^{-2}}$ in the y, and 13 $\mathrm{m\,s^{-1}}$ and 393 $\mathrm{m\,s^{-2}}$ in the z-direction. Furthermore, we can also derive the settling time of the particle. This is the time that it takes for the particle oscillations to decay after a jump of the trap. When users control the particle to move it to a small target, this settling time will influence their performance. Settling time in x direction for jumps of the trap of around 3 mm is approx. 12 ms (see Figure 4.13).

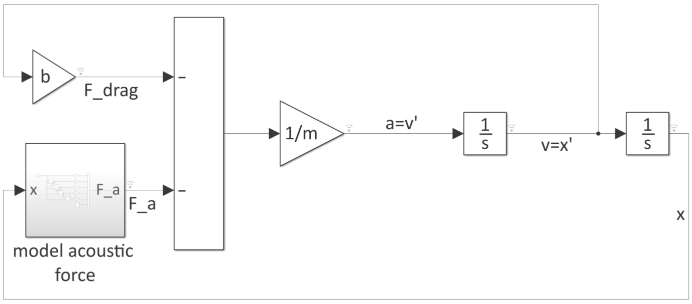


Figure 4.12 Implementation of the semi-analytical model in Simulink.

4.8 Discussion

Traditionally, design of interactive systems in Human-Computer Interaction is centred around an iterative design process. Designers would build a prototype and try it with users, improving the system over several iterations. Models of system behaviour and of the interaction are not always used. This can be contrasted with other engineering disciplines, such as electrical, mechanical, or automotive engineering, which have traditionally centred their design processes much more around models of the system to be designed.

The iterative design process centred around prototypes and user studies has been tremendously successful in HCI. However, we argue that, as HCI is starting to investigate the dynamics of interactive systems and other complex phenomena, this process needs to be augmented by a model-based design process. For example, the design of interactive levitation systems is highly complex, and such systems are very difficult to debug. In case of problems, the levitated particle would usually simply be ejected from the levitation volume. There is a large number of possible causes for problems. The observed phenomena are complex, including oscillations and instability of the particle. The number of system parameters is large, and system behaviour depends on parameter settings in non-trivial ways.

For interactive systems of this complexity level, availability of a good system model is mandatory. The model helps the designer to think about system behaviour, drive design decisions, and track down problems. The model also enables the designer to investigate system behaviour before even building a prototype. This is especially useful for complex systems, such as ultrasonic levitation, where implementation even of a simple prototype is difficult and costly. Before building the system, designers could simulate important questions, such as: What would be the amount and duration of oscillations during interaction? What would be the maximum achievable particle speed and acceleration? Based on these, what is the theoretically achievable throughput? What is the necessary accuracy of system calibration? How do the different frame rates in the system, including motion capture,

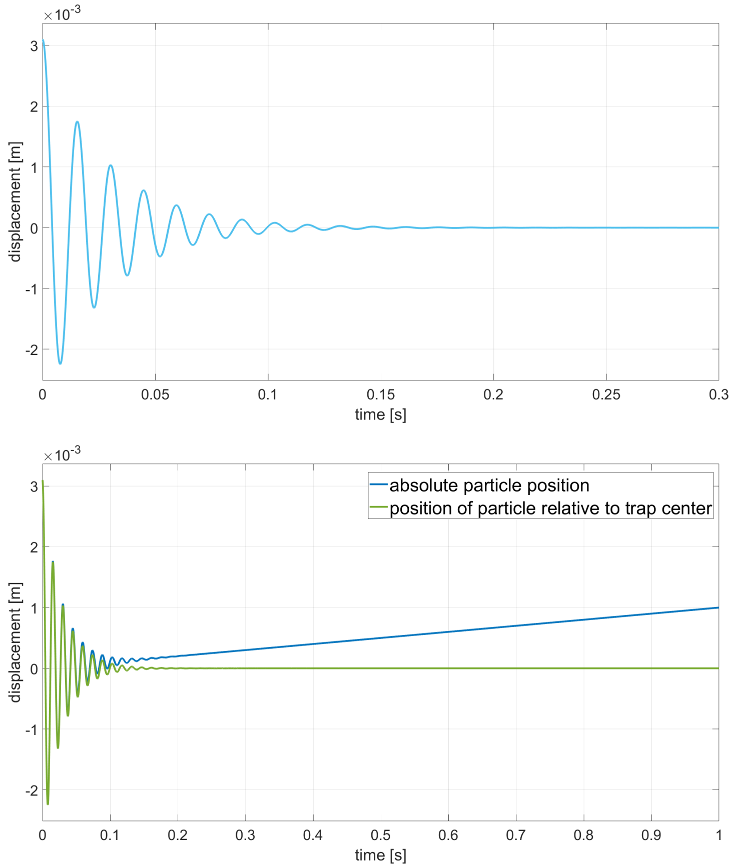


Figure 4.13 Simulation of the system behaviour when the particle is displaced 3.1 mm from the trap centre in a stationary acoustic field (up). Simulation of a moving trap along the x-direction (down).

processing, and transducer driver board, impact system performance? What is the maximum size and mass of levitated particles?

The model of system dynamics could also be connected to a model of human interaction dynamics, such as the one presented in [92]. This would allow an automatic simulation of the entire human-computer interaction loop, without involving real users. Such a simulation could be used, for example, together with computational optimisation in order to optimise system parameters, such as transfer functions that map from user input to particle behaviour.

In this chapter, we demonstrated two fundamental kinds of dynamical systems models: Black-box and semi-analytical (grey-box). Black-box models have the benefit that we do not need to know anything about how the system works. As a consequence, while they allow us to simulate the system and make predictions, their contribution to our understanding of the system is limited. With semi-analytical models, part of the dynamics is known from theory and the other part is completed with data, e.g. from existing literature or from experiments.

These models can serve as mental models for the designer, guiding intuition, design, and debugging.

While in this chapter, we develop such models for the special case of interactive ultrasonic levitation, we hope that the modelling of system dynamics will also become more popular in other areas of HCI. Depending on how much we understand about the system and what our requirements are, black-box or (semi-)analytical models can be more useful. Obvious cases include those that involve real-world physics, including shape-changing and robotic interfaces, and simulated physics, such as in Virtual Reality. However, the approach also generalises to general interactive systems, including desktop, mobile, and ubiquitous devices.

4.9 Conclusion

In this chapter, we show how HCI can benefit from modelling the dynamics of interactive systems. We do this on the application example of interactive ultrasonic levitation. First, we identify an empirical model of particle dynamics from experimental data. Next, we provide an analysis of the forces acting on a levitated particle in an ultrasound field. Finally, we derive a semi-analytical model of the particle dynamics. We hope that our models can help researchers in interactive ultrasonic levitation better understand the underlying dynamics of such interfaces, simulate their behaviour, and adjust the interaction accordingly. An example of how such dynamic interactive simulation could look like and be used in testing with users, is described in Chapter 5. We believe that this mathematical approach to the "feel" of interfaces will strengthen the focus on the dynamical aspects of HCI. Lastly, we hope that our approach also inspires researchers in other areas of HCI to build models of the dynamics of interactive systems.

Chapter 5

Prototyping Acoustic Levitation Interfaces in Virtual Reality

5.1 Introduction

Ultrasonic Levitation Interfaces promise a future in which the computer can control the existence of matter in our environment, truly merging real and virtual worlds. This is similar to approaches such as Programmable Matter [44] and Radical Atoms [57]. However, with Ultrasonic Levitation, power supply, actuation, and computation are placed in the environment making the individual atoms simpler and cheaper.

Despite the potential of such interfaces and the accumulated body of research [53, 89, 86, 37, 101, 9], not many applications for levitation interfaces have been developed thus far. Among the few examples are a two-platform jump game [126], and augmented static objects (e.g. volcano) [36]. The main reason that so few applications exist is that it is difficult for designers, artists, game developers and even researchers to work with ultrasonic levitation interfaces.

Building and maintaining an interactive levitation interface, requires time, resources and very specific technical expertise. The underlying physics are nontrivial. The interface requires microsecond synchronisation of all ultrasonic transducers to ensure a correct sound field. It also requires exact calibration of all physical parts. Such systems are also difficult to debug. In case of problems, the only observable effects are usually the levitating particle dropping or shooting out in an uncontrolled way. The sources of such problems can be various. It could be a bug in the software calculating the phase shift for each transducer. It could be a bug in the underlying libraries, firmware or hardware controlling the ultrasonic transducers. Finally, it could be physical effects such as reflections of sound waves off

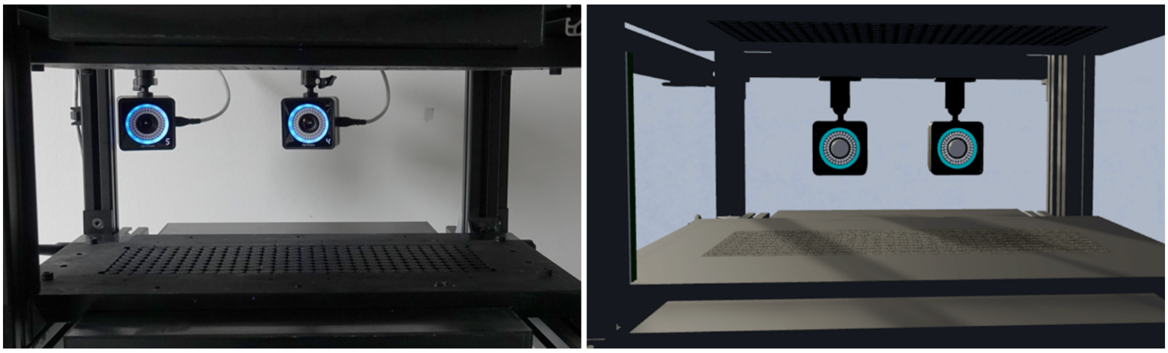


Figure 5.1 The *Levitation Simulator* allows to simulate interaction with a levitation interface in VR. To enable accurate and realistic simulation, we develop a model of the movement of a particle in an ultrasound field. Two user studies, comparing the real prototype (left) with the VR simulator (right), show that the VR simulation provides a good approximation of the interaction with the levitating particle in the real prototype. We share our *Levitation Simulator* as Open Source (www.ai8.uni-bayreuth.de/en/projects/Levisim), thereby democratising levitation research and facilitating the design of applications for levitation interfaces, without the need for a levitation apparatus.

neighbouring surfaces. The physical and technical challenges potentially lead to a lack of applications, progress and replication.

Further, when developing such systems, it is uncertain how far one is from the performance ceiling. If one would invest in higher frame-rates, better synchronisation, better transducers etc., how much would user performance improve? Would there be a noticeable improvement in the user experience when interacting with the upgraded interface? Lack of this knowledge leads to missed opportunities in some areas and futile iteration in others.

To address these problems, we present the *Levitation Simulator*, a prototyping tool aiming to facilitate levitation research and content creation for levitating interfaces. Instead of having to build an ultrasonic levitation apparatus, designers, artists, software developers and researchers can develop applications and interaction techniques, and perform evaluations and user studies in VR. The volumetric nature of levitation interfaces is preserved in VR, so a comparable visual experience can be provided to the user. Only after the development has converged, the resulting system can be validated using a real levitation apparatus, possibly built by another group.

We derive a model of a particle levitating in an ultrasonic field from first principles. Thus the *Levitation Simulator* is able to physically accurately simulate the dynamics of levitating particles. We validate the Levitation Simulator through a series of two user studies. First we conduct a Fitts' Law - type pointing study in the *Levitation Simulator* and on the real prototype, to obtain a quantitative measure of the pointing performance that can be achieved with both systems. Next, we develop two levitation minigames in the *Levitation Simulator*, which are then implemented on the real prototype. In the second user study, User Engagement levels when playing the games on the real and virtual interface, are compared. With this study, we want to better understand how users engage in the interaction with levitating matter in the simulator and in the physical world, and what are the most frequently observed differences and similarities.

In summary, the contributions of this chapter are:

- A model that describes the dynamics of movement of a levitating particle in an acoustic field.
- An interactive simulation of the levitating interface in Virtual Reality which exhibits the dynamical properties of the real interface - the *Levitation Simulator*.
- A Fitts' Law pointing study involving aimed movements of a 3D levitating particle between two spherical 3D targets, on the real prototype and in the *Levitation Simulator*.
- A user experience study comparing user engagement levels when playing interactive levitation games, on the real prototype and in the *Levitation Simulator*.

More generally, we believe that the *Levitation Simulator* is a good example to promote modelling and simulation of UIs in HCI. Nowadays, the dominant method to develop user interfaces is (physical) prototyping. Physical prototyping , however, can be difficult, time-consuming, and expensive to build, limiting the number of design iterations. In these cases, modelling and simulation of the UI can help to increase the number of design iterations at lower costs.

5.2 Related Work

5.2.1 Acoustic Levitation

Ultrasonic levitation interfaces use acoustic radiation force to counteract gravity and trap small objects in mid-air. This effect can be achieved by using phased arrays of ultrasonic transducers emitting the appropriate phase to create acoustic nodes in mid-air, where the objects are trapped. Typically, these traps are generated by creating an acoustic focus. To create the focus, the phases for each transducer are computed, such that, all acoustic waves constructively interfere at the position, where the levitation trap should be generated.

Even though most research focuses on the technical implementation, several concepts concerning applications for levitating interfaces have been introduced. For example, a game where acoustically transparent structures (e.g. tubes made of metallic mesh) are passed around a particle levitating in a standing wave levitator, similar to BigLev [82], is presented in [85]. *Floating Charts* [105] is a modular display, where levitating particles are used to encode data points on a dynamic chart in mid-air. An example of a levitating particle being used to trace hiking routes and annotate summits across a model of a mountain range, is shown in [36]. The *Pixie Dust* [101] system proposed using levitating particles in combination with a projector to form 2D graphics. The system can be employed to create raster and vector graphics in mid-air (e.g. logos), as well as to animate physical objects (e.g. products in a store window). *LeviProps* [89] are tangible structures composed of an acoustically transparent lightweight fabric and levitating beads as anchors. *LeviProps* can be used as free-form interactive elements and as projection surfaces. A volumetric acoustophoretic display, where a levitated particle is rapidly displaced, while being illuminated to create 3D shapes, is presented in [40]. Hirayama et al. [53] propose a levitating volumetric display that can simultaneously deliver visual, auditory and tactile content, using acoustophoresis as the single operating principle.

Recently, new interaction techniques allowing users to manipulate levitating particles almost in real time, using gestures, have been developed. With the *Point-and-Shake* [37] method, users are able to select levitating objects by pointing their finger at them. Visual feedback is provided in the form of a continuous side-to-side (*shake*) movement. *LeviCursor* [9] is a method for interactively moving a levitating, physical cursor in 3D with high agility. The user controls the levitating cursor with finger gestures. Freeman et al. [36] propose employing these techniques to use the levitating particle as the user representation in interactive applications, such as, to explore landmarks in a miniature model.

5.2.2 VR Prototyping

Prototyping in Virtual Reality is a standard practice in many industries, in particular automotive and aerospace. It is important to notice that there seem to be two different definitions of the term Virtual Prototyping [47]. In engineering, it is sometimes understood as any kind of computer model or simulation of a product. In this chapter, we assume the more narrow computer science interpretation, which specifically relates to the visualisation of prototypes in Virtual Reality. A recent survey of the use of Virtual Prototyping in industry is provided by Berg and Vance [10]. Gomes de Sa and Zachmann [47] explore Virtual Prototyping in the Automotive Industry. Seth et al. [129] provide a comprehensive review of Virtual Prototyping techniques for assembly processes. From the HCI perspective, comparably less studies involving Virtual Prototyping have been conducted. Aromaa and Väänänen [7] conducted a comparative study of AR vs. VR prototyping for ergonomics evaluation of a maintenance panel of a rock crusher machine.

5.2.3 Summary

In general, the number of applications that have been developed for ultrasonic levitation interfaces has been very limited. Most of the applications for levitation interfaces so far, have been introduced as proof-of-concept by groups working on improving technical aspects of levitation. To our knowledge, no interactive content involving complex actions and information, tailored for levitation displays has been developed. We are also not aware of formal studies investigating how users interact with more complex content or applications. With our virtual prototyping tool for levitation, the *Levitation Simulator*, we hope to bridge this gap and make prototyping, testing and evaluation of levitation applications accessible to a much wider audience.

5.2.4 Model-based Design in HCI

In contrast to engineering, model-based design in HCI has traditionally not focused on modelling the physics of interactive devices, but rather on modelling tasks, users, or interactive systems. An overview of classical modelling work in HCI is provided in [111]. Regarding middling tasks, [112] explains how they can be decomposed into individual actions. Based on such models, user interfaces have been formally specified and validated, for example through Petri nets [94]. Further work included the automated generation of user interfaces for different platforms from an abstract description [25]. A very good overview of modern modelling work in HCI is provided in [106]. Most of this modelling work was based on the notion of discrete events and actions. The dynamical behaviour of the user interface, that is, how it continuously behaves over time, has received comparatively less attention.

Even though there have been some publications in the HCI community related to modelling of dynamical systems, we consider that we are still scratching the surface in regard to the potential of this powerful tool. The chapter compares different models for the human part of the continuous interaction loop, but does not investigate the dynamics of the computer side beyond a unit gain. [90] extends this model to model different functions that map mouse movement to pointer movement, including pointer acceleration functions. Thus, a small number of dynamics models of interactive systems exist in particular in the case of scrolling and zooming.

5.2.5 Summary

In summary, one core competency of HCI is developing interactive user interfaces. However, developing such systems, especially the ones involving physical components, is challenging. For example, when observing the videos of the developed levitating interfaces, phenomena such as oscillations and instabilities of the levitating particles are visible. We argue that a deeper understanding of the dynamics of interactive ultrasonic levitation would greatly help the development and research of such systems. Thus, we hope that our models are useful both to researchers and practitioners working with interactive ultrasonic levitation and the ones interested in exploring this novel technology.

5.3 Real Prototype

Our physical ultrasonic levitation system generates the acoustic field that applies forces to a particle, which levitate and move it. A motion capture system provides real-time feedback concerning the position of the particle in the levitation volume, the position of the users' fingertip and any optical markers in the observable volume (e.g., the ones defining a rigid body of a controller).

The system uses two rectangular arrays of 14x9 ultrasonic transducers each. The arrays oppose each other and generate a sound field in a levitation volume of 14x9x10.6 cm. Both arrays are connected to a single controller board, which ensures accurate synchronisation of the acoustic signals. The controller board receives a stream of signals through a USB 2.0 interface from a high-end workstation optimised for real-time performance. Controller boards and transducer arrays are manufactured by Ultraleap Ltd. We wrote an application in C, which receives input from the motion capture system. The application generates the new levitation trap position, as well as runs the experiment.

5.4 Model of the Movement of a Levitating Particle

In Chapter 4, we analysed the dynamics of the acoustic levitation system and derived a semi-analytical model of the movement of a particle levitated in a static acoustic field. In this section, we summarise the findings and present a linear force approximation that will later enable us to simulate the behaviour of a levitated particle in the virtual environment.

Figure 5.2 shows the individual force components, in each spatial dimension. We observe strong acoustic force vectors in the y-direction, pushing the particle from above and below towards the centre of the trap. Also in the vicinity of the levitation trap, in all three directions,

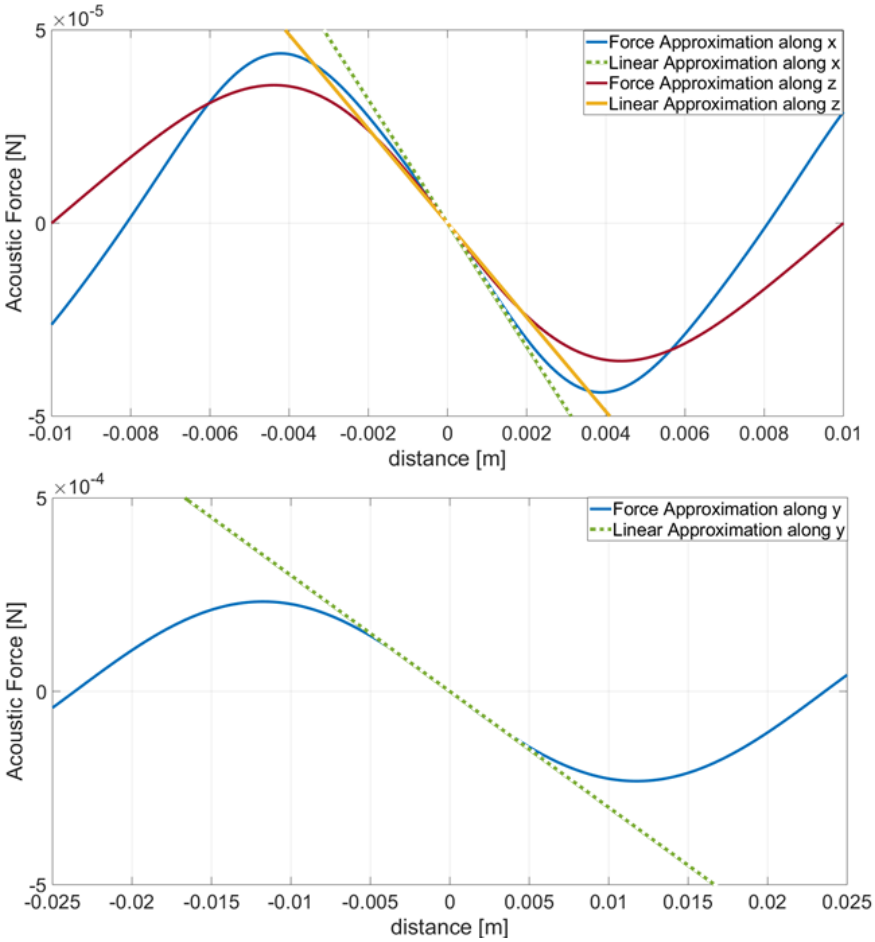


Figure 5.2 Decomposition of the acoustic force in each spatial dimension. Linear approximation in the vicinity of the levitation trap (dashed green and yellow line).

the force profile is of a sinusoidal shape, meaning that the acoustic force acts as a restoring force, pushing the particle towards the centre of the trap, while air friction dampens its motion. The force profiles indicate the maximal distance we can push a particle away from the trap centre and it would still be pulled back at its original (equilibrium) position, i.e. the trap size (ellipsoid with diameters $d_x \approx 16\,\text{mm}$, $d_y \approx 5\,\text{mm}$ and $d_z \approx 20\,\text{mm}$). The maximal force in the x-direction is $4.4 \times 10^{-5}\,\text{N}$, $2.2 \times 10^{-4}\,\text{N}$ in the y and $3.5 \times 10^{-5}\,\text{N}$ in the z-direction. The maximal force that the acoustic field can exert to the levitating particle sets the physical limit for its velocity and acceleration.

Note that the acoustic force in the y-direction is an order of magnitude higher than the two horizontal forces, implying that the levitating particle can achieve higher accelerations and velocities in this direction compared to the other two directions. This result can have implications for performing pointing tasks in 3D on the levitating display. While in pointing we often assume homogeneity of movement in all directions, this assumption is clearly violated for ultrasonic levitation.

As described in Chapter 4, the acoustic force, typically described in physics as the gradient of the Gor'kov potential [48], in the neighbourhood of the trap, can be approximated by a linear function i.e. $F_a = bx$ (see Figure 5.2). We use Taylor series expansion, to identify the vector of coefficients $b = [0.016, 0.26, 0.011]$.

The diameter of the levitating particle is measured to be $d = 0.002\,\mathrm{m}$ and the material density of expanded polystyrene is estimated to be $\rho_{EPS} = 25\,\mathrm{kg\,m^{-3}}$, thus we obtain a particle mass of $m \approx 1.05 \times 10^{-7}\,\mathrm{kg}$. The linear drag constant $c \approx 9.42\,\mathrm{kg\,s^{-1}}$ was empirically obtained.

According to our experiences, the particle movement in VR appears to be indistinguishable from the particle movement in the real prototype. A comparison of simulated particle movements to measured data from the real prototype is provided in Figure 5.3.

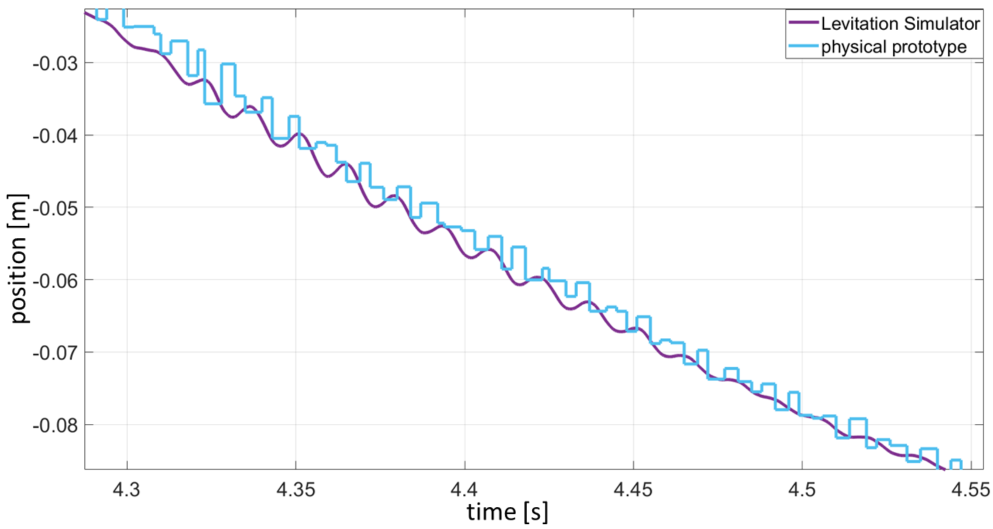


Figure 5.3 Comparison of output data of the *Levitation Simulator* to the measured data from the real prototype.

5.5 The Levitation Simulator

The *Levitation Simulator* consists of an interaction and a simulation module. The interaction module is implemented in C# within the Unity Game engine. Users can interact with the simulated levitation interface either via a motion capture system or via a VR controller. The levitated particles move realistically, as they would with a physical levitation system.

Applications and experiments can be implemented through Unity scripts. One script we provide, records the details of the movement, including particle and finger position per frame with timestamps, to a CSV file. Another script reads the input, e.g., from the OptiTrack Streaming Client via NatNet or from the VR controller. It transforms the input, e.g., by applying C:D gain, and sends the new trap position via UDP to the simulation module. It receives the new particle position via UDP from the simulation module and renders the scene. The entire system runs at 90 FPS, synchronised with the update rate of the HMD.

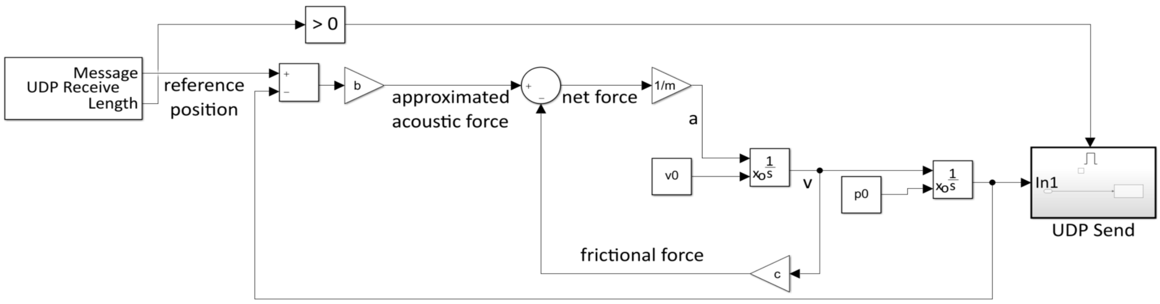


Figure 5.4 Model of the dynamics of the levitating particle in *Simulink*.

The simulation module (Figure 5.4) is implemented in the *Matlab Simulink* environment. It reads the new trap position from a UDP socket. It maintains the velocity and position of levitated particles in two integrators. From the new position of the trap and the current position of the particle, it calculates the force that is acting on the particle. To calculate the force, it uses the model described above. It then simulates the effect of that force on the particle. The new particle position is transmitted to the interaction module via UDP. In order to minimise latency, we use no UDP buffers. We use a variable-step solver (Runge-Kutta (4,5)).

We currently use an HTC Vive Pro HMD and OptiTrack Motion Capture System with Prime 13 cameras.

This design makes the *Levitation Simulator* flexible and easy-to-operate, and makes changes to the underlying model trivial to implement. In particular, we can swap models of particle behaviour to simulate different levitation apparatuses, different levitated particles, or different techniques to generate the sound field. In addition, it allows us to simulate and test application ideas that would take months to implement on the real prototype, such as the Levitating Piano (see Figure 5.5).

The *Levitation Simulator* is simple to use, if one does not want to change interface or simulation. Extending it might require some expertise in Unity and/or Simulink, depending whether one wants to change the interface or the simulation. As such skills are available in the HCI community, we believe that the *Levitation Simulator* will be easy to work with and

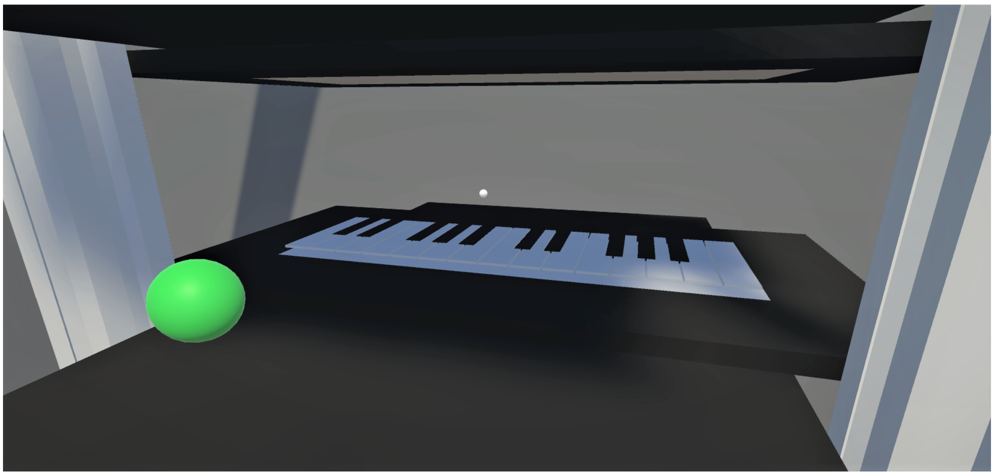


Figure 5.5 The “Levitating Piano” is an application developed in the *Levitation Simulator*. The user has a 3 mm wide retroreflective marker placed on their fingernail. With their finger (green sphere), they control the levitating particle (white sphere) in the levitating display. In this way, they can play a virtual levitating piano, made of acoustically transparent fabric. Whenever the levitating particle touches the fabric, the corresponding piano tone sounds.

extend for research groups in the HCI community. For users that do not wish to experiment with different models of the movement dynamics, we provide a stand-alone Unity version of the *Levitation Simulator*, where the model in Figure 5.4 is integrated into the Unity environment.

5.6 Pointing Study

We are interested in how far the interaction with the real and virtual prototype enables similar pointing performance, and if not, what the differences are. Further, we want to investigate whether the control of a levitating cursor with the real and virtual prototype, is perceived similarly by users. Thus we evaluate the performance achieved by interacting with the *Levitation Simulator*, compared to the real prototype, in a Fitts’ Law type repetitive pointing study.

5.6.1 Experimental Design

Participants performed repetitive aimed mid-air movements between two three-dimensional, spherical targets of varying size (see Figure 5.6). We used a within subjects experimental design with two interface conditions (real and VR), two movement directionality conditions (left-right and front-back), and three index of difficulty conditions (target amplitude 5 cm and

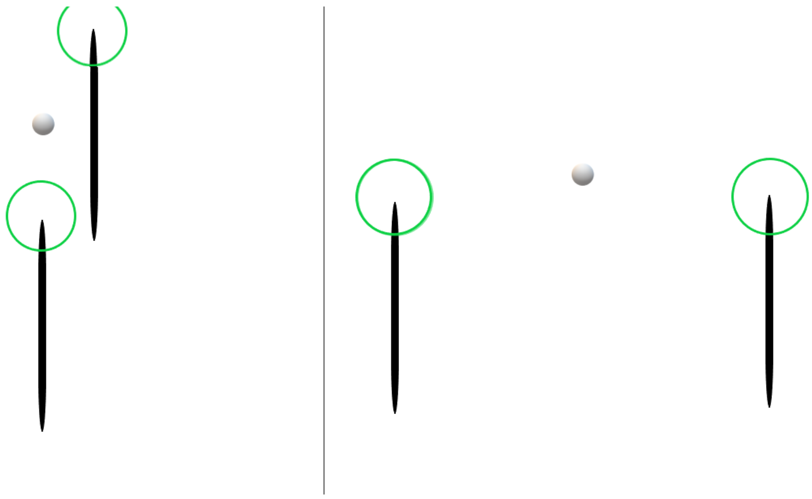


Figure 5.6 Participants perform repetitive aimed movements in two movement directionality conditions: front-back and left-right. The tip of the black needles represents the centre of the the virtual 3D spherical targets, denoted in green.

three target diameters: 4 , 8 and 16 mm). The indices of difficulty covered in the experiment are 2.04, 2.85 and 3.75. The range of possible indices was constrained by the properties of the real prototype: the distance between the targets was constrained by the sound field region allowing stable levitation and agile control, the target size was limited by the size of the cursor and the persistent oscillations of the particle within the trap. The participants performed a set of equivalent Fitts' law reciprocal pointing tasks, both on the real prototype and in the *Levitation Simulator*. The order of interfaces was counterbalanced by randomisation, and the order of conditions on each interface was fully counterbalanced by a Latin square. The participants performed 70 aimed movements for each condition. We recorded both, the position of the fingertip marker and the levitating particle, on a frame-by-frame basis, as well as the timings of the auditory feedback and the individual aimed movement durations.

5.6.2 Participants

Twelve healthy participants (4 females), aged between 20 and 39 (mean 27.25, SD 5.15), were recruited for the experiment. They all had normal or corrected-to-normal vision. All preferred to control the levitating particle with their right hand. Before the experiment, they were provided with basic information about the study and signed a consent form. The experiment was approved by the Ethical Committee of the University of Bayreuth and followed the ethical standards of the Helsinki Declaration. All participants received monetary reimbursement for their participation.

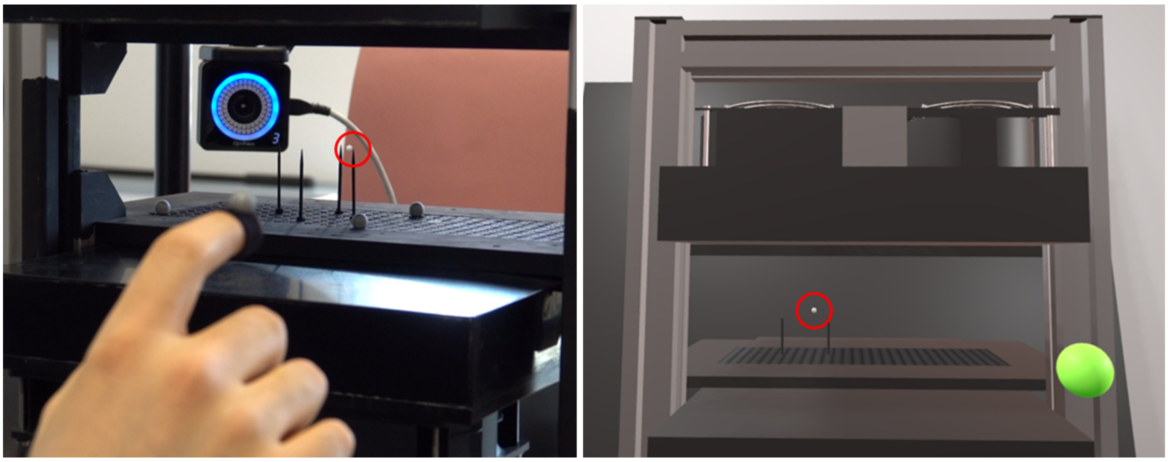


Figure 5.7 Real acoustic levitation interface (left) and *Levitation Simulator* (right). For orientation, the green sphere indicates the position of the user's fingertip.

5.6.3 Apparatus

In the experiment, we have used the real prototype and the *Levitation Simulator*, described in the previous sections, as the basis for control of the levitating particle. The experimental setup is illustrated in Figure 5.7. The targets for the Fitts' law tasks were implemented as thin pointy black sticks, placed between the transducer arrays, to reduce interference with the acoustic field and the optical tracking system. In the *Levitation Simulator*, equivalent sticks were placed in the scene. Internally, in both systems, the targets were represented as spheres of three different sizes, centred at the tips of the sticks. While the actual spheres were invisible, the system produced an auditory feedback as soon as the levitating particle entered the target sphere. Both applications processed and recorded the motion capture data describing the movement of the finger and the position of the (real or simulated) levitated particle. Participants could control the particle motion in 3D with their fingertip, using a control-to-display ratio of 3.

5.6.4 Procedure

Before beginning the experiment, the participants read basic information about the study and a description of the tasks. The experimenter provided clarifications, if the participants had any questions. The participants sat on a chair in front of the apparatus. In the VR condition, they were asked to wear the HTC Vive Pro and adjust it according to their comfort (see Figure 5.8). They were asked to take a comfortable position, adjust the chair height and location to ensure good visibility of the targets both in the *Levitation Simulator* and the real prototype. A retroreflective marker of 9 mm diameter was attached to the fingertip of the index finger of the dominant hand of each participant. In the real condition, the experimenter placed a white polystyrene particle into the levitation volume. We started with an exploration phase, in which the participants explored the interaction with the interface and moved the levitating particle in different directions for approximately 30 s. We asked the participants to place the particle at each of the tips of the target indicators, as accurately as possible. We then calibrated the target locations according to the participants' individual perception of the target. Next, we conducted a training phase for each condition. The participants were instructed to move between the targets as quickly and accurately as possible, for 20 repetitions. In the experiment, participants were asked to perform 70 repetitions for each condition. During the experiment, our software was continuously recording the time-stamped 3D position of the levitated particle at each frame. We also recorded the time stamps when the user reached each target and was notified by the sound. After each condition, the participants were asked to take a short break. After the completion of all tasks in the first interface condition, the participant was moved to another room to continue with the other condition. Subsequent to completing all pointing tasks in each environment, the participants filled in the NASA Task Load Index questionnaire [51]. At the end, we performed a semi-structured interview with each participant.

Figure 5.8 Participant of the user study in the VR condition. The participant was comfortably seated, while wearing an HMD and a retroreflective marker on their index finger, with which they were able to control the virtual cursor in the VR scene. Infrared motion tracking cameras were mounted facing the interaction space.

5.6.5 Analysis

We performed two types of analyses on the collected data. We used Fitts' law modelling for comparing the performance achieved on the real prototype and in the *Levitation Simulator*. We used statistical analysis for the impact of conditions on movement time, as well as for the NASA TLX data.

As preprocessing, we average the movement time values per condition. The movement times were not normally distributed according to the Kolmogorov-Smirnov test ($p > 0.05$), so we opted for a non-parametric statistical analysis. We performed a Friedman test to compare movement times between the interfaces and IDs. We use a Wilcoxon signed-rank test to analyse the NASA TLX data.

The Fitts' law modelling was performed according to the common HCI practice [134]. While the pointing tasks and the targets were represented in 3D space and multivariate Fitts' law models could be applied [49], with spherical targets they are equivalent to the application of Fitts' law in the Shannon formulation:

$$ID = \log_2\left(\frac{D}{W}+1\right) \tag{5.1}$$

$$MT = a + b \times ID \tag{5.2}$$

$$TP = \frac{1}{y}\sum_{i=1}^{y}\left(\frac{1}{x}\sum_{j=1}^{x}\frac{ID}{MT}\right) \tag{5.3}$$

where ID is index of difficulty, D and W are the movement amplitude and target width respectively, MT is movement time, a and b are regression coefficients, and TP is information throughput. In the preprocessing step, we discard the first 20 movements from each trial, as the participants were adjusting to the new size of the invisible target. We then compute ID and average MT for each trial. Before fitting the regression lines, we group all trials according to the ID and compute average MT and ID for each group. Similarly, we use the group values to compute the movement performance TP.

5.6.6 Results

Pointing performance

The results of the Friedman test show no significant differences in movement times between the conditions. We can however observe some trends in the data, which we describe in the following. We can see a trend, according to which with increase in index of difficulty, the movement time increases faster in the VR condition than in the real condition. For increasing

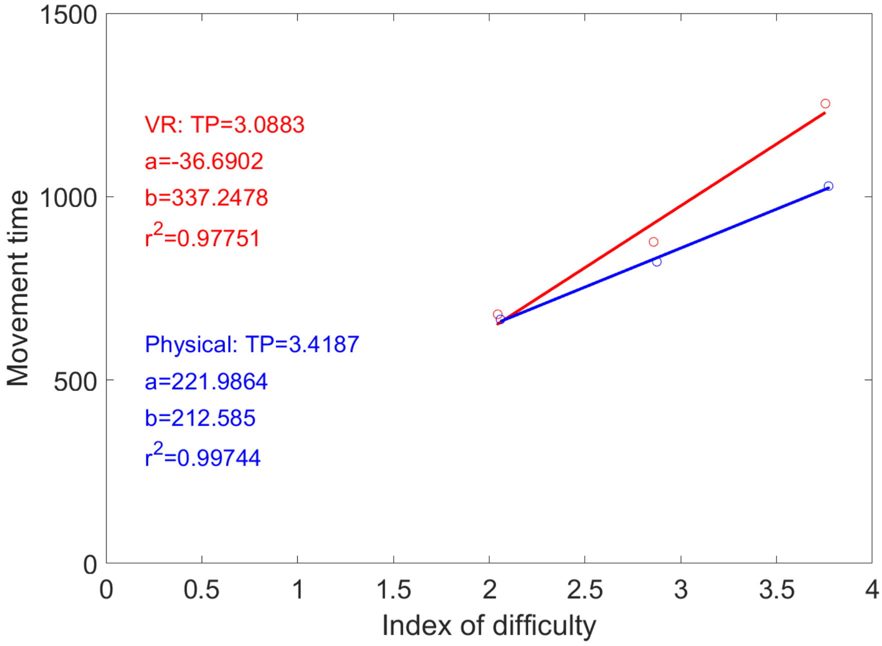


Figure 5.9 Fitts' law models for *Levitation Simulator* vs. physical apparatus conditions. The circles represent the average MT corresponding to the three IDs, in each condition.

IDs of 2.04, 2.85, and 3.75 bits, the average movement time in the real condition is 0.665, 0.823, and 1.028 s, respectively, compared to the respective movement times of 0.679, 0.876, and 1.253 s, in the VR condition.

When considering the effect of movement direction, we observe another trend. With increasing index of difficulty, the movement time increases faster in the left-right condition than in the front-back condition, in the *Levitation Simulator*. For the same increasing sequence of IDs as above, the average movement times in the front-back condition are 0.686, 0.863, and 1.206 s. In the left-right condition, the movement times are 0.671, 0.890, and 1.301 s.

We can see no such trend in the real condition. Both left-right and front-back movements exhibit almost the same movement times for each index of difficulty. There is only a small constant difference. For the same increasing sequence of IDs as above, the average movement times in the front-back condition are 0.674, 0.814, and 1.038 s. In the left-right condition the movement times are 0.656, 0.830, and 1.019 s.

We computed Fitts' law models for the different interfaces, and further considered the effects of movement direction for the whole data. The Fitts' law models comparing the real and VR condition are shown in Figure 5.9. Models in both the *Levitation Simulator* and real prototype conditions, as well as for all four condition combinations provided a good fit for the data with $R^2 > 0.97$. The models reflect the trends in the movement time between conditions described in the previous paragraph. We can see that both conditions are characterised by similar movement times and similar throughput, with the throughput of the real prototype being slightly higher. In the VR condition, the throughput is 3.08 bits/s. In the real condition, the throughput is 3.41 bits/s. The regression line of the *Levitation Simulator* condition is steeper ($b = 337$) than for the real condition ($b = 212$). Movement time increases faster in the VR than in the real condition.

Figure 5.10 shows the Fitts' law analysis, split by interface and direction. We can observe that movement direction does not affect the movement times in the real prototype condition (slope $b = 211$ in left-right condition and $b = 213$ in front-back condition). It does however affect the movement times in the *Levitation Simulator* condition. The movement times of the left-right movements in the VR condition grow faster with increasing index of difficulty, than the movement time of the front-back movements ($b = 369$ for left-right vs. $b = 305$ for front-back conditions).

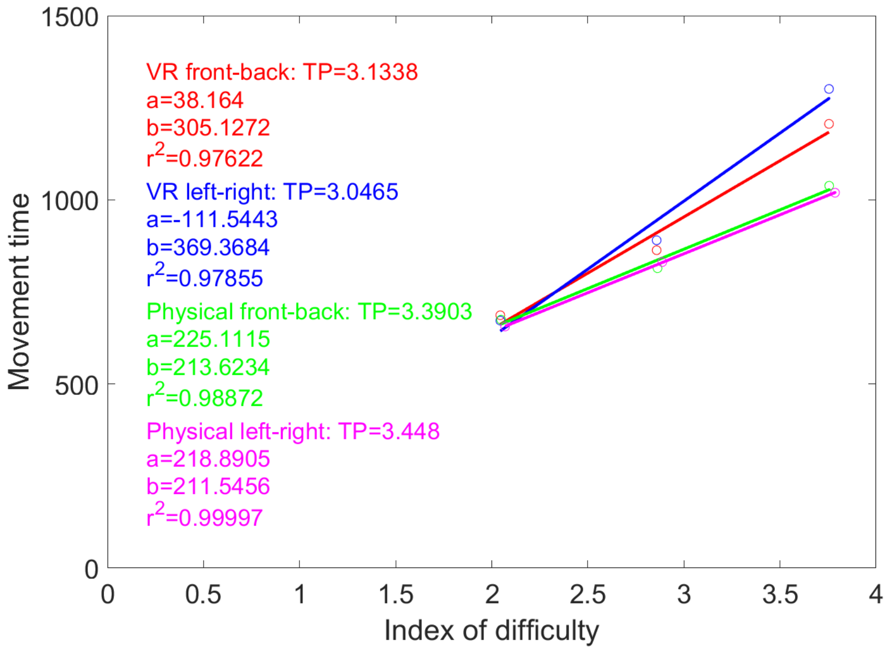


Figure 5.10 Fitts' law models for each combination of interface and direction conditions. The circles represent the average MT corresponding to the three IDs, in each condition.

Task Load Index (TLX)

The results of the NASA TLX questionnaire are shown in Figure 5.11. Overall, the workload in VR was reported as slightly higher (mean=10.5), compared to the workload in the physical environment (mean=9.0). The Wilcoxon signed-rank did not show any significant difference in the mental demand ($p = 0.066$), physical demand($p = 0.92$), temporal demand ($p = 0.089$), performance ($p = 0.12$), effort ($p = 0.37$) and frustration ($p = 0.15$) scores, for the real and VR conditions.

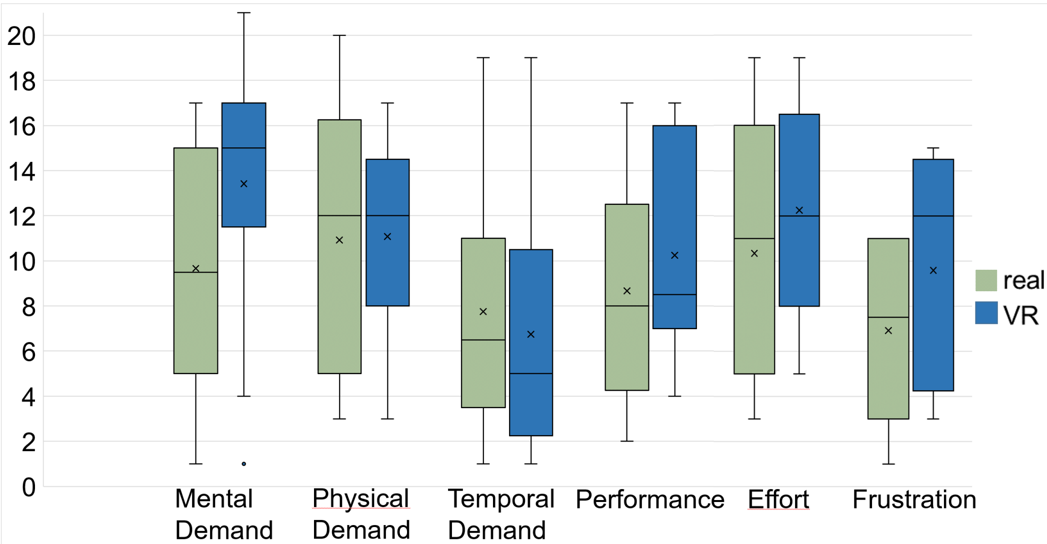


Figure 5.11 Boxplot of the scores on the NASA TLX factors for the Fitt's Law pointing task, in the real and VR condition.

Observations and Interviews

During the experiment, we observed different strategies to complete the pointing task. In tasks with a low index of difficulty, the participants relied more on the visual cues when trying to hit the target. With a high index of difficulty, the exact depth of the target was difficult to see, so some participants relied more on their muscle memory. They tried to memorise and reproduce the gesture that produced a successful target hit. Some participants reported to have used the collision of the levitating particle with the target indicators, in both the VR and the real condition, as a visual cue. On the physical apparatus, some participants also used the increase in oscillations of the particle around the physical targets as additional visual feedback. These oscillations occur as the target indicators slightly disturb the sound field. We did not model this effect in the *Levitation Simulator* yet.

Almost all participants reported in the interviews, that the interaction felt very similar in both conditions. In both conditions, they felt high control over the levitating particle. There was an interesting difference between participants who had experience in VR and those who had not. Those who had previous VR experience, were very impressed by the physically realistic movement of the particle in the *Levitation Simulator*. They had not had such a realistic experience in VR before. In contrast, those who had not experienced VR before, were rather underwhelmed by the overall realism of the *Levitation Simulator* experience. Apparently, they had much higher expectations of the realism of VR experiences in general.

On the real levitation interface, all participants showed amazement over real levitation. They expressed this with phrases such as that when they controlled the particle with their finger they felt as a *wizard with a magic wand*. This might be due to the fact that none of our participants have experienced a levitation interface before. This amazement was not expressed with the *Levitation Simulator*.

The most frequent criticism of the *Levitation Simulator* was the limited quality of the depth cues in VR. When performing the experimental task, participants clearly had difficulty judging the depth of the levitating particle relative to the targets. This was the case particularly for the left-right movements. Participants found this particularly frustrating when they compared it with the real prototype, which provided all natural depth cues perfectly. Some participants also complained about the quality of the resolution of the HMD Vive Pro headset.

In both conditions, the participants reported that they felt they could perform the task better if the targets were visible. Participants made further suggestions to increase the levitation volume, to add visible targets and to improve the depth perception in VR.

5.7 User Engagement Study

The objective of our second user study was to investigate differences and similarities in the way users engage and interact with the same applications, presented in the *Levitation Simulator* and on a physical levitation apparatus. For this purpose, two different games for levitation interfaces, a ball-and-racket game and a first person shooter, were developed in the *Levitation Simulator*, and then implemented on the real prototype. The ideas for the games were generated in a brainstorming session. The exact game design and movement parameters were determined in the *Levitation Simulator* in an iterative testing process with pilot participants. The biggest challenge was to find the optimal bead velocity, such that the game is challenging enough for the player, but not too difficult so they become frustrated and quit. With the *Levitation Simulator* it was possible to efficiently test many velocities and collect user responses.

The first game developed was *BeadBounce*. The goal of the game is to prevent the levitating particle from going into the 'danger zone'. The game is played in the left half of the levitation interface. The right half is the danger zone. The entrance to the danger zone is marked by a black pole, positioned at the back of the levitation volume, to allow for free movement of the racket. During the game, the levitating particle moves in a straight 3D line

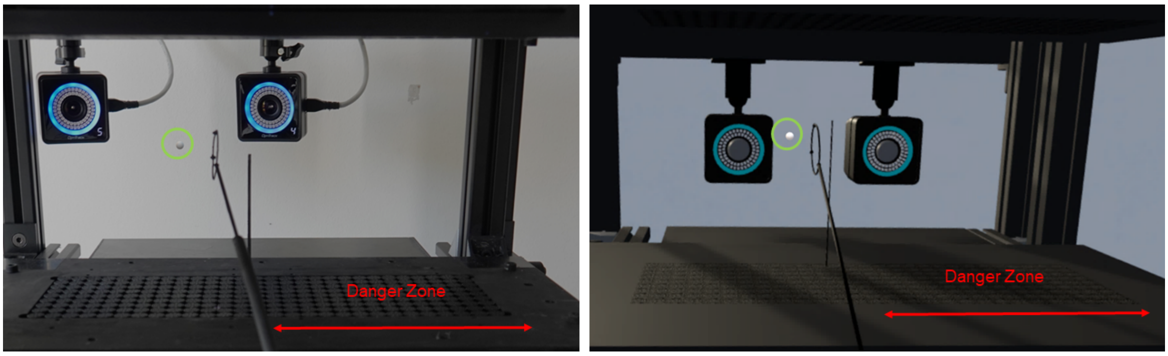


Figure 5.12 In the BeadBounce game, the player has to prevent the levitating particle (marked with a green circle) from bouncing off into the *danger zone* (red arrow), by hitting it with the racket.

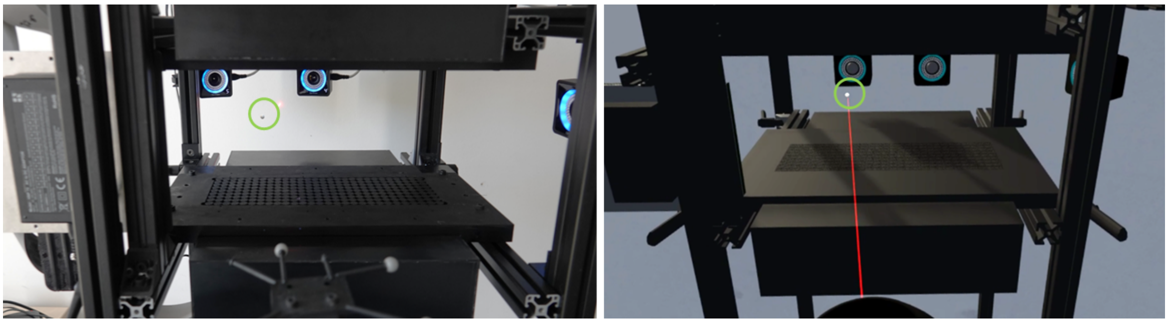


Figure 5.13 In the LeviShooter game, the player aims at the levitating particle (marked with a green circle), using a laser gun. The goal is to successfully hit the particle, as often as possible, while with every hit, the particle velocity increases.

and bounces off the walls of the levitating volume. When the particle starts moving towards the danger zone, the player has to hit it with the racket controller, so that it bounces back to the left (see Figure 5.12). Initially the particle starts moving with 0.09 m/s. When the particle is hit, part of the racket momentum transfers to the levitating particle, hence the particle moves with varying velocity, during the course of the game. Audio feedback is given whenever the levitating particle bounces off the wall or the racket. Each time the particle is hit with the racket, the player gains 1 point, when the particle wonders off to the danger zone, they lose 2 points.

The second game was *LeviShooter*. In this game, the player shoots at a levitating particle with a laser gun controller (see Figure 5.13). The levitating particle moves in a straight 3D line, with an initial velocity of 0.05 m/s. The player needs to aim at it accurately. Visual feedback that the gun controller is aiming correctly, is given by the reflection of the laser beam from the particle. When the particle is successfully hit, the player has to wait 2 s until they can shoot again. When the player hits the particle, they receive 5 points, when they miss it, they lose 1 point. With every hit, the difficulty of the game increases, by increasing the velocity of the particle, by an increment of 0.001 m/s. After ten misses, the levitating particle lowers its velocity again. Audio feedback is given whenever the laser gun is fired, and when the particle is successfully hit.

5.7.1 Experimental Design

The user study consisted of a within-groups experimental design, with two interface conditions (real and VR), for each of the two games. The order of conditions was fully counterbalanced by a Latin square.

To measure user engagement levels, we adopted the long version of the User Engagement Scale (UES) [107]. The scale uses four dimension identifiers: Focused Attention - measures the degree of being absorbed in the experience and losing track of time, Perceived Usability - evaluates interface usability, level of control and how demanding the experience was, Aesthetic Appeal - appeal of the interface to the visual senses, and Reward - assesses whether the experience was fun, rewarding and worthwhile. The UES questionnaire consists of 30 items, where each is answered on a five-point rating scale. We randomised the order of items prior to administering.

We also measured how long participants played the games, and conducted semi-structured interviews with them.

5.7.2 Participants

Twenty four participants (10 females and 14 males) aged between 19 and 32 (mean 24.17, SD 3.95), were recruited for the experiment. All had normal or corrected-to-normal vision and no previous experience with levitating interfaces. Before the experiment started, the participants read basic information about the study and signed a consent form. The study was approved by the Ethical Committee of the University of Bayreuth. All participants received a monetary reimbursement for their participation.

5.7.3 Apparatus

The study was conducted using the *Levitation Simulator* and the real prototype, as previously described. For the BeadBounce game on the physical apparatus, we designed and 3D printed a racket that can interact directly with the levitating particle within the levitating volume, without disturbing the acoustic field and that can be tracked by the motion capture system. The racket head is hollow, with a diameter of 3 cm and frame thickness of 1 mm. The racket handle is 40 cm long with thickness of 6 mm. We attached five 3 mm wide retroreflective markers to the racket handle, to enable optical motion tracking. In the LeviShooter game, the input device consisted of a 3D printed laser-gun-shaped casing. We placed a laser pointer and a trigger inside of the casing, and mounted five 9 mm wide retroreflective markers on top.

5.7.4 Procedure

Upon arrival, participants were given basic information about the study and the rules of the levitation games were explained to them, by the experimenter. They were asked to take a comfortable position, either standing up or sitting down, that allowed them a good overview over the levitation volume. In the VR condition, participants were asked to adjust the HTC Vive Pro headset to their comfort. Then they were given either a racket or a laser gun controller and started playing BeadBounce, or respectively LeviShooter. The participants were instructed to play the games as long as they liked. We recorded the gameplay duration and the score. If the gameplay exceeded 5 minutes, we asked them to move to the next condition. After each condition, the participants took a short break and filled in the UES questionnaire. At the end of the experiment, we conducted a semi-structured interview, where we asked participants about their game experience in the *Levitation Simulator* and on the real prototype.

5.7.5 Analysis

The UES questionnaire data and the gameplay times were statistically analyzed. The normality assumption of the UES data was confirmed by performing a Shapiro-Wilk test (p>0.05). We analyze the data using a paired t-test. Using a Shapiro-Wilk test, we also checked the normality of the distribution of gameplay times data. According to the test, we cannot assume normality for the data obtained in the VR condition of the BeadBounce game ($W = 0.75, p < 0.01$) and in the VR condition of LeviShooter ($W = 0.78, p < 0.01$). Thus we opt for non-parametric statistical analysis and perform a Wilcoxon signed-rank test.

5.7.6 Results

User Engagement

The UES scores obtained for the BeadBounce game, in each subscale, are shown in Figure 5.14. There was a significant difference in the Reward scores for the real (M=4.20, SD=0.56) and VR (M=3.89, SD=0.70) conditions; ($t(23) = 4.05, p < 0.001$). No significant

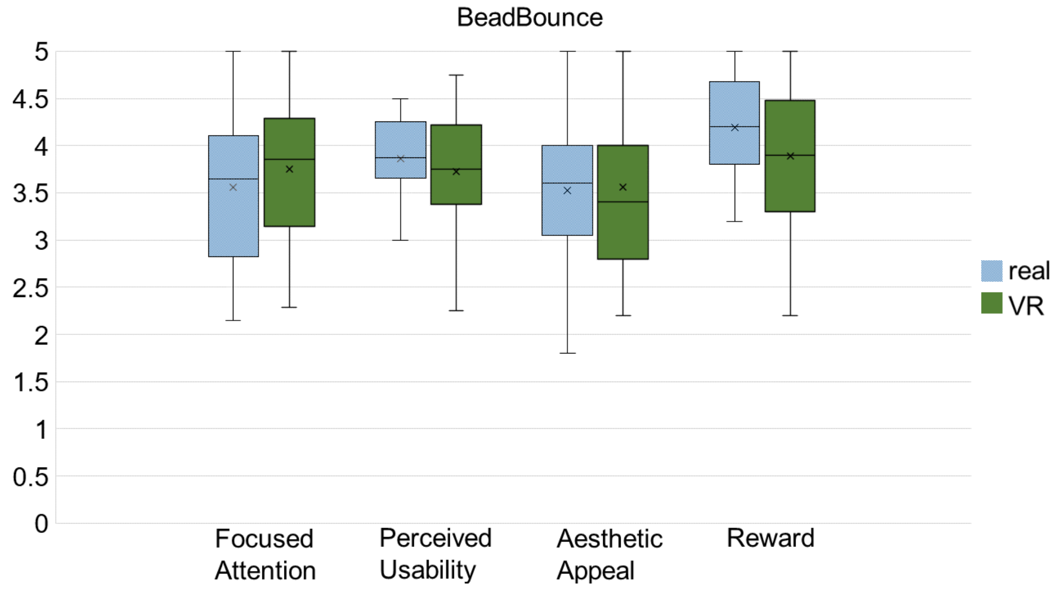


Figure 5.14 Scores for the BeadBounce game on the physical apparatus and in the *Levitation Simulator*, in each subscale of the UES Questionnaire.

difference was found in the Focused Attention ($p = 0.17$), Perceived Usability ($p = 0.32$) and Aesthetic Appeal ($p = 0.80$) scores, for the real and VR conditions.

The scores in each UES subscale, for the LeviShooter game, are presented in Figure 5.15. There was a significant difference in the Aesthetic Appeal scores for the real (M=3.15, SD=0.84) and VR (M=3.61, SD=0.82) conditions; ($t(23) = -3.0731, p = 0.005$). No significant difference was found in the Focused Attention ($p = 0.85$), Perceived Usability ($p = 0.29$) and Reward ($p = 0.40$) scores, for the real and VR conditions.

Gameplay Times

Means and standard deviations for the gameplay times in each condition, are shown in Table 5.1. A Wilcoxon signed-rank test indicates that participants spent significantly more time playing the BeadBounce game in the VR (Mdn= 302.5), compared to the real condition (Mdn= 235); ($p < 0.001$). The scores are presented in Figure 5.16. No significant difference was found for the gameplay times in the LeviShooter game ($p = 0.39$).

Observations and Interviews

When we asked participants which game and in which setting they liked the most, there was no clear preference regarding the game. Most of the participants, however, preferred the interaction with the real prototype. Namely, nine participants preferred BeadBounce

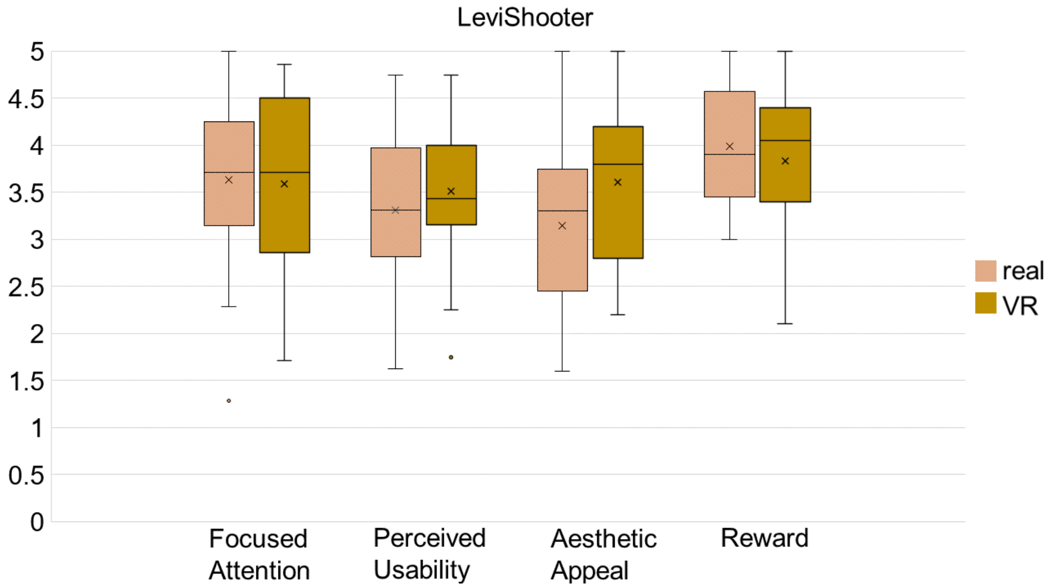


Figure 5.15 Scores for the LeviShooter game on the physical apparatus and in the *Levitation Simulator*, in each subscale of the UES Questionnaire.

Game	BeadBounce		LeviShooter	
Interface Condition	real	VR	real	VR
Mean Playtime	236.92(±47.34)	269.96(±58.29)	255.60(±54.99)	268.16(±54.79)

Table 5.1 Means and standard deviations of the times participants spent playing the two games on the physical apparatus and in the *Levitation Simulator*.

in the real condition, seven participants preferred LeviShooter in the real condition, and both games in the VR condition were preferred by four participants each. Participants used words like *new* and *fun* to describe the game experience with the real prototype, and *cool* and *immersive* for the *Levitation Simulator*. Almost all participants stated that the real and the virtual environment looked very similar, and in both cases they had a strong feeling of control. Few participants noted that even though the experience was very similar, they had a 'different feeling' when playing the game in the real world and in VR - the interaction with the real prototype felt more intuitive to them. Similarly as in the pointing user study, participants most frequently criticised the depth cues in VR and the resolution of the HMD, and reacted with more amazement when observing real levitation.

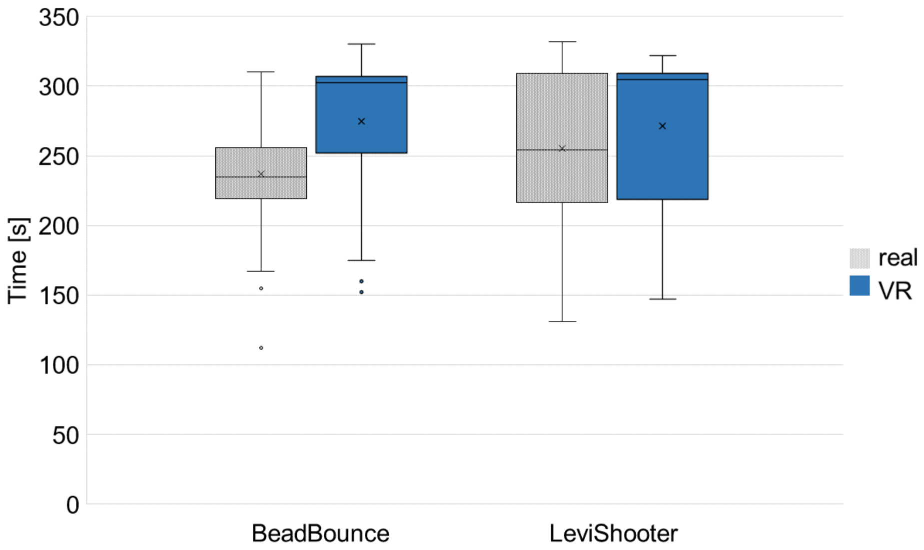


Figure 5.16 Boxplot for the gameplay times per game and interface condition.

5.8 Discussion

Overall, the two user studies yielded similar results for the interaction in the *Levitation Simulator* and with the real prototype. The overall results indicate that if a levitation user study is conducted in the *Levitation Simulator*, and then validated on the real prototype, there is a high probability that similar results will be obtained.

Certainly, differences might become significant with a larger sample size. It is more important, however, to look at the effective sizes. In the pointing study, for example, the average movement times and throughputs are relatively similar. Thus we are quite confident, that any differences between the interfaces will not be very large in magnitude.

In both studies, we experienced two main limitations of the virtual environment. These are important to keep in mind when developing or evaluating levitation applications in the *Levitation Simulator*.

First, the quality of the depth cues with current VR headsets is still far from the depth cues one can perceive in a real user interface. This effect is visible in the Fitts' law analysis and in the interview data of both user studies. It might explain the trend in the pointing data, where smaller targets lead to a higher increase of movement times in the *Levitation Simulator* compared to the real condition (Fig. 5.9). In the User Engagement study, this phenomenon was more prominent in the BeadBounce game, where the levitating particle and the head of the racket need to be at the same depth, for a hit to be registered. This effect might also explain the higher mental demand and frustration in the VR condition, observable in the NASA TLX scores. The difference in depth perception might not be perceivable by users at first. However, in tasks in which very accurate depth perception is necessary, the performance of users in VR might be lower than with a real prototype. In the pointing study, this was the case particularly for left-right movements with very small targets in VR. In these cases, the depth of the levitating particle had to be matched with the depth of the target to a few millimetres. We also observed that in the *Levitation Simulator* the increase in the movement time was steeper in the left-right direction, compared to front-back (Fig. 5.10). One possible explanation is that because of the lacking depth cues in VR, participants relied less on visual, and more on proprioceptive cues to complete the task. After identifying the correct target depth, in subsequent trials, they tried to reproduce the movement that resulted in a target hit the first time. From biomechanics perspective, however, it is easier to reproduce the front-back, then the left-right movement, since there are fewer joints involved (i.e. wrist in front-back, wrist and elbow or shoulder in left-right).

There are three ways to deal with the depth perception problem. First, for tasks that require accurate depth perception, it is important to realise that performance in the *Levitation Simulator* might underestimate the performance that might be achieved with a real prototype. Second, when evaluating virtual prototypes, it might be beneficial to choose tasks which do not rely as much on accurate depth perception. This can be done by using larger targets overall, or by using targets which are larger in the depth dimension. Third, we might hope that future VR headsets will provide improved depth perception.

The second shortcoming of the virtual prototype is that the "wow effect" seems to be gone. The interview results and study observations indicate that the thrilling experience of interacting with a levitation interface, for the first time, cannot be replicated in VR. This effect needs to be considered when estimating the user experience of a real levitation interface from a virtual prototype. The novelty appeal of physical levitation might partially explain the result that participants found the experience of playing the BeadBounce in the real condition more rewarding, compared to the VR condition (Fig. 5.14). This result, however, was not obtained for the LeviShooter game. One possible reason might be a greater sense of involvement, through direct interaction in BeadBounce. Participants described the experience of reaching into the levitation volume with the racket and interacting with the levitating particle as unique and exciting, which possibly made the overall experience more rewarding and worthwhile. Another reason might be the physical limitation of not being able to see the laser ray in the real condition of LeviShooter, but only its reflection from the particle and the background.

Even though, in the interviews, the majority of the participants reported that they preferred playing the levitation games in the real condition, on average, they spent more time playing in VR (mean playtime = 270 s for BeadBounce, 268 s for LeviShooter). The immersion property of VR, as well as the blocking of distractions, might be a possible explanation for this result. Thus it should be taken into consideration that the *Levitation Simulator* might overestimate the gameplay times, that would be obtained with the real prototype.

However, overall, the participants considered the interaction with the *Levitation Simulator* as highly realistic when comparing it to the real prototype. We believe that, when accounted for differences in depth perception, immersion and the “wow effect”, a virtual prototype can provide a very good prediction of the interaction performance and experience that would be achieved with a real prototype.

5.9 Applications

Virtual prototyping allows us to investigate applications *before* we build a physical prototype. This is particularly useful in cases where it is difficult, or expensive, to build a physical prototype. For some applications, building a functioning prototype might require years of research with a full research team. Prototyping in VR allows us to explore whether it is worth investing in a physical prototype, before we make the investment.

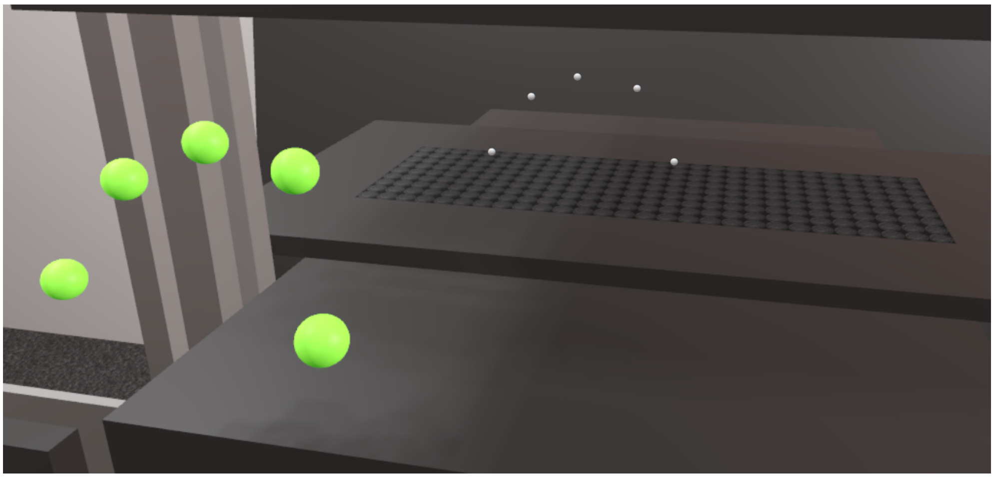


Figure 5.17 In the *Levitation Simulator*, the user is able to explore applications where multiple beads can be simultaneously controlled, independent of each other.

A good example of this is levitation of multiple, independently controllable particles. To our knowledge, fully independent movement of multiple levitating particles has not been achieved yet. We can already now investigate what it would be possible to create, if we were able to achieve this. This is trivial and straightforward to do in the *Levitation Simulator*. The simulation module can easily be extended to simulate an arbitrary number of levitating particles.

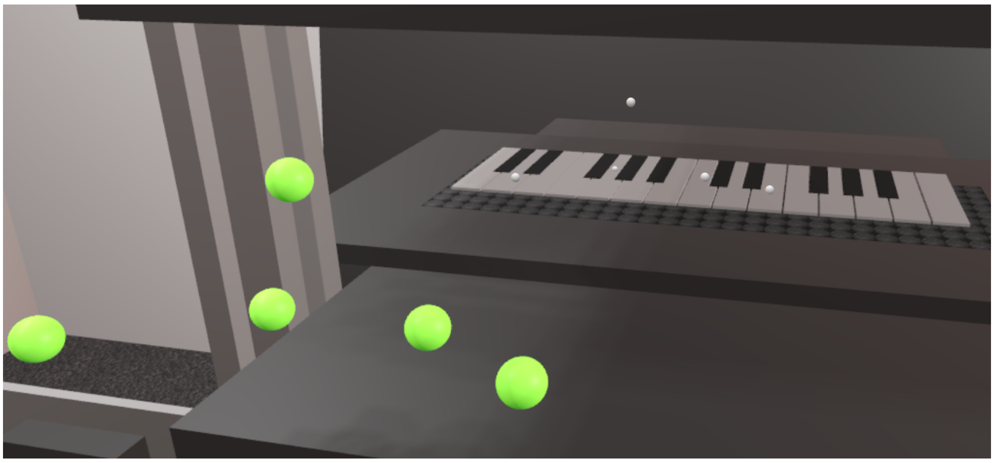


Figure 5.18 In the *Levitation Simulator*, the user is able to play the piano using levitating particles to play the piano keys.

To showcase this possibility, we implemented a “levitating hand” application. Each fingertip of the user is represented by a levitating particle. The prototype is shown in Figure 5.17. The levitating hand does have a number of interesting properties. For example, it is not possible to actually touch a levitating particle with a real hand. This is because the hand disturbs the sound field, causing the levitating particle to drop. However, with a levitating hand, we can virtually touch other levitating objects. We can employ this prototype to make a version of the “Levitating Piano” in Figure 5.5 that can be played with all five fingers, shown in Figure 5.18.

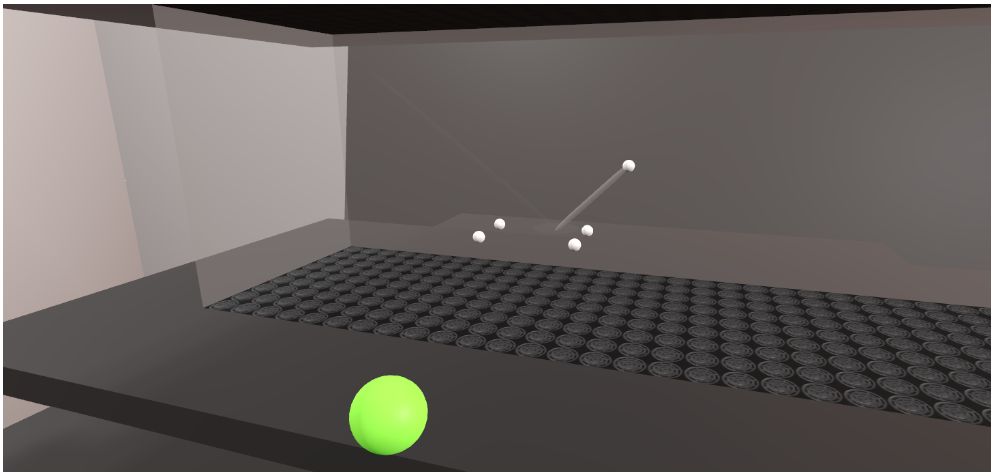


Figure 5.19 Concept of a Pong game in the *Levitation Simulator*, where the user controls a plane made up of levitating particles with which she needs to prevent the shooting particle from hitting the bottom.

Beyond these implemented applications, we developed several concepts, which can be implemented in the *Levitation Simulator*. The first concept is a “levitating Pong game” (Figure 5.19). Users control a paddle made out of four levitating particles by moving their fingertip. A fifth particle serves as the ball and bounces from the edges of the play area and the paddle.

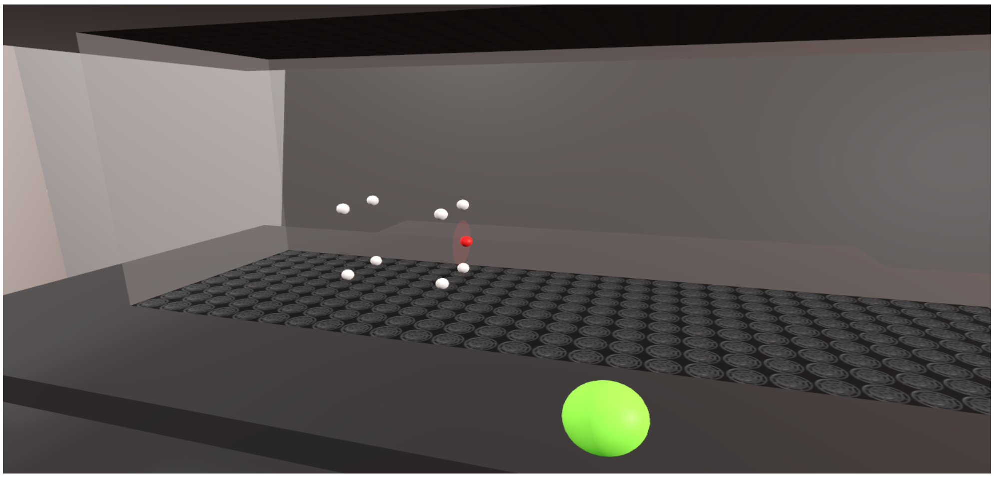


Figure 5.20 Concept of an object manipulation interface. In the *Levitation Simulator*, the user is manipulating a 3D cuboid, which has levitating particles as vertices (white spheres), using a levitating cursor (red sphere).

Another concept is a system in which users can manipulate levitating 3D “objects” with a levitating cursor (Figure 5.20). The levitating “objects” are made of levitating particles which represent the vertices of that object. The user can move the object around by “touching” it with a levitating cursor.

Applications, such as those presented above, are easy and quick to implement using the *Levitation Simulator*. We plan to distribute the *Levitation Simulator* to artists, who are interested in developing applications and artworks involving ultrasonic levitation. Artists could work from their ateliers, without needing access to the physical prototypes. For the promising pieces, it would be straightforward to port them to the physical levitation apparatus, because the *Levitation Simulator* and the physical prototype share very similar APIs.

5.10 Conclusion

In this chapter, we have presented the *Levitation Simulator*. We derived a model of the movement of a levitated particle in a sound field from first principles. This makes the virtual levitating particle behave seemingly identical to a real particle. Our user studies show that the *Levitation Simulator* provides performance as well as levels of user engagement and user experience comparable to the real levitation apparatus. Further, our qualitative results show that the user experience is similar in the *Levitation Simulator*, compared to the real levitation apparatus. We hope that virtual prototyping in the *Levitation Simulator* can speed up progress and improve replicability of ultrasonic levitation research, as well as inspire more content creation for this novel type of interfaces. Here we demonstrated how virtual prototyping can be helpful in the design of user interfaces. With this, we hope to inspire future research in modelling, simulation, and virtual prototyping of UIs in HCI.

Chapter 6

Presenting Information in Braille using Acoustic Mid-Air Haptics

6.1 Introduction

There are several challenges that blind people face when engaging with interactive systems in public spaces. Firstly, it is more difficult for the blind to maintain their personal privacy when engaging with public displays. Audio feedback is easily overheard by bystanders and can be perceived as obtrusive, since it contributes to the environmental noisescape. Some interfaces, such as ATMs, feature a headphone plug. In this case, however, users need to remember to bring headphones and once they start the interaction, they might have more difficulty monitoring events in their surroundings. Refreshable Braille displays, consisting of lines of actuated pins, also have some shortcomings. The information they can convey is limited to patterns of dots, which is suitable for text, but not sufficient for content involving shapes and objects (e.g. data charts). It can be difficult to detect them from a distance, since the user has to already touch them to know they are there. The physical contact with these interfaces could potentially cause hygiene problems in public spaces, e.g. hospitals. They contain moving parts, which can become clogged by dirt in public spaces.

As a potential solution for these challenges, we present HaptiRead - a concept of the first public display that presents Braille information as touchless stimulation, using mid-air haptic technology [21]. The feedback generated by HaptiRead is delivered directly to the user, without disturbing the environment. The system can detect the user's hand using a built-in Leap Motion sensor, and render the Braille text where the hand is, improving detectability. Its contactless nature means that it could prevent hygiene-related issues. Because it contains no moving parts, it is potentially more robust for public spaces. The combination of easily-

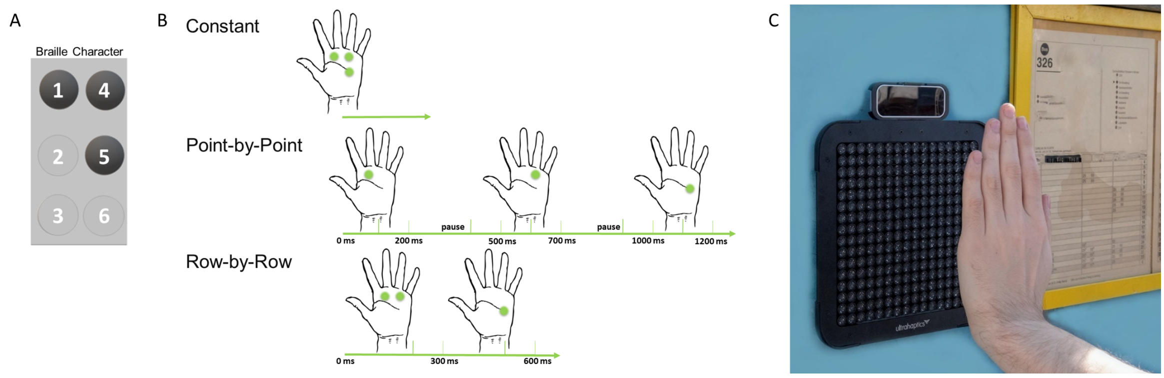


Figure 6.1 With HaptiRead we evaluate for the first time the possibility of presenting Braille information as touchless haptic stimulation using ultrasonic mid-air haptic technology. We present three different methods of generating the haptic stimulation: Constant, Point-by-Point and Row-by-Row. (A) depicts the standard ordering of cells in a Braille character, and (B) shows how the character in (A) is displayed by the three proposed methods. HaptiRead delivers the information directly to the user, through their palm, in an unobtrusive manner. Thus the haptic display is particularly suitable for messages communicated in public, e.g. reading the departure time of the next bus at the bus stop (C).

detectable, yet unobtrusive interface could potentially encourage more blind people to use the accessibility feature, thus granting them more autonomy and independence in their daily life. Lastly, the volumetric interaction space of the interface, allows for content versatility, beyond Braille text.

To our knowledge, there are no formal studies exploring the potential of using mid-air haptic technology to convey Braille information. Questions, such as: Do mid-air haptic displays provide enough spatial resolution to discriminate between different Braille characters? How do blind people experience the haptic stimulation? Can they transfer their Braille knowledge on paper to correctly interpret the haptic information in mid-air? etc., remained unanswered thus far. In this chapter, we evaluate different methods for presenting the haptic stimuli in mid-air, in an iterative design process. Then we test with users the three most promising methods: Constant (emission of all haptic points at the same time), Point-by-Point, and Row-by-Row. We first conduct a preliminary study with sighted participants, investigating whether HaptiRead can provide enough haptic cues to differentiate between different dot patterns. Then we evaluate the performance and user experience in a user study with blind participants, proficient in Braille.

Our main contributions are:

1. We present the first user study that investigates the use of a mid-air haptic interface with blind participants.

2. Through user studies, we demonstrate that it is possible to effectively distinguish between different Braille patterns, where each dot of the pattern is represented by a mid-air haptic stimulation point.

3. We present and compare three different haptic stimulation methods for generating Braille characters in mid-air.

6.2 Related Work

6.2.1 Mid-air Haptics

Ultrasonic mid-air haptics [59, 21] is a technology that allows for haptic feedback to be projected directly onto users' unadorned hands. Focused ultrasonic waves are emitted by a phased array of transducers at a frequency of 40 kHz. By modulating the waves with a frequency detectable by the receptors in the human skin, it is possible to create a perceivable haptic sensation in mid-air [21].

Early prototypes were able to generate a single haptic point using linear focusing [54]. Wilson et al. [150] investigated the perception of an ultrasonic haptic point experimentally. The results show that users were able to localise a static haptic point on their hand, with an average error of 8.5 mm. Both localisation and sensitivity were better near the centre of the hand. Alexander et al. [2] introduced multi-point haptic feedback, using spatial multiplexing, where different portions of the array contribute towards different focus points, and temporal multiplexing, where the system rapidly switches between different points. Later prototypes used optimisation algorithms to generate multiple haptic points [41, 21]. User studies involving multi-point haptic feedback, carried out by Carter et al. [21], show that differentiability between two haptic points improves, when they are modulated with different frequencies, and the accuracy of determining the correct number of points increases with the distance - for distances of 3 cm and above the accuracy was over 85%. An algorithm for creating volumetric shapes using the mid-air haptic technology was presented by Long et al. [75]. Haptogram [68] is an alternative method to generate 2D and 3D tactile shapes in mid-air, using a point-cloud representation. In addition to points and shapes, a rendering technique for the creation of haptic textured surfaces has been demonstrated in [35].

By tuning parameters, such as the location of the mid-air haptic stimulus, number of haptic points, modulating the frequency, among other factors, it is possible to generate haptic patterns that are suitable for different applications. For example, Vi et al. [145] used mid-air haptic technology to enhance the experience of visual art in a museum, Martinez et al. [79] generated haptic sensations that mimic supernatural experiences in VR and in [50] buttons and sliders were augmented with mid-air haptic feedback in a driving simulator, to reduce off-road glance time. Gil et al. [43] explored the perception of mid-air haptic cues on the face, across different parameters and in a practical notification task.

6.2.2 Braille Interfaces

Today Braille is mostly read from a nonrefreshable embossed medium (e.g. paper [15]). This is the more widely-spread and affordable option. Refreshable Braille displays, made of actuated plastic or metal pins, embody a more flexible, but also a more pricey alternative, ranging up to 10000$[1].

In the past, several methods have been developed for reading Braille on mainstream devices, such as mobile phones and tablets. Rantala et al. [119] presented three interaction methods: scan, sweep and rhythm for reading Braille on mobile devices with a touchscreen. Al-Quidah et al. [1] optimized the temporal rhythm method further, by developing an encoding scheme for each possible column combination in a Braille character, similar to the Morse code. The encoding scheme lowers the time it takes to represent a character. The users are required, however, to learn a new mapping. The accuracy ranged from 61 to 73%. *HoliBraille* [96] is a system consisting of six vibrotactile motors and dampening elements that can be attached to mobile devices in order to enable interaction in Braille, in the form of multipoint localized feedback. Another method for presenting Braille characters on a mobile phone is *VBraille* [62]. The touchscreen of the phone is divided into six cells in the usual Braille order (Figure 6.1(A)). When a cell representing a raised dot is touched, the phone vibrates.

UbiBraille [95] is a wearable device consisting of six aluminum rings that transmit vibrotactile feedback. The device is able to simultaneously actuate the index, middle and ring finger of both hands of the user, each corresponding to one Braille cell. The order of the cells is analogous to writing on a Braille typewriter. Luzhnica et al. [76] investigated encoding text using a wearable haptic display, in a hand, forearm and two-arms configuration. Tactile information transfer on the ear was explored with *ActivEarring* [74], a device able to stimulate six different locations on the ear using vibration motors.

6.2.3 Summary

In the past, as an alternative to Braille displays consisting of individually actuated pins, a variety of methods and devices relaying on vibrotactile feedback have been researched. The area of touchless mid-air haptics for Braille applications has been unexplored up to now. In

[1] https://canasstech.com/collections/blindness-products/braille-displays

this chapter we propose HaptiRead, an interface for blind users with the potential to provide improved privacy, detectability, hygiene and variability of displayable content. Our work presented here contributes the first validation of the use of mid-air haptic technology for the purpose of conveying Braille characters. We evaluate three methods of haptic stimulus generation: Constant, Point-by-Point and Row-by-Row, by measuring performance indicators, as well as collecting system usability feedback from members of the blind community.

6.3 The System

For providing the haptic feedback in mid-air we use the Stratos Explore development kit from Ultraleap[2]. The hardware is equipped with 256 transducers, that emit ultrasonic waves to create up to eight perceivable points at a maximum range of approx. 70 cm, as well as a Leap Motion hand tracking module. The board's update rate for the ultrasound is 40 kHz, which implies that the diameter of the generated points is 8.6 mm (the wavelength of sound at 40 kHz). Such high frequencies are above the threshold of human tactile perception in the hand [24]. Thus the ultrasonic waves are modulated, using frequencies between 100 and 200 Hz (recommended by the manufacturer). Additionally, the intensity for each point can be configured with values from 0 to 1. Per default, we chose the highest intensity for each point since lower intensities increased the difficulty of detecting different points. For better differentiability, we modulate each haptic focus point, representing a different cell in a Braille character, with a different frequency. We chose a modulation frequency of 200 Hz for cell 1, 140 Hz for cell 2, 120 Hz for cell 3,160 Hz for cell 4,180 Hz for cell 5 and 100 Hz for cell 6 (see Figure 6.1(A) for ordering convention). For consistency, the chosen modulation frequency for each cell was fixed throughout all characters. For our application we chose a distance of 3 cm between the centres of the points, since in our pilot tests, it showed the best trade off between the overall size of the pattern and the ability to detect single points.

6.4 Iterative Design Process

6.4.1 Interview with a Braille Teacher

To gain expert feedback on the HaptiRead concept, we conducted an exploratory interview with a local Braille teacher, with 20 years of teaching experience. In her daily life, the teacher uses a mixture of various voice systems, as well as refreshable Braille lines and books. She relies on Braille for completing tasks that require precision, like reading a

[2]www.ultraleap.com

phone number or correcting a text. She responded favourably to the HaptiRead concept and appreciated the compactness and mobility of the device. The teacher could envision using it in public spaces for reading timetables, menus in a restaurant or doctor's prescriptions. In all of these cases, text-to-speech devices are not suitable, because they can be overheard by bystanders and other existing solutions require the user to have a minimum amount of visual capability. The teacher suggested that the HaptiRead device might be especially useful for young, congenitally blind people, for training the recognition of dots and their location, in the Braille learning process.

6.4.2 Pilot Study

We carried out a brainstorming session where different methods, specially designed for presenting Braille characters on a mid-air haptic device, were generated. The refined list of potential methods to display Braille via touchless haptics is presented in Table 6.1. These methods were evaluated in a pilot study with six sighted participants (1 female, 5 male) with no previous experience in Braille or with mid-air haptic systems. Most of them reported they felt more comfortable using the methods where part of the pattern or the individual points are sequentially presented. For these methods, they reported higher levels of confidence in their ability to correctly identify the patterns. The preferred methods were Row-by-Row and Point-by-Point. In iterative tests with other pilot participants, we determined the best timespans for displaying the feedback for these methods. In the Point-by-Point method, the best results were achieved when individual dots were displayed for 200 ms, with a 300 ms pause between subsequent dots and a 500 ms pause at the end of a character. Performance with the Row-by-Row method was the best, when the rows were displayed in 300 ms intervals. An illustration of the pattern presentation timelines per method, is given in Figure 6.1(B).

6.4.3 Interview with a Proficient Braille Reader

When presented with the haptic stimulation methods in Table 6.1, the interviewee reported that the method rendering all the haptic points simultaneously (Constant), was the most in line with her expectations of reading Braille. In addition, the process of transferring her previous Braille knowledge onto the novel system was the most fluent with this method. She also stated, however, that her fluency improved rapidly (with the other methods as well) after a few training sessions. In her opinion, the interface could particularly be useful for Braille beginners, for whom the refreshable Braille lines are too fast. With HaptiRead they can take the time to explore the individual dots and patterns. In a later consultation, the Braille teacher also stated the Constant method as her preferred one.

Method	Description
Constant	all dots are simultaneously displayed
Pulsating	the dots are flashing in sync
Rotating	each dot is rotating clockwise
Expanding	the dots move away from each other
Varying Intensity	dot intensity is fluctuating over time
Row-by-Row	rows are subsequently displayed
Column-by-Column	columns are subsequently displayed
Point-by-Point	only one dot is displayed at a time
Morse-Like	dots are presented in a time sequence, at the same position

Table 6.1 Haptic stimulation methods evaluated during the design phase.

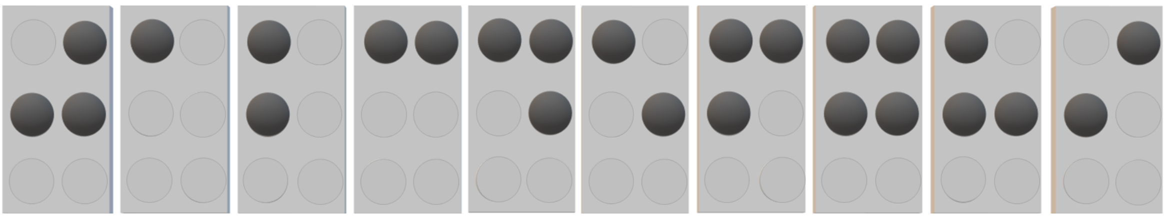


Figure 6.2 Visual representation of the dot patterns tested.

6.5 Pre-Study

We conduct a pre-study with eighteen sighted participants (11 females and 7 males; 2 left and 16 right-handed), aged between 20 and 40 years (mean 25.29, SD 5.12), to test whether the HaptiRead system provides enough haptic cues for dot pattern recognition. The participants reported no previous experience with mid-air haptics or knowledge in Braille. The experimental task was chosen after careful consideration and consultation with Braille experts. As this was the first time the participants came in contact with mid-air haptic technology, in order to avoid overwhelming the user with the study protocol, we opted for a simple experimental task that ensures high internal validity and experimental control. The task consisted of correctly identifying a pattern of dots being presented in the form of mid-air haptic stimulation. The possible patterns were limited to 4-cell Braille characters (see Figure 6.2). Using the methods identified as most promising, in the pilot study - Row-by-Row and Point-by-Point, and in the interview - Constant, the participants were presented with ten dot patterns per method (30 trials in total).

To avoid potential learning and ordering effects, a fully counterbalanced design was used, i.e. all possible combinations of experimental orders were presented the same number of times. Each participant was seated in front of a computer and asked to place their left hand

20 cm above the ultrasonic array and focus on perceiving the pattern presented on their palm. They were provided with earmuffs to prevent auditory influences. Participants had to indicate which pattern they were perceiving on their left hand by selecting the visual equivalent, i.e. visual representation of the pattern, on a screen (see Figure 6.3). Before completing the actual trials of the study, participants underwent a training session that included four trials for each haptic stimulation method. No time limit was given on the response time for each trial, however the time taken to answer was recorded for each trial. In the training trials, performance feedback was given. The actual trials of the study did not include feedback, so participants were not aware of their performance. After completing the ten trials of each type of Haptic Stimulation, participants reported their subjective opinion of how comfortable they felt using the system. Each question was answered on a 7 point Likert scale, with 1 denoting *not at all* and 7 *extremely comfortable*. At the end of the study, the participants were given a chance to express their experiences, comments or suggestions, concerning the mid-air haptic interface, in a semi-structured interview. The experiment lasted approximately 30 min in total per participant. The pre-study was approved by the Ethical Committee of University of Bayreuth and all participants received monetary compensation for their participation.

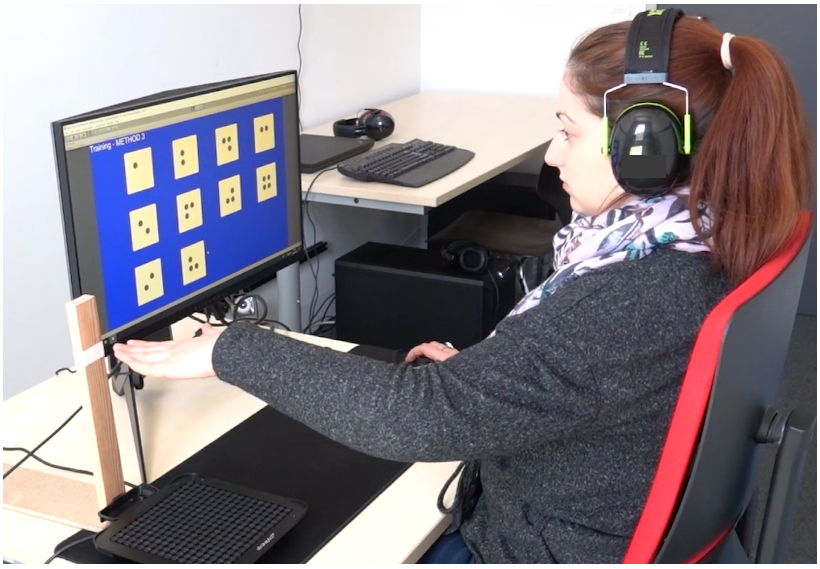


Figure 6.3 The users of the pre-study were instructed to place their left-hand approximately 20 cm above the ultrasonic array. Their task was to correctly identify the haptic pattern being presented to them and select the corresponding visual equivalent on the screen, where ten possible pattern choices were provided.

6.5.1 Results

The average pattern recognition accuracy rate over all methods was 86%. The highest average accuracy score of 94% (SD 7) was achieved with the Point-by-Point method, whereas for both the Constant and the Row-by-Row method the average score was 82% (SD 19.21). The average time it took the participants to recognise a pattern was 10.89 s (SD 4.75) for the Constant, 8.55 s (SD 2.36) for the Point-by-Point, and 10.55 s (SD 4.34) for the Row-by-Row method. Note that the participants were instructed to focus on correctly identifying the dot patterns, rather than providing fast answers.

The confusion matrix in Figure 6.4 provides information about the most frequently mistaken patterns. In general, the participants identified the correct pattern most of the time, yet, it appears, identifying the second half of the digits, i.e. from to , was more challenging. The most frequent mistake, occurring nine times in the study, was in identifying as . Figure 6.5 gives a detail account of the distribution of the pattern mistakes per method.

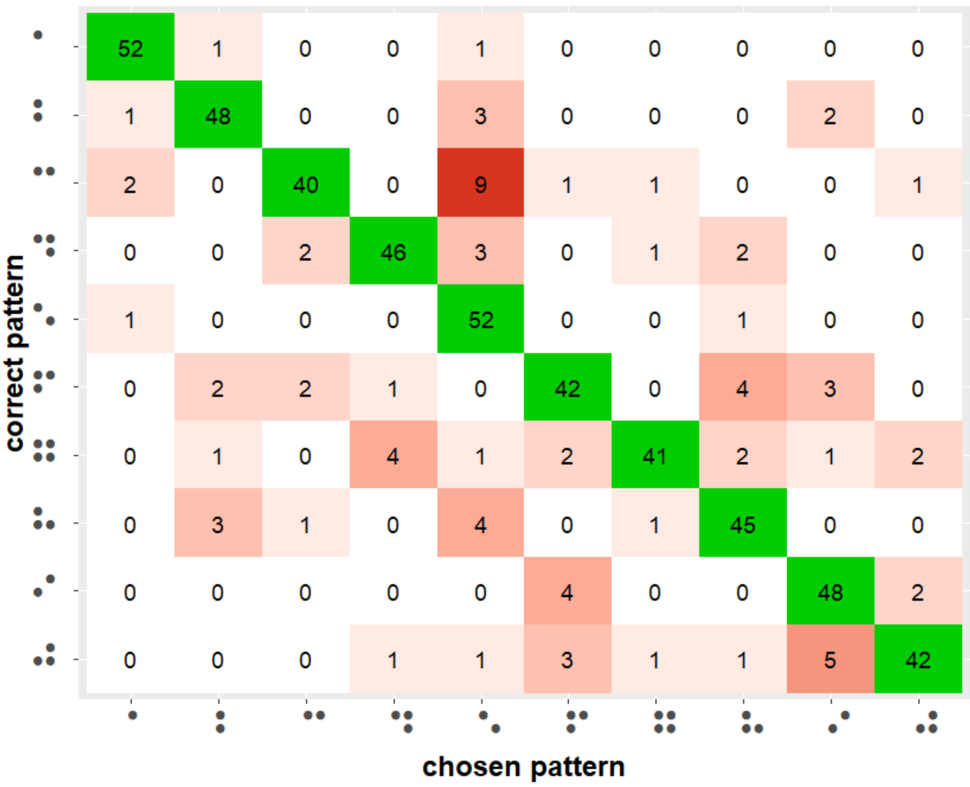
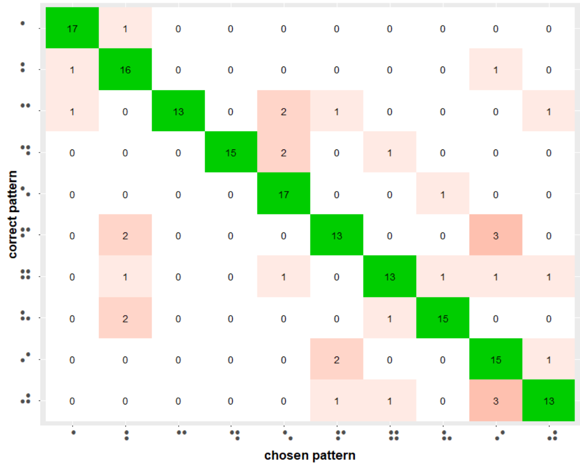
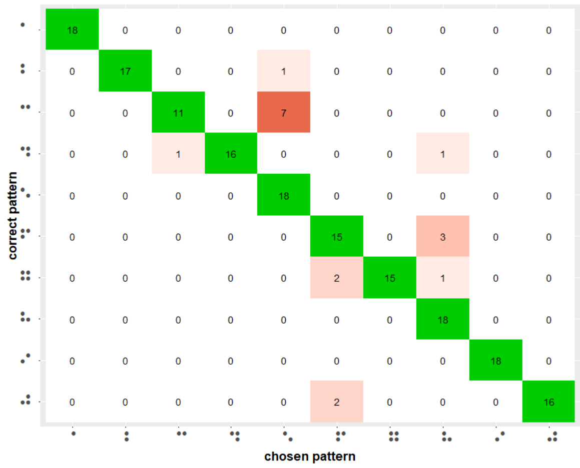


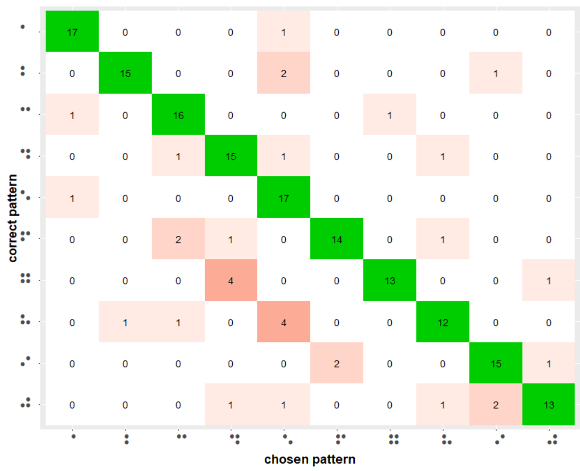
Figure 6.4 Confusion matrix displaying the total number of trials each number was correctly identified (in green) in the pre-study. The red cells show the number of trails a number was incorrectly identified as some other.



(a) Constant



(b) Point-by-Point



(c) Row-by-Row

Figure 6.5 Confusion Matrix per haptic stimulation method in the pre-study.

The high accuracy rates indicate that it is possible to communicate different dot patterns as touchless haptic stimulation, using all three methods. Using the Friedman test, no significant difference was found between the haptic stimulation methods ($\chi^2 = 5.15$, $df = 2$, $p = 0.059$) and in the Mean Time to Respond ($\chi^2 = 3.11$, $df = 2$, $p = 0.21$).

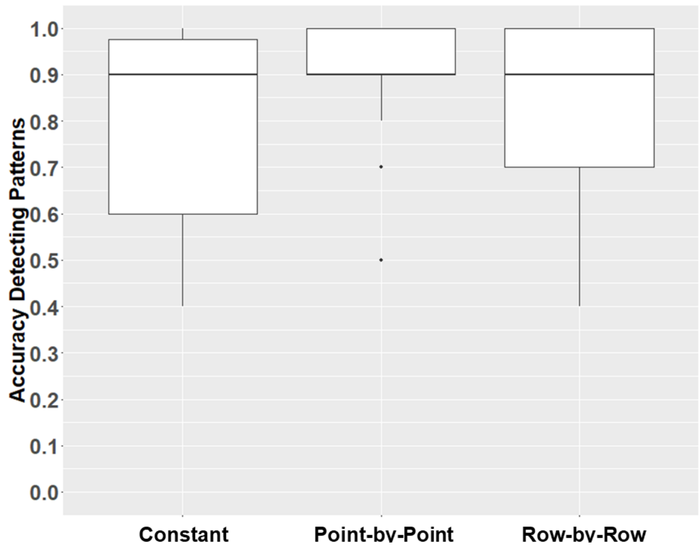


Figure 6.6 Boxplot of the Accuracy for the three haptic stimulation methods (Constant, Point-by-Point and Row-by-Row) in the pre-study.

6.6 User Study

Since in the pre-study all three haptic stimulation methods showed potential to be used for dot pattern presentation, we test all of them in a user study with blind participants.

6.6.1 Experimental Design

The user study consisted of a within-groups experimental design. The participants experienced three possible types of haptic stimulation: 1) Point-by-Point, 2) Constant, and 3) Row-by-Row. The different methods were presented in a randomised order.

6.6.2 Participants

Eleven blind participants (5 females and 6 males) aged between 19 and 70 (mean 42, SD 13.45) were recruited for the experiment. Their demographic data is given in Table 6.2. Before the experiment started, the participants were read basic information about the study and they signed a consent form. The study was approved by the Ethical Committee of University of Bayreuth and followed ethical standards as per the Helsinki Declaration. All participants received a monetary reimbursement for their participation.

6.6.3 Measures and Procedure

To better accommodate participants' needs, the study was conducted in the familiar environment of their homes. All potential distractions (e.g. phones) were removed from the vicinity. The participant was comfortably seated and the HaptiRead interface was placed on a table in front of them. An information sheet, explaining the device and the experimental task, as well as a consent form was read to the participant. The participant was encouraged to raise any questions regarding the study and the technology. After the consent form was signed, a demographic questionnaire was verbally administered. Then the participant was asked to complete a short task to verify their proficiency in reading Braille. The task consisted of five 5-digit numbers in Braille, that they had to read out loud. Next, the participant was instructed to place their dominant hand 20 cm above the ultrasonic array and focus on perceiving the haptic sensation on the palm of their hand. The participant was asked to wear headphones during the experiment, to control for any potential auditory influence on their responses. Similarly as in the pre-study, before completing the actual trials of the study, the participant underwent a training session that included four trials for each haptic stimulation method. The experimental task consisted of a random presentation of trials (10 trials per method, 30 in total), where the participant had to identify the Braille digit presented via mid-air haptics. No time limit to respond was given, however the time taken to answer was recorded for each trial. The participant was permitted to actively explore the haptic sensation, but instructed to approximately keep the recommended vertical distance to the array. When the participant recognised the Braille pattern, they stated the corresponding character out loud. At this moment the timer was halted, but the feedback continued. After completing the experiment, the participant was asked to indicate their subjective opinion of how mentally demanding

ID	1	2	3	4	5	6	7	8	9	10	11
Gender	m	f	f	m	f	m	f	m	m	f	m
Handedness	r	r	r	l	r	l	r	r	r	l	r
Age	19	45	36	46	52	48	45	29	31	41	70
BE in Years	13	34	30	28	2	41	40	24	22	35	56

Table 6.2 Demographic data of the participants. BE = Braille Experience

the task was, as well as how comfortable they felt using each of the haptic stimulation methods. The questions were answered on a 7 point Likert scale (1 meaning *not mentally demanding at all/not comfortable at all*, 7 meaning *extremely mentally demanding/extremely comfortable*). Next, the System Usability Scale [17] questionnaire was verbally administered. The participant was asked to answer the questionnaire considering the HaptiRead system with their preferred haptic stimulation method. Finally, a semi-structured interview was conducted. The user study lasted approximately one hour per participant.

6.6.4 Results

Accuracy and Time to Respond

The average accuracy was 81% (SD 17) for the Constant, 88% (SD 14) for the Point-by-Point, and 75% (SD 23) for the Row-by-Row method. Figure 6.7 shows that the Point-by-Point method achieved the highest mean accuracy score, followed by the Constant, and the Row-by-Row method. Using the Friedman test, no significant difference in the Accuracy between the haptic stimulation methods was found ($\chi^2 = 4.92$, $df = 2$, $p = 0.08$). The mean time to identify a character totals 7.19 s (SD 4.02) for the Constant, 7.30 s (SD 2.44) for the Point-by-Point and 7.31 s (SD 3.45) for the Row-by-Row method. The Friedman test indicated no significant differences between the three ($\chi^2 = 1.64$, $df = 2$, $p = 0.44$).

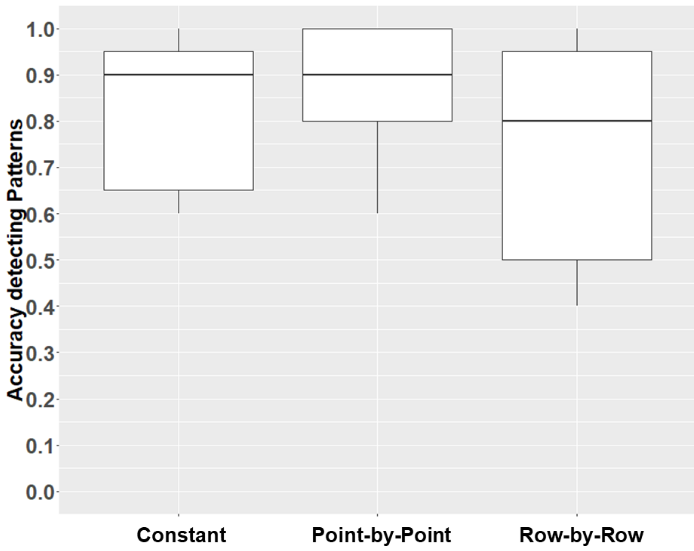


Figure 6.7 Boxplots of the Accuracy for the three haptic stimulation methods (Constant, Point-by-Point and Row-by-Row).

Mental Demand and Perceived Comfort

On average, the participants reported slightly lower Mental Demand when using the Point-by-Point method (median = 3) to read the Braille characters, compared to the Constant and Row-by-Row methods (median = 4 for both). Lower levels of Comfort were reported for the Row-by-Row method (median = 4), compared to the Constant and Point-by-Point method (median = 5 for both). The scores are presented in Figure 6.8. However, using the Friedman test, no significant difference for Mental Demand ($\chi^2 = 2.34$, $df = 2$, $p = 0.30$) or Perceived Comfort ($\chi^2 = 1.90$, $df = 2$, $p = 0.39$) was found.

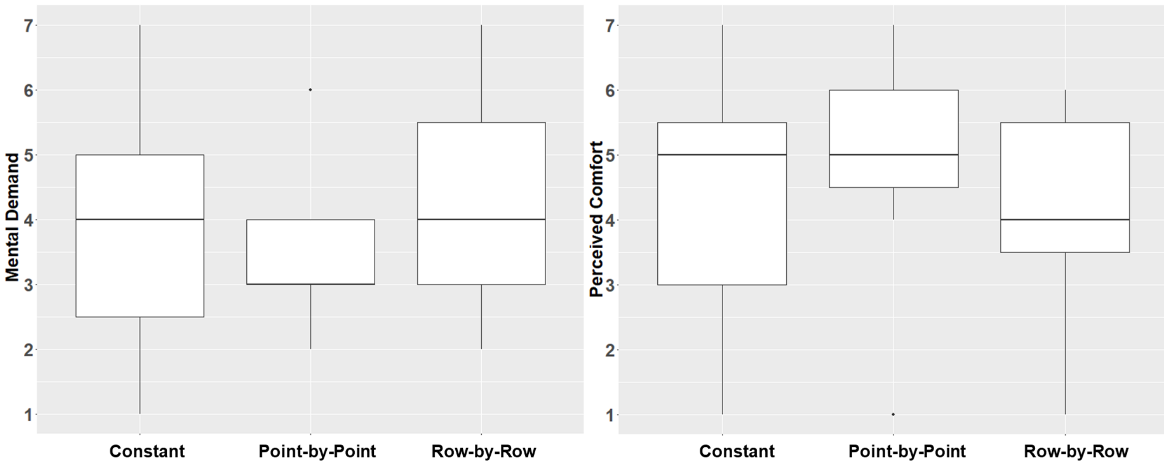


Figure 6.8 Boxplot of Mental Demand and Perceived Comfort for the three haptic stimulation methods (Constant, Point-by-Point and Row-by-Row).

Confusion Matrix Analysis

The confusion matrix, providing information about the most frequently mistaken patterns, is shown in Figure 6.9. The pattern ⠛ was identified correctly the least amount of times (19 out of 33), whereas the pattern ⠁ consisting of only one haptic point was identified correctly almost always (32 out of 33 trials). The pattern ⠛ was most often mistaken for ⠋ and ⠓. The majority of the errors (61%), occurred due to misperception of a single haptic stimulation point. In 30 trials, the error was due to a false negative (e.g. ⠓ identified as ⠑), and in 8 trials, due to a false positive (e.g. ⠊ identified as ⠋). In 31% of the errors, both a false positive and false negative occurred (e.g. ⠝ identified as ⠚). The remaining 8% of the errors, were due to the omission of two or more points, i.e identifying ⠛ as ⠁ and ⠉. The confusion matrices per method are provided in Figure 6.10.

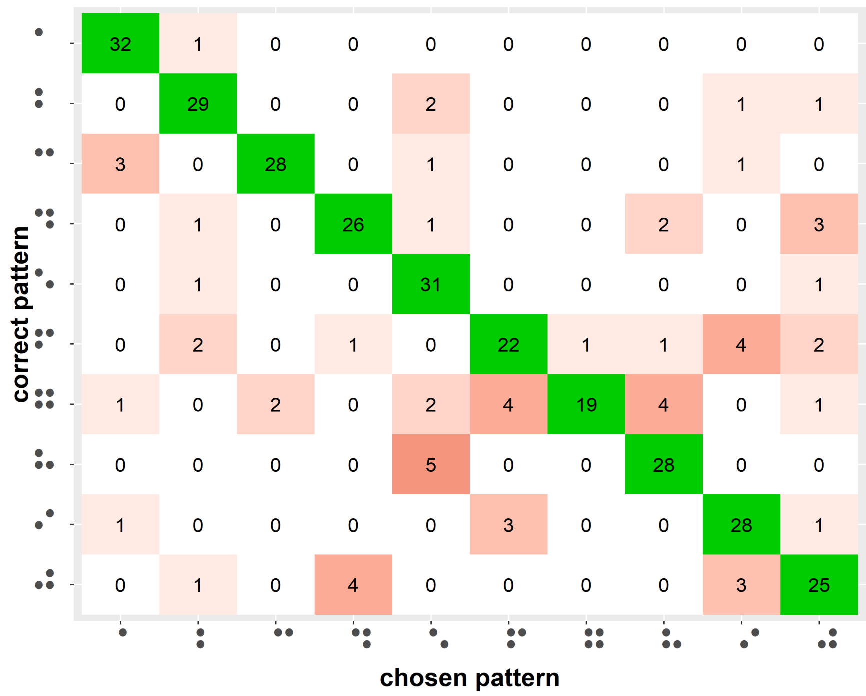


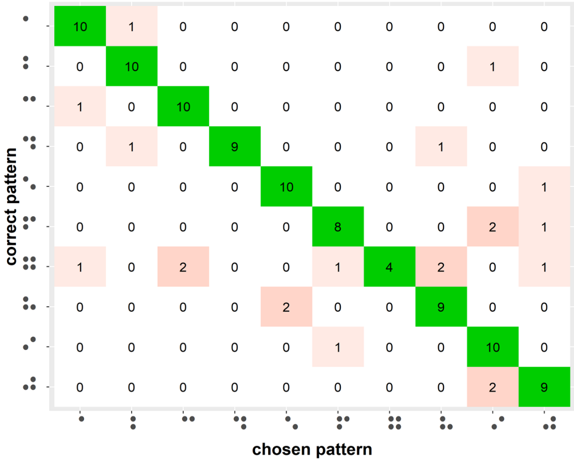
Figure 6.9 Visualisation of the correctly and incorrectly identified patterns in the user study with the respective frequencies.

System Usability

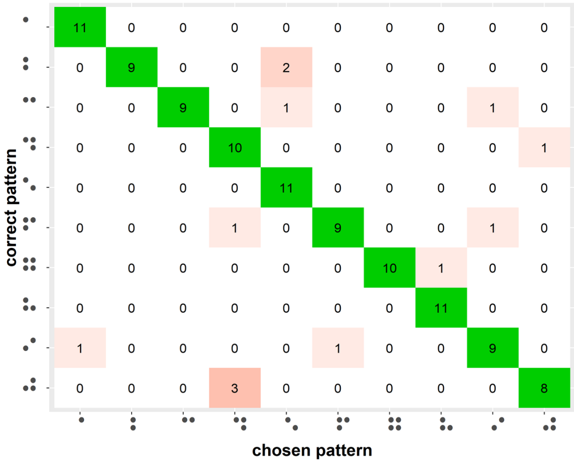
The results of the System Usability Questionnaire are shown in Figure 6.11. The HaptiRead system scored a SUS score of 78.6 (SD 7.6), meaning that the participants rated the system as above average in terms of usability. The main concern participants expressed was that, for example, older members of the blind community might not be able to learn quickly how to use and operate the interface.

Semi-Structured Interview

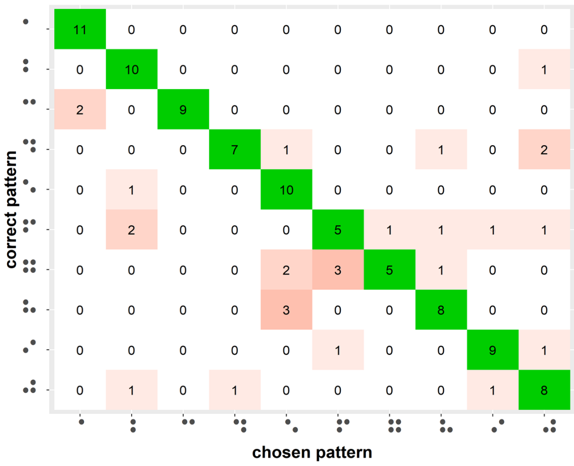
There was no overall preferred haptic stimulation method by the participants in the user study. Some liked the familiarity of the Constant method: *the Constant method felt closest to regular Braille*. Others felt more comfortable with the temporally modulated methods because they provided them with additional spatial cues: *I liked that with the Row-by-Row method I had an indication how to orient my hand.* Regarding the Point-by-Point method, participants stated: *It was more like a flow and the image was constructed over time, like an image is constructed over time in real life.*; *I found the method too slow, but I felt the most secure*. Most participants reported they felt comfortable with using the palm, instead of the finger, for reading the Braille numbers. None of the participants reported feelings of fatigue.



(a) Constant



(b) Point-by-Point



(c) Row-by-Row

Figure 6.10 Confusion Matrix per haptic stimulation method in the user study.

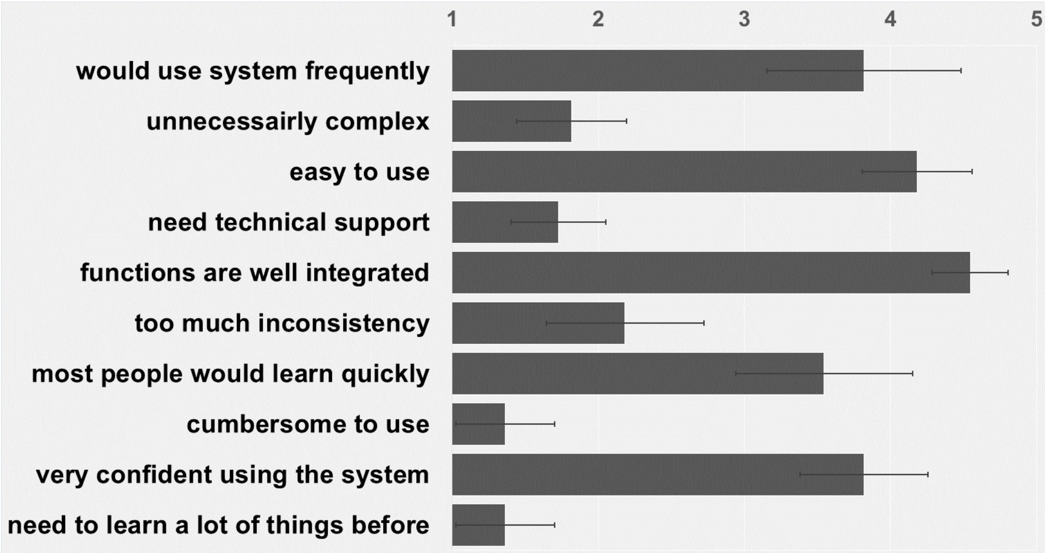


Figure 6.11 Scores on the System Usability Scale for the HapiRead system (1 = strongly disagree, 5 = strongly agree, n = 11).

One participant reported a light tingling sensation in the hand, after experiencing the haptic feedback. As scenarios where they would use HaptiRead, participants listed: to read door signs in public spaces, at the self-checkout register in the supermarket, at a ticket machine, as a small portable clock, to read relief maps etc. One participant stated: *I could use it at work for the punch clock, to see the time I worked*, and another: *I could imagine using the system at home, because my mechanical Braille lines got too slow over time.* They also expressed great interest in a low-priced, ’pocket’ version of HaptiRead that they could carry with them and use it on the spot to convert any text into Braille.

6.7 Discussion

6.7.1 Reading Braille with Mid-Air Haptics

In this chapter, we investigated the possibility of conveying Braille characters using ultrasonic haptics and evaluated three haptic stimulation methods. With the small sample size, we were not able to identify a clear difference between the methods, but we still see value in reporting the scores and the quantitative feedback, as well as the finding that haptic information can be conveyed with all three. The problem needs to be revisited with a larger sample size, to be able to draw clear conclusions about significant differences in the accuracy between the methods. Taking into account the presentation times and the expert feedback, the Point-by-Point and Row-by-Row methods could be beneficial in the initial Braille learning phase, whereas proficient users could potentially prefer the Constant method. An interesting finding is that all participants reported no difficulties in transferring their Braille reading skills to the mid-air haptic interface, after only four training trails for each method. The participants had an overwhelmingly positive reaction to the system and provided a list of scenarios and concrete tasks in their everyday life that could potentially be facilitated by such device. A selection of the possible applications is illustrated in Figure 6.12. Note that further testing and development of the HaptiRead interface is required to achieve them.

6.7.2 Limitations

This first validation study was conducted using a small subset of Braille characters, limited to four cells, individually presented, to ensure internal validity and experimental control. Further studies are required to validate the findings using a full 6-cell layout, as well as presenting the information in context (e.g. words and sentences). Due to these limitations, it is difficult to compare the obtained results to the prior work. Testing with 6-cell characters, might result in lower accuracy rates, they could, however, potentially be compensated by longer training sessions. Rendering 6 or even 8-cell Braille characters could potentially be facilitated in the future, by manufacturing mid-air haptic displays with smaller transducers and thus better spatial resolution. In our extensive testing process with domain experts and users, we did not come across any major challenge or criticism that would pose a doubt that with sufficient testing and development, the HaptiRead system would not work for more complex information.

6.7.3 Future Work

The user study presented here focused on testing whether mid-air haptic feedback alone can be sufficient to communicate Braille. For future applications, involving more complex content (e.g. city maps, platform games etc.), we plan to explore strategies for multisensory integration, where haptic and audio cues are used to encode different types of information. As mentioned by one of the interviewees, the ability of the system to temporally modulate the dot patterns can be beneficial for Braille learners, thus we would like to investigate methods that could optimise the Braille learning process with HaptiRead.

6.8 Conclusion

In this chapter, we evaluate the possibility of using ultrasonic mid-air haptic technology to convey Braille. The obtained results hold importance for the field of Human Computer

Figure 6.12 Potential applications scenarios for HaptiRead (left to right, top to bottom): to read the account balance at the ATM, display landmark names and direction on navigation maps, to facilitate item localisation in restrooms, to provide floor information in elevators, to provide directions to pedestrians in traffic, and support navigation in large public buildings (e.g. hospitals).

Interaction, because they provide the first empirical validation of employing mid-air haptics for developing interfaces for blind people. We conduct performance and system usability tests and evaluate three different methods for generating the haptic stimulation. Our results show that it is possible to convey Braille as touchless haptic stimulation in mid-air with all of the proposed methods. The participants responded favourably to the concept, however, further testing and development is needed. We hope that our study will spark research into using mid-air haptics to potentially make the everyday multisensory experience of visually impaired and blind people richer.

Chapter 7

Optimal Trap Trajectories for Acoustic Levitation Displays

7.1 Introduction

Acoustic levitation has recently demonstrated the creation of volumetric content in mid-air by exploiting the PoV effect. This is achieved by using acoustic traps to rapidly move single [53] or multiple [115] particles along a periodic reference path, revealing a shape within $0.1s$ (i.e., the integration interval of human eyes [14]).

However, the way to define such PoV content remains unsolved. That is, there are no automated approaches to compute the *trap trajectories*, i.e., the positioning and timing of the acoustic traps that will allow us to reveal the desired shape.

Currently, content creators can only rely on trial and error to find physically feasible trap trajectories resembling their intended target shapes, resulting in a time consuming and challenging process. For instance, the creator will need to define the timing of the path (i.e., not only *where* the particle must be, but also *when*). Such timing is not trivial and will affect the overall rendering time and the accelerations applied to the particle, which must be within the capabilities of the levitator. The way forces distribute around the acoustic trap will also need to be considered to decide where traps must be located.

Considering these challenges is crucial to design feasible trap trajectories, as a single infeasible point along the trajectory typically results in the particle being ejected from the levitator (e.g., approaching a sharp corner too quickly). Thus, while it has been theorised that linear PoV paths of up to 4m and peak speeds of 17m/s are possible [38], actual content demonstrated to date has been limited to simple shapes with almost constant curvature along

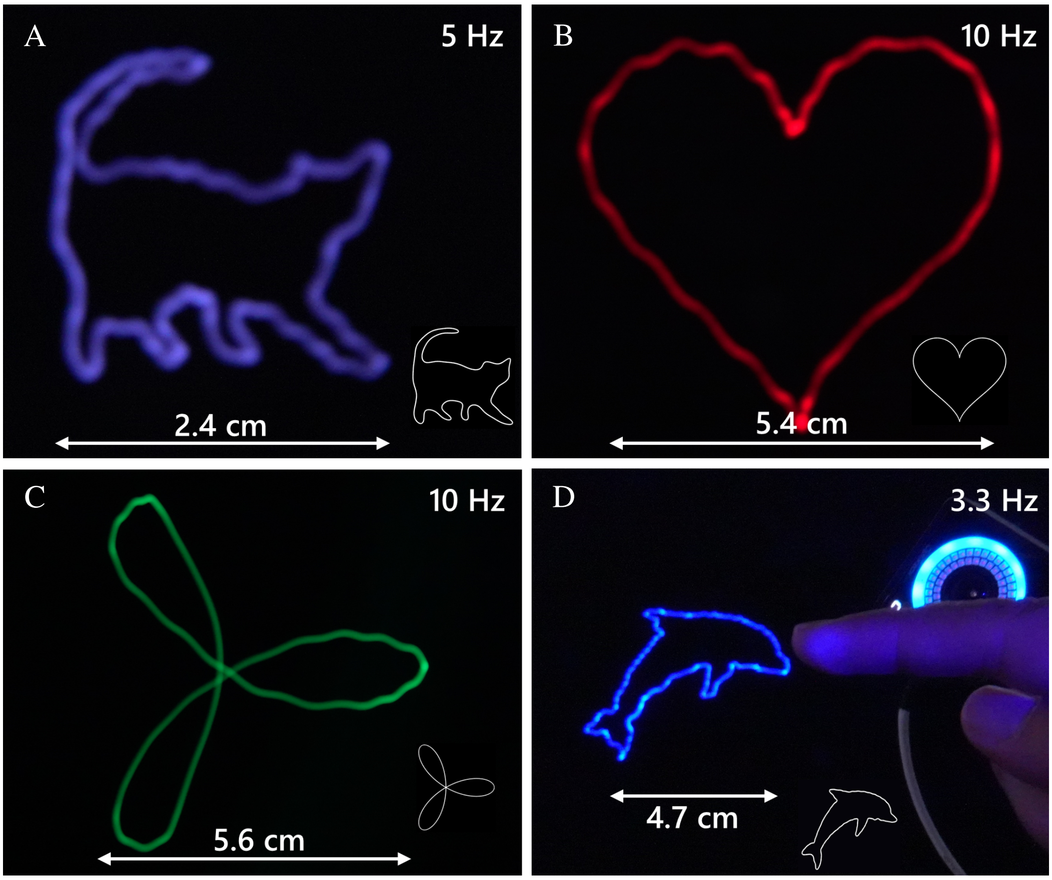


Figure 7.1 While previously, shapes demonstrated on levitation displays were limited to simple shapes with almost constant curvature [40, 53, 115], our approach allows to render generic complex paths. Shapes that have not been demonstrated before include sharp edges as with the heart (B) and dolphin (D), as well as significant changes of the curvature, as with the cat (A) and flower (C). Photos of the physical particle are made with an exposure time ranging from 0.2 to 1 s, such that each photo shows multiple periods of the orbit.

the path and much lower speeds (e.g., 0.72m/s in [101]; 1.2m/s in [53]; 2.25m/s in [115], combining six particles).

In this chapter, we present *OptiTrap*, the first structured numerical approach to compute trap trajectories for acoustic levitation displays. *OptiTrap* automates the definition of levitated PoV content, computing physically feasible and nearly time-optimal trap trajectories given only a reference path, i.e., the desired shape. As shown in Figure 7.1, this allows for larger and more complex shapes than previously demonstrated, as well as shapes featuring significant changes in curvature and/or sharp corners.

Our approach is summarised in Figure 7.2. *OptiTrap* assumes only a generic reference path $\mathbf{q}(\theta)$ as an input, with no temporal information (see Figure 7.2(A)). *OptiTrap* formulates this as a path following problem (see Figure 7.2(B)), computing the optimum timing in which

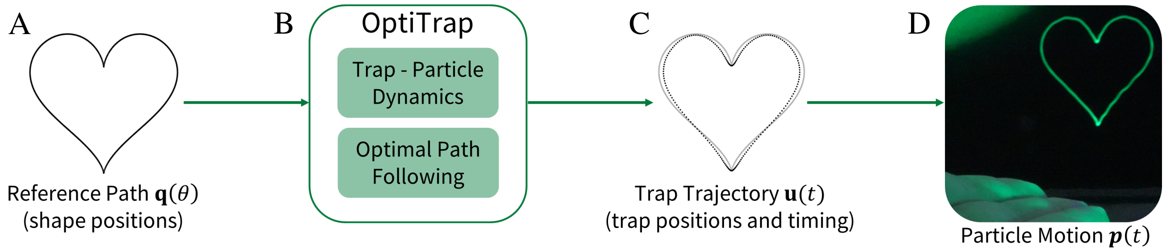


Figure 7.2 *OptiTrap* is an automated method to compute trap trajectories to reveal generic mid-air shapes on levitation displays. The method accepts a reference path (e.g., the shape of a heart) without any timing information as an input (A). Our approach considers the capabilities of the device and its trap-particle dynamics and combines these with a path following approach (B). The approach produces feasible trap trajectories, describing when and where the traps must be created (C). The resulting particle motion can be presented on the actual device, yielding feasible paths and supporting complex objects with sharp edges and/or significant changes of the curvature (D).

a particle can traverse such path. Our formulation considers the *Trap-Particle Dynamics* of the system, using a 3D model of acoustic forces around a trap. This results in a non-invertible model, which cannot exploit differential flatness [32], on which most path following approaches rely. We instead provide a coupling stage for the dynamics of our system and numerically invert the system. This approach produces a trap trajectory $\mathbf{u}(t)$ (Figure 7.2(C)) that results in the intended and nearly time-optimal particle motion. That is, $\mathbf{u}(t)$ defines the positions and timing of the traps that cause the particle to reveal the target shape $\mathbf{q}(\theta)$, according to the capabilities of the actual device, cf. Figure 7.2(D).

In summary, we contribute the first structured numerical approach to compute physically feasible and nearly time-optimal trap trajectories for levitation displays, given only a reference path that the particle should follow. As a core contribution, we provide a theoretical formulation of the problem in terms of path following approaches, allowing optimum timing and device properties (e.g., trap-particle dynamics) to be jointly considered. All the details of the approach we propose are provided in Section 7.4.

We illustrate the potential of our approach by rendering shapes featuring straight lines, sharp corners, and complex shapes, such as those in Figure 7.1. We then provide an experimental validation of our approach, demonstrating increases of up to 563% in the size of rendered objects, up to 150% in the rendering frequency, and improvements in accuracy (e.g., shape revealed more accurately). While the baseline shapes we compare against require trial and error to determine optimal sizes or frequencies, our approach always yields feasible paths (i.e., working in at least 9 out of 10 attempts) and makes consistent use of accelerations very close to (but not exceeding) the maximum achievable by the device, independently of the target shape.

These features allow *OptiTrap* to render complex objects involving sharp edges and significant changes in curvature that have never been demonstrated before. Even more importantly, it provides a tool to systematically explore the range of contents that levitation displays can create, as a key step to exploit their potential.

7.2 Background and Challenges

Our goal is is to minimise complexity for content creators. Thus, given a reference path (i.e., a shape defined by the content creator), our approach must produce physically feasible trap trajectories, which accurately reveal the desired path, while making optimal use of the capabilities of the device.

We formalise our content as a reference path Q, provided by the content creator without any preassigned time information. It is an explicitly parametrised curve

$$Q := \{\xi \in \mathbb{R}^3 \mid \theta \in [\theta_0, \theta_f] \mapsto \mathbf{q}(\theta)\}, \tag{7.1}$$

where $\mathbf{q}: \mathbb{R} \to \mathbb{R}^3$, as later justified in Section 7.4.2, must be a twice continuously differentiable function with respect to θ, which is called the path parameter. Increasing values of θ denote forward movement along the path Q. The starting point on the path is $\mathbf{q}(\theta_0)$, while $\mathbf{q}(\theta_f)$ marks the end of the path. For periodic paths, such as a particle cyclically revealing a PoV shape, $\mathbf{q}(\theta_0) = \mathbf{q}(\theta_f)$ and $\dot{\mathbf{q}}(\theta_0) = \dot{\mathbf{q}}(\theta_f)$ hold. Please note that while splines would provide a generic solution to this parametrisation (i.e., twice continuously differentiable fitting the points in Q), any other parametric functions can be used. For example, taking $\theta \in [0, 2\pi]$ and $r > 0$, a cardioid as in Figure 7.3, can be described by:

$$\mathbf{q}(\theta) = (0, r\sin(\theta)(1+\cos(\theta)), -r\cos(\theta)(1+\cos(\theta))+r)^\top. \tag{7.2}$$

Whatever the approach used, revealing such reference paths typically involves rapid particle movements, as PoV content must be revealed within about $0.1s$ [14]) and must consider the feasibility of the path (i.e., the trap-particle dynamics, the topology of the trap, and the capabilities of the levitator). This results in the following two main challenges for *OptiTrap*: 1) determining the optimal timing for the particle revealing the path; and 2) computing the trap positions generating the required forces on the particle.

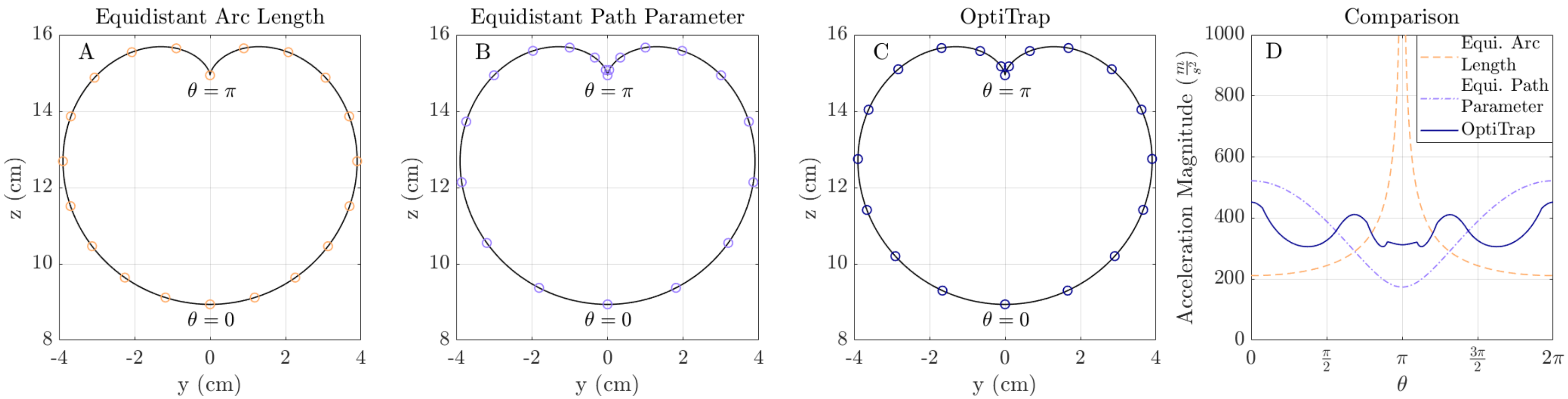


Figure 7.3 Three different path timings for the cardioid shape with fixed traversal time: (A) equidistant sampling of the arc length, (B) equidistant sampling of the path parameter θ, and (C) *OptiTrap*. The corresponding acceleration magnitudes are depicted in (D). Strategy (A) is physically infeasible due to infinite required acceleration at the corner ($\theta = \pi$). Strategy (B) does not share this problem, but requires careful parametrisation of the path, while *OptiTrap* (C) yields the timing automatically.

7.2.1 Challenge 1: Determining an Optimal Path Timing

The reference path Q defines the geometry but not the timing (i.e., it describes *where* the particle needs to be, but not *when* it needs to be there). Our approach must compute such a timing, and the strategy followed will have important implications on the velocity and accelerations applied to the particle and, hence, on the physical feasibility of the content.

Different timing strategies and their effects on the feasibility can be illustrated using the cardioid example from (7.2). Figure 7.3 depicts various timing strategies applied to this shape, all of them traversing the same path in the same overall time.

A straight-forward approach would be to sample equidistantly along the path, as shown in Figure 7.3(A). This works well for lines and circles, where a constant speed can be maintained, but fails at sudden changes in curvature due to uncontrolled accelerations (see the acceleration at the corner of cardioid in Figure 7.3(D), at $\theta = \pi$).

The second example in Figure 7.3(B) shows a simple equidistant sampling on the path parameter θ. As shown in Figure 7.3(D), this results in areas of strong curvature naturally involving low accelerations, and can work particularly well for low-frequency path parametrisations in terms of sinusoidals (e.g., [38] showed such sinusoidal timing along straight paths, to theorise optimum/optimistic content sizes). In any case, the quality of the result obtained by this approach will vary depending on the specific shape and parametrisation used.

As an alternative, the third example shows the timing produced by our approach, based on the shape and capabilities of the device (i.e., forces it can produce in each direction). Please note how this timing reduces the maximum accelerations required (i.e., retains them below the limits of the device), by allowing the particle to travel faster in parts that were unnecessarily slow (e.g., Figure 7.3(D), at $\theta = \pi$). For the same maximum accelerations (i.e., the same device), this timing strategy could produce larger shapes or render them in shorter times, for better refresh rates. This illustrates the impact of a careful timing strategy when revealing any given reference path. We address this challenge in Section 7.4.2.

7.2.2 Challenge 2: Computation of Trap Positions

In addition to the timing, the approach must also compute the location of the traps. Previous approaches [53, 115, 38] placed the traps along the shape to be presented, under the implied assumption that the particle would remain in the centre of such trap.

However, the location where the traps must be created almost never matches the location of the particle. Acoustic traps feature (almost) null forces at the centre of the trap and high restorative forces around them [86]. As such, the only way a trap can accelerate a particle is by having it placed at a distance from the centre of the trap. Such trap-particle distances were measured by [53] to assess performance achieved during their speed tests. Distortions related to rendering fast moving PoV shapes were also shown in [40]. However, none of them considered or corrected for such displacements.

Please note the specific displacement between the trap and the particle will depend on several factors, such as the particle acceleration required at each point along the shape, as well as the trap topology (i.e., how forces distribute around it). No assumptions can be made that the traps will remain constrained to positions along or tangential to the target shape. We describe how our approach solves this challenge in Section 7.4.3.

7.3 Related Work

OptiTrap addresses the challenges above by drawing from advances in the fields of acoustic levitation and control theory, which we review in this section.

7.3.1 Acoustic Levitation

Single frequency sound-waves were first observed to trap dust particles in the lobes of a standing wave more than 150 years ago [136]. This has been used to create mid-air displays with particles acting as 3D voxels [101, 103, 105, 126], but such standing waves do not allow control of individual particles. Other approaches have included Bessel beams [99], self-bending beams [100], boundary holograms [55] or near-field levitation [6].

However, most display approaches have relied on the generic levitation framework proposed by [86], combining a focus pattern and a levitation signature. Although several trap topologies (i.e., twin traps, vortex traps or bottle beams) and layouts (i.e., one-sided, two-sided, v-shape) are possible, most displays proposed have adopted a top-bottom levitation setup and twin traps, as these result in highest vertical trapping forces. This has allowed individually controllable particles/voxels [84] or even particles attached to other props and projection surfaces for richer types of content [89, 31]. The use of single [40, 53] or multiple [115] fast moving particles have allowed for dynamic and free-form volumetric content, but is still limited to small sizes and simple vector graphics [38].

Several practical aspects have been explored around such displays, such as selection [37] and manipulation techniques [9], content detection and initialisation [31] or collision avoidance [121].

However, no efforts have been made towards optimising content considering the capabilities (i.e., the dynamics) of such displays, particularly for challenging content such as the one created by PoV high-speed particles. [53] showed sound-fields must be updated at very high rates for the trap-particle system to engage in high accelerations, identifying optimum control for rates above 10kHz. However, they only provided a few guidelines (e.g., maximum speed in corners, maximum horizontal/vertical accelerations) to guide the definition of the PoV content. [38] provided a theoretical exploration of this topic, looking at maximum achievable speeds and content sizes, according to the particle sizes and sound frequency used. While the dynamics of the system were considered, these were extremely simplified, using a model of acoustic trap forces and system dynamics that are only applicable for oscillating recti-linear trajectories along the vertical axis of the levitator. [108] proposed a generic system simulating the dynamics of a top-bottom levitation system. This operates as a forward model simulating the behaviour of the particle given a specific path for the traps, but unlike our approach it does not address the inverse problem. Thus, *OptiTrap* is the first algorithm allowing the definition of PoV content of generic shapes, starting only from a geometric definition (i.e., shape to present, no timing information) and optimising it according to the capabilities of the device and the dynamics of the trap-particle system.

7.3.2 Path Following and Optimal Control

The particle in the trap constitutes a dynamical system, which can be controlled by setting the location of the trap. As such, making the particle traverse the reference path can be seen as an optimal control problem (OCP).

In the context of computer graphics, optimal control approaches have been studied particularly in the areas of physics-based character animation [42] and aerial videography [93]. Such OCPs can be considered as function-space variants of nonlinear programs, whereby nonlinear dynamics are considered as equality constraints. In engineering applications, OCPs are frequently solved via direct discretization [147, 13], which leads to finite-dimensional nonlinear programs. In physics-based character animation, such direct solution methods are known as spacetime constraints [151, 124].

If framed as *a levitated particle along a given geometric reference path (i.e., the PoV shape)*, the problem can be seen as an instance of a path following problem [28]. Such problems also occur in aerial videography, when a drone is to fly along a reference path [93, 122]. In order to yield a physically feasible trajectory, it is necessary to adjust the timing along that trajectory. This can be done by computing feed-forward input signals [28, 122] or, if the system dynamics and computational resources allow, using closed-loop solutions, such as Model Predictive Control in [93].

However, optimal control of acoustically levitated particles cannot be approached using such techniques. All of the above approaches require *differential flatness* [32]. That is, the underlying system dynamics must be invertible, allowing for the problem to be solved by projecting the dynamics of the moving object onto the path manifold [97]. This inversion approach also enables formulating general path following problems in the language of optimal control, encoding desired objectives (e.g., minimum time, see [130, 144, 29]).

As discussed later in Section 7.4.1, the distribution of forces around the acoustic trap (i.e., our system dynamics) are not invertible, requiring approaches that have not been widely developed in the literature. We do this by coupling the non-invertible particle dynamics into a virtual system, solvable with a conventional path following approach. We then use these coupling parameters and our model of particle dynamics to solve for the location of the traps.

7.4 Optimal Control for Levitation Displays

This section provides a description of our *OptiTrap* approach, which automates the definition of levitated content, computing physically feasible and nearly time-optimal trap trajectories using only a reference path as an input. In Section 7.4.1, we first describe our specific hardware setup and the general mathematical framework, then we consider existing models of the trap-particle dynamics (Subsection 7.4.1), before introducing our proposed model (Subsection 7.4.1). Next, we describe the two stages in our algorithm, which match the challenges identified in Section 7.2. That is, Section 7.4.2 describes the computation of optimum timing, approached as an optimal path following problem, while Section 7.4.3 explains how to compute the trap locations. Please note that Section 7.4 mainly focuses on the general case of presenting levitated content, that is, particles cyclically traversing a reference path (shape) as to reveal it. Other cases, such as a particle accelerating from rest as to reach the initial state (i.e., initial position and speed) required to render the content can be easily derived from the general case presented here, and are detailed at the end, in Section 7.4.4.

7.4.1 Modelling the Trap-Particle Dynamics

We start by describing the model of the trap-particle dynamics used for our specific setup, shown in Figure 7.4. This setup uses two opposed arrays of 16×16 transducers controlled by an FPGA and an OptiTrack tracking system (Prime 13 motion capture system at a frequency of 240Hz). The design of the arrays is a reproduction of the setup in [88], modified to operate at 20Vpp and higher update rates of 10kHz. The device generates a single twin-trap using the method described in [53], allowing for vertical and horizontal forces of $4.2 \cdot 10^{-5}$N and $2.1 \cdot 10^{-5}$N, respectively, experimentally computed using the linear speed tests in [53] (i.e., 10cm paths, binary search with 9 out 10 success ratios, with particle mass $m \approx 0.7 \cdot 10^{-7}$kg).

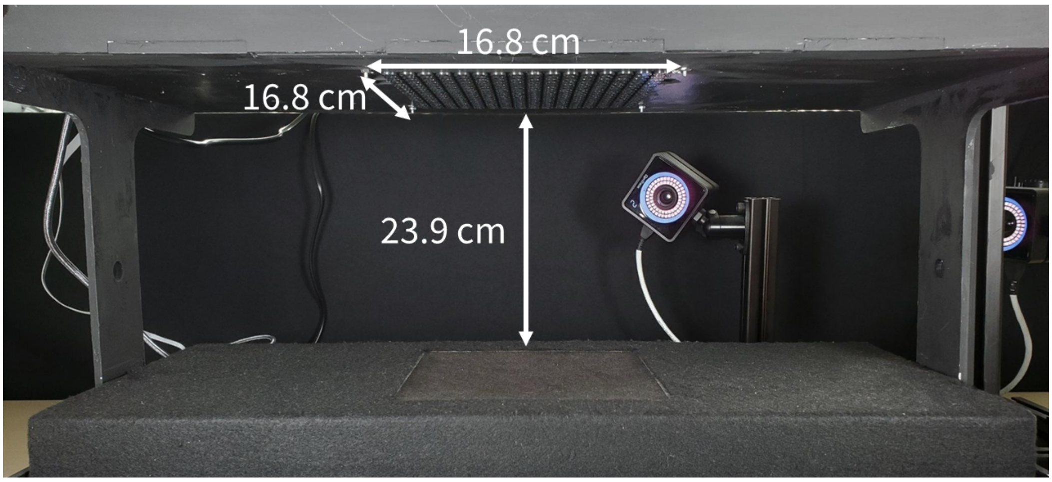


Figure 7.4 Overview of the components in our setup. We used two opposed arrays of transducers at a distance of 23.9 cm and an OptiTrack system to track the position of levitated particles in real time.

Note the substantial difference between the maximum forces in the vertical and horizontal directions. Our approach will need to remain aware of direction as this determines maximum accelerations.

In general, the trap-particle dynamics of such system can be described by simple Newtonian mechanics, i.e.,

$$m\ddot{\mathbf{p}}(t) = F(\mathbf{p}(t), \dot{\mathbf{p}}(t), \mathbf{u}(t)). \tag{7.3}$$

Here, $\mathbf{p}(t) = (p_x, p_y, p_z)^\top \in \mathbb{R}^3$ represents the particle position in Cartesian (x, y, z) coordinates at time $t \in \mathbb{R}_0^+$, and $\dot{\mathbf{p}}(t)$ and $\ddot{\mathbf{p}}(t)$ are the velocity and acceleration of the particle, respectively. The force acting on the particle is mostly driven by the acoustic radiation forces and, as such, drag and gravitational forces can be neglected [53]. Therefore, the net force acting on the particle will depend only on $\mathbf{p}(t)$ and on the position of the acoustic trap at time t denoted by $\mathbf{u}(t) = (u_x(t), u_y(t), u_z(t))^\top \in \mathbb{R}^3$.

As a result, our approach requires an accurate and ideally invertible model of the acoustic forces delivered by an acoustic trap. That is, we seek an accurate model predicting forces at any point around a trap only in terms of $\mathbf{u}(t)$ and $\mathbf{p}(t)$. Moreover, an invertible model would allow us to analytically determine where to place the trap as to produce a specific force on the particle given the particle location.

For spherical particles considerably smaller than the acoustic wavelength and operating in the far-field regime, such as those used by our device, the acoustic forces exerted can be modelled by the gradient of the Gor'kov potential [18]. This is a generic model, suitable to

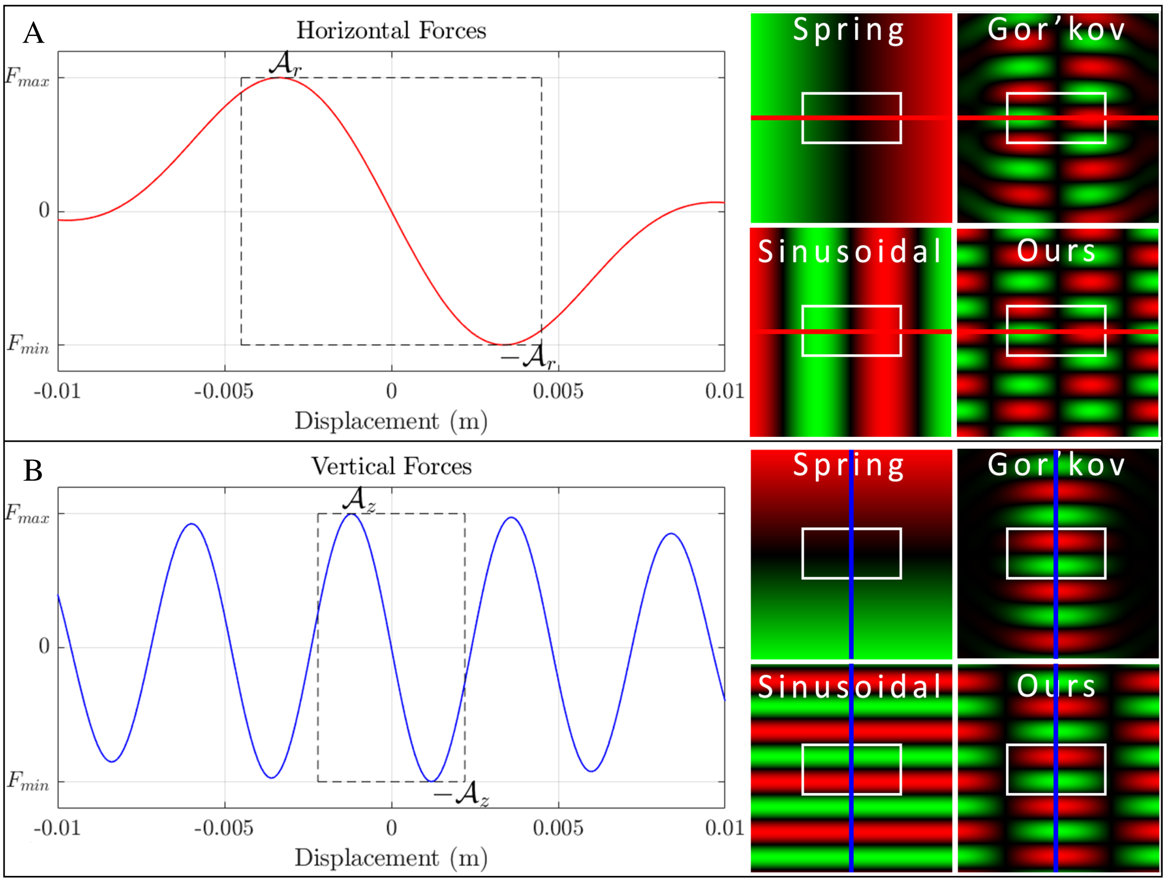


Figure 7.5 Analytical models of horizontal (A) and vertical (B) acoustic forces around a trap. The plots on the left show forces along the main axes X and Z, and analytical models provide a good fit. The plots on the right show horizontal (A) and vertical (B) forces across 2D slices through the trap centre along the X and Z axes. Spring and Sinusoidal models fail to predict forces outside the main axes, while our proposed model provides accurate reconstruction within the region of interest (highlight).

model acoustic forces resulting from any combination of transducer locations and transducer activations, but it also depends on all these parameters, making it inadequate for our approach.

Our case, using a top-bottom setup and vertical twin traps is much more specific. This allows for simplified analytical models with forces depending only on the relative position of the trap and the particle, which we compare to forces as predicted from the Gor'kov potential in Figure 7.5.

Existing Models of the Trap-Particle Dynamics

Simple *spring* models have been used extensively [108, 39, 38], modelling trapping forces according to a *stiffness* parameter $\mathcal{K}_i$, that is, with forces being proportional to the distance of the particle to the centre of the trap:

$$F_i(\mathbf{p},\mathbf{u}) := \mathscr{K}_i \cdot |u_i - p_i|, \quad i \in \{x,y,z\}.$$

Such models are usually refined by providing specific stiffness values for each dimension, but they are only suitable for particles remaining in close proximity to the centre of the trap, i.e., the linear region near the centre of the trap. As a result, *spring* models are only accurate for systems moving particles slowly, requiring low acceleration and forces (so that particles remain within the linear region).

Sinusoidal models have been proposed as an alternative [39, 38], providing accurate fitting from the centre of the trap to the peaks of the force distribution in Figure 7.5, according to the peak trapping force $\mathscr{A}_i$ and characteristic frequency $\mathscr{V}_i$:

$$F_i(\mathbf{p},\mathbf{u}) := \mathscr{A}_i \cdot \sin(\mathscr{V}_i \cdot (u_i - p_i)), \quad i \in \{x,y,z\},$$

However, both of these models (i.e., *spring* and *sinusoidal*) are only suitable for particles placed along one of the main axes of the acoustic trap, not for particles arbitrarily placed at any point around it. This is illustrated on the right of Figure 7.5, which provides an overview of how forces distribute on a horizontal and vertical 2D plane around a trap, according to each model (i.e., Gor'kov, *spring*, *sinusoidal*, and *Ours*, detailed in the next subsection).

Please note how the three previous models show good matching along the horizontal and vertical axes (represented as blue and red lines). However, the *spring* and *sinusoidal* models are of one-dimensional nature (e.g., force F_x only depends on X distance $(u_x - p_x)$) and become inaccurate at points deviating from the main axes.

Our Model

The complexity of the force distribution modelled by the Gor'kov potential increases as the distance to the centre of the trap increases. However, we only need to derive a model allowing us to predict where to place the trap to produce a specific force on the particle. This allows us to limit our considerations to the region corresponding to the peaks designated by $\mathscr{A}_r$ and $\mathscr{A}_z$ in Figure 7.5.

Table 7.1 Fit parameters for the Spring, Sinusoidal and Axis-symmetric models and the respective average relative errors when compared to Gor'kov.

Model	Spring		Sinusoidal		Axis-symmetric	
Fit	$\mathscr{K}_x$	-0.0071	$\mathscr{A}_x$	0.00009	$\mathscr{A}_r$	0.0004636
	$\mathscr{K}_y$	-0.0071	$\mathscr{A}_y$	0.00009	$\mathscr{A}_z$	0.0002758
	$\mathscr{K}_z$	-0.94	$\mathscr{A}_z$	-0.0019	$\mathscr{V}_z$	1307.83
			$\mathscr{V}_x$	-68.92	$\mathscr{V}_{xr}$	-476.49
			$\mathscr{V}_y$	-68.92	$\mathscr{V}_{zr}$	287.87
			$\mathscr{V}_z$	1307.83		
Error	63.3%		35.9%		4.3%	

For points within this region, forces around twin traps distribute in a mostly *axis-symmetric* fashion, which can be approximated as:

$$F_r(\mathbf{p},\mathbf{u}) := \mathscr{A}_r \cdot \cos\left(\mathscr{V}_z \cdot (u_z - p_z)\right) \cdot \sin\left(\mathscr{V}_{xr} \cdot \sqrt{(u_x - p_x)^2 + (u_y - p_y)^2}\right), \tag{7.4a}$$

$$F_z(\mathbf{p},\mathbf{u}) := \mathscr{A}_z \cdot \sin\left(\mathscr{V}_z \cdot (u_z - p_z)\right) \cdot \cos\left(\mathscr{V}_{zr} \cdot \sqrt{(u_x - p_x)^2 + (u_y - p_y)^2}\right), \tag{7.4b}$$

where $\mathscr{A}_r, \mathscr{A}_z$ represent peak trapping forces along the radial and vertical directions of the trap, respectively. $\mathscr{V}_z$, $\mathscr{V}_{xr}$, $\mathscr{V}_{zr}$ represent characteristic frequencies of the sinusoidals describing how the forces evolve around the trap. The resulting forces can be converted into acoustic forces in 3D space from these cylindrical coordinates, with azimuth $\phi = \arctan((u_y - p_y)/(u_x - p_x))$:

$$F(\mathbf{p},\mathbf{u}) = \begin{pmatrix} F_x(\mathbf{p},\mathbf{u}) \\ F_y(\mathbf{p},\mathbf{u}) \\ F_z(\mathbf{p},\mathbf{u}) \end{pmatrix} := \begin{pmatrix} F_r(\mathbf{p},\mathbf{u})\cos\phi \\ F_r(\mathbf{p},\mathbf{u})\sin\phi \\ F_z(\mathbf{p},\mathbf{u}) \end{pmatrix}. \tag{7.5}$$

We validated our model by comparing its accuracy against the forces predicted by the gradient of the Gor'kov potential. More specifically, we simulated 729 single traps homogeneously distributed across the working volume of our levitator (i.e., 8x8x8cm, in line with [53, 40]), testing 400 points around each trap for a total of 400x729 force estimations.

Table 7.1 summarises the error distribution achieved by these three models when compared to Gor'kov, showing an average relative error as low as 4% for our model and much poorer fitting for the other two models. To validate our force model, we tested its accuracy by comparing it to the forces predicted by the Gor'kov potential. More specifically, the Gor'kov potential was derived by considering our trap (twin trap), particle (radius 1mm; density 19kg/m^3; speed of sound in particle 900m/s), setup (top and bottom arrays of 16×16 transducers, each modelled using a piston model) and assuming 346m/s and 1.18 kg/m^3 as the speed and density of air, respectively.

We also compared these predictions to those from other previously existing analytical models (i.e., *Spring* and *Sinusoidal* models). We simulated 729 single traps, each of them homogeneously distributed across a 8x8x8cm lattice at the centre of our levitator (i.e., at -4, -3, -2, -1, 0, 1, 2, 3, and 4cm along each axis). The force was computed at 400 points distributed around each trap, with distances of up to 4.5mm in the horizontal and 2.2mm in the vertical direction, thus covering the region from the centre of the trap to the peak forces in each direction (see Figure 7.5). This resulted in a total of 400 x 729 force estimations, where each estimation is a 3D vector.

We used this data to compute parameters for each analytical model (i.e., *Spring*, *Sinusoidal*, and *Axis-symmetric*) minimising the fitting error. More specifically, we used *lsqnonlin*, a non-linear least squares fitting algorithm in Matlab, resulting in the fitting parameters summarised in Table 7.1.

As a final summary, this results in a model for our trap-particle dynamics that only depends on $\mathbf{p}$ and $\mathbf{u}$ and which fits the definition of a second-order ordinary differential equation:

$$m\ddot{\mathbf{p}}(t) = F(\mathbf{p}(t), \mathbf{u}(t)). \tag{7.6}$$

While the resulting model is accurate (4% relative error), we note that it is not invertible, which will complicate the formulation of our solution approach. For instance, for a particle at $\mathbf{p} = (0,0,0)$, force $F(\mathbf{p},\mathbf{u}) = (0,0,\mathscr{A}_z \cdot \cos(\mathscr{V}_{zr} \cdot x))$ can be obtained with either $\mathbf{u} = (x, 0, \pi/(2\mathscr{V}_z))$ or $\mathbf{u} = (-x, 0, \pi/(2\mathscr{V}_z))$.

7.4.2 Open-Loop Optimal Path Following

This section computes the timing for the particle, so that it moves along the given reference path Q from (7.1) in minimum time and according to the dynamics of the system. We approach this as an open-loop (or feed-forward) path following problem, as typically done in robotics [28]. That is, we design an optimal control problem that computes the timing $t \mapsto \theta(t)$ as to keep the levitated particle on the prescribed path, to traverse the path in optimum (minimum) time while considering the trap-particle dynamics (i.e., minimum *feasible* time).

Our non-invertible model of particle dynamics calls for a more complex treatment of the problem, which we split in two parts. In the first part we derive a virtual system for the timing law, which we describe as a system of first-order differential equations, solvable with traditional methods. In the second part we couple the timing law with our non-invertible dynamics, introducing auxiliary variables that will later enable pseudo-inversion (i.e., to compute trap placement for a given force) as described in Section 7.4.3.

Error Dynamics and the Timing Law.

The requirement that the particle follows the path Q exactly and for all times means that the deviation from the path equals zero for all $t \in \mathbb{R}_0^+$, i.e.,

$$\mathbf{e}(t) := \mathbf{p}(t) - \mathbf{q}(\theta(t)) \equiv 0.$$

If the path deviation $\mathbf{e}(t)$ is 0 during the whole interval $[t_0, t_1]$, this implies that the time derivatives of $\mathbf{e}(t)$ also have to vanish on (t_0, t_1). Considering this (i.e., $\dot{e}(t) \equiv 0$ and $\ddot{e}(t) \equiv 0$) results in:

$$\mathbf{p}(t) = \mathbf{q}(\theta(t)), \tag{7.7a}$$

$$\dot{\mathbf{p}}(t) = \dot{\mathbf{q}}(\theta(t)) = \frac{\partial \mathbf{q}}{\partial \theta}\dot{\theta}(t), \tag{7.7b}$$

$$\ddot{\mathbf{p}}(t) = \ddot{\mathbf{q}}(\theta(t)) = \frac{\partial^2 \mathbf{q}}{\partial \theta^2}\dot{\theta}(t)^2 + \frac{\partial \mathbf{q}}{\partial \theta}\ddot{\theta}(t). \tag{7.7c}$$

Thus, for particles on the path Q, system dynamics of the form (7.6), and provided we are able to express $\mathbf{u}$ as a function of $\mathbf{p}$ and $\ddot{\mathbf{p}}$ (i.e., the system inversion described in Section 7.3.2), the position, the velocity, and the acceleration of the particle can be expressed via $\theta, \dot{\theta}, \ddot{\theta}$, respectively. For details, a formal derivation, and tutorial introductions we refer to [28, 30].

Observe that in (7.7) the partial derivatives $\frac{\partial^2 \mathbf{q}}{\partial \theta^2}$ and $\frac{\partial \mathbf{q}}{\partial \theta}$, as well as $\dot{\theta}$ and $\ddot{\theta}$ appear. This leads to two additional observations: First, the parametrisation $\mathbf{q}$ from (7.1) should be at least twice continuously differentiable with respect to θ, so that the partial derivatives are well-defined. This justifies our constraint in Section 7.2. Please note this does not rule out corners in the path Q, as shown by the parametrisation (7.2) of the cardioid. Second, the time evolution of θ should be continuously differentiable, as otherwise large jumps can occur in the acceleration.

To avoid said jumps, we generate the timing $t \mapsto \theta(t)$ via the double integrator

$$\ddot{\theta}(t) = v(t), \tag{7.8}$$

where $v(t) \in \mathbb{R}$ is a computational degree of freedom, used to control the progress of the particle along Q. The function $v(t)$ enables us to later cast the computation of time-optimal motions along reference paths as an OCP, see (7.16).

The in-homogeneous second-order ordinary differential equation (7.8) takes care of jumps, but needs to be augmented by conditions on θ and $\dot{\theta}$ at initial time $t = 0$ and final time $t = T$. This is done to represent the periodic nature of our content (i.e., the particle reveals the same path many times per second):

$$\theta(0) = \theta_0, \quad \theta(T) = \theta_f, \quad \dot{\theta}(0) = \dot{\theta}(T), \tag{7.9}$$

where θ_0 and θ_f are taken from (7.1) and the total time T will be optimally determined by the OCP. The first two equations in (7.9) ensure that the path Q is fully traversed, while the third one ensures the speeds at the beginning and end of the path match.

With these additional considerations, we define the (state) vector $\mathbf{z}(t) := (\theta(t), \dot{\theta}(t))^\top$, which finally allows us rewrite our second-order differential equations in (7.8) as a system of first-order differential equations:

$$\dot{\mathbf{z}}(t) = \begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix} \mathbf{z}(t) + \begin{pmatrix} 0 \\ 1 \end{pmatrix} v(t), \quad \mathbf{z}(0) = \mathbf{z}_0, \; \mathbf{z}(T) = \mathbf{z}_T. \tag{7.10}$$

Please note that (7.10) is an equivalent *virtual* system to (7.8) and (7.9), solvable using standard Runge-Kutta methods [19]. The system (7.10) is suitable to generate the timing law, but the particle dynamics (7.6) still need to be included, as described next.

Coupling of non-invertible particle dynamics:

To include the particle dynamics (7.6) in the computation of the timing, we need to couple them with the system (7.10). The typical approach is to rewrite (7.6) as:

$$M(\ddot{\mathbf{p}}(t), \mathbf{p}(t), \mathbf{u}(t)) := m\ddot{\mathbf{p}}(t) - F(\mathbf{p}(t), \mathbf{u}(t)) = 0 \tag{7.11}$$

and make sure that this equation locally admits an inverse function:

$$\mathbf{u}(t) = M^{-1}(\ddot{\mathbf{p}}(t), \mathbf{p}(t)), \tag{7.12}$$

This would allow us to easily compute the location of our traps. However, such inversion is not straightforward for our model, and to the best of our knowledge, standard solution methods do not exist for such cases.

We deal with this challenge by introducing constraints related to our particle dynamics. These will allow us to compute feasible timings while delaying the computation of the exact trap location to a later stage in the process.

More specifically, we introduce auxiliary variables $\zeta_1, \ldots, \zeta_6$ for each trigonometric term in the force F, and we replace $p_i = q_i(\theta)$ for $i \in \{x, y, z\}$ (i.e., particle position exactly matches our reference path):

$$\begin{aligned}
\zeta_1 &= \sin\left(\mathscr{V}_{xr}\sqrt{(u_x - q_x(\theta))^2 + (u_y - q_y(\theta))^2}\right), && (7.13a)\\
\zeta_2 &= \cos\left(\mathscr{V}_z \cdot (u_z - q_z(\theta)\right), && (7.13b)\\
\zeta_3 &= \sin\left(\mathscr{V}_z \cdot (u_z - q_z(\theta)\right), && (7.13c)\\
\zeta_4 &= \cos\left(\mathscr{V}_{zr}\sqrt{(u_x - q_x(\theta))^2 + (u_y - q_y(\theta))^2}\right), && (7.13d)\\
\zeta_5 &= \sin\phi, && (7.13e)\\
\zeta_6 &= \cos\phi. && (7.13f)
\end{aligned}$$

With this, we are now able to formally express the force (7.5) in terms of $\zeta := (\zeta_1, \ldots, \zeta_6)$, i.e.,

$$\tilde{F}(\zeta) := \begin{pmatrix} \mathscr{A}_r\zeta_1\zeta_2\zeta_6 \\ \mathscr{A}_r\zeta_1\zeta_2\zeta_5 \\ \mathscr{A}_z\zeta_4\zeta_3 \end{pmatrix}. \tag{7.14}$$

Using these auxiliary variables, we can now couple the trap-particle dynamics (7.6) with the virtual system (7.10). To this end, similar to (7.11), we define the following constraint along path Q:

$$\widetilde{M}(\theta(t), \dot{\theta}(t), v(t), \zeta(t)) := m\ddot{\mathbf{q}}(\theta(t)) - \tilde{F}(\zeta(t)) = 0. \tag{7.15}$$

Observe that i) we need $\dot{\theta}(t)$ due to (7.7c); and ii) $\ddot{\theta}(t)$ can be replaced by $v(t)$ due to (7.8).

Finally, combining this with the prior first-order differential system allows us to conceptually formulate the problem of computing a minimum-time motion along the path Q

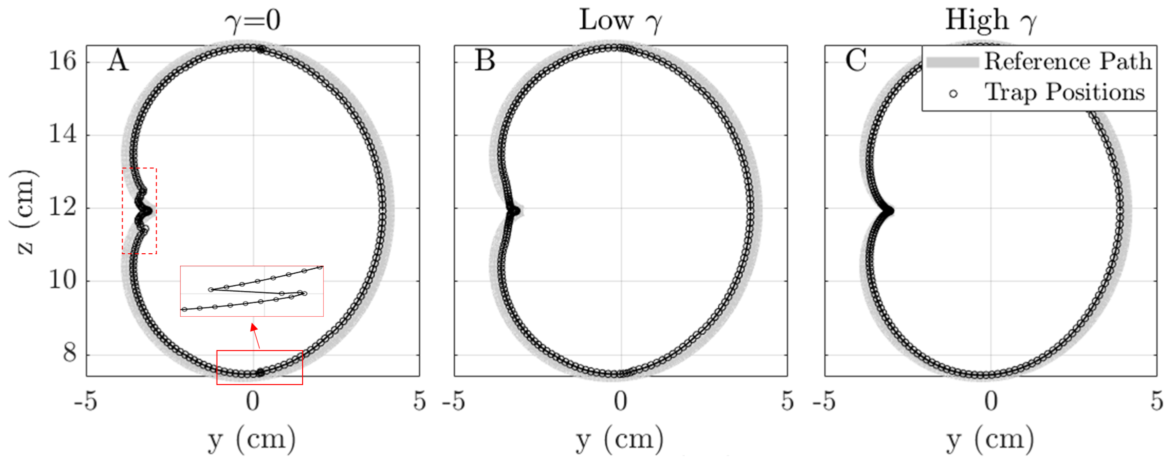


Figure 7.6 Effect of the regularisation parameter γ on the position of the optimised traps.

as:

$$\begin{aligned} &\min_{v,T,\zeta} T + \gamma \int_0^T v(t)^2 \mathrm{d}t \\ &\text{subject to} \\ &\dot{\mathbf{z}}(t) = \begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix} \mathbf{z}(t) + \begin{pmatrix} 0 \\ 1 \end{pmatrix} v(t), \quad \mathbf{z}(0) = \mathbf{z}_0,\ \mathbf{z}(T) = \mathbf{z}_T, \\ &0 = \widetilde{M}(\mathbf{z}(t), v(t), \zeta(t)), \\ &\zeta(t) \in [-1,1]^6. \end{aligned} \tag{7.16}$$

The constraint on $\zeta(t)$ is added since the trigonometric structure of (7.13) is not directly encoded in the OCP, while $\gamma \geq 0$ is a regularisation parameter.

The case $\gamma = 0$ corresponds to strictly minimising time, which typically leads to an aggressive use of the forces around the trap. The example in Figure 7.6(A) shows the trap accelerating the particle before arriving at the corner, then applying aggressive deceleration before it reaches the corner. At the top and bottom of the shape, again the particle is accelerated strongly, then it is suddenly decelerated by the traps placed behind it, in order to move around the curve. This would be the optimum solution in an ideal case (i.e., a device working *exactly* as per our model), but inaccuracies in the real device can make them unstable. Moreover, we observed the overall reductions in rendering time to usually be quite small.

The regularisation ($\gamma > 0$) enables *nearly* time-optimal solutions. More specifically, this is a strictly convex regularisation that penalises high magnitudes of the virtual input $v = \ddot{\theta}$. This is most closely related to the accelerations applied to the particle, hence avoiding aggressive acceleration/deceleration as shown in Figures 7.6(B) and (C). Several heuristics could be proposed to automate selection of a suitable value for γ (e.g., use smallest γ ensuring that the dot product of $\dot{\mathbf{p}}(t)$ and $\dot{\mathbf{u}}(t)$ remains always positive), but this step is considered beyond the scope of this body of work.

Note that the OCP in (7.16) yields the (nearly) optimal timing along the path Q, respecting the particle dynamics and hence solving *Challenge 1*. Moreover, it yields the required forces through $\zeta(t)$. However, it does not give the trap trajectory $\mathbf{u}(t)$ that generates these forces. To obtain the trap trajectory and thus solve *Challenge 2*, we refine (7.16) in the following section.

7.4.3 Computing the Trap Trajectory

The second stage in our approach deals with *Challenge 2*, computing the trap trajectory $\mathbf{u}(t)$ based on the solution of (7.16). In theory, we would use the values for $\zeta_i(t)$, $i = 1,\ldots,6$ and $q_j(\theta(t))$, $j \in \{x,y,z\}$ to determine the particle position and force required at each moment in time, obtaining the required trap position $\mathbf{u}(t)$ by solving (7.13).

In practice, solving $\mathbf{u}(t)$ from (7.13) numerically is not trivial, particularly for values of ζ_i close to ± 1, where numerical instabilities could occur, particularly for ζ_1 and ζ_4.

We attenuate these difficulties by providing further structure to the OCP (7.16), specifically regarding ζ:

$$\zeta_2^2 + \zeta_3^2 = 1, \quad \zeta_5^2 + \zeta_6^2 = 1. \tag{7.17}$$

We also constrain the solvability of (7.13) by using a constant back-off $\varepsilon \in \,]0,1[$ in order to avoid numerical instabilities:

$$-1+\varepsilon \leq \zeta_i \leq 1-\varepsilon, \quad i = 1,\ldots,6. \tag{7.18}$$

For the sake of compact notation, we summarise the above constraints in the following set notation

$$\mathscr{Z} := \left\{ \zeta \in \mathbb{R}^6 \mid (7.17) \text{ and } (7.18) \text{ are satisfied} \right\}. \tag{7.19}$$

The final OCP is then given by

$$\begin{aligned}
&\min_{v,T,\zeta} \; T + \gamma \int_0^T v(t)^2 \mathrm{d}t \\
&\text{subject to} \\
&\quad \dot{\mathbf{z}}(t) = \begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix} \mathbf{z}(t) + \begin{pmatrix} 0 \\ 1 \end{pmatrix} v(t), \quad \mathbf{z}(0) = \mathbf{z}_0,\ \mathbf{z}(T) = \mathbf{z}_T, \\
&\quad 0 = \widetilde{M}(\mathbf{z}(t), v(t), \zeta(t)), \\
&\quad \zeta(t) \in \mathscr{Z}.
\end{aligned} \tag{7.20}$$

Finally, given $\zeta(t) \in \mathscr{Z}$ and $\theta(t)$ from the solution of (7.20), we numerically solve (7.13) for u_i, $i \in \{x, y, z\}$, using Powell's dog leg method [78]. Please note that higher values of ε will limit the magnitudes of the forces exploited by our approach. A simple solution is to perform a linear search over ε, only increasing its value and recomputing the solution if Powell's method fails to converge to a feasible trap location. We provide details on discretising the OCP (7.20) in the continuation.

The optimal control problem (7.20) is written in continuous time. We discretise it as follows. Given a number of traps N, we introduce the time steps $t_k := k\frac{T}{N}$, $k = 0, \ldots, N$, which covers the interval $[0, T]$. We stress that this is done symbolically, without needing to know the value of T in advance. The cost function to minimise becomes

$$T + \frac{\gamma}{N}\mathbf{v}^\top \mathbf{v},$$

where $\mathbf{v} := (v(t_0), \ldots, v(t_{N-1}))$ is an N-dimensional vector. Analogously, we consider the state $\mathbf{z}(t)$ and the auxiliary variable $\zeta(t)$ at $N+1$ respectively N discrete points in time. Hence, the ordinary differential equation in (20) becomes

$$\mathbf{z}(t_{k+1}) = RK4(\mathbf{z}(t_k), v(t_k)), \quad \mathbf{z}(t_0) = \mathbf{z}_0,\ \mathbf{z}(t_N) = \mathbf{z}_T,$$

where $RK4$ is the Runge-Kutta 4 integration scheme and $\mathbf{z}_0$ and $\mathbf{z}_T$ are the same as in (7.20). Similarly, the equality constraint covering the trap-particle dynamics in (20) becomes

$$0 = \widetilde{M}(\mathbf{z}(t_k), v(t_k), \zeta(t_k)), \quad k = 0, \ldots, N-1.$$

Likewise, the constraints on $\zeta(t)$ translate to

$$\zeta(t_k) \in \mathscr{Z}, \quad k = 0, \ldots, N-1.$$

This constitutes a (large) nonlinear program (NLP), where the variables T as well as $v(t_k)$ and $\zeta(t_k)$ for $k = 0, \ldots, N-1$, and $\mathbf{z}(t_k)$ for $k = 0, \ldots, N$ are all optimisation variables $(1 + N + 6N + 3(N+1)$ optimisation variables in total).

7.4.4 Ramp-Up, Ramp-Down, and Rest-to-Rest

A further unsolved problem for acoustic levitation displays is the computation of ramp-up trajectories. That is, even if a physically feasible periodic trap trajectory is available, it is only feasible if the particle can be brought to a set location on the path with the correct velocity vector. Since a particle placed in the levitator is usually at rest, this is not trivial.

The problem is commonly addressed as follows. First, place the particle at a position on the path at rest. Play back a much slower version of the trap trajectory (i.e., update the traps less frequently) to slowly move the particle along the path. Increase the update rate until the final update rate is reached. This manual process is both time-consuming and unreliable.

Our structured approach naturally allows the computation of physically feasible and nearly time-optimal ramp-up trajectories. To achieve this, we merely need to adjust the conditions (7.9) to

$$\theta(0) = \theta_0, \quad \theta(2T/3) = \theta_f, \quad \theta(T) = 2\theta_f, \tag{7.21a}$$
$$\dot{\theta}(0) = 0, \quad \dot{\theta}(2T/3) = \dot{\theta}(T). \tag{7.21b}$$

This causes the particle to start from rest and traverse the path twice, where the first round is given double the time of the second round, which closely corresponds to the periodic solution. From here, we can use the periodic trap trajectory indefinitely. Recall that the total time T does not need to be known, but is optimised (and is of course longer for the ramp-up trajectory than for the periodic movement).

In a similar fashion, to ramp-down the particle from the periodic solution to rest, we adjust the conditions (7.9) to

$$\theta(0) = \theta_0, \quad \theta(T/3) = \theta_f, \quad \theta(T) = 2\theta_f, \tag{7.22a}$$
$$\dot{\theta}(0) = \dot{\theta}(T/3), \quad \dot{\theta}(T) = 0, \tag{7.22b}$$

and for a rest-to-rest movement, to

$$\theta(0) = \theta_0, \quad \theta(T) = \theta_f, \quad \dot{\theta}(0) = 0, \quad \dot{\theta}(T) = 0. \tag{7.23}$$

7.5 Evaluation

This section evaluates the capability of our algorithm to support content creation by generating physically feasible trap trajectories given only a reference path (i.e., a shape) as an input. We select a range of shapes and analyse the effects each step in our approach has on the final achievable size, rendering frequency, and reconstruction error, for each shape. We also inspect how these steps influence the presence of visual distortions in the end result. Finally, we analyse the resulting acceleration profiles.

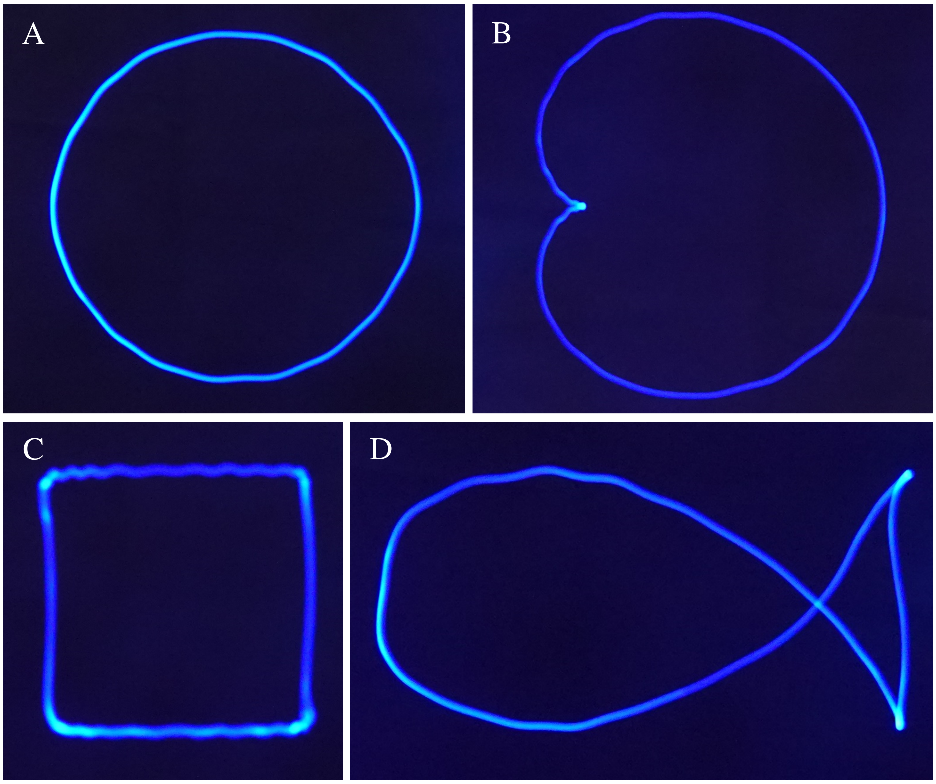


Figure 7.7 Test shapes used for evaluation, as rendered by our *OptiTrap* approach. The circle (A) provides a trivial timing case. The cardioid (B) and squircle (C) represent simple shapes featuring sharp corners and straight lines. The fish (D) is used as an example of composite shape featuring corners, straight lines, and curves.

7.5.1 Test Shapes

We used four test shapes for the evaluation: a circle, a cardioid, a squircle, and a fish, all shown in Figure 7.7.

The circle is selected as a trivial case in terms of timing. It can be optimally sampled by using a constant angular speed (i.e., constant acceleration), and is there to test whether our solution converges towards such optimum solutions. The cardioid and squircle represent simple shapes featuring sharp corners (cardioid) and straight lines (squircle), both of them challenging features. Finally, the fish is selected as an example composite shape, featuring all elements (i.e., corners, straight lines, and curves). For $\theta \in [0, 2\pi]$, we define the test trajectories as follows:

$$\mathbf{q}_{circle}(\theta) = \begin{pmatrix} r \cdot sin(\theta) \\ r \cdot cos(\theta) + \frac{h}{2} \end{pmatrix}, \tag{7.24}$$

$$\mathbf{q}_{cardioid}(\theta) = \begin{pmatrix} a \cdot cos(\theta) \cdot (1 + cos(\theta)) \\ a \cdot sin(\theta) \cdot (1 + sin(\theta)) + \frac{h}{2} \end{pmatrix}, \tag{7.25}$$

$$\mathbf{q}_{squircle}(\theta) = \begin{pmatrix} b \cdot cos(\theta) \\ \frac{b \cdot sin(\theta)}{\sqrt{1 - s \cdot cos^2(\theta)}} + \frac{h}{2} \end{pmatrix}, \tag{7.26}$$

$$\mathbf{q}_{fish}(\theta) = \begin{pmatrix} c \cdot \left(cos(\theta) - \frac{sin^2(\theta)}{\sqrt{2}} \right) \\ c \cdot sin(\theta) \cdot cos(\theta) + \frac{h}{2} \end{pmatrix}, \tag{7.27}$$

where $s = 0.99$ is the squareness factor, $h = 23.9$cm is the total height of the levitation volume, and r, a, b, and c are scaling parameters. (Note that we omit the second horizontal component, which always equals zero.)

As we see above, all of these shapes are parameterised by low-frequency sinusoids, ensuring the path parameter θ progresses slowly in areas of high curvature. This provides a good starting point for our baseline comparison that we explain in Section 7.5.2. While our approach works for content in 3D, we perform the evaluation on shapes in a 2D plane of the 3D space to facilitate the analysis of accelerations in Section 7.5.6, which we will perform in terms of horizontal and vertical accelerations (i.e., the independent factors given our axis-symmetric model of forces), allowing us to validate *OptiTrap*'s awareness of particle directions and how close it can get to the maximum accelerations allowed by the dynamics of acoustic traps, as discussed in Section 7.4.1.

Notice in Figure 7.7, that due to the timing differences, different shape elements exhibit different levels of brightness. To obtain homogeneous brightness along the shape, please refer to the solution proposed by [53], where the particle illumination is adjusted to the particle speed.

7.5.2 Conditions Compared

We compare three approaches to rendering levitated shapes, where each approach subsequently addresses one of the challenges introduced in Section 7.2. This will help us assess the impact that each challenge has on the final results obtained.

The first condition is a straight-forward *Baseline*, with homogeneous sampling of the path parameter, which matches the example strategy shown in Figure 7.3(B), and placing traps where the particle should be. The *Baseline* still does not address any of the challenges identified (i.e., optimum timing or trap placement), but it matches approaches used in previous works [53, 115, 38] and will illustrate their dependence on the specific shape and initial parametrisation used.

The second condition, *OCP_Timing*, makes use of our *OptiTrap* approach to compute optimum and feasible timing (see Subsection 7.4.2), but still ignores *Challenge 2*, assuming that particles match trap location (i.e., it skips Subsection 7.4.2). The third condition is the full *OptiTrap* approach, which considers feasibility and trap-particle dynamics and deals with both the timing and trap placement challenges.

These conditions are used in a range of comparisons involving size, frequency, reconstruction error as well as comparative analysis of the effects of each condition on the resulting particle motion, detailed in the following sections.

7.5.3 Maximum Achievable Sizes

This section focuses on the maximum sizes that can be achieved for each test shape and condition, while retaining an overall rendering time of 100ms[1] (i.e., PoV threshold). For each of these cases, we report the maximum achievable size and reliability, which were determined as follows.

Maximum size is reported in terms of shape width and in terms of meters of *content per second* rendered [115], and their determination was connected to the feasibility of the trap trajectories, particularly for the *Baseline* condition.

More specifically, we determined maximum feasible sizes for the *Baseline* condition by conducting an iterative search. That is, we increased the size at each step and tested the reliability of the resulting shape. We considered the shape feasible if it could be successfully rendered at least 9 out of 10 times in the actual levitator. On a success, we would increase size by increasing the shape width by half a centimetre. On a failure, we would perform a binary search between the smallest size failure and the largest size success, stopping after two consecutive failures and reporting the largest successful size. This illustrates the kind of trial and error that a content designer would need to go through without our approach and it was the most time consuming part of this evaluation.

For the other two conditions (i.e., *OCP_Timing* and *OptiTrap*), maximum sizes could be determined without needing to validate their feasibility in the final device. Again, we iteratively increased the shape size using the same search criteria as before (i.e., 5mm increases, binary search). We assumed any result provided would be feasible (which is a reasonable assumption; see below) and checked the resulting total rendering time, increasing the size if the time was still less than 100ms. At each step, a linear search was used, iteratively increasing ε, until feasible trap locations could be found (i.e., equation (7.13) could be solved without numerical instabilities). Adjusting the value of the regularisation parameter γ was

[1]The circle rendered at 100ms would exceed the size of our device's working volume. Thus, this shape was rendered at 67ms.

done by visually inspecting the resulting trap trajectories, until no discontinuities could be observed (see Figure 7.6). Please note that this is a simple and quick task, which, unlike the *Baseline* condition, does not involve actual testing on the levitation device and could even be automated. All solutions provided can be assumed feasible, and the designer only needs to choose the one that better fits their needs.

Once the maximum achievable sizes were determined, these were tested in the actual device. We determined the feasibility of each shape and condition by conducting ten tests and reporting the number of cases where the particle succeeded to reveal the shape. This included the particle accelerating from rest, traversing/revealing the shape for 6 seconds and returning to rest.

Table 7.2 summarises the results achieved for each test shape and condition. First of all, it is worth noting that all trials were successful for *OptiTrap* and *OCP_Timing* approaches, confirming our assumption that the resulting paths are indeed feasible and underpinning *OptiTrap*'s ability to avoid trial and error on the actual device during content creation.

The results and relative improvements in terms of size vary according to the particular condition and shape considered. For instance, it is interesting to see that all conditions yield similar final sizes for the circle, showing that *OptiTrap* (and *OCP_Timing*) indeed converge towards optimum solutions.

More complex shapes where the *Baseline* parametrisation is not optimal (i.e., cardioid, squircle, and fish), show increases in size when using *OCP_Timing* and *OptiTrap* (see Figure 7.8). More interestingly, the increases in size vary greatly between shapes, showing increases of around 12% for the fish and cardioid, and up to 562% for the squircle. This is the result of the explicit parametrisation used, with reduced speeds at corners in the fish and cardioid cases, but not in the case of the squircle.

It is also interesting to see that *OCP_Timing* and *OptiTrap* maintain high values of *content per second* rendered, independently of the shape. That is, while the performance of the *Baseline* approach is heavily determined by the specific shape (and parametrisation) used, the OCP-based approaches (*OCP_Timing* and *OptiTrap*) yield results with consistent *content per second*, determined by the capabilities of the device (but not so much by the specific shape). Finally, please note that no changes in terms of maximum size can be observed between *OCP_Timing* and *OptiTrap*.

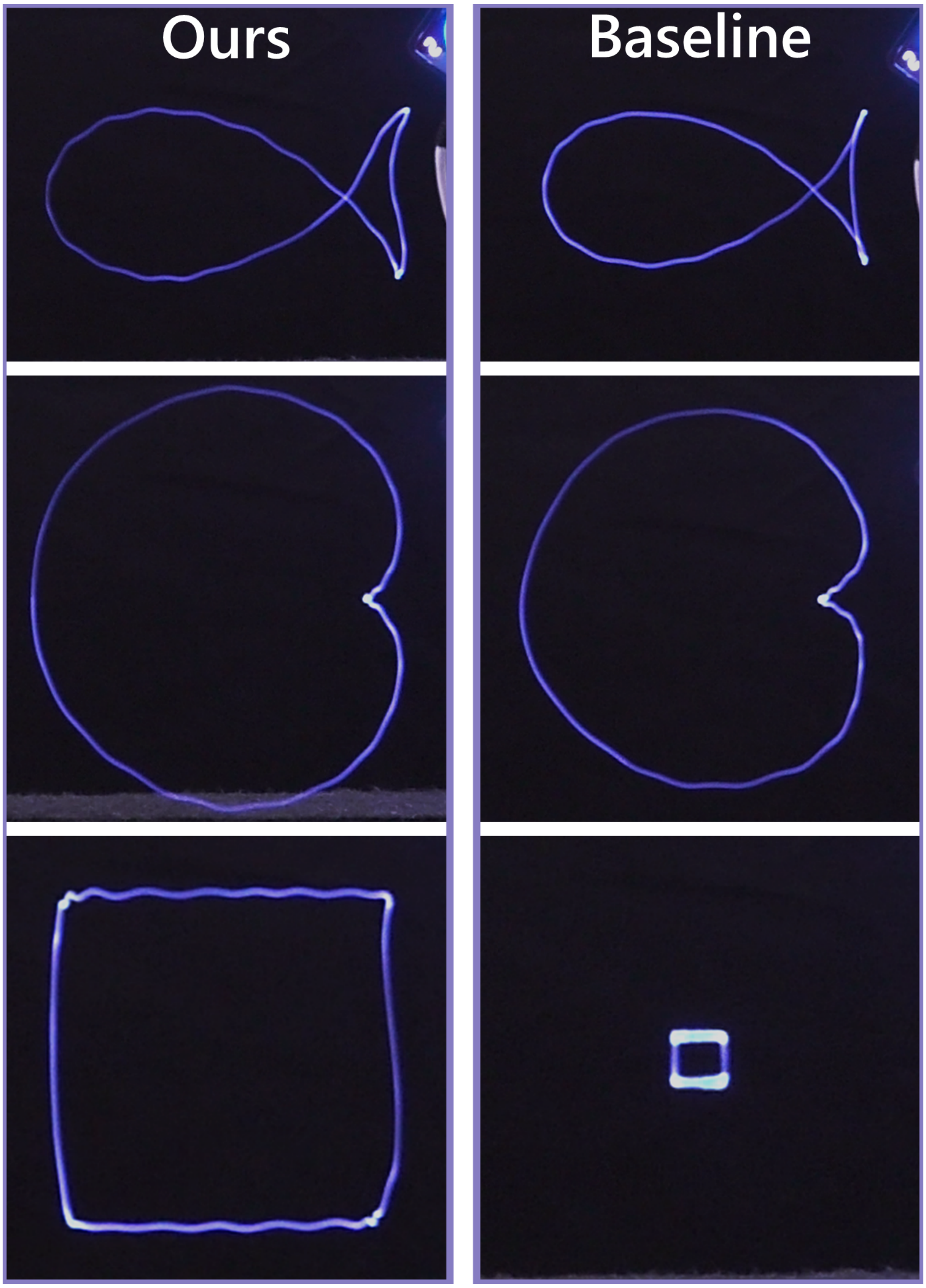


Figure 7.8 Visualisation of the results for the evaluation of the maximum achievable shape sizes at 10Hz, for the cardioid, squircle, and fish. The shapes in the first column were generated using our approach, whereas the second column was generated using the *Baseline*. Note that the most striking size increase (562%) was obtained for the squircle, the shape that features the highest number of sharp corners.

Table 7.2 Maximum shape width and meters of content per second achieved with the *Baseline*, *OCP_Timing*, and *OptiTrap* approach at constant rendering frequency. Successful trials out of 10.

		Baseline			*OCP_Timing*			*OptiTrap*			Increase in
Shape	Freq. (Hz)	Width (cm)	Content per Second (m)	Success Rate	Width (cm)	Content per Second (m)	Success Rate	Width (cm)	Content per Second (m)	Success Rate	Width w.r.t. the *Baseline*
Circle	15	7.00	3.30	10/10	7.00	3.30	10/10	7.00	3.30	10/10	0%
Cardioid	10	8.05	2.48	10/10	9.09	2.80	10/10	9.09	2.80	10/10	12.9%
Squircle	10	0.80	0.29	9/10	5.30	1.90	10/10	5.30	1.90	10/10	562.5%
Fish	10	7.77	2.42	10/10	8.76	2.75	10/10	8.76	2.75	10/10	12.7%

Table 7.3 Maximum rendering frequencies achieved with the *Baseline*, *OCP_Timing*, and *OptiTrap* approach, for shapes of equal size. Successful trials out of 10.

	Baseline		*OCP_Timing*		*OptiTrap*	
Shape	Freq. (Hz)	Success Rate	Freq. (Hz)	Success Rate	Freq. (Hz)	Success Rate
Circle	15	10/10	15	10/10	15	10/10
Cardioid	8	10/10	10	10/10	10	10/10
Squircle	4	10/10	10	10/10	10	10/10
Fish	9	9/10	10	10/10	10	10/10

Table 7.4 RMSE and Path-normalised (PN) RMSE with respect to the total path length of each test shape, for the *Baseline*, *OCP_Timing*, and *OptiTrap* approach, at constant rendering frequency.

	Baseline		*OCP_Timing*		*OptiTrap*	
Shape	RMSE (cm)	PN RMSE	RMSE (cm)	PN RMSE	RMSE (cm)	PN RMSE
Circle	0.196	0.893	0.146	0.666	0.114	0.521
Cardioid	0.142	0.575	0.153	0.545	0.120	0.429
Squircle	0.056	1.956	0.0824	0.434	0.080	0.421
Fish	0.111	0.458	0.147	0.536	0.0581	0.212

7.5.4 Maximum Rendering Frequencies

In this second evaluation, we assessed the effect of the timing strategy on the maximum achievable rendering frequencies. To do this, we selected the maximum achievable sizes obtained in the prior evaluation for each shape. We then reproduced such sizes with the *Baseline* approach, searching for the maximum frequency at which this approach could reliably render the shape (using the same searching and acceptance criteria as above).

That is, while we knew *OCP_Timing* and *OptiTrap* could provide 10Hz for these shapes and sizes, we wanted to determine the maximum frequency at which the *Baseline* would render them, as to characterise the benefits provided by optimising the timing.

The results are summarised in Table 7.3. As expected, no changes are produced for the circle. However, there is a 25% increase for the cardioid, 150% for the squircle, and 11% increase for the fish using the *OptiTrap* method. Again, please note that no differences can be observed in terms of maximum frequency between *OCP_Timing* and *OptiTrap*.

Figure 7.9 shows a plot of the particle trajectories, measured using the motion capturing system for this as well as for the previous evaluation of maximum achievable sizes, for the *Baseline* and the *OptiTrap* methods.

7.5.5 Reconstruction Accuracy

To evaluate the reconstruction accuracy, we recorded each of the trials in our maximum size evaluations (see Section 7.5.3) using an OptiTrack camera system. We then computed the RMSE between the recorded data and the intended target shape, taking 2s of shape rendering into account (exclusively during the cyclic part, ignoring ramp-up/ramp-down). For a fair comparison across trials, we normalise the RMSE with respect to the traversed paths, by dividing the RMSE by the total path length of each individual test shape. The results are summarised in Table 7.4.

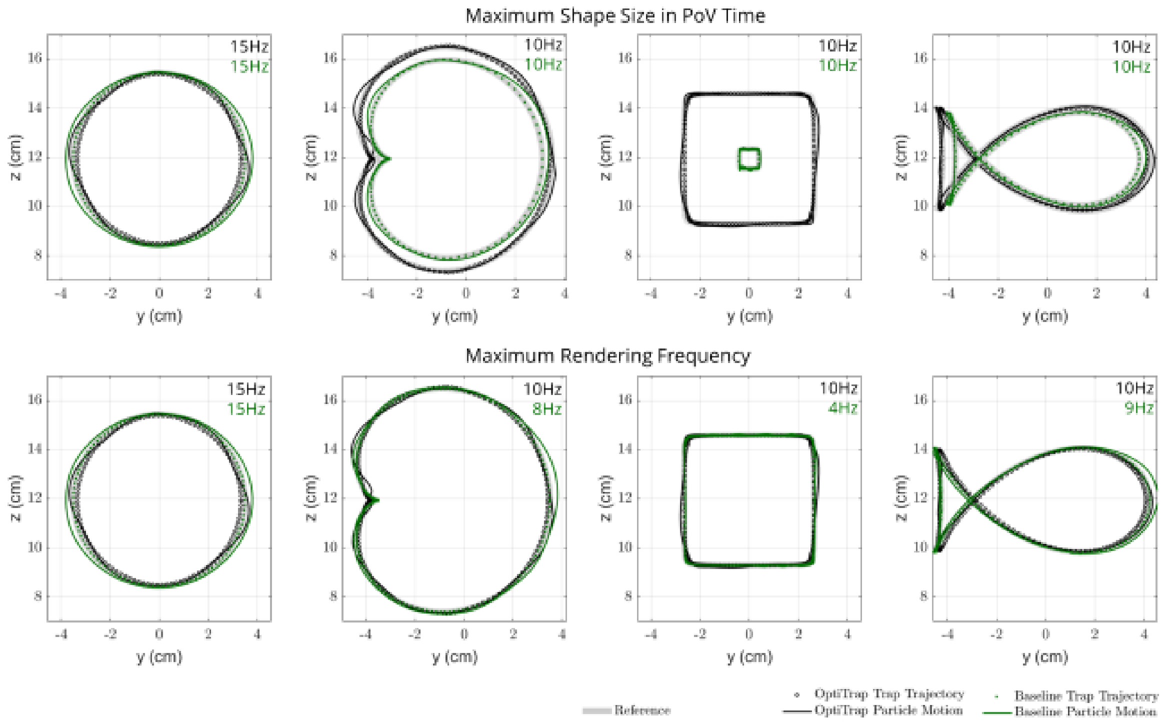


Figure 7.9 The particle trajectories, as captured by the motion tracking system, for the *Baseline* (green) and *OptiTrap* (black) method. The dotted lines represent the trap distribution for the respective method. The reference trajectories are shown in light grey.The top row shows the trajectories measured for the evaluation of the maximum achievable sizes, while the bottom row for the maximum rendering frequencies.

On average, both *OCP_Timing* and *OptiTrap* provide better results in terms of accuracy, when compared to the *Baseline*. It is particularly worth noting that the accuracy results, in terms of the raw RMSE for *OptiTrap* and *OCP_Timing*, are better for the cardioid when compared to the *Baseline*, even though a larger shape is being rendered. While the raw RMSE of *OCP_Timing* and *OptiTrap* is slightly larger for the squircle, we need to take into account that the size rendered by *OCP_Timing* and *OptiTrap* is almost 6 times larger. Comparing accuracy in terms of normalised RMSE shows consistent increases of accuracy for *OptiTrap* compared to the *Baseline*, with overall decreases in the RMSE of 41.7% (circle), 25.4% (cardioid), 78.5% (squircle), and 53.8% (fish). *OCP_Timing* shows a decrease of 25.4%, 5.14%, and 77.8% of the normalised RMSE for the circle, cardioid, and squircle, with the exception of the fish, where the normalised RMSE increased by 17.1%, when compared to the *Baseline*.

Comparing the reconstruction accuracy of *OCP_Timing* and *OptiTrap* highlights the relevance of considering trap dynamics to determine trap locations (i.e., Section 7.4.3). *OptiTrap* consistently provides smaller RMSE than *OCP_Timing*, as a result of considering

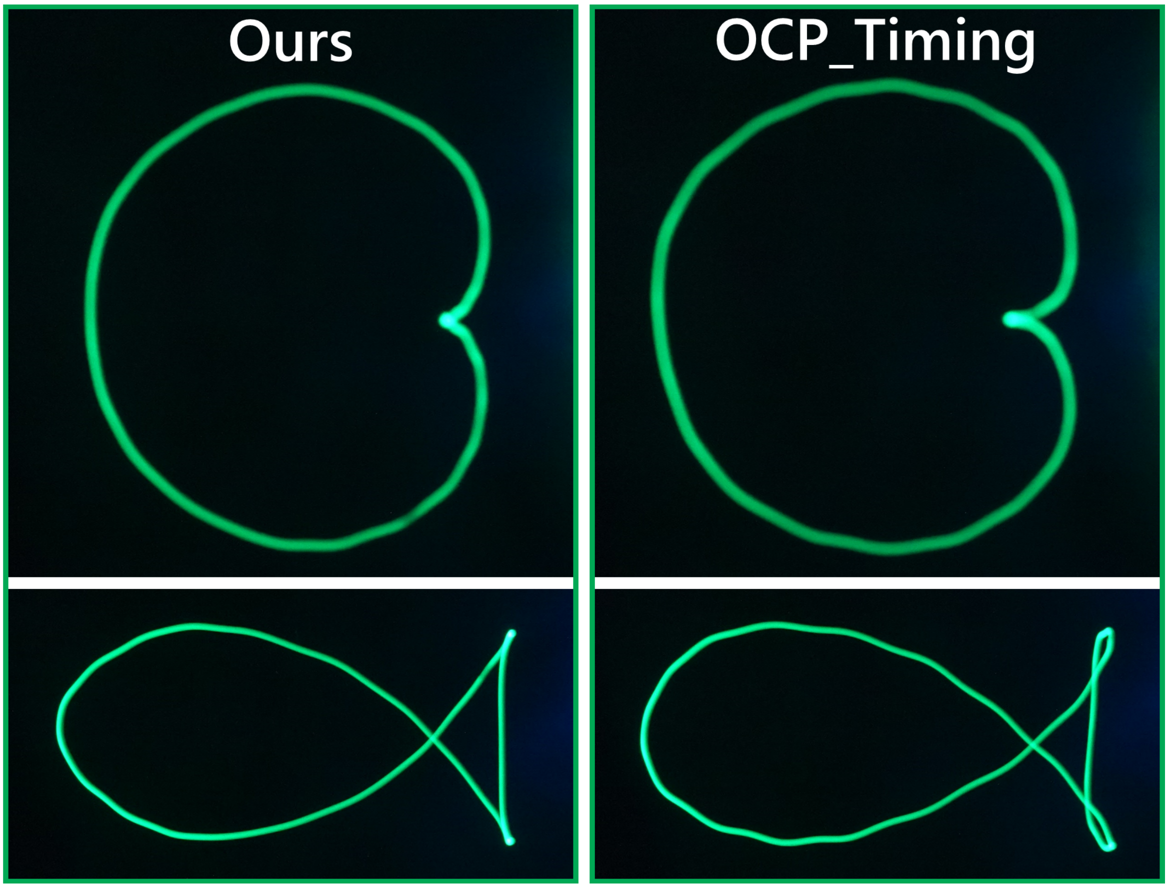


Figure 7.10 Visual comparison between shapes rendered with traps located according to particle dynamics using *OptiTrap* (left) and traps placed along the reference path using *OCP_Timing* (right) for the cardioid (top) and the fish (bottom) test shapes. Please note undesired increases in size and error in sharp features, such as corners.

and accounting for the trap-to-particle displacements required to apply specific accelerations. We obtain a 21.8%, 21.3%, 2.9% and 60.5% decrease in the normalised RMSE, for the circle, cardioid, squircle, and fish, respectively. Such differences are visually illustrated in Figure 7.10, showing the effects on the cardioid and fish, for the *OCP_Timing* and *OptiTrap* approaches. It is worth noting how placing the traps along the reference path results in the cardioid being horizontally stretched, as the particle needs to retain larger distances to the trap to keep the required acceleration (horizontal forces are weaker than vertical ones). This also results in overshooting of the particle at corner locations, which can be easily observed at the corner of the cardioid and in the fins of the fish. For completeness, the visual comparison for the remaining two test shapes is provided in Figure 7.11.

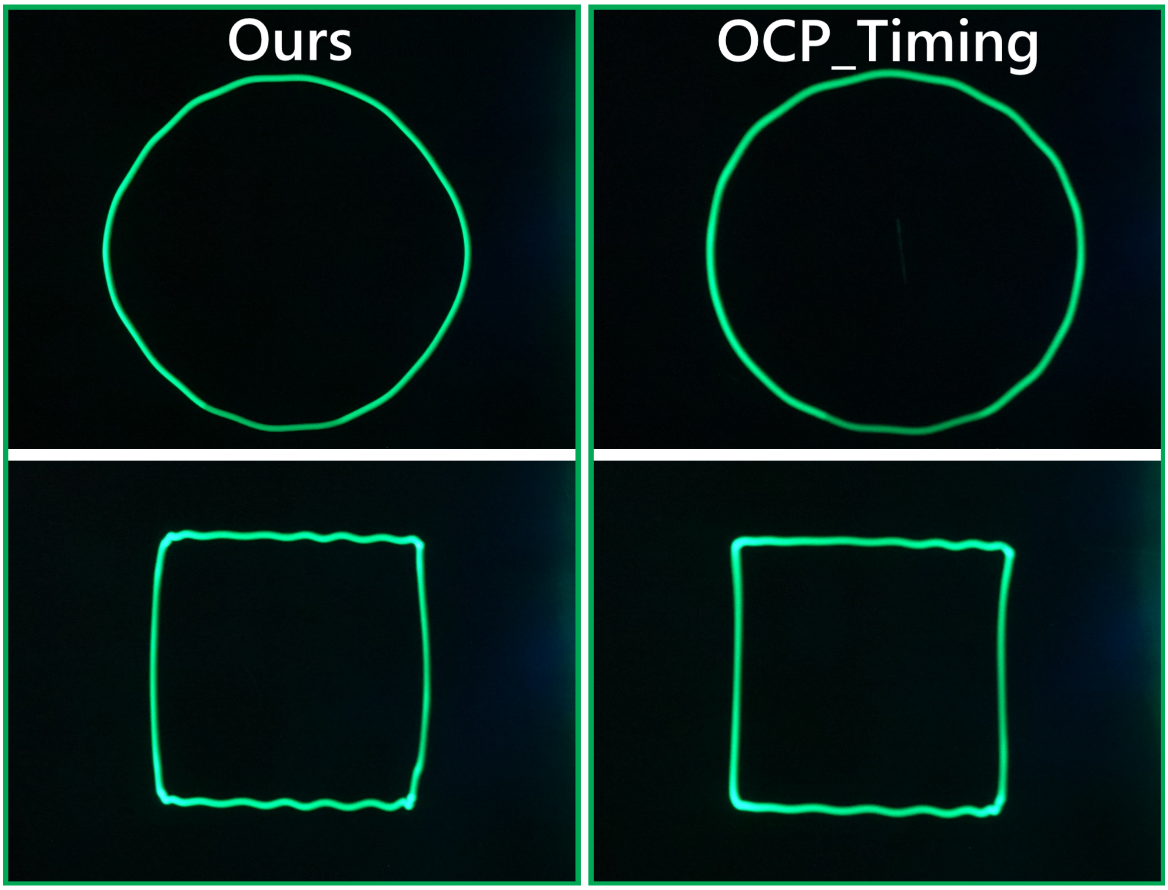


Figure 7.11 Visual comparison between shapes rendered with traps located according to particle dynamics using *OptiTrap* (left) and traps placed along the reference path using *OCP_Timing* (right) for the circle (top) and squircle (bottom) test shapes.

7.5.6 Analysis of Acceleration profiles

Finally, we examined the acceleration profiles produced by *OptiTrap* and how these differ from the pre-determined acceleration profiles of the *Baseline*. Please note that we only compare and discuss *OptiTrap* and *Baseline* in Figure 7.12, and only for the fish shape. *OCP_Timing* is not included, as it results in similar acceleration profiles as *OptiTrap*. Only the fish is discussed, as it already allows us to describe the key observations that can be derived from our analysis. For completeness, the acceleration profiles for all remaining test shapes are included in Figures 7.12,7.13,7.14,7.15.

As introduced above, Figure 7.12 shows the particle accelerations and speeds of a particle revealing a fish shape of maximum size (i.e., a width of 8.76cm), rendered over 100ms using the *OptiTrap* and *Baseline* approaches. It is worth noting that while *OptiTrap* succeeded in rendering this shape, *Baseline* did not (the maximum width for *Baseline* was 7.77cm).

A first interesting observation is that although *OptiTrap* provides lower total accelerations than the *Baseline* during some parts of the path (see Figure 7.12(A)), it still manages to reveal the shape in the same time. The key observation here is that the regions where the total acceleration is lower for *OptiTrap* match with the parts of the path where the horizontal acceleration is very close to its maximum; for example, note the flat regions in Figure 7.12(B) of around $\pm 300 m/s^2$). This is an example of *OptiTrap*'s awareness of the dynamics and capabilities of the actual device, limiting the acceleration applied as to retain the feasibility of the path.

Second, it is worth noting that maximum horizontal accelerations are significantly smaller than vertical accelerations. As such, it does make sense for horizontal displacements to become the limiting factor. In any case, neither acceleration exceeds its respective maximum value. The *OptiTrap* approach recovers any missing time by better exploiting areas where acceleration is unnecessarily small. It is also worth noting that the value of the horizontal and vertical accelerations are not simply being capped to a maximum acceleration value per direction (e.g., simultaneously maxing out at $\pm 300 m/s^2$ and $\pm 600 m/s^2$ in the horizontal and vertical directions), as such cases are not feasible according to the dynamics of our acoustic traps.

Third, it is interesting to note that the final acceleration profile is complex, retaining little resemblance with the initial parametrisation used. This is not exclusive for this shape and can be observed even in relatively simple shapes, such as the cardioid in Figure 7.3(D) or the remaining shapes, provided in the Figures 7.12,7.13,7.14,7.15.

The complexity of these profiles is even more striking if we look at the final speeds resulting from both approaches (see Figure 7.12(D)). Even if the acceleration profiles of *Baseline* and *OptiTrap* are very different, their velocity profiles show only relatively subtle differences. But even if these differences are subtle, they mark the difference between a feasible path (i.e., *OptiTrap*) and a failed one (i.e., *Baseline*). These complex yet subtle differences also illustrate how it is simply not sensible to expect content designers to deal with such complexity, and how *OptiTrap* is a necessary tool to enable effective exploitation of PoV levitated content.

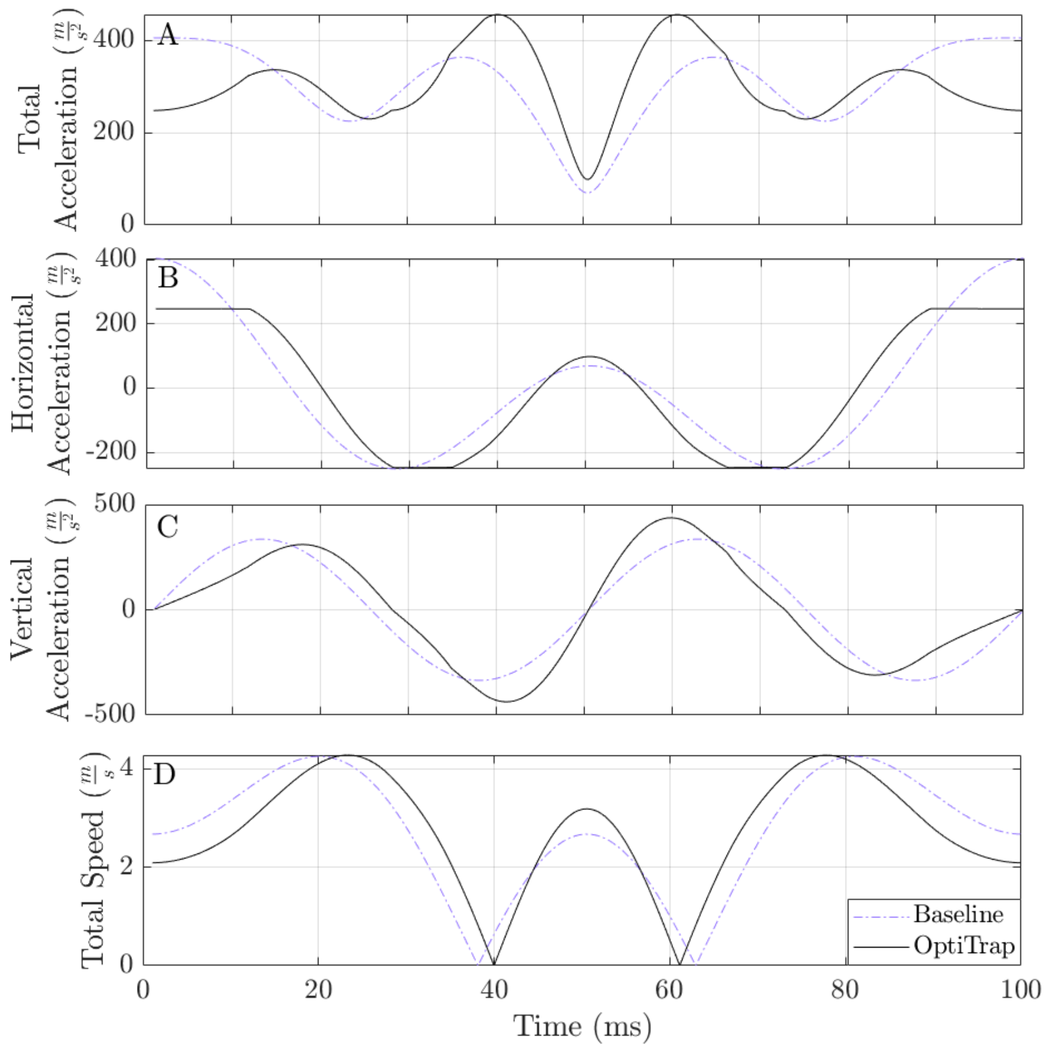


Figure 7.12 Particle acceleration and speed for the fish rendered at 10 Hz using *OptiTrap* (solid black) and the *Baseline* (dash-dotted purple). While the speed profiles (D) of both trap trajectories are similar, the trap trajectory generated by our approach is feasible, whereas the one generated by the *Baseline* is not. The main reason is that the *Baseline* exceeds the feasible horizontal acceleration, while our approach caps it to feasible values (B). Our approach compensates by using higher (available) vertical acceleration (C). Note that the times where the *Baseline* applies a higher total acceleration (A) than our approach are those where our approach respects the constraints on the feasible horizontal acceleration (B).

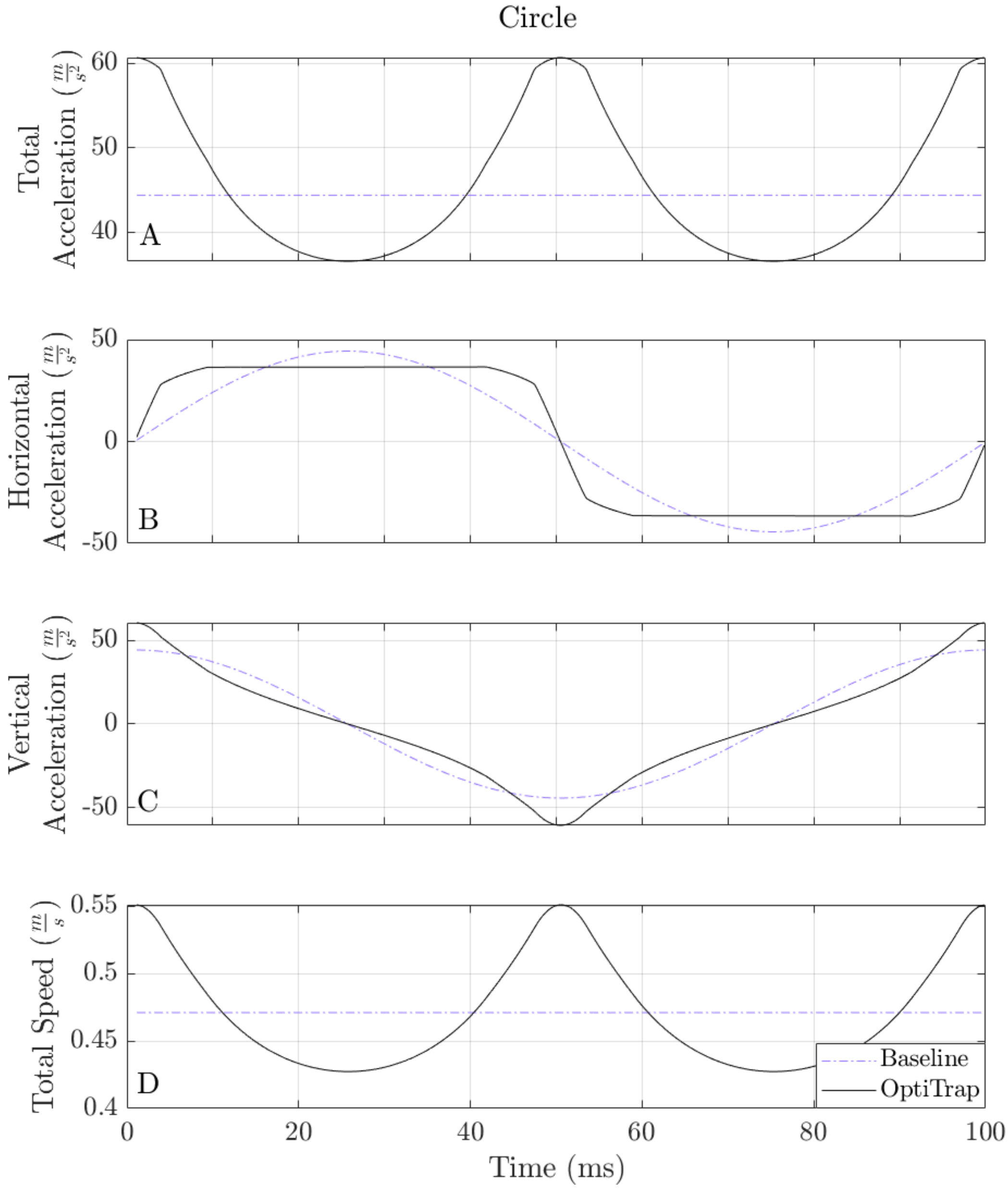


Figure 7.13 Particle acceleration and speed for the circle rendered at 15Hz using *OptiTrap* (solid black) and the *Baseline* (dash-dotted purple).

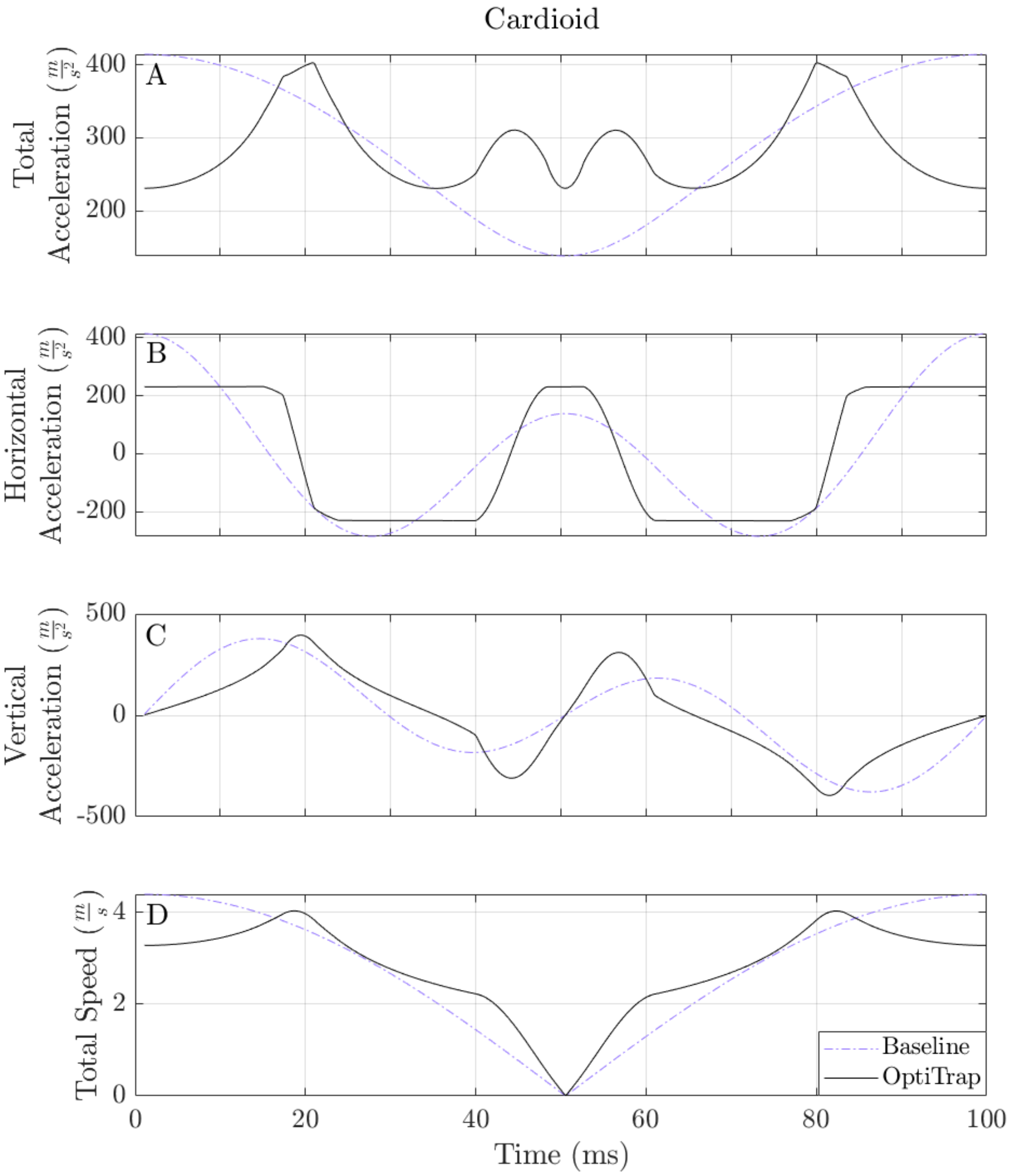


Figure 7.14 Particle acceleration and speed for the cardioid rendered at 10Hz using *OptiTrap* (solid black) and the *Baseline* (dash-dotted purple).

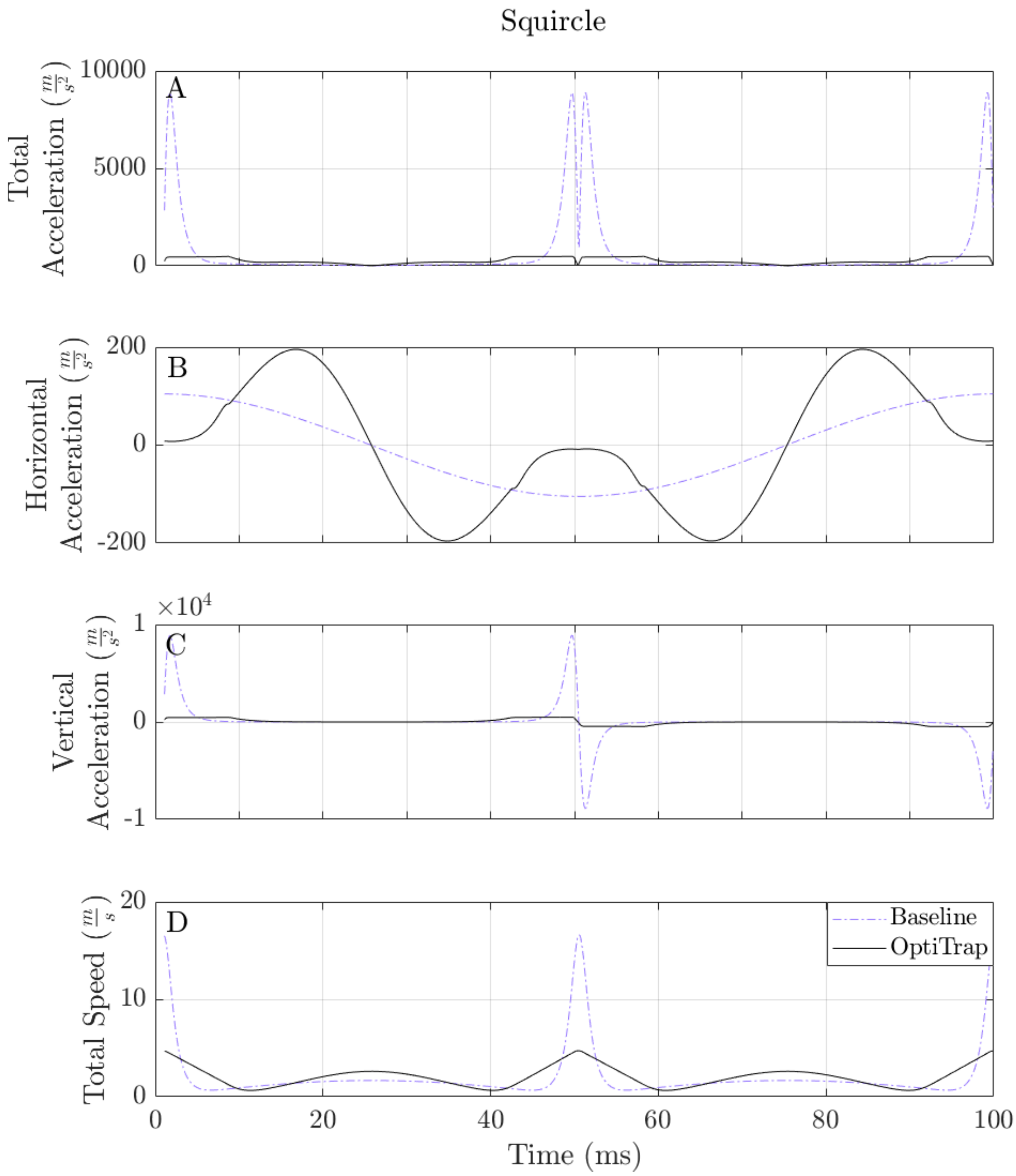


Figure 7.15 Particle acceleration and speed for the squircle rendered at 10Hz using *OptiTrap* (solid black) and the *Baseline* (dash-dotted purple).

7.6 Discussion

This chapter presented *OptiTrap*, an automated approach for optimising timings and trap placements, as to achieve feasible target shapes. We believe this is a particularly relevant step for the adoption of levitation PoV displays, as it allows the content creator to focus on the shapes to present, with feasible solutions being computed automatically, while making effective usage of the capabilities of the device. As such, we hope *OptiTrap* to become an instrumental tool in helping explore the actual potential of these displays.

However, *OptiTrap* is far from a complete content creation tool. Such a tool should consider the artist's workflows and practices. Similarly, testing and identifying most useful heuristics to tune our approach (e.g., the regularisation) or visualisation tools identifying tricky parts of the shapes (i.e., requiring high accelerations) should be included as a part of this process. Our goal is simply to provide the base approach enabling this kind of tools.

Even this base approach can be extended in a variety of ways. Our levitator prototype is built from off-the-shelf hardware, and is still subject to inaccuracies that result in distortions in the sound-fields generated [40]. As such, more accurate levitation hardware or, alternatively, a model reflecting the dynamics of the system in a more accurate manner would be the most obvious pathway to improve our approach. It is worth noting that both factors should be advanced jointly. A more accurate model could also be less numerically stable, potentially leading to worse results if the hardware is not accurate enough.

The high update rates of 10kHz required by the levitator and the millisecond delays introduced by optical tracking systems indicate that closed-loop approaches can be both promising and challenging avenues to explore. Assuming a tracking device synchronised with the levitation device (as to map current trap locations with real particle positions), *OptiTrap* could be combined with learning-based approaches. These learning-based approaches will require example paths (i.e., with initial timing and trap placement), and their achievable complexity and convergence will be limited by the examples provided. In such cases, *OptiTrap* can be used to always generate feasible initial trajectories (i.e., to avoid system restarts on failure). Thus learning-based approaches could be used to further refine *OptiTrap* beyond our current model, as to account for device inaccuracies such as those discussed by [40].

However, higher gains can be obtained from more radical changes in the approach. For instance, our approach optimises the timing and trap placement, but it does not modify the target shapes. As illustrated in Figure 7.12, slight changes in speed have significant effects on the acceleration profiles (and feasibility) of the shapes. This is even more prominent for position, where small changes can heavily influence the acceleration and feasibility of target shapes. As such, approaches exploiting subtle modifications to the shape could lead to

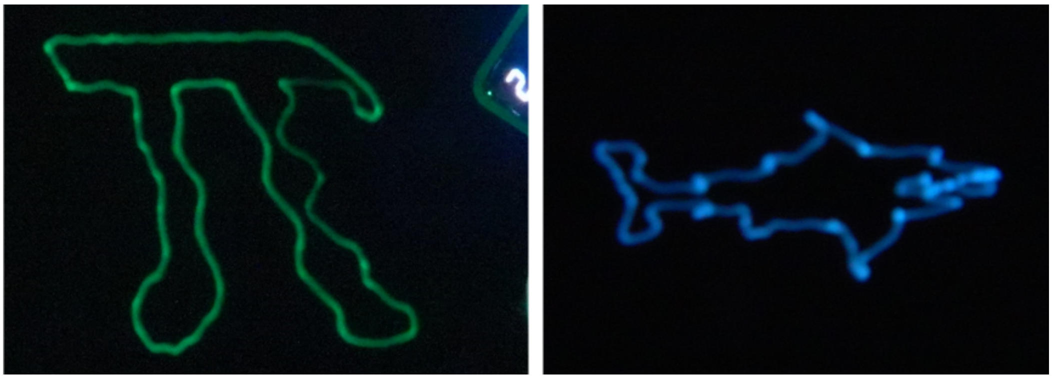


Figure 7.16 Example of complex shapes rendered on an acoustic levitation display for the first time using *OptiTrap* method. However, we can notice how the system inaccuracies have a significant impact on the rendering quality.

significant gains in rendering performance. For example, in Figure 7.16 we can see how the different inaccuracies heavily impact the rendering of the letter π and the shark. Such cases could potentially greatly benefit from an adjustment of the reference path.

Another interesting possibility would be extending our approach to use several particles. As shown in [115], while the use of several particles does not increase the overall power that can be leveraged, it does allow for increased flexibility. That is, the intensity/forces of each trap can be individually and dynamically adjusted, as to match the needs of the region of the path that each particle is revealing. Also, particles can each be rendering specific independent features, so the content is not limited to a single connected path, and the particles do not waste time/accelerations traversing parts of the path that will not be illuminated (i.e., visible). This approach, however, entails significant challenges. The first obvious challenge is the reliability of the intensity control of the traps. [115] demonstrate accurate control of the stiffness at the centre of the traps, but the effects of multiple (interfering) traps in each trap's topology (i.e., how forces distribute around the trap) is yet to be studied. A second challenge is that each particle is not forced to traverse the path at the same speed/rates, with such independent timing progression becoming an additional degree of freedom to account for.

Finally, further extensions to our work can come from its application to domains other than PoV displays. An obvious next step would be to adapt *OptiTrap* to photophoretic displays, which trap particles using optical traps instead of acoustic traps [132, 71]. This would involve including a model of the dynamics of such optical traps, but it can also involve further challenges, such as modelling the response times of galvanometers and LC panels involved in creating the trap.

Our approach can also be adapted to applications requiring objects to be transported quickly and accurately. For example, contactless transportation of matter has a wealth of applications in areas such as the study of physical phenomena, biochemical processes, materials processing, or pharmaceutics [34]. Our method can help solving such problems by directly computing a rest-to-rest solution of the matter to be transported, given an estimation of the mass of that matter, the acoustic force acting on the matter, and a path from start to target position.

7.7 Conclusion

In this chapter we proposed the first structured numerical approach to compute trap trajectories for acoustic levitation displays. *OptiTrap* automatically computes physically feasible and nearly time-optimal trap trajectories to reveal generic mid-air shapes, given only a reference path. Building on a novel multi-dimensional approximation of the acoustic forces around the trap, we formulate and show how to solve a non-linear path following problem without requiring or exploiting differential flatness of the system dynamics. We demonstrate increases of up to 563% in size and up to 150% in frequency for several shapes. Additionally, we obtain better reconstruction accuracy with up to a 79% decrease in the path-normalised RMSE. While previously, feasible trap trajectories needed to be tuned manually for each shape and levitator, our approach requires calibration of each individual levitator just once. We are confident that the ideas in this chapter could form the basis for future content authoring tools for acoustic levitation displays and bring them a key step closer to real-world applications.

Chapter 8

Future Work and Conclusion

8.1 Future Work

Beyond the contributions of this dissertation, there is a vast scope for further investigation and discovery within the field of acoustic levitation displays and their potential applications. Looking ahead, I point out several areas that hold great potential for further scientific advancements and innovation.

In my research, I delved into a model-based open loop approach for the control of the dynamics of acoustically levitated particles (e.g., *Levitation Simulator* and *OptiTrap*). Using this approach, I was able to obtain significant performance improvements. However, it is important to note that the mathematical models I employed are only approximations of the real system dynamics. They were derived using certain simplifying assumptions in order to streamline the computation, and as such, they do not fully capture the complex behaviour of the actual system. For instance, one assumption is that acoustic traps generated at different locations within the acoustic volume have equal strength. However, in reality, the strength of the acoustic field (i.e., the amount of exerted acoustic pressure) decreases as the angle between acoustic trap and the transducer normal increases. This means that an acoustic trap generated at the centre receives more contribution from each transducer compared to traps generated at the periphery of the acoustic volume. Furthermore, these models do not account for hardware inaccuracies or external factors like heat dissipation, air flow, or the presence of other objects in the volume. To improve this approach, there are two straightforward strategies to consider. The first is to **refine the mathematical model** to more accurately reflect the dynamics of the levitation system. The second strategy is to **develop more accurate levitation hardware** that can better align with the mathematical model. It is worth noting that both aspects should be advanced together, as a more accurate model may introduce numerical instabilities, and potentially lead to worse results if the hardware is not accurate enough. In addition, there exist specialised models that provide a good description of a particular phenomena or interaction patterns within the levitation system. For example, recently developed models detail the sound scattering off stationary objects' surfaces within the working volume of the levitator [52], and sound reflections off the fingertip during collocated direct particle manipulation [61]. Although these models broaden the potential applications of acoustic levitation displays, they can be challenging to generalise to different scenarios and may have limitations in their applicability.

Despite the notable progress, open loop approaches still have the inherent limitation that they do not allow for automatic corrections of inaccuracies and system disturbances. **Closed loop approaches** could offer a promising solution to this issue, as they can enhance the system's adaptability and responsiveness. Implementing closed loop control for acoustic levitation requires information about the actual particle position corresponding to the current location of the acoustic trap. Yet, currently, using this information is challenging due to the millisecond delays introduced by the optical motion tracking system. Assuming improved tracking solutions become available in the future, model-based approaches could be combined with **machine learning approaches** for better performance. The learning-based approaches will require example paths including initial timing and trap placement information to serve as training data for the learning algorithms. The complexity and convergence of the learning-based approaches will be constrained by the quality and quantity of the provided examples. As such, careful consideration must be given to ensure that the dataset encompasses a wide range of scenarios and captures the desired behaviour of the levitation system. In scenarios where there are too few examples available, the existing model-based algorithms can be used to consistently generate physically feasible initial trajectories, thereby mitigating the need for system restarts in the event of failure. Following this, the learning-based closed loop approach can be deployed to further refine the system dynamics model, account for device inaccuracies, and improve the overall performance. By effectively combining the strengths of both model-based and learning-based approaches, we can advance the capabilities of acoustic levitation displays, paving the way for more efficient and intelligent control strategies.

The **optimal control of multiple-particle levitated graphics** is another promising avenue for enhancing the capabilities of acoustic levitation interfaces. One of the main challenges in levitating multiple particles simultaneously has been determining the phase delays for each transducer that would result in multiple focus points, while avoiding destructive interference of the acoustic waves and the consequent loss of acoustic pressure. Recent research has made significant progress in addressing this issue, most notably by adapting phase retrieval methods from optics to acoustic levitation. The *Holographic Acoustic Tweezers* [84] approach used an iterative backpropagation algorithm to achieve simultaneous levitation and dynamic manipulation of 12 particles, and partial manipulation of 25 particles. This method allowed for the rendering of simple sparse 3D shapes. Additionally, by attaching lightweight cloth and threads to the levitated particles that serve as anchors, levitated props and mid-air projection screens were created [89, 31]. However, the slow computation rates of the algorithm limited the rendering of continuous content using multiple particles. To overcome this limitation, the *GS-PAT* [115] algorithm was developed as a fast phase retrieval method for multi-particle acoustic levitation displays. It enabled the fast computation of up to 32 levitation traps at update rates exceeding 10 kHz, allowing the rendering of continuous levitated 3D content, as multiple particles rapidly scan the acoustic volume.

This development opens up great potential for extending the single-particle optimisation routine presented in Chapter 7 to multiple levitated particles. While the use of several particles does not increase the overall interface power that can be leveraged, it does allow for increased flexibility and versatility. That is, each acoustic trap's forces can be individually and dynamically adjusted to match the requirements of the specific path region revealed by each particle. Moreover, multiple particles can render distinct independent features, enabling the creation of content beyond a single connected path and preventing unnecessary traversal of unilluminated sections that are not visible to the user. To fully explore the potential of multiparticle displays, future research should focus on various aspects. Firstly, there is a need to address the reliability of intensity control for the acoustic traps. The generation of multiple acoustic traps, which may potentially interfere with one another, could lead to changes in the individual trap's topology. Consequently, this could result in inaccurate predictions of the trap intensity around its centre. This phenomenon has yet to be thoroughly studied and hence calls for additional investigation, as achieving consistent and precise control over the intensity of each trap is essential for ensuring accurate and reliable levitation of multiple particles. Thus, the development of robust control algorithms and hardware improvements are necessary to secure the stability and consistency of the acoustic traps. Another important area of exploration involves accounting for the independent timing progression of each particle. As different particles traverse their respective paths, their speeds and rates of progression may differ. Adapting the control algorithms to accommodate these variations will enable synchronised and coordinated movement of multiple particles. This requires developing techniques that account for the individual dynamics of each particle, allowing for precise control and coordination among them.

To enhance the **interactivity** aspect of the interface, it is crucial to reduce the computational complexity of the algorithm, enabling almost real-time execution of the optimisation routine. This optimisation should be performed online, allowing for interactive control and responsiveness. By achieving efficient computation, the system can accommodate interaction almost in real time, providing a seamless and engaging user experience.

In addition to enhancing interactivity, further investigation is needed in the realm of computationally supported **content design** for multiparticle levitation displays. It is important to continue developing and expanding the features of content creation tools and software frameworks that are specifically tailored for levitation displays, such as the *Levitation Simulator*. These tools should serve to streamline the design process, empowering content creators to efficiently manipulate and control multiple particles, levitated props, and PoV content. Manipulation methods and techniques need to be developed to create visually appealing and interactive content that takes advantage of the presence of multiple particles. By providing intuitive interfaces, real-time visual feedback, and simulation capabilities, designers can experiment with different configurations, particle arrangements, and interactive behaviours. Ultimately, this will facilitate the creation of visually captivating and immersive experiences that seamlessly integrate digital content into our physical environment, thereby pushing the boundaries of interactive display technology.

8.2 Conclusion

Traditional displays have limited ability to seamlessly integrate digital experiences with our everyday lives. The lack of depth, tangibility, and immersion creates a significant barrier between the virtual and physical realms, restricting users from perceiving and interacting with digital content in a natural and intuitive manner. Ideally, interfaces would enable users to engage with virtual content as effortlessly as they do with objects in their physical reality. One promising solution to bridge this divide comes from acoustic levitation technology. Acoustic levitation displays use ultrasound waves to generate focus points, and suspend and manipulate small objects, enabling the creation of 3D graphics and the generation of haptic feedback in mid-air. This combination allows for dynamic visual and haptic experiences to be realised in the real physical space. By fully leveraging its potential we can potentially bring display technologies closer to the *ultimate display* vision of **bridging the gap between physical and virtual worlds**. However, numerous challenges must be addressed on the path to achieving this ambitious goal. In the Section 1.1 of the introduction, I identified several key challenges relating to optimising the control, content generation, haptic stimulation, and user interaction aspects of acoustic levitation displays. In this dissertation, I tackled these challenges by developing a novel method for smooth, agile and low latency interaction with acoustic levitation displays, content generation facilitated via virtual prototyping, user study investigating different stimulation methods for enhancing the perception of multi-point

Figure 8.1 Visual summary of the key implementations presented in this dissertation. These include *LeviCursor*, for precise low-latency manipulation of a levitated object and agile selection of 3D targets, the deployment of the *Levitation Simulator* for virtual prototyping of acoustic levitation display applications, the investigation of different mid-air haptic stimulation methods with a specific focus on conveying Braille haptic information in *HaptiRead*, and the implementation of the *OptiTrap* algorithm for computing optimal trap trajectories for rendering levitated graphics of up to approx. 10cm in width in PoV time, featuring sharp corners and large curvature changes.

haptic feedback by users, and a structured numerical approach for optimal particle control. An overview of the key results is provided in Figure 8.1. By offering solutions to these challenges, this dissertation contributes to the advancement of acoustic levitation displays, opening up opportunities for developing cutting-edge interface technologies that enable immersive applications and enhanced experiences, foster a more direct transfer of skills and information, and facilitate workflows and collaboration. Below, I provide a summary of the contributions of each system.

In particular, I developed a method for interactively moving a levitated 3D pointer in mid-air, addressing previous challenges of limited indirect (input device-mediated) display interactivity and, unstable and jittery particle movement. By achieving a round-trip latency of 15ms through precomputation of transducer phases, implementing a 3D interpolation scheme for precise control, and employing a particle stabilisation mechanism, this method enabled agile dexterous interaction with a levitated object, and continuous jitter-free particle motion. A series of user studies involving repeated pointing and selection of virtual 3D targets in mid-air demonstrated the effectiveness of the levitated cursor and its comparable performance to traditional input devices.

In addition, I provided an open source prototyping tool for acoustic levitation displays. This comprehensive tool includes a built-in simulator that uses a semi-analytical mathematical model to efficiently simulate the dynamic movement of levitated particles within a sound field. By providing a realistic representation of particle motion, the simulator allows for performance evaluations and predictions of user engagement, comparable to real-world levitation setups. The tool's main advantage lies in its ability to expedite design iterations during the early stages of development, facilitating faster and more efficient virtual prototyping, user interface design, and application testing processes.

Through empirical research, I explored various haptic stimulation methods involving spatial and temporal modulation to present multipoint haptic patterns in mid-air. The specific focus of the study was on improving the resolution of the haptic interface for conveying information in Braille. In a broader context, the study results can also be helpful for designing and evaluating composite haptic representations of individual particles within levitated content for future multimodal applications. Notably, the study included the participation of blind individuals, marking an important milestone in examining the accessibility opportunities and limitations of this emerging display technology. These findings provided valuable insights and guidelines for future implementations using multimodal acoustic displays to create inclusive and enriching user experiences.

Finally, I proposed an optimisation algorithm for computing trap trajectories in acoustic levitation displays that outputs physically feasible and nearly time-optimal rendering of generic mid-air shapes. The algorithm solves a non-linear path following problem, computing the optimum (minimum) timing in which the levitated particle can traverse a given path while adhering the prescribed trap-particle dynamics. The dynamics are prescribed using a multi-dimensional approximation of acoustic forces, marking a significant improvement building upon the 1D semi-analytical model used in the virtual prototyping tool. Notably, this algorithmic approach outperformed traditional trial and error methods, as demonstrated in comprehensive performance tests, eliminating the need for laborious manual tuning and calibration for each shape and levitator. As a result, the proposed algorithm offered a structured solution to the problem of optimal shape rendering in acoustic levitation displays.

Bibliography

[1] Al-Qudah, Z., Doush, I. A., Alkhateeb, F., Al Maghayreh, E., and Al-Khaleel, O. (2011). Reading braille on mobile phones: A fast method with low battery power consumption. In *2011 International Conference on User Science and Engineering (i-USEr)*, pages 118–123. IEEE.

[2] Alexander, J., Marshall, M. T., and Subramanian, S. (2011). Adding haptic feedback to mobile tv. In *CHI '11 Extended Abstracts on Human Factors in Computing Systems*, CHI EA '11, pages 1975–1980, New York, NY, USA. ACM.

[3] Alexander, J., Roudaut, A., Steimle, J., Hornbæk, K., Bruns Alonso, M., Follmer, S., and Merritt, T. (2018). Grand challenges in shape-changing interface research. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, CHI '18, page 1–14, New York, NY, USA. Association for Computing Machinery.

[4] Alonso-Mora, J., Breitenmoser, A., Rufli, M., Siegwart, R., and Beardsley, P. (2012). Image and animation display with multiple mobile robots. *The International Journal of Robotics Research*, 31(6):753–773.

[5] Alrøe, T., Grann, J., Grönvall, E., Petersen, M. G., and Rasmussen, J. L. (2012). Aerial tunes: Exploring interaction qualities of mid-air displays. In *Proceedings of the 7th Nordic Conference on Human-Computer Interaction: Making Sense Through Design*, NordiCHI '12, page 514–523, New York, NY, USA. Association for Computing Machinery.

[6] Andrade, M. A. B., Bernassau, A. L., and Adamowski, J. C. (2016). Acoustic levitation of a large solid sphere. *Applied Physics Letters*, 109(4):044101.

[7] Aromaa, S. and Väänänen, K. (2016). Suitability of virtual prototypes to support human factors/ergonomics evaluation during the design. *Applied Ergonomics*, 56:11 – 18.

[8] Bachynskyi, M., Palmas, G., Oulasvirta, A., and Weinkauf, T. (2015). Informing the design of novel input methods with muscle coactivation clustering. *ACM Trans. Comput.-Hum. Interact.*, 21(6):30:1–30:25.

[9] Bachynskyi, M., Paneva, V., and Müller, J. (2018). Levicursor: Dexterous interaction with a levitating object. In *Proceedings of the 2018 ACM International Conference on Interactive Surfaces and Spaces*, ISS '18, pages 253–262, New York, NY, USA. ACM.

[10] Berg, L. P. and Vance, J. M. (2017). Industry use of virtual reality in product design and manufacturing: a survey. *Virtual Reality*, 21(1):1–17.

[11] Berthelot, J. and Bonod, N. (2019). Free-space micro-graphics with electrically driven levitated light scatterers. *Opt. Lett.*, 44(6):1476–1479.

[12] Blundell, B. G. and Schwarz, A. J. (2000). *Volumetric Three-Dimensional Display Systems*. John Wiley & Sons, New York.

[13] Bock, H. G. and Plitt, K.-J. (1984). A multiple shooting algorithm for direct solution of optimal control problems. *IFAC Proceedings Volumes*, 17(2):1603–1608.

[14] Bowen, R. W., Pola, J., and Matin, L. (1974). Visual persistence: Effects of flash luminance, duration and energy. *Vision Research*, 14(4):295 – 303.

[15] Braille, L. (1829). Method of writing words, music, and plain songs by means of dots, for use by the blind and arranged for them. *Paris, France: l'Institution Royale des Jeunes Aveugles*.

[16] Braley, S., Rubens, C., Merritt, T., and Vertegaal, R. (2018). Griddrones: A self-levitating physical voxel lattice for interactive 3d surface deformations. In *Proceedings of the 31st Annual ACM Symposium on User Interface Software and Technology*, UIST '18, page 87–98, New York, NY, USA. Association for Computing Machinery.

[17] Brooke, J. et al. (1996). Sus-a quick and dirty usability scale. *Usability evaluation in industry*, 189(194):4–7.

[18] Bruus, H. (2012). Acoustofluidics 7: The acoustic radiation force on small particles. *Lab Chip*, 12:1014–1021.

[19] Butcher, J. C. (2016). *Numerical methods for ordinary differential equations*. John Wiley & Sons.

[20] Büttner, S., Prilla, M., and Röcker, C. (2020). Augmented reality training for industrial assembly work - are projection-based ar assistive systems an appropriate tool for assembly training? In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*, CHI '20, page 1–12, New York, NY, USA. Association for Computing Machinery.

[21] Carter, T., Seah, S. A., Long, B., Drinkwater, B., and Subramanian, S. (2013). Ultrahaptics: Multi-point mid-air haptic feedback for touch surfaces. In *Proceedings of the 26th Annual ACM Symposium on User Interface Software and Technology*, UIST '13, pages 505–514, New York, NY, USA. ACM.

[22] Chainani, E. T., Choi, W.-H., Ngo, K. T., and Scheeline, A. (2014). Mixing in colliding, ultrasonically levitated drops. *Analytical Chemistry*, 86(4):2229–2237. PMID: 24460103.

[23] Costello, P. J. (1997). Health and safety issues associated with virtual reality: a review of current literature.

[24] Dalecki, D., Child, S. Z., Raeman, C. H., and Carstensen, E. L. (1995). Tactile perception of ultrasound. *The Journal of the Acoustical Society of America*, 97(5):3165–3170.

[25] Eisenstein, J., Vanderdonckt, J., and Puerta, A. (2001). Applying model-based techniques to the development of uis for mobile computers. In *Proceedings of the 6th International Conference on Intelligent User Interfaces*, IUI '01, pages 69–76, New York, NY, USA. ACM.

[26] Engert, S., Klamka, K., Peetz, A., and Dachselt, R. (2022). Straide: A research platform for shape-changing spatial displays based on actuated strings. In *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems*, CHI '22, New York, NY, USA. Association for Computing Machinery.

[27] Eslambolchilar, P. and Murray-Smith, R. (2004). Tilt-based automatic zooming and scaling in mobile devices – a state-space implementation. In Brewster, S. and Dunlop, M., editors, *Mobile Human-Computer Interaction - MobileHCI 2004*, pages 120–131, Berlin, Heidelberg. Springer Berlin Heidelberg.

[28] Faulwasser, T. (2012). *Optimization-based solutions to constrained trajectory-tracking and path-following problems*. PhD thesis, Otto-von-Guericke-Universität Magdeburg, Universitaetspl. 2.

[29] Faulwasser, T., Hagenmeyer, V., and Findeisen, R. (2014). Constrained reachability and trajectory generation for flat systems. *Automatica*, 50(4):1151–1159.

[30] Faulwasser, T., Weber, T., Zometa, P., and Findeisen, R. (2017). Implementation of nonlinear model predictive path-following control for an industrial robot. *IEEE Trans. Contr. Syst. Techn.*, 25(4):1505–1511.

[31] Fender, A. R., Martinez Plasencia, D., and Subramanian, S. (2021). *ArticuLev: An Integrated Self-Assembly Pipeline for Articulated Multi-Bead Levitation Primitives*. Association for Computing Machinery, New York, NY, USA.

[32] Fliess, M., Lévine, J., Martin, P., and Rouchon, P. (1995). Flatness and defect of non-linear systems: introductory theory and examples. *Int. J. Contr.*, 61(6):1327–1361.

[33] Follmer, S., Leithinger, D., Olwal, A., Hogge, A., and Ishii, H. (2013). Inform: Dynamic physical affordances and constraints through shape and object actuation. In *Proceedings of the 26th Annual ACM Symposium on User Interface Software and Technology*, UIST '13, page 417–426, New York, NY, USA. Association for Computing Machinery.

[34] Foresti, D., Nabavi, M., Klingauf, M., Ferrari, A., and Poulikakos, D. (2013). Acoustophoretic contactless transport and handling of matter in air. *Proceedings of the National Academy of Sciences*, 110(31):12549–12554.

[35] Freeman, E., Anderson, R., Williamson, J., Wilson, G., and Brewster, S. A. (2017). Textured surfaces for ultrasound haptic displays. In *Proceedings of the 19th ACM International Conference on Multimodal Interaction*, ICMI '17, pages 491–492, New York, NY, USA. ACM.

[36] Freeman, E., Marzo, A., Kourtelos, P. B., Williamson, J. R., and Brewster, S. (2019). Enhancing physical objects with actuated levitating particles. In *Proceedings of the 8th ACM International Symposium on Pervasive Displays*, PerDis '19, pages 2:1–2:7, New York, NY, USA. ACM.

[37] Freeman, E., Williamson, J., Subramanian, S., and Brewster, S. (2018). Point-and-shake: Selecting from levitating object displays. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, CHI '18, pages 18:1–18:10, New York, NY, USA. ACM.

[38] Fushimi, T., Drinkwater, B. W., and Hill, T. L. (2020). What is the ultimate capability of acoustophoretic volumetric displays? *Applied Physics Letters*, 116(24):244101.

[39] Fushimi, T., Drinkwater, B. W., Marzo, A., and Hill, T. L. (2018). Nonlinear trapping stiffness of mid-air single-axis acoustic levitators. *Applied Physics Letters*, 113(3):034102.

[40] Fushimi, T., Marzo, A., Drinkwater, B. W., and Hill, T. L. (2019). Acoustophoretic volumetric displays using a fast-moving levitated particle. *Applied Physics Letters*, 115(6):064101.

[41] Gavrilov, L. R. (2008). The possibility of generating focal regions of complex configurations in application to the problems of stimulation of human receptor structures by focused ultrasound. *Acoustical Physics*, 54(2):269–278.

[42] Geijtenbeek, T. and Pronost, N. (2012). Interactive character animation using simulated physics: A state-of-the-art review. *Comput. Graph. Forum*, 31(8):2492–2515.

[43] Gil, H., Son, H., Kim, J. R., and Oakley, I. (2018). Whiskers: Exploring the use of ultrasonic haptic cues on the face. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, CHI '18, New York, NY, USA. Association for Computing Machinery.

[44] Goldstein, S. C., Campbell, J., and Mowry, T. C. (2005). Programmable matter. *Computer*, 38(6):99–101.

[45] Goldstein, S. C. and Mowry, T. C. (2004). Claytronics: An instance of programmable matter. In *Wild and Crazy Ideas Session of ASPLOS*, Boston, MA.

[46] Gomes, A., Rubens, C., Braley, S., and Vertegaal, R. (2016). Bitdrones: Towards using 3d nanocopter displays as interactive self-levitating programmable matter. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*, CHI '16, page 770–780, New York, NY, USA. Association for Computing Machinery.

[47] Gomes de Sá, A. and Zachmann, G. (1999). Virtual reality as a tool for verification of assembly and maintenance processes. *Computers & Graphics*, 23(3):389 – 403.

[48] Gor'kov, L. P. (1962). On the forces acting on a small particle placed int an acoustic field in an ideal liquid. *Sov. Phys. Dokl.*, 6(9):773.

[49] Grossman, T. and Balakrishnan, R. (2004). Pointing at trivariate targets in 3d environments. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '04, pages 447–454, New York, NY, USA. ACM.

[50] Harrington, K., Large, D. R., Burnett, G., and Georgiou, O. (2018). Exploring the use of mid-air ultrasonic feedback to enhance automotive user interfaces. In *Proceedings of the 10th International Conference on Automotive User Interfaces and Interactive Vehicular Applications*, AutomotiveUI '18, pages 11–20, New York, NY, USA. ACM.

[51] Hart, S. G. and Staveland, L. E. (1988). Development of nasa-tlx (task load index): Results of empirical and theoretical research. In Hancock, P. A. and Meshkati, N., editors, *Human Mental Workload*, volume 52 of *Advances in Psychology*, pages 139 – 183. North-Holland.

[52] Hirayama, R., Christopoulos, G., Plasencia, D. M., and Subramanian, S. (2022). High-speed acoustic holography with arbitrary scattering objects. *Science Advances*, 8(24):eabn7614.

[53] Hirayama, R., Plasencia, D. M., Masuda, N., and Subramanian, S. (2019). A volumetric display for visual, tactile and audio presentation using acoustic trapping. *Nature*, 575(7782):320–323.

[54] Hoshi, T., Takahashi, M., Iwamoto, T., and Shinoda, H. (2010). Noncontact tactile display based on radiation pressure of airborne ultrasound. *IEEE Transactions on Haptics*, 3(3):155–165.

[55] Inoue, S., Mogami, S., Ichiyama, T., Noda, A., Makino, Y., and Shinoda, H. (2019). Acoustical boundary hologram for macroscopic rigid-body levitation. *The Journal of the Acoustical Society of America*, 145(1):328–337.

[56] Ishii, H. (2008). The tangible user interface and its evolution. *Commun. ACM*, 51(6):32–36.

[57] Ishii, H., Lakatos, D., Bonanni, L., and Labrune, J.-B. (2012). Radical atoms: Beyond tangible bits, toward transformable materials. *interactions*, 19(1):38–51.

[58] Ishikawa, H. and Saito, H. (2008). Point cloud representation of 3d shape for laser-plasma scanning 3d display. In *2008 34th Annual Conference of IEEE Industrial Electronics*, pages 1913–1918.

[59] Iwamoto, T., Tatezono, M., and Shinoda, H. (2008). Non-contact method for producing tactile sensation using airborne ultrasound. In Ferre, M., editor, *Haptics: Perception, Devices and Scenarios*, pages 504–513, Berlin, Heidelberg. Springer Berlin Heidelberg.

[60] Jagacinski, R. J. and Flach, J. M. (2003). *Control Theory for Humans: Quantitative approaches to modeling performance*. Lawrence Erlbaum, Mahwah, New Jersey.

[61] Jankauskis, E., Elizondo, S., Montano Murillo, R., Marzo, A., and Martinez Plasencia, D. (2022). Tiptrap: A co-located direct manipulation technique for acoustically levitated content. In *Proceedings of the 35th Annual ACM Symposium on User Interface Software and Technology*, UIST '22, New York, NY, USA. Association for Computing Machinery.

[62] Jayant, C., Acuario, C., Johnson, W., Hollier, J., and Ladner, R. E. (2010). V-braille: Haptic braille perception using a touch-screen and vibration on mobile phones. In *ASSETS*, volume 10, pages 295–296.

[63] Jota, R., Ng, A., Dietz, P., and Wigdor, D. (2013). How fast is fast enough?: A study of the effects of latency in direct-touch pointing tasks. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '13, pages 2291–2300, New York, NY, USA. ACM.

[64] Kelley, C. R. (1968). *Manual and Automatic Control: A Theory of Manual Control and Its Applications to Manual and to Automatic Systems*. Academic Press.

[65] Kimura, H., Uchiyama, T., and Yoshikawa, H. (2006). Laser produced 3d display in the air. In *ACM SIGGRAPH 2006 emerging technologies*, pages 20–es.

[66] Kleinman, D. L., Baron, S., and Levison, W. H. (1971). A control theoretic approach to manned-vehicle systems analysis. *IEEE Trans. Automatic Control*, 16:824–832.

[67] Knierim, P., Kosch, T., Achberger, A., and Funk, M. (2018). Flyables: Exploring 3d interaction spaces for levitating tangibles. In *Proceedings of the Twelfth International Conference on Tangible, Embedded, and Embodied Interaction*, TEI '18, page 329–336, New York, NY, USA. Association for Computing Machinery.

[68] Korres, G. and Eid, M. (2016). Haptogram: Ultrasonic point-cloud tactile stimulation. *IEEE Access*, 4:7758–7769.

[69] Kratz, S., Brodien, I., and Rohs, M. (2010). Semi-automatic zooming for mobile map navigation. In *Proceedings of the 12th International Conference on Human Computer Interaction with Mobile Devices and Services*, MobileHCI '10, pages 63–72, New York, NY, USA. ACM.

[70] Kumagai, K., Hasegawa, S., and Hayasaki, Y. (2017). Volumetric bubble display. *Optica*, 4(3):298–302.

[71] Kumagai, K., Miura, S., and Hayasaki, Y. (2021). Colour volumetric display based on holographic-laser-excited graphics using drawing space separation. *Scientific reports*, 11(1):1–9.

[72] Le Goc, M., Kim, L. H., Parsaei, A., Fekete, J.-D., Dragicevic, P., and Follmer, S. (2016). Zooids: Building blocks for swarm user interfaces. In *Proceedings of the 29th Annual Symposium on User Interface Software and Technology*, UIST '16, page 97–109, New York, NY, USA. Association for Computing Machinery.

[73] Lee, J., Post, R., and Ishii, H. (2011). Zeron: Mid-air tangible interaction enabled by computer controlled magnetic levitation. In *Proceedings of the 24th Annual ACM Symposium on User Interface Software and Technology*, UIST '11, pages 327–336, New York, NY, USA. ACM.

[74] Lee, M., Je, S., Lee, W., Ashbrook, D., and Bianchi, A. (2019). Activearring: Spatiotemporal haptic cues on the ears. *IEEE Transactions on Haptics*, 12(4):554–562.

[75] Long, B., Seah, S. A., Carter, T., and Subramanian, S. (2014). Rendering volumetric haptic shapes in mid-air using ultrasound. *ACM Trans. Graph.*, 33(6):181:1–181:10.

[76] Luzhnica, G., Veas, E., and Pammer, V. (2016). Skin reading: Encoding text in a 6-channel haptic display. In *Proceedings of the 2016 ACM International Symposium on Wearable Computers*, ISWC '16, page 148–155, New York, NY, USA. Association for Computing Machinery.

[77] MacKenzie, I. S. (1992). Fitts' law as a research and design tool in human-computer interaction. *Human-Computer Interaction*, 7(1):91–139.

[78] Mangasarian, O. L. (1994). *Nonlinear programming*. SIAM.

[79] Martinez, J., Griffiths, D., Biscione, V., Georgiou, O., and Carter, T. (2018). Touchless haptic feedback for supernatural vr experiences. In *2018 IEEE Conference on Virtual Reality and 3D User Interfaces (VR)*, pages 629–630.

[80] Marzo, A. (2016). Gauntlev: A wearable to manipulate free-floating objects. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*, CHI '16, pages 3277–3281, New York, NY, USA. ACM.

[81] Marzo, A. (2017). Ultraino. https://github.com/asiermarzo/Ultraino.

[82] Marzo, A., Barnes, A., and Drinkwater, B. W. (2017). Tinylev: A multi-emitter single-axis acoustic levitator. *Review of Scientific Instruments*, 88(8):085105.

[83] Marzo, A., Caleap, M., and Drinkwater, B. W. (2018). Acoustic virtual vortices with tunable orbital angular momentum for trapping of mie particles. *Phys. Rev. Lett.*, 120:044301.

[84] Marzo, A. and Drinkwater, B. W. (2019). Holographic acoustic tweezers. volume 116, pages 84–89. National Academy of Sciences.

[85] Marzo, A., Kockaya, S., Freeman, E., and Williamson, J. (2019). Tangible interactions with acoustic levitation. In *Extended Abstracts of the 2019 CHI Conference on Human Factors in Computing Systems*, CHI EA '19, pages INT005:1–INT005:4, New York, NY, USA. ACM.

[86] Marzo, A., Seah, S. A., Drinkwater, B. W., Sahoo, D. R., Long, B., and Subramanian, S. (2015). Holographic Acoustic Elements for Manipulation of Levitated Objects. *Nature communications*, 6(1):8661.

[87] McRuer, D. T. and Jex, H. R. (1967). A review of quasi-linear pilot models. *IEEE Trans. on Human Factors in Electronics*, 8(3):231–249.

[88] Morales, R., Ezcurdia, I., Irisarri, J., Andrade, M. A. B., and Marzo, A. (2021). Generating airborne ultrasonic amplitude patterns using an open hardware phased array. *Applied Sciences*, 11(7).

[89] Morales, R., Marzo, A., Subramanian, S., and Martínez, D. (2019). Levipropss: Animating levitated optimized fabric structures using holographic acoustic tweezers. In *Proceedings of the 32Nd Annual ACM Symposium on User Interface Software and Technology*, UIST '19, pages 651–661, New York, NY, USA. ACM.

[90] Müller, J. (2017). Dynamics of pointing with pointer acceleration. In Bernhaupt, R., Dalvi, G., Joshi, A., K. Balkrishan, D., O'Neill, J., and Winckler, M., editors, *Human-Computer Interaction – INTERACT 2017*, pages 475–495, Cham. Springer International Publishing.

[91] Müller, J., Eberle, D., and Schmidt, C. (2015). Baselase: An interactive focus+context laser floor. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*, CHI '15, page 3869–3878, New York, NY, USA. Association for Computing Machinery.

[92] Müller, J., Oulasvirta, A., and Murray-Smith, R. (2017). Control theoretic models of pointing. *ACM Trans. Comput.-Hum. Interact.*, 24(4):27:1–27:36.

[93] Nägeli, T., Meier, L., Domahidi, A., Alonso-Mora, J., and Hilliges, O. (2017). Real-time planning for automated multi-view drone cinematography. *ACM Transactions on Graphics (TOG)*, 36(4):1–10.

[94] Navarre, D., Palanque, P., Ladry, J.-F., and Barboni, E. (2009). Icos: A model-based user interface description technique dedicated to interactive systems addressing usability, reliability and scalability. *ACM Trans. Comput.-Hum. Interact.*, 16.

[95] Nicolau, H., Guerreiro, J., Guerreiro, T., and Carriço, L. (2013). Ubibraille: designing and evaluating a vibrotactile braille-reading device. In *Proceedings of the 15th International ACM SIGACCESS Conference on Computers and Accessibility*, page 23. ACM.

[96] Nicolau, H., Montague, K., Guerreiro, T., Rodrigues, A., and Hanson, V. L. (2015). Holibraille: Multipoint vibrotactile feedback on mobile devices. In *Proceedings of the 12th Web for All Conference*, W4A '15, pages 30:1–30:4, New York, NY, USA. ACM.

[97] Nielsen, C. and Maggiore, M. (2008). On local transverse feedback linearization. *SIAM Journal on Control and Optimization*, 47(5):2227–2250.

[98] Nisser, M., Makaram, Y., Faruqi, F., Suzuki, R., and Mueller, S. (2022). Selective self-assembly using re-programmable magnetic pixels.

[99] Norasikin, M. A., Martinez-Plasencia, D., Memoli, G., and Subramanian, S. (2019). Sonicspray: A technique to reconfigure permeable mid-air displays. In *Proceedings of the 2019 ACM International Conference on Interactive Surfaces and Spaces*, pages 113–122.

[100] Norasikin, M. A., Martinez Plasencia, D., Polychronopoulos, S., Memoli, G., Tokuda, Y., and Subramanian, S. (2018). Soundbender: Dynamic acoustic control behind obstacles. In *Proceedings of the 31st Annual ACM Symposium on User Interface Software and Technology*, UIST '18, page 247–259, New York, NY, USA. Association for Computing Machinery.

[101] Ochiai, Y., Hoshi, T., and Rekimoto, J. (2014). Pixie dust: Graphics generated by levitated and animated objects in computational acoustic-potential field. *ACM Trans. Graph.*, 33(4).

[102] Ochiai, Y., Kumagai, K., Hoshi, T., Rekimoto, J., Hasegawa, S., and Hayasaki, Y. (2016). Fairy lights in femtoseconds: Aerial and volumetric graphics rendered by focused femtosecond laser combined with computational holographic fields. *ACM Trans. Graph.*, 35(2).

[103] Omirou, T., Marzo, A., Seah, S. A., and Subramanian, S. (2015a). Levipath: Modular acoustic levitation for 3d path visualisations. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*, CHI '15, page 309–312, New York, NY, USA. Association for Computing Machinery.

[104] Omirou, T., Marzo, A., Seah, S. A., and Subramanian, S. (2015b). Levipath: Modular acoustic levitation for 3d path visualisations. https://www.youtube.com/watch?v=ahE37cvr1uo.

[105] Omirou, T., Perez, M. A., Subramanian, S., and Roudaut, A. (2016). Floating charts: Data plotting using free-floating acoustically levitated representations. In *2016 IEEE Symposium on 3D User Interfaces (3DUI)*, pages 187–190.

[106] Oulasvirta, A., Kristensson, P. O., Bi, X.-J., and Howes, A. (2018). *Computational Interaction*. Oxford University Press 2018.

[107] O'Brien, H. L., Cairns, P., and Hall, M. (2018). A practical approach to measuring user engagement with the refined user engagement scale (ues) and new ues short form. *International Journal of Human-Computer Studies*, 112:28 – 39.

[108] Paneva, V., Bachynskyi, M., and Müller, J. (2020a). Levitation simulator: Prototyping ultrasonic levitation interfaces in virtual reality. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*, CHI '20, New York, NY, USA. ACM.

[109] Paneva, V., Fleig, A., Plasencia, D. M., Faulwasser, T., and Müller, J. (2022). Optitrap: Optimal trap trajectories for acoustic levitation displays. *ACM Trans. Graph.*, 41(5).

[110] Paneva, V., Seinfeld, S., Kraiczi, M., and Müller, J. (2020b). Haptiread: Reading braille as mid-air haptic information. In *Proceedings of the 2020 ACM Designing Interactive Systems Conference*, page 13–20, New York, NY, USA. ACM.

[111] Paterno, F. (1999). *Model-based design and evaluation of interactive applications*. Springer Science & Business Media.

[112] Paterno, F., Mancini, C., and Meniconi, S. (1997). *ConcurTaskTrees: A Diagrammatic Notation for Specifying Task Models*, pages 362–369. Springer US, Boston, MA.

[113] Patten, J. and Ishii, H. (2007). Mechanical constraints as computational constraints in tabletop tangible interfaces. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '07, page 809–818, New York, NY, USA. Association for Computing Machinery.

[114] Perlin, K. and Jefferson, H. Y. (2006). Volumetric display with dust as the participating medium. US Patent 6,997,558.

[115] Plasencia, D. M., Hirayama, R., Montano-Murillo, R., and Subramanian, S. (2020). Gs-pat: High-speed multi-point sound-fields for phased arrays of transducers. *ACM Trans. Graph.*, 39(4).

[116] Putra, H. A. and Ren, X. (2016). Airvis: An air-based physical visual and tactile display. *ICIC express letters. Part B, Applications: an international journal of research and surveys*, 7(12):2511–2518.

[117] Qamar, I. P. S., Groh, R., Holman, D., and Roudaut, A. (2018). Hci meets material science: A literature review of morphing materials for the design of shape-changing interfaces. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, CHI '18, page 1–23, New York, NY, USA. Association for Computing Machinery.

[118] Quinn, P., Malacria, S., and Cockburn, A. (2013). Touch scrolling transfer functions. In *Proceedings of the 26th Annual ACM Symposium on User Interface Software and Technology*, UIST '13, pages 61–70, New York, NY, USA. ACM.

[119] Rantala, J., Raisamo, R., Lylykangas, J., Surakka, V., Raisamo, J., Salminen, K., Pakkanen, T., and Hippula, A. (2009). Methods for presenting braille characters on a mobile device with a touchscreen and tactile feedback. *IEEE Transactions on Haptics*, 2(1):28–39.

[120] Rasmussen, M. K., Pedersen, E. W., Petersen, M. G., and Hornbæk, K. (2012). Shape-changing interfaces: A review of the design space and open research questions. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '12, page 735–744, New York, NY, USA. Association for Computing Machinery.

[121] Reynal, M., Freeman, E., and Brewster, S. (2020). Avoiding collisions when interacting with levitating particle displays. In *Extended Abstracts of the 2020 CHI Conference on Human Factors in Computing Systems*, CHI EA '20, page 1–7, New York, NY, USA. Association for Computing Machinery.

[122] Roberts, M. and Hanrahan, P. (2016). Generating dynamically feasible trajectories for quadrotor cameras. *ACM Trans. Graph.*, 35(4).

[123] Romanishin, J. W., Gilpin, K., and Rus, D. (2013). M-blocks: Momentum-driven, magnetic modular robots. In *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 4288–4295.

[124] Rose, C., Guenter, B., Bodenheimer, B., and Cohen, M. F. (1996). Efficient generation of motion transitions using spacetime constraints. In *Proceedings of the 23rd Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '96, page 147–154, New York, NY, USA. Association for Computing Machinery.

[125] Rubenstein, M., Cornejo, A., and Nagpal, R. (2014). Programmable self-assembly in a thousand-robot swarm. *Science*, 345(6198):795–799.

[126] Sahoo, D. R., Nakamura, T., Marzo, A., Omirou, T., Asakawa, M., and Subramanian, S. (2016). Joled: A mid-air display based on electrostatic rotation of levitated janus objects. UIST '16, page 437–448, New York, NY, USA. Association for Computing Machinery.

[127] Saito, H., Kimura, H., Shimada, S., Naemura, T., Kayahara, J., Jarusirisawad, S., Nozick, V., Ishikawa, H., Murakami, T., Aoki, J., Asano, A., Kimura, T., Kakehata, M., Sasaki, F., Yashiro, H., Mori, M., Torizuka, K., and Ino, K. (2008). Laser-plasma scanning 3D display for putting digital contents in free space. In Woods, A. J., Holliman, N. S., and Merritt, J. O., editors, *Stereoscopic Displays and Applications XIX*, volume 6803, page 680309. International Society for Optics and Photonics, SPIE.

[128] Schwerdtfeger, B., Pustka, D., Hofhauser, A., and Klinker, G. (2008). Using laser projectors for augmented reality. In *Proceedings of the 2008 ACM Symposium on Virtual Reality Software and Technology*, VRST '08, page 134–137, New York, NY, USA. Association for Computing Machinery.

[129] Seth, A., Vance, J. M., and Oliver, J. H. (2011). Virtual reality for assembly methods prototyping: a review. *Virtual Reality*, 15(1):5–20.

[130] Shin, K. and McKay, N. (1985). Minimum-time control of robotic manipulators with geometric path constraints. *IEEE Trans. Automat. Contr.*, 30(6):531 – 541.

[131] Siu, A. F., Gonzalez, E. J., Yuan, S., Ginsberg, J. B., and Follmer, S. (2018). Shapeshift: 2d spatial manipulation and self-actuation of tabletop shape displays for tangible and haptic interaction. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, CHI '18, page 1–13, New York, NY, USA. Association for Computing Machinery.

[132] Smalley, D., Nygaard, E., Squire, K., Van Wagoner, J., Rasmussen, J., Gneiting, S., Qaderi, K., Goodsell, J., Rogers, W., Lindsey, M., et al. (2018). A photophoretic-trap volumetric display. *Nature*, 553(7689):486.

[133] Sorkine-Hornung, O. and Rabinovich, M. (2017). Least-squares rigid motion using svd. *no*, 3:1–5.

[134] Soukoreff, R. W. and MacKenzie, I. S. (2004). Towards a standard for pointing device evaluation, perspectives on 27 years of fitts' law research in hci. *International Journal of Human-Computer Studies*, 61(6):751 – 789. Fitts' law 50 years later: applications and contributions from human-computer interaction.

[135] Steinicke, F. (2016). *The Science and Fiction of the Ultimate Display*, pages 19–32. Springer International Publishing, Cham.

[136] Stevens, W. L. C. (1899). A text-book of physics–sound.

[137] Sturdee, M. and Alexander, J. (2018). Analysis and classification of shape-changing interfaces for design and application-based research. *ACM Comput. Surv.*, 51(1).

[138] Sutherland, I. E. (1965). The Ultimate Display. *Multimedia: From Wagner to Virtual Reality*, pages 506–508.

[139] Suzuki, R., Kato, J., Gross, M. D., and Yeh, T. (2018a). Reactile: Programming swarm user interfaces through direct physical manipulation. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, CHI '18, page 1–13, New York, NY, USA. Association for Computing Machinery.

[140] Suzuki, R., Yamaoka, J., Leithinger, D., Yeh, T., Gross, M. D., Kawahara, Y., and Kakehi, Y. (2018b). Dynablock: Dynamic 3d printing for instant and reconstructable shape formation. UIST '18, page 99–111, New York, NY, USA. Association for Computing Machinery.

[141] Suzuki, R., Zheng, C., Kakehi, Y., Yeh, T., Do, E. Y.-L., Gross, M. D., and Leithinger, D. (2019). Shapebots: Shape-changing swarm robots. In *Proceedings of the 32nd Annual ACM Symposium on User Interface Software and Technology*, UIST '19, page 493–505, New York, NY, USA. Association for Computing Machinery.

[142] Teather, R. J. and Stuerzlinger, W. (2011). Pointing at 3d targets in a stereo head-tracked virtual environment. In *2011 IEEE Symposium on 3D User Interfaces (3DUI)*, pages 87–94.

[143] Toffoli, T. and Margolus, N. (1991). Programmable matter: concepts and realization. *Physica. D, Nonlinear phenomena*, 47(1-2):263–272.

[144] Verscheure, D. (2009). *Contribution to contact modeling and identification and optimal robot motion planning*. PhD thesis, Katholieke Universiteit Leuven, Faculteit Ingenieurswetenschappen, Department Werktuigkunde.

[145] Vi, C. T., Ablart, D., Gatti, E., Velasco, C., and Obrist, M. (2017a). Not just seeing, but also feeling art: Mid-air haptic experiences integrated in a multisensory art exhibition. *International Journal of Human-Computer Studies*, 108:1 – 14.

[146] Vi, C. T., Marzo, A., Ablart, D., Memoli, G., Subramanian, S., Drinkwater, B., and Obrist, M. (2017b). Tastyfloats: A contactless food delivery system. In *Proceedings of the 2017 ACM International Conference on Interactive Surfaces and Spaces*, ISS '17, page 161–170, New York, NY, USA. Association for Computing Machinery.

[147] Von Stryk, O. (1993). Numerical solution of optimal control problems by direct collocation. In *Optimal control*, pages 129–143. Springer.

[148] Wenjun, G., Seungwook, K., Sangung, Y., Minkyu, C., Seungha, Y., and Kyungwon, L. (2017). Garden aqua: Tangible user interface study using levitation. *International Journal of Asia Digital Art and Design Association*, 21(2):87–93.

[149] Williamson, J. (2006). *Continuous Uncertain Interaction*. PhD thesis, Dept. of Computing Science, University of Glasgow.

[150] Wilson, G., Carter, T., Subramanian, S., and Brewster, S. A. (2014). Perception of ultrasonic haptic feedback on the hand: Localisation and apparent motion. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '14, pages 1133–1142, New York, NY, USA. ACM.

[151] Witkin, A. and Kass, M. (1988). Spacetime constraints. In *Proceedings of the 15th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '88, page 159–168, New York, NY, USA. Association for Computing Machinery.

[152] Xiao, R., Cao, T., Guo, N., Zhuo, J., Zhang, Y., and Harrison, C. (2018). Lumiwatch: On-arm projected graphics and touch input. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, CHI '18, page 1–11, New York, NY, USA. Association for Computing Machinery.

[153] Yamada, W., Manabe, H., Ikeda, D., and Rekimoto, J. (2020). Raygraphy: Aerial volumetric graphics rendered using lasers in fog. In *Symposium on Spatial User Interaction*, SUI '20, New York, NY, USA. Association for Computing Machinery.

[154] Yim, M., Shen, W.-m., Salemi, B., Rus, D., Moll, M., Lipson, H., Klavins, E., and Chirikjian, G. S. (2007). Modular self-reconfigurable robot systems [grand challenges of robotics]. *IEEE Robotics Automation Magazine*, 14(1):43–52.

[155] Yui, T. and Hashida, T. (2016). Floatio: Floating tangible user interface based on animacy perception. In *Proceedings of the 29th Annual Symposium on User Interface Software and Technology*, UIST '16 Adjunct, page 43–45, New York, NY, USA. Association for Computing Machinery.

[156] Zhang, X., Braley, S., Rubens, C., Merritt, T., and Vertegaal, R. (2019). Lightbee: A self-levitating light field display for hologrammatic telepresence. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, CHI '19, page 1–10, New York, NY, USA. Association for Computing Machinery.

List of Figures

List of Tables

Appendix A

Levitation Simulator: User Study Materials

This appendix contains the following forms and questionnaires (provided in German and English) used in the user studies, presented in Chapter 5:

- Consent Form
- Demographics Questionnaire
- Study Information Sheet
- Realism and Presence Questionnaire
- UES
- Semi-structured Interview Questions

EINVERSTÄNDNISERKLÄRUNG DES TEILNEHMERS

Dies ist von den Freiwilligen auszufüllen. Wir bitten Sie, die nachfolgenden Fragen sorgfältig zu lesen.

Haben Sie das Informationsblatt dieser Studie gelesen? JA/NEIN

Haben Sie die Möglichkeit gehabt, Fragen zu stellen und über die Studie zu sprechen? JA/NEIN

Haben Sie befriedigende Antworten auf alle Ihre Fragen erhalten? JA/NEIN

Haben Sie genügend Informationen über diese Studie erhalten? JA/NEIN

Mit welchem/r MitarbeiterIn haben Sie über diese Studie gesprochen? _______________________________

Ist Ihnen bewusst, dass es Ihnen jederzeit möglich ist, von dieser Studie zurückzutreten?

- Jederzeit JA/NEIN
- Ohne Angabe von Gründen JA/NEIN

Sind Sie damit einverstanden, dass wir eine Audioaufnahme des Interviews am Ende des Experiments anfertigen? Das Interview bezieht sich auf Ihre Erfahrungen mit der Verwendung des Levitators. JA/NEIN

Erklären Sie sich bereit, an der Studie teilzunehmen? JA/NEIN

Ich bestätige hiermit:

-Ich bin älter als 18 Jahre

-Ich leide an keiner neurologischen Erkrankung, Sehstörung oder Hörstörung

-Ich habe keinen Herzschrittmacher

-Ich leide nicht an Epilepsie und hatte in meinem Leben noch nie einen epileptischen Anfall.

-Ich nehme keine Psychopharmaka.

-Ich habe in den vergangenen 8 Stunden keinen Alkohol (oder andere psychoaktive Substanzen) konsumiert.

Falls Sie Fragen oder Anregungen zu diesem Experiment haben, kontaktieren Sie bitte:

Prof. Jörg Müller, Universität Bayreuth, Serious Games, Universitätsstraße 30, 95447 Bayreuth

Unterschrift…………………………………..…………Datum……………………………..

Name in Druckbuchstaben..…………………………..…………………………………..

Die von uns gesammelten Informationen werden nie so abgespeichert, dass Personen identifiziert werden können. Die Informationen werden in zusammengefasster Form publiziert. Alle verbalen Äußerungen Ihrerseits werden in den Publikationen anonymisiert dargestellt. Sie haben jederzeit das Recht, uns aufzufordern, Ihre Daten aus unseren Datenbanken zu entfernen.

**Diese Fragen werden nur gestellt, wenn es im Sinne der Studie erforderlich ist, die Teilnehmenden auf Video- und Tonaufnahmen aufzuzeichnen, um die subjektive Meinung der Teilnehmenden zu Protokoll zu nehmen.*

PARTICIPANT INFORMED CONSENT

To be completed by volunteers. We would like you to read the following questions carefully.

Have you read the information sheet about this study? YES/NO

Have you had an opportunity to ask questions and discuss this study? YES/NO

Have you received satisfactory answers to all your questions? YES/NO

Have you received enough information about this study? YES/NO

Which investigator have you spoken to about this study? ____

Do you understand that you are free to withdraw from this study?

- At any time YES/NO
- Without giving a reason for withdrawing YES/NO

Do you agree that we make an audio recording of the interview at the end of the experiment? The interview will consist of questions regarding your experience when interacting with the Levitator. YES/NO

Do you agree to take part in this study? YES/NO

I certify that:
-I am older than 18 years
-I do not suffer any neurological disorder, visual impairment or auditory impairment
-I do not have a pacemaker
-I do not have a history of epilepsy
-I do not take psychoactive medications
-I have not drink alcohol (or other substances) in the last 8 hours

In case you have any question or comment concerning this experiment please contact:
Prof. Jörg Müller, Universität Bayreuth, Serious Games, Universitätsstraße 30, 95447 Bayreuth

Signed…………………………………..…………**Date**……………………………...

Name in block letters……………………………..…………………………………...

Information that we collect will never be reported in a way that individuals can be identified. Information will be reported in aggregate, and any verbal comments that you make, if written about in subsequent papers, will be presented anonymously. You have the right to request us to eliminate your data from our databases at any time.

**This questions will only be asked if the study requires to record physiological signals or videotape interviews made to the participants to better understand their subjective opinion about what they experienced during the experiment.*

Fragebogen (Demographics)

Teilnehmer Nummer:____________________________________

Alter: ____

Geschlecht: ____

Studiengang/Beruglicher Hintergrund:_____________________

____

Besitzen Sie eine Virtual-Reality-Brille? JA/NEIN

Wie oft nutzen sie Virtual-Reality-Inhalte?

□ Täglich/mehrmals pro Woche

□Mehrmals/einmal pro Monat

□ Mehrmals/einmal pro Jahr

□Nie

Bitte geben Sie das Niveau Ihrer Computerkenntnisse auf einer Skala von 1 bis 7 an (Einsteiger) 1 ◯ 2 ◯ 3 ◯ 4 ◯ 5 ◯ 6 ◯ 7 ◯ (Experte)
Bitte bewerten Sie Ihre Programmierkentnise: (Einsteiger) 1 ◯ 2 ◯ 3 ◯ 4 ◯ 5 ◯ 6 ◯ 7 ◯ (Experte)
Haben Sie bereits Erfahrungen mit 'virtual reality' gemacht? (Keine Erfahrung) 1 ◯ 2 ◯ 3 ◯ 4 ◯ 5 ◯ 6 ◯ 7 ◯ (umfangreiche Erfahrung)

Wie oft haben Sie im letzten Jahr Videospiele gespielt (zu Hause, bei der Arbeit, in der Schule oder in Spielhallen)?	(* mal im letzten Jahr) Nie ◯ 1 - 5 ◯ 6 - 10 ◯ 11 - 15 ◯ 16 - 20 ◯ 21 - 25 ◯ > 25 ◯ Welche Spiele haben Sie gespielt?
Wie viele Stunden pro Woche verbringen Sie mit Videospielen?	0 ◯ < 1 ◯ 1 - 3 ◯ 3 - 5 ◯ 5 - 7 ◯ 7 - 9 ◯ > 9 ◯

Questionnaire (Demographics)

Participant's ID:_______________________________________

Age: ____

Gender: ____

Field of Study/Occupation:______________________________

____

Do you own a Virtual Reality Headset? YES/NO

How often do you use Virtual Reality content?

- □ daily/few times a week
- □ once/ few times a month
- □ once/ few times a year
- □ never

Please state your level of computer literacy on a scale of (1...7)

(novice) 1 ◯ 2 ◯ 3 ◯ 4 ◯ 5 ◯ 6 ◯ 7 ◯ (expert)

Please rate your level of experience with computer programming:

(novice) 1 ◯ 2 ◯ 3 ◯ 4 ◯ 5 ◯ 6 ◯ 7 ◯ (expert)

Have you ever experienced 'virtual reality' before?

(no experience) 1 ◯ 2 ◯ 3 ◯ 4 ◯ 5 ◯ 6 ◯ 7 ◯ (extensive experience)

How many times did you play video games (at home, work, school, or arcades) in the last year?	Never ◯ 1 - 5 ◯ 6 - 10 ◯ 11 - 15 ◯ 16 - 20 ◯ 21 - 25 ◯ > 25 ◯ What video games do you play?
How many hours per week do you spend playing video games?	0 ◯ < 1 ◯ 1 - 3 ◯ 3 - 5 ◯ 5 - 7 ◯ 7 - 9 ◯ > 9 ◯

Information about the Study

In this study you will be asked to move a levitating cursor (spherical particle) between two targets **as fast and as accurately** as possible. The cursor is gesture controlled at a distance, that is, we will place a retroreflective spherical marker on your index finger, with the help of which, you will be able to control the cursor.

You will perform the same task in two different environments: in the real world and in virtual reality. Before starting the task in each environment, you will have a short training session. During the training, you will have the opportunity to familiarize yourself with the system and explore its functions. Please ask any questions you have to the experimenter during the training session.

The tasks in both in the real and virtual environment consists of six sessions. Before each session, you will be asked to place your index finger in a comfortable position. Make sure you have a good view both of the cursor and the two targets and you feel comfortable with your arm placement. Between the sessions, you have the opportunity to take a break and rest your hand, if you wish to do so. If not, you can tell the experimenter to continue to the next session.

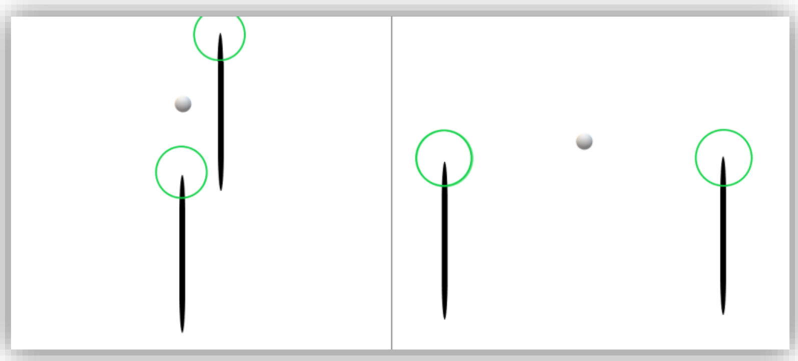


Figure 1. The two different spatial target arrangements. When the cursor is placed entirely within the virtual target (green circle), you will hear an audio feedback and you can move to the next target.

The targets are virtual spheres (green circles in **Figure 1**) marked by two black vertical needles. The center of the spherical targets is at the needle tip. When you place the cursor entirely within the virtual target and shortly stop, you will hear an audio signal, which means that you have successfully hit the target and you can move to the next. If you did not hear the sound, but you already moved on to the next target, you will have to go back and correctly hit the previous target. In each session, the targets will appear in one of the two possible spatial arrangements, shown in **Figure 1**, and in one of three possible target sizes.

At the end of the experiment we will also kindly ask you to answer some questions to better understand your experience.

We will combine your speed and accuracy into a single performance measure, and the participant who achieves the highest performance out of all participants, will receive extra *10 euros*, on top of the regular reimbursement.

Informationen zur Studie

In dieser Studie werden Sie gebeten, einen schwebenden Cursor (sphärisches Teilchen) zwischen zwei Zielen **so schnell und so genau** wie möglich zu bewegen. Der Cursor wird aus einer Entfernung mithilfe eines an Ihrem Zeigefinger platzierten retroreflektierenden Kugelmarkers durch eine Geste gesteuert.

Sie werden dieselbe Aufgabe in zwei verschiedenen Umgebungen ausführen: in der realen Welt und in der virtuellen Realität. Bevor Sie die Aufgabe in der jeweiligen Umgebung starten, werden Sie ein kurzes Trening durchführen. Während des Trainings haben Sie die Möglichkeit, sich mit dem System vertraut zu machen und seine Funktionen zu erkunden. Bitte stellen Sie alle Fragen, die Sie während des Trainings haben.

Die Aufgaben in der realen und virtuellen Umgebung bestehen aus sechs Einheiten. Vor jeder Sitzung werden Sie gebeten, Ihren Zeigefinger in eine bequeme Position zu bringen. Stellen Sie sicher, dass Sie einen guten Überblick über die beiden Ziele haben und sich mit Ihrer Armposition wohl fühlen. Zwischen den Sitzungen haben Sie die Möglichkeit, eine Pause einzulegen und Ihre Hand auszuruhen, wenn Sie dies wünschen. Wenn nicht, können Sie den Versuchleiter anweisen, mit der nächsten Sitzung fortzufahren.

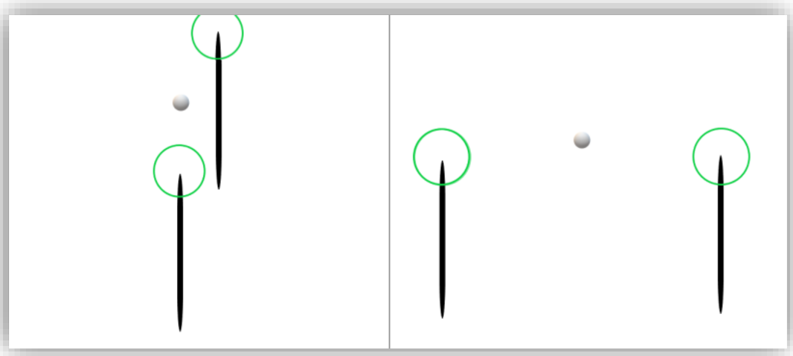


Abbildung 1. Die zwei verschiedenen räumlichen Zielanordnungen. Wenn sich der Cursor innerhalb des virtuellen Ziels (grüner Kreis) befindet, werden Sie ein Audio-Feedback hören und Sie können zum nächsten Ziel wechseln.

Die Ziele sind virtuelle Kugeln (grüne Kreise in **Abbildung 1**), die durch zwei schwarze vertikale Nadeln markiert sind. Die Mitte der kugelförmigen Ziele befindet sich an der Nadelspitze. Wenn Sie den Cursor in das virtuelle Ziel setzen und kurz anhalten, werden Sie ein Audiosignal hören. Das heißt, Sie haben das Ziel erfolgreich getroffen und können zum nächsten wechseln. Wenn Sie den Ton nicht gehört haben, aber bereits zum nächsten Ziel weitergegangen sind, gehen Sie bitte zurück um das vorherige Ziel richtig zu treffen. In jeder Sitzung erscheint das Ziel in einer der zwei möglichen räumlichen Anordnungen (siehe **Abbildung 1**) und in einer von drei möglichen Zielgrößen.

Am Ende des Experiments bitten wir Sie, einige Fragen zu beantworten, um uns Ihre Erfahrung mitzuteilen.

Wir werden Ihre Geschwindigkeit und Genauigkeit zu einer einzelnen Leistungskennzahl kombinieren, und der Teilnehmer, der unter allen Teilnehmern die höchste Leistung erzielt, erhält zusätzlich zur regulären Erstattung **10 Euro**.

VORGEHEN, WELCHES JEDER TEILNEHMER VERSTEHEN UND AKZEPTIEREN MUSS.

- Sie werden gebeten, die Einwilligungserklärung zu lesen, zu verstehen und zu unterschreiben. Wenn Sie dieses Formular unterschreiben, werden wir mit Ihrer Teilnahme am Experiment rechnen. Denken Sie daran, dass Sie das Experiment jederzeit ohne Angabe von Gründen abbrechen können.
- Wir werden Sie bitten, Ihr Handy auszuschalten.
- Wir können Sie bitten, einige Fragen in Echtzeit zu beantworten, damit wir Ihre Antworten während der Studie besser nachvollziehen können.
- Vielen Dank für Ihre Teilnahme! Bitte sprechen Sie im Zeitraum von 3 Monaten, in denen das Experiment durchgeführt wird, mit niemandem über das Experiment.
- Wenn Sie irgendwelche Fragen haben, können Sie sich jeder Zeit an das Versuchspersonal wenden.

PROCEDURES THAT EACH PARTICIPANT HAS TO UNDERSTAND AND ACCEPT

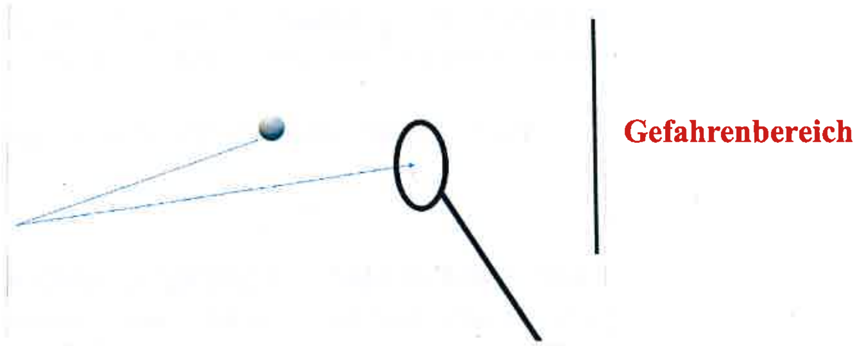
- You are asked to read, understand and sign the informed consent. If you sign this form, we will count with your participation for the experiment. Remember that you can leave the experiment at any time without giving a reason.
- We are going to ask you to please turn off your mobile phone.
- We can ask you to answer some question in real time so we can understand your responses during the study.
- Thanks for your participation! Please don`t talk about the experiment with anyone during 3 months while the experiment continues.
- If you have any question you can ask the experimenters.

Informationen zur Studie

In dieser Studie wird es Ihre Aufgabe sein zwei Minispiele (Games) mit einem schwebenden Partikel zu spielen. Sie werden die Gelegenheit haben diese Spiele in zwei verschiedenen Umgebungen (Settings) zu spielen: In der echten Welt und virtueller Realität.

Bitte stellen Sie dem Versuchsleiter alle eventuellen Fragen, die sie haben, vor dem Beginn jedes Spiels. Sie können zwischen den Versuchen Pausen nehmen.

BeadBounce



In diesem Minispiel müsse sie das Partikel (die weiße Kugel im Bild) von dem Gefahrenbereich fernhalten. Das Spiel wird in der linken Hälfte des Levitators gespielt. Die rechte Hälfte ist der Gefahrenbereich. Das schwebende Partikel wird sich in zufällige Richtungen bewegen und von den Wänden abprallen. Wenn es sich nach rechts bewegt, müssen Sie es mit dem Schläger (Controller) berühren, damit es sich wieder zurück nach links bewegt. Jedes Mal, wenn Sie das Partikel treffen, bekommen Sie einen Punkt. Schafft das Partikel es in den Gefahrenbereich, verlieren Sie zwei Punkte.

LeviShooter

In diesem Minispiel werden Sie mit einer Laserpistole (Controller) auf das schwebende Partikel schießen. Das schwebende Partikel wird sich wieder in eine zufällige Richtung bewegen. Sie müssen genau auf das Partikel zielen. Das Partikel reflektiert den Laser, wenn Sie genau darauf zielen. Daran erkennen Sie, ob Sie richtig zielen. Treffen Sie das Partikel, bleibt es kurze Zeit stehen bevor es sich wieder bewegt und sie erneut schießen können. Treffen Sie das Partikel erhalten sie fünf Punkte, bei einem Fehler verlieren sie einen Punkt.

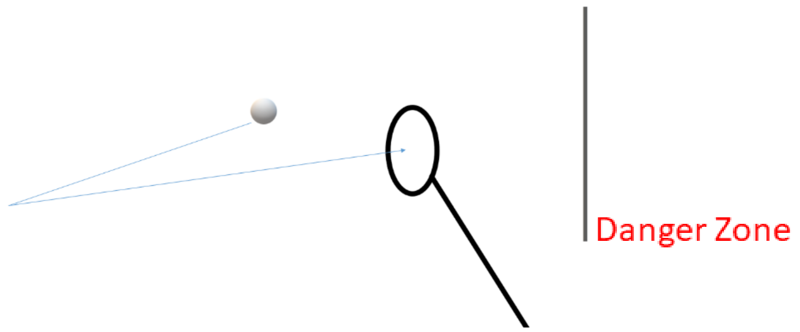
Sie können jedes Spiel solange spielen wie sie möchten. Informieren sie nur dem Versuchsleiter, wenn sie bereit für das nächste Spiel sind. Das Ziel in jedem Spiel ist es, so viele Punkte wie möglich zu sammeln.

Information of the Study

In this study, you will be asked to play two mini-games involving a levitating particle. You will have the opportunity to play these games in two different environments: in the real world and in virtual reality.

Please ask any questions you have to the experimenter before the start of each game session. You can take breaks between the sessions.

BeadBounce



In this mini-game you need to prevent the levitating particle (white sphere in the figure) from going into the Danger Zone. The game is played in the left half of the levitation interface. The right half is the danger zone. The levitating particle will start moving in random directions and bounce off the walls. When the particle is moving towards the right, you have to hit it with your racket (controller), so that it bounces back to the left. Each time you hit the particle with the racket you receive 1 point, when the particle wonders off to the danger zone you lose 2 points.

LeviShooter

In this mini-game you will shoot at levitating particles with a laser gun controller. The levitating particle will again move in random directions. You will need to aim at it accurately. You will notice that your gun controller is aiming at the particle, by the reflection of the laser beam from it. When you successfully hit the particle, you have to wait 2s until you can shoot again. When you hit the particle you receive 5 points, when you miss it you lose 1.

You can play as long as you wish to. Just inform the experimenter when you are ready to move on to the next game. The goal of each game is to collect as many points as possible.

Inwiefern haben Sie das Gefühl, dass das VR-Interface dem echten ähnelt?

Kein bisschen 1 2 3 4 5 6 7 sehr ähnlich

Bis zu welchem Grad hatten Sie das Gefühl, Sie hätten den schwebenden Cursor in VR unter Kontrolle?

Kein bisschen 1 2 3 4 5 6 7 sehr unter Kontrolle

Bis zu welchem Grad hatten Sie das Gefühl, den schwebenden Cursor in der realen Welt unter Kontrolle zu haben?

Kein bisschen 1 2 3 4 5 6 7 sehr unter Kontrolle

Inwiefern hatten Sie in der VR-Umgebung das Gefühl, dass Sie sich wirklich in einem Raum befinden, in dem sie eine Aufgabe mit einem schwebenden Cursor ausführen?

Kein bisschen 1 2 3 4 5 6 7 komplett

To what degree did you feel the VR interface was similar to the real one?

Not at all 1 2 3 4 5 6 7 Extremely similar

To what degree did you feel you were in control of the levitating cursor in VR?

Not at all 1 2 3 4 5 6 7 Extremely in control

To what degree did you feel you were in control of the levitating cursor in the real world?

Not at all 1 2 3 4 5 6 7 Extremely in control

In the VR environment, to what extend did you feel you were really inside of a room performing a task with a levitating cursor?

Not at all 1 2 3 4 5 6 7 Completely

BeadBounce in VR

Mit den folgenden Aussagen werden Sie gebeten, über Ihre Erfahrungen mit dem Levitator nachzudenken. Verwenden Sie für jede Aussage die angegebene Skala, um anzugeben, inwieweit diese auf Sie zutrifft. The following statements ask you to reflect on your experience of engaging with the Levitator. For each statement, please use the provided scale to indicate what is truest for you.

Stimme überhaupt nicht zu *Strongly Disagree*	**Stimme nicht zu** *Disagree*	**Stimme weder zu noch nicht zu** *Neither agree not disagree*	**Stimme zu** *Agree*	**Stimme komplett zu** *Strongly agree*
1	**2**	**3**	**4**	**5**

Ich war so in dieser Erfahrung vertieft, dass ich das Zeitgefühl verloren habe. *I was so involved in this experience that I lost track of time.*

1 2 3 4 5

Ich fand es nervig den Levitator zu bedienen. *I felt annoyed while interacting with the Levitator.*

1 2 3 4 5

Während ich den Levitator benutzte verlor ich die Welt um mich aus dem Blick. *When I was interacting with the Levitator, I lost track of the world around me.*

1 2 3 4 5

Die Zeit ist während der Interaktion mit dem Levitator schnell vergangen. *The time I spend interacting with the Levitator just slipped away.*

1 2 3 4 5

Ich würde den Levitator meiner Familie und meinen Freunden empfehlen. *I would recommend the Levitator to my family and friends.*

1 2 3 4 5

Die Erfahrung war bereichernd für mich. *My experience was rewarding.*

1 2 3 4 5

Ich konnte, während ich mit dem Levitator interagierte, einige der Dinge, die ich tun musste, nicht tun. *I could not do some of the things I needed to do while interacting with the Levitator.*

1 2 3 4 5

Diese Erfahrung hat Spaß gemacht. *This experience was fun.*

1 2 3 4 5

Den Levitator zu nutzen hat sich gelohnt. *Using the Levitator was worthwhile.*

1 2 3 4 5

Ich habe die Dinge um mich ausgeblendet während ich den Levitator benutzte. *I blocked out things around me when I was interacting with the Levitator.*

1 2 3 4 5

Stimme überhaupt nicht zu *Strongly Disagree*	Stimme nicht zu *Disagree*	Stimme weder zu noch nicht zu *Neither agree not disagree*	Stimme zu *Agree*	Stimme komplett zu *Strongly agree*
1	**2**	**3**	**4**	**5**

Ich war frustriert im Umgang mit dem Levitator. *I felt frustrated while interacting with the Levitator.*

1 2 3 4 5

Ich fühlte mich in diese Erfahrung hineingezogen. *I was really drawn into this experience.*

1 2 3 4 5

Den Levitator zu bedienen war anstrengend. *Interacting with the Levitator was taxing.*

1 2 3 4 5

Mich hat es demotiviert den Levitator zu bedienen. *I felt discouraged while interacting with the Levitator.*

1 2 3 4 5

Ich fand den Levitator ästhetisch ansprechend. *This Levitator was aesthetically appealing.*

1 2 3 4 5

Der Levitator war attraktiv. *This Levitator was attractive.*

1 2 3 4 5

Ich würde den Levitator aus Neugier weiter benutzen (nach Abschluss der offiziellen Aufgabe). *I would continue to use the Levitator out of curiosity (after finishing the official task).*

1 2 3 4 5

Ich fühlte mich einbezogen in diese Erfahrung. *I felt involved in this experience.*

1 2 3 4 5

Mir haben die Graphiken und die Bilder des Levitators gefallen. *I liked the graphics and images of the Levitator.*

1 2 3 4 5

Die Erfahrung verlief nicht wie geplant. *This experience did not work out the way I planned.*

1 2 3 4 5

Ich betrachte meine Erfahrung als Erfolg. *I consider my experience a success.*

1 2 3 4 5

Ich habe während dieser Erfahrung losgelassen. *During this experience I let myself go.*

1 2 3 4 5

Ich habe mich in dieser Erfahrung verloren. *I lost myself in this experience.*

1 2 3 4 5

Der Levitator sprach meine visuellen Sinne an. *The Levitator appealed to my visual senses.*

1 2 3 4 5

Stimme überhaupt nicht zu *Strongly Disagree*	Stimme nicht zu *Disagree*	Stimme weder zu noch nicht zu *Neither agree not disagree*	Stimme zu *Agree*	Stimme komplett zu *Strongly agree*
1	**2**	**3**	**4**	**5**

Das Bildschirm-Layout des Levitators war optisch ansprechend. *The screen layout of the Levitator was visually pleasing.*

1 2 3 4 5

Die Erfahrung war fordernd. *This experience was demanding.*

1 2 3 4 5

Ich hatte das Gefühl bei der Interaktion mit dem Levitator die Kontrolle zu haben. *I felt in control while interacting with the Levitator.*

1 2 3 4 5

Ich war vertieft in diese Erfahrung. *I was absorbed in this experience.*

1 2 3 4 5

Ich fand den Levitator verwirrend zu benutzen. *I found the Levitator confusing to use.*

1 2 3 4 5

Der Inhalt des Levitators hat meine Neugier geweckt. *The content of the Levitator incited my curiosity.*

1 2 3 4 5

Interview-Fragen

Benutzer-Studie: User Engagement mit einem Levitations-Simulator

1. Beschreiben Sie die Erfahrung beim Spielen jedes Spiels in Ihren eigenen Worten.

2. Welches Spiel und in welcher Umgebung hat Ihnen am besten gefallen? Was hat am meisten Spaß gemacht?

3. Haben Sie in beiden Situationen das Gefühl, den Schläger/die Waffe unter Kontrolle zu haben? Haben sie sich so verhalten, wie Sie es erwartet hätten?

4. Wie ähnlich waren sich die beiden Umgebungen? Was hat sich gleich angefühlt? Was war anders?

5. Wenn man Ihnen eine Demo der Spiele in VR gegeben hätte, würden Sie sich dann auf die Spiele auf dem echten Levitator vorbereitet fühlen?

6. Was würden Sie verbessern (in jeder Umgebung und jedem Spiel)? Gibt es etwas, das Sie ändern würden, um das Spielerlebnis in VR dem Erlebnis mit dem echten Levitator ähnlicher zu machen?

Interview Questions

User Study: User Engagement with a Levitation Simulator

1. Describe the experience of playing each game in your own words.
2. Which game and in which setting did you like the best? What was the most fun?
3. Did you feel in control of the racket/gun in both settings? Did they behave the way you would expect them to?
4. How similar were the two settings? What felt the same? What was different?
5. If you were given a demo of the games in VR, would you feel prepared for the games on the real levitator?
6. What would you improve (in any setting and any game)? Is there anything that you would change to make the gameplay experience in VR more similar to the experience with the real levitator?

Appendix B

HaptiRead: User Study Materials

This appendix contains the following forms and questionnaires (provided in German and English) used in the user studies presented in Chapter 6:

- Demographics Questionnaire
- Study Information Sheet
- Questions During and After each Session
- SUS Questionnaire
- Semi-structured Interview Questions

Please note that we omit the Consent Form here, as it is similar to the one provided in Appendix A.

DEMOGRAPHIC INFORMATION (English)

ID Number	
Age	
Gender	Male Female
Handedness	Right Left
Occupational status	
Are you taking any medication?	Yes No If yes, please specify
Do you play any musical instrument?	Yes No If yes, please specify
If yes, with what frequency do you play the instrument?	Everyday Every week Every month Sometimes during the year
Do you practice any sport?	Yes No If yes, please specify
If yes, with what frequency do you do sports?	Everyday Every week Every month Sometimes during the year

How often do you use Braille in your daily life?	Everyday Every week Every month Sometimes during the year Never
For what do you currently use Braille?	
How many years of Braille experience do you have?	
Were you born blind or was your visual impairment caused later in your life by an accident or an illness?	born blind accident illness If illness, please specify
How much percent visual capability do you have per eye?	Left eye Right eye

DEMOGRAPHIC INFORMATION (Deutsch)

ID Nummer	
Alter	
Geschlecht	Männlich Weiblich
Händigkeit	Rechts Links
Aktueller Beruf	
Nehmen Sie Medikamente?	Ja Nein Falls ja, bitte spezifizieren:
Spielen Sie ein Musikinstrument	Ja Nein Falls ja, bitte spezifizieren
Falls Ja, wie häufig spielen Sie das Instrument	täglich wöchentlich monatlich machmal im Jahr
Treiben Sie Sport?	Ja Nein Falls ja, bitte spezifizieren
Falls ja, wie häufig treiben Sie Sport?	täglich wöchentlich monatlich machmal im Jahr
Wie häufig benutzen Sie Braille im täglichen Leben?	täglich wöchentlich monatlich machmal im Jahr Never

Wofür benutzen Sie Braille momentan?	
Seit wie vielen Jahren können Sie Braille lesen?	
Wurden Sie blind geboren oder ist ihre visuelle Einschränkung von einem Unfall oder einer Krankheit?	geburtsblind Unfall Krankheit Falls Krankheit, bitte spezifizieren
Wie viel Prozent Sehvermögen haben Sie auf jedem Auge?	Linkes Auge Rechtes Auge

Information of the study

At first, we will verify your Braille reading capabilities. You will be presented with 5 random Braille numbers. Please read the numbers out loud.

After that we will present numbers with a ultrasonic technique. We will use different methods of tactile feedback to represent the different numbers. You will have to place your dominant hand in mid-air to perceive these tactile sensations. More specifically, we will ask you to place your hand roughly 20 cm above a board that provides the tactile feedback. However, for each trial you are free to move your hand around the board to explore the tactile sensations. During emitting the sensation, the board will produce a buzzing sound. To prevent you from relying on the sound instead of the tactile sensation we will ask you to wear headphones. If you feel uncomfortable at any time during the experiment, please say so and the experiment will halt until you are ready to continue.

Your task is to recognize which number the tactile feedback is representing. When you have recognized a number say the number out loud. Before beginning with the experiment, we will first have a training session where you can familiarize with the tactile sensation, the different methods used, and your task. In this training session, we will give you feedback of your performance by indicating whether your answer was correct or not. However, during the actual experiment trials, no more feedback of your performance will be provided. Please ask any questions you have to the experimenter during the training session.

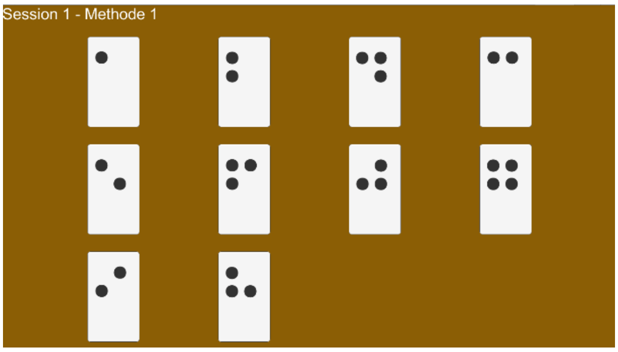


Figure 1. Different patterns that can be represented by the tactile feedback. You will have to select on a computer screen which is the pattern/shape you are perceiving on your hand (tactile perceptions).

The experiment will consist of three different sessions, where we are testing three different methods to provide the feedback. The methods will appear in the same order, as they appeared in the training session. Between each of the sessions you will have to take a break to rest you hand. You will be told when to take a break. However, try to keep your hand above the board while the session is in progress. After each session and at the end of the experiment we will also kindly ask you to answer some questions to better understand your experience. If you do not have inconvenience, we will audio record a short interview that will carry out at the end of the study to better understand your opinion.

Informationen zur Studie

In dieser Studie werden wir Sie bitten, Muster zu erkennen, indem Sie Ihren Tastsinn nutzen. Wir werden unterschiedliche Methoden der taktilen Feedback verwenden, um verschiedene Formen darzustellen. Sie werden Ihre Hand in der Luft platzieren, um diese taktilen Empfindungen wahrzunehmen. Genauer gesagt, werden wir Sie bitten, Ihre linke Hand ca. 20 cm über einem Feld zu platzieren, welches das haptische Feedback ausgibt. Bei jedem Versuch steht es Ihnen jedoch frei, Ihre Hand über dem Feld zu bewegen, um auf diese Weise das Muster zu ertasten.

Ihre Aufgabe ist es, zu erkennen, welcher Form das haptische Feedback auf dem Computerbildschirm entspricht. Sie geben Ihre Antwort ein, indem Sie die grafische Entsprechung auf dem Computerbildschirm anklicken, von der Sie denken, dass sie der haptischen Form, die sie spüren, entspricht. Dem eigentlichen Experiment geht ein Training voraus, in dem Sie das haptische Empfinden, die verschiedenen verwendeten Methoden und Ihre Aufgabe kennenlernen können. In dieser Trainingseinheit geben wir Ihnen zudem noch Feedback über Ihre Leistung, indem wir Ihnen mitteilen, ob Ihre Antwort richtig war oder nicht. Während des eigentlichen Experimentes wird jedoch kein Feedback mehr zu Ihrer Leistung gegeben. Bitte stellen Sie dem Versuchspersonal während der Trainingseinheit alle Fragen, die Sie haben.

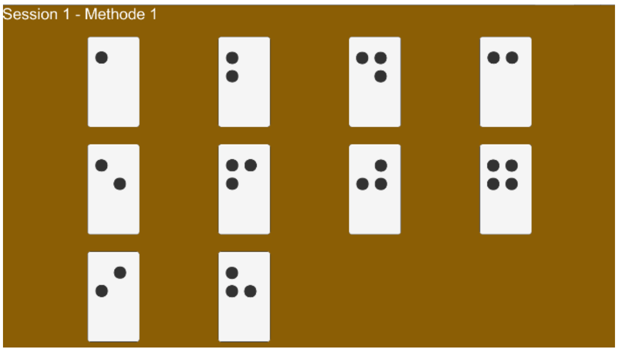


Abbildung 1. Verschiedene Muster, die durch haptisches Feedback dargestellt werden können. Sie müssen auf einem Computerbildschirm auswählen, welches das Muster/ die Form ist, das/die Sie auf Ihrer Hand wahrnehmen (taktile Wahrnehmungen).

Das Experiment wird aus drei verschiedenen Phasen bestehen, in denen wir drei verschiedene Methoden testen, um das haptische Feedback auszugeben. Die verschiedenen Methoden werden in derselben Reihenfolge wie in den Trainingssessions abgespielt. Zwischen den einzelnen Phasen werden Sie gebeten, eine Pause einzulegen, um Ihre Hand auszuruhen. Es wird Ihnen auf dem Bildschirm angezeigt, wann Sie diese Pause einlegen sollen. Versuchen Sie jedoch während der aktiven Phase Ihre Hand über dem Brett zu halten. Nach jeder Sitzung und am Ende des Experiments bitten wir Sie außerdem, einige Fragen zu beantworten, um Ihre Erfahrungen besser zu verstehen. Wenn Sie keine Einwände haben, werden wir am Ende ein kurzes Interview mit Ihnen durchführen, um Ihre Meinung besser verstehen zu können. Bei diesem Interview wird der Ton elektronisch mitgeschnitten werden.

After each session

How well were you able to differentiate between different patterns using method 1?

not at all 1 2 3 4 5 6 7 perfectly

How well were you able to differentiate between different patterns using method 2?

not at all 1 2 3 4 5 6 7 perfectly

How well were you able to differentiate between different patterns using method 3?

not at all 1 2 3 4 5 6 7 perfectly

At the end

How well were you able to differentiate between different patterns using method 1?

not at all 1 2 3 4 5 6 7 perfectly

How well were you able to differentiate between different patterns using method 2?

not at all 1 2 3 4 5 6 7 perfectly

How well were you able to differentiate between different patterns using method 3?

not at all 1 2 3 4 5 6 7 perfectly

How comfortable did you feel using method 1?

not comfortable at all 1 2 3 4 5 6 7 extremely comfortable

How comfortable did you feel using method 2?

not comfortable at all 1 2 3 4 5 6 7 extremely comfortable

How comfortable did you feel using method 3?

not comfortable at all 1 2 3 4 5 6 7 extremely comfortable

Please report any positive or negative experiences you had during method 1.

Please report any positive or negative experiences you had during method 2.

Please report any positive or negative experiences you had during method 3.

System Usability Scale (SUS)

	Strongly Disagree	Somewhat Disagree	Neutral	Somewhat Agree	Strongly Agree
1. I think I would like to use this system frequently.	☐	☐	☐	☐	☐
2. I found the system unnecessarily complex.	☐	☐	☐	☐	☐
3. I thought the system was easy to use.	☐	☐	☐	☐	☐
4. I think that I would need the support of a technical person to be able to use this system.	☐	☐	☐	☐	☐
5. I found the various functions in this system were well integrated.	☐	☐	☐	☐	☐
6. I thought there was too much inconsistency in this system.	☐	☐	☐	☐	☐
7. I would imagine that most people would learn to use this system very quickly.	☐	☐	☐	☐	☐
8. I found the system very cumbersome to use.	☐	☐	☐	☐	☐
9. I felt very confident using the system.	☐	☐	☐	☐	☐
10. I needed to learn a lot of things before I could get going with this system.	☐	☐	☐	☐	☐

System Usability Scale (SUS)

	Stimme überhaupt nicht zu	Stimme eher nicht zu	Neutral	Stimme einigermaßen zu	Stimme voll zu
1. Ich kann mir sehr gut vorstellen, das System regelmäßig zu nutzen.	☐	☐	☐	☐	☐
2. Ich empfinde das System als unnötig komplex.	☐	☐	☐	☐	☐
3. Ich empfinde das System als einfach zu nutzen.	☐	☐	☐	☐	☐
4. Ich denke, dass ich technischen Support brauchen würde, um das System zu nutzen.	☐	☐	☐	☐	☐
5. Ich finde, dass die Kernfunktionen des Systems gut integriert sind.	☐	☐	☐	☐	☐
6. Ich finde, dass es im System zu viele Inkonsistenzen gibt.	☐	☐	☐	☐	☐
7. Ich kann mir vorstellen, dass die meisten Leute das System schnell zu beherrschen lernen.	☐	☐	☐	☐	☐
8. Ich empfinde die Bedienung als sehr umständlich.	☐	☐	☐	☐	☐
9. Ich habe mich bei der Nutzung des Systems sehr sicher gefühlt.	☐	☐	☐	☐	☐
10. Ich musste eine Menge Dinge lernen, bevor ich mit dem System arbeiten konnte.	☐	☐	☐	☐	☐

1. What is your preferred method and why?
2. What do you think can be improved in the way the Braille numbers were presented by the device? What would you change to facilitate the reading?
3. How comfortable did you feel that the numbers were projected onto your palm compared to the fingers?
4. Did you experience fatigue or other impleasentries during the study?
5. What kind of technical aids do you use in your daily life to read Braille?
6. How useful do you think mid-air haptic technology in general can be for visually-impaired people?
7. In what type of context or scenarios do you see yourself using this type of haptic interface?

1. Was ist Ihre bevorzugte Methode und warum?
2. Was glauben Sie könnte an der Art wie die Zahlen dargestellt werden verbessert werden? Was würden Sie ändern, um das lesen zu vereinfachen?
3. Wie angenehm fanden Sie die Projektion der Zahlen auf ihre Handfläche im Vergleich zu ihrem Finger?
4. Haben Sie während der Studie Ermüdung oder andere Unannehmlichkeiten verspürt?
5. Welche sonstigen technischen Hilfsmittel benutzen Sie in ihrem täglichen Leben?
6. Wie nützlich glauben Sie ist haptische Technologie in der Luft für Menschen mit Sehbehinderung?
7. In welchem Kontext oder Szenario können Sie sich vorstellen diese Technologie zu nutzen?

Appendix C

OptiTrap: Trajectory Optimisation Algorithm

This appendix contains the following Matlab implementation files of the *OptiTrap* algorithm, presented in Chapter 7:

- OCP formulation
- Optimiser Setup
- Auxiliary Functions
- Shape Definitions

C.1 OCP Formulation

```
%% Minimum Time Path for a Levitating Particle in an a Acoustic Field

% ---- initialize ----------------
clear vars; close all; clc;
cd 'C:\Optimal Control\optitrap_alg_final'

import casadi.*
opti = casadi.Opti();

addpath(genpath('shapes'))
addpath(genpath('helperfunctions'))

set(groot, 'defaultAxesTickLabelInterpreter','latex');
set(groot, 'defaultLegendInterpreter','latex');

save_yes = true;
load_yes = false;

%1. CONFIGURE INPUT (Shape, speed, size)
% ---- pick a shape ----------------
POV_freq = 10;
num_shape = 1;

switch num_shape
    case 1
        shape = "circle";
        size_coeff = 0.035;
        POV_freq = 15;
    case 2
        shape = "cardioid";
        size_coeff = 0.035;
    case 3
        shape = "squircle";
        size_coeff = 0.0265;
    case 4
        shape = "fish";
        size_coeff = 0.0425;
    case 5
        shape = "flower";
        size_coeff = 0.036;
    case 6
        shape = "cat";
        size_coeff = 3e-5;
    case 7
        shape = "dolphin";
        size_coeff = 2e-5;
    case 8
        shape = "heart";
        size_coeff = 0.0017;
    otherwise
        warning('Unexpected number.')
end

% ---- pick a mode ----------------
num_mode = 1;

switch num_mode
    case 0
        mode = 'rest_to_rest';
    case 1
```

```
        mode = 'periodic';
    case 2
        mode = 'rampup';
    case 3
        mode = 'rampdown';
end

% ---- parameters----------------
if strcmp(mode,'periodic')
     N = ceil(10000/POV_freq);
else
     N = ceil(10000/POV_freq)*3;
end

%2. CONFIGURE SYSTEM

parameters_nlp_values = struct( ...
    'm',        0.07  ,... % mass of the particle [mg]
    'A_r',      0.0243, ... % maximum horizontal force [mN]
    'A_z',      0.042, ... % maximum vertical force [mN]
    'c_par',    1e8,... % weight of the control cost term in the cost
function [no unit]
    'arc_lim',  0.85    ...);

%2.b. Parameters for inversion
parameters_simulation_values = struct( ...
    'm',        parameters_nlp_values.m, ...
    'A_r',      parameters_nlp_values.A_r, ...
    'A_z',      parameters_nlp_values.A_z, ...
    'c_z1',     13.23667588,   ...  % 2*pi/c_z1 = 474.68
    'c_z2',     004.901845302,  ...  % 2*pi/c_z2 = 1281.8
    'c_z5',     004.901845302,  ...  % 2*pi/c_z5 = 1281.8 [same as c_z2]
    'c_z6',     21.56428358 ... % 2*pi/c_z6 = 291.37
);
% ---- parameters END -------------

%3. CREATE FUNCTIONS DEFINING THE SHAPE

shape_name = str2func(sprintf('OT%s',shape));
thSym = SX.sym('th'); %Create a symbolic variable (thSym)

fCas = 1e3*shape_name(thSym, size_coeff);
fCasJ = jacobian(fCas,thSym);
fCasJJ = jacobian(fCasJ,thSym);
fCasfun = Function('fCasfun',{thSym},{fCas});
fCasJfun = Function('fCasJfun',{thSym},{fCasJ});
fCasJJfun = Function('fCasJJfun',{thSym},{fCasJJ});

% ---- setup NLP --------------------
disp('(1/6) Setting up NLP')
tic
decision_vars = setup_final(mode, opti, N, parameters_nlp_values,
fCasJfun, fCasJJfun);
toc

% decision variables
Z = decision_vars.Z;        % states of virtual system
v = decision_vars.v;        % virtual control input
T = decision_vars.T;        % final time
```

```
z_N = decision_vars.z_N;    % auxiliary variables for the force
computation

% ---- initial guess for solver ----

if load_yes
    filename = sprintf('data/ocp_valid_%s_%d_%s.mat', shape, N, mode);
    load(filename);
    opti.set_initial(Z, Z_sol);
    opti.set_initial(v, v_sol);
    opti.set_initial(T, opt_T);
    opti.set_initial(z_N, z_N_sol);
else
    opti.set_initial(Z,zeros(size(Z)));
    opti.set_initial(v,zeros(size(v)));
    opti.set_initial(T, 500);

    tmpinit = min(0.5,parameters_nlp_values.arc_lim);
    opti.set_initial(z_N(1,:), zeros(1,N));
    opti.set_initial(z_N(2,:), tmpinit*ones(1,N));
    opti.set_initial(z_N(3,:), sqrt(1-(tmpinit*ones(1,N)).^2));
    opti.set_initial(z_N(4,:), zeros(1,N));
end

% ---- solve NLP -------------------
disp('(2/6) Solving NLP')
tic
p_opts = struct('expand',true);
s_opts = struct('max_iter',50000);
opti.solver('ipopt',p_opts, s_opts);
sol = opti.solve();
toc

% ---- Get and save solution ---------
Z_sol = sol.value(Z);
theta_sol = Z_sol(1,:);
theta_dot_sol = Z_sol(2,:);
opt_T = sol.value(T);
v_sol = sol.value(v);
z_N_sol = sol.value(z_N);

% if save_yes
%     filename = sprintf('data/ocp_valid_%s_%d_%s.mat', shape, N, mode);
%     save(filename,'v_sol','Z_sol','z_N_sol','opt_T');
% end

% ---- post-processing ------------
% 1a) Preliminaries: extract particle position, velocity, and
acceleration
bead_nlp_pos = full(fCasfun(theta_sol));
bead_nlp_vel = full(fCasJfun(theta_sol)) .* theta_dot_sol;
bead_nlp_acc = full(fCasJJfun(theta_sol)) .* theta_dot_sol.^2 +
full(fCasJfun(theta_sol)) .* [v_sol v_sol(1)]; % AF: dim(v_sol) =
dim(theta_sol) - 1

% 1b) Preliminaries: extract forces
force_nlp_unscaled = [zeros(size(z_N(1,:))); ...
                      z_N_sol(1,:).*z_N_sol(2,:); ...
                      z_N_sol(3,:).*z_N_sol(4,:)];
```

```
force_nlp = [parameters_nlp_values.A_r; parameters_nlp_values.A_r;
parameters_nlp_values.A_z] .* force_nlp_unscaled;

% 3) Perform system inversion
disp('(4/6) System inversion')
[dd_num, force_num_unscaled] = sys_inv_num(N,
parameters_simulation_values, force_nlp_unscaled);
force_num = [parameters_nlp_values.A_r; parameters_nlp_values.A_r;
parameters_nlp_values.A_z] .* force_num_unscaled;

% plot deltas
figure
plot(dd_num','-o')
legend('dy-num','dz-num');

% plot forces
figure
plot(force_num','-')
title('Acoustic Force (F_x, F_y, F_z) after numerical system inversion')
legend('Fx-num', 'Fy-num', 'Fz-num')

% plot force differences
figure
plot(abs(force_num' - force_nlp'))
title('|F_{num} - F_{nlp}| (NOT scaled with A_r, A_z)')
legend('Fx-diff', 'Fy-diff', 'Fz-diff')

% 4) Calculate, save, and plot trap positions
% 4a) Calculate trap positions
disp('(5/6) Calculate and save trap positions')
traps_num = zeros(3,N);
traps_num(2:3, :) = bead_nlp_pos(2:3, 1:end-1) + dd_num;

% 4b) Save trap positions
csvwrite(sprintf('data/u_trap_solutions_%s_fsolve.csv', shape),
traps_num)
traps = traps_num./1000;
traps(4,:) = 1;
traps_line = reshape(traps,[],1)'/1;
filename = sprintf('data/%s_%s_%d_ready_to_play.csv', shape, mode, N);
csvwrite(filename,round(traps_line,5))

if save_yes
    filename = sprintf('data/ocp_valid_%s_%d_%s.mat', shape, N, mode);
    save(filename,'v_sol','Z_sol','z_N_sol','opt_T', 'traps_num');
end

% 4c) Plot trap positions
scaleFactor = 1e0;
figure
plot3(scaleFactor*traps_num(1,:), scaleFactor*traps_num(2,:),
scaleFactor*(traps_num(3,:)), '-o')
hold on
plot3(scaleFactor*bead_nlp_pos(1,:), scaleFactor*bead_nlp_pos(2,:),
scaleFactor*(bead_nlp_pos(3,:)),'-o')
grid on
xlabel('x')
ylabel('y')
zlabel('z')
for i = 1:10:length(traps_num(1,:))
```

```
    text(scaleFactor*traps_num(1,i), scaleFactor*(traps_num(2,i)+0.0001),
scaleFactor*(traps_num(3,i)+0.0001), num2str(i));
end
text(scaleFactor*traps_num(1,N), scaleFactor*(traps_num(2,N)+0.0001),
scaleFactor*(traps_num(3,N)+0.0001), num2str(N));
hold off
title('trap and bead [in mm]')
legend('trap','bead')
view(90,0)
saveas(gcf,'Bead-Trap-Num.png')

% 5) Simulation with obtained trap position and corresponding plots
% 5a) Simulation
disp('(6/6) Simulation with obtained trap positions')
scaleFactor = 1e0;
deltaT = opt_T / N;
force_trap = zeros(3, N);
force_trap_nlp = zeros(3, N);
bead_sim_trap_pos = zeros(3, N+1);
bead_sim_trap_vel = zeros(3, N+1);
bead_sim_trap_pos(:,1) = scaleFactor * bead_nlp_pos(:,1);
bead_sim_trap_vel(:,1) = bead_nlp_vel(:,1);
trap_u = traps_num;% - [0;0;1e3*0.1194];
sim_state = [bead_sim_trap_pos(:,1); bead_sim_trap_vel(:,1)];
for k=1:N
    force_trap(:,k) = F_acoustic(sim_state(1:3), trap_u(:,k),
parameters_simulation_values);
    force_trap_nlp(:,k) = F_acoustic(bead_nlp_pos(:,k), trap_u(:,k),
parameters_simulation_values);

    k1 = ode_rhs_new((k-1)*deltaT, deltaT, sim_state,
trap_u, parameters_simulation_values);
    k1(1:3) = k1(1:3) / scaleFactor;
    k2 = ode_rhs_new((k-1)*deltaT, deltaT, sim_state + deltaT/2*k1,
trap_u, parameters_simulation_values);
    k2(1:3) = k2(1:3) / scaleFactor;
    k3 = ode_rhs_new((k-1)*deltaT, deltaT, sim_state + deltaT/2*k2,
trap_u, parameters_simulation_values);
    k3(1:3) = k3(1:3) / scaleFactor;
    k4 = ode_rhs_new((k-1)*deltaT, deltaT, sim_state + deltaT*k3,
trap_u, parameters_simulation_values);

    sim_state = sim_state + deltaT/6*(k1+2*k2+2*k3+k4);
    bead_sim_trap_pos(:,k+1) = sim_state(1:3);
    bead_sim_trap_vel(:,k+1) = sim_state(4:6);
end

% 5b) Plots of the simulation
figure
quiver(bead_sim_trap_pos(2,:), bead_sim_trap_pos(3,:),
bead_sim_trap_vel(2,:), bead_sim_trap_vel(3,:), 0)
title('Particle position and velocity (trap simulation)')

figure()
scaleFactor = 1;
plot3(scaleFactor*trap_u(1,:), scaleFactor*trap_u(2,:),
scaleFactor*(trap_u(3,:)), '-o')
hold on
plot3(scaleFactor*bead_nlp_pos(1,:), scaleFactor*bead_nlp_pos(2,:),
scaleFactor*(bead_nlp_pos(3,:)),'-o')
```

```
hold on
plot3(scaleFactor*bead_sim_trap_pos(1,1:end),
scaleFactor*bead_sim_trap_pos(2,1:end),
scaleFactor*(bead_sim_trap_pos(3,1:end)),'-o')
grid on
xlabel('x')
ylabel('y')
zlabel('z')
hold off
title(sprintf('[SIMULATION] trap and bead [in mm] - %s, N=%d traps,
optT=%dms',shape, N, round(opt_T,3,'significant')))
legend('trap','bead','bead (sim)')
view(90,0)
saveas(gcf,'Bead-Trap-Sim.png')
```

C.2 Optimiser Setup

```
function decision_vars = setup_final(mode, opti, N,
parameters_nlp_values, fCasJfun, fCasJJfun)
    import casadi.*

    nZ = 2;
    n_zeta = 4;

    % initial conditions
    theta_0 = 0;
    % terminal constraints
    if strcmp(mode,'rampup') || strcmp(mode,'rampdown')
        theta_final = 4*pi;
    else
        theta_final = 2*pi;
    end

    % ---- decision variables ---------
    Z = opti.variable(nZ,N+1); % states of virtual system
    theta = Z(1,:);
    theta_dot = Z(2,:);
    v = opti.variable(N)';   % virtual control input
    T = opti.variable();      % final time
    z_N = opti.variable(n_zeta,N);

    q_d = fCasJfun(theta);
    q_dd = fCasJJfun(theta);

    decision_vars = struct( ...
        'Z',Z,       ... % virtual state [m, m/s]
        'v',v,       ... % virtual control [m/s^2]
        'T',T,       ... % time [s]
        'z_N',z_N    ... % virtual variables for sin/cos terms.
    );

    % ---- objective ------------------
    opti.minimize(T+parameters_nlp_values.c_par/N*sum(v.^2));

    % ---- dynamic constraints --------
    f1 = @(x,v) [x(2); v]; % ODE virtual system

    dt = T/N;
    for k=1:N
        % Runge-Kutta 4 integration
        k1 = f1(Z(:,k),         v(k));
        k2 = f1(Z(:,k)+dt/2*k1, v(k));
        k3 = f1(Z(:,k)+dt/2*k2, v(k));
        k4 = f1(Z(:,k)+dt*k3,   v(k));

        z_next = Z(:,k) + dt/6*(k1+2*k2+2*k3+k4);
        con=Z(:,k+1)==z_next;
        opti.subject_to(con);

        % --- !!! constrains on the z_N terms !!! ---
        % 1) Constrain relevant z_N's to make inversion feasible:
        opti.subject_to(-parameters_nlp_values.arc_lim <= z_N(1:4,k) <=
parameters_nlp_values.arc_lim);

        % 2) trigonometric Pythagoras to give some knowledge about the
        % problem structure to the optimizer:
        opti.subject_to(z_N(2,k)^2 + z_N(3,k)^2 == 1)
```

```
        % --- !!! Particle dynamics as constraints !!! ---
        dyn_lhs = parameters_nlp_values.m*(q_dd(:,k).*theta_dot(k)^2 +
q_d(:,k).*v(k));
        con6 = dyn_lhs(2:3) -
[parameters_nlp_values.A_r*z_N(1,k)*z_N(2,k);
parameters_nlp_values.A_z*z_N(3,k)*z_N(4,k)];
        opti.subject_to(con6(:) == 0);
    end

    % ---- boundary conditions --------
    % theta position
    opti.subject_to(theta(1)==theta_0);
    opti.subject_to(theta(N+1)==theta_final);
    % theta velocity
    if strcmp(mode,'periodic')
        opti.subject_to(theta_dot(1)==theta_dot(N+1));
    elseif strcmp(mode,'rampup')
        opti.subject_to(theta(ceil(2/3*N))==2*pi);
        opti.subject_to(theta_dot(1)==0);
        opti.subject_to(theta_dot(ceil(2/3*N))==theta_dot(N+1));
    elseif strcmp(mode,'rampdown')
        opti.subject_to(theta(ceil(1/3*N))==2*pi);
        opti.subject_to(theta_dot(N+1)==0);
        opti.subject_to(theta_dot(1)==theta_dot(ceil(1/3*N)));
    else
        opti.subject_to(theta_dot(1)==0); % from stand-still
        opti.subject_to(theta_dot(N+1)==0); % to stand-still
    end

    % ---- misc. constraints  ----------
    opti.subject_to(T>=0); % time must be positive
end
```

C.3 Auxiliary Functions

```
function force = F_acoustic(bead_p, trap_u, params)
    dr = sqrt((trap_u(1) - bead_p(1))^2 + (trap_u(2) - bead_p(2))^2);
    dz = trap_u(3) - bead_p(3);
    F_z = params.A_z * sin(dz * (2*pi)/params.c_z2) * cos(dr *
(2*pi)/params.c_z6);
    if abs(dr) <= 1e-16
        force = [0; 0; F_z];
    else
        F_r = params.A_r * sin(dr * (2*pi)/params.c_z1) * cos(dz *
(2*pi)/params.c_z2);
        force = [F_r * (trap_u(1) - bead_p(1))/dr; F_r * (trap_u(2) -
bead_p(2))/dr; F_z];
    end
end
```

```
function result = ode_rhs_new(t, trap_dt, state, trap_u, params)
    scaleTemp = 1e0;
    trap_number = round(t/trap_dt)+1;
    if trap_number == length(trap_u(1,:))+1
        trap_number = length(trap_u(1,:));
    end
    result = [state(4:6); scaleTemp*F_acoustic(state(1:3)/scaleTemp,
trap_u(:,trap_number)/scaleTemp, params) / params.m];
end
```

```
function [bead_sim_forces_pos, bead_sim_forces_vel] =
simulation_force(opt_T, N, forces, params, initial_state)
    deltaT = opt_T / N;
    bead_sim_forces_pos = zeros(3, N+1); with forces from the NLP
solver
    bead_sim_forces_vel = zeros(3, N+1); % bead simulated velocity
with forces from the NLP solver
    bead_sim_forces_pos(:,1) = initial_state(1:3); % initial
condition added
    bead_sim_forces_vel(:,1) = initial_state(4:6); % initial
condition added
    sim_state = [bead_sim_forces_pos(:,1);
    bead_sim_forces_vel(:,1)];
    for k=1:N
        tmp = sim_state;
        k1 = [tmp(4:6); forces(:,k) / params.m];
        tmp = sim_state + deltaT/2*k1;
        k2 = [tmp(4:6); forces(:,k) / params.m];
        tmp = sim_state + deltaT/2*k2;
        k3 = [tmp(4:6); forces(:,k) / params.m];
        tmp = sim_state + deltaT*k3;
        k4 = [tmp(4:6); forces(:,k) / params.m];
        sim_state = sim_state + deltaT/6*(k1+2*k2+2*k3+k4);
        bead_sim_forces_pos(:,k+1) = sim_state(1:3);
        bead_sim_forces_vel(:,k+1) = sim_state(4:6);
    end
end
```

```
function [deltas_num, force_num] = sys_inv_num(N, params,
force_nlp_unscaled) % numerically solving for delta_x, delta_y,
delta_z by solving 0=M(...).
    flaags = 10*ones(1,N);
    options2 =
optimoptions('fsolve','Display','off','Algorithm','levenberg-
marquardt','OptimalityTolerance',1e-12,'FunctionTolerance',1e-8);
    options = optimoptions('fsolve','Display','final-
detailed','Algorithm','trust-region-dogleg','OptimalityTolerance',1e-
12,'FunctionTolerance',1e-8);

    force_num = zeros(3,N);
        deltas_num = zeros(2,N);
        [deltas_num(:,1),fval,flaags(1),ouput] =
fsolve(@(dd)eqs_2d_ver(dd, params, force_nlp_unscaled(:,1)),
0.5*ones(2,1), options);
        force_num(2:3,1) = f_2d_ver(deltas_num(:,1), params);
        for k=2:N
            [deltas_num(:,k),fval,flaags(k),ouput] =
fsolve(@(dd)eqs_2d_ver(dd, params, force_nlp_unscaled(:,k)),
0.5*ones(2,1), options);
            if flaags(k) < 1
                [deltas_num(:,k),fval,flaags(k),ouput] =
fsolve(@(dd)eqs_2d_ver(dd, params, force_nlp_unscaled(:,k)),
deltas_num(:,k-1), options2);
            end
            force_num(2:3,k) = f_2d_ver(deltas_num(:,k), params);
        end
end

function eqs = eqs_2d_ver(dd, params, force_nlp_current)
    force = f_2d_ver(dd, params);
    eqs = force - force_nlp_current(2:3);
end

function force_unscaled = f_2d_ver(dd, params)
    force_unscaled =
[sin((2*pi)/params.c_z1*sqrt(dd(1)^2))*cos((2*pi)/params.c_z2*dd(2))*
dd(1)/sqrt(dd(1)^2);

sin((2*pi)/params.c_z2*dd(2))*cos((2*pi)/params.c_z6*sqrt(dd(1)^2))];
end
```

C.4 Shape Definitions

```
function q=OTcardioid(theta,c)
h_center = 0.1194;
q=[0*theta;c*cos(theta).*(1+cos(theta))-
c;c*sin(theta).*(1+cos(theta))+h_center];
end
```

```
function q = OTcircle(theta,r)
h_center = 0.1194;
theta=theta+pi/2;
q=[0*theta; r*sin(theta); r*(1-cos(theta))+h_center-r];
end
```

```
function q = OTfish(theta,r)
h_center = 0.1194;
q_y=cos(theta)-(sin(theta).^2/sqrt(2));
q_z=sin(theta).*cos(theta);
q=[0*theta; r*q_y; r*q_z+h_center];
end
```

```
function q = OTflower(theta,r)
h_center = 0.1194;
sc = 0.5;
q_y = sin(3*sc*theta).*cos(sc*theta);
q_z = sin(sc*theta).*sin(3*sc*theta);
q=[0*theta; r*q_y; r*q_z+h_center];
end
```

```
function q = OTheart(theta,r)
h_center = 0.1194;
q_y = 16*sin(theta).^3;
q_z = 13*cos(theta)-5*cos(2*theta)-2*cos(3*theta)-cos(4*theta);
q=[0*theta; r*q_y; r*q_z+h_center];
end
```

```
function q=OTsquircle(theta,r)
s = 0.9;
h_center = 0.1194;
q=[0*theta; r*cos(theta); r*sin(theta)./(sqrt(1-
s*cos(theta).^2))+h_center];
end
```

```
function q = OTdolphin(t, r)
h_center = 0.1194;
q_y=4/23*sin(62/33-58*t)+8/11*sin(10/9-56*t)+17/24*sin(38/35-
55*t)+30/89*sin(81/23-54*t)+3/17*sin(53/18-53*t)+21/38*sin(29/19-
52*t)+11/35*sin(103/40-51*t)+7/16*sin(79/18-50*t)+4/15*sin(270/77-
49*t)+19/35*sin(59/27-48*t)+37/43*sin(71/17-47*t)+sin(18/43-
45*t)+21/26*sin(37/26-44*t)+27/19*sin(111/32-42*t)+8/39*sin(13/25-
41*t)+23/30*sin(27/8-40*t)+23/21*sin(32/35-37*t)+18/37*sin(91/31-
36*t)+45/22*sin(29/37-35*t)+56/45*sin(11/8-33*t)+4/7*sin(32/19-
32*t)+54/23*sin(74/29-31*t)+28/19*sin(125/33-30*t)+19/9*sin(73/27-
29*t)+16/17*sin(737/736-28*t)+52/33*sin(130/29-27*t)+41/23*sin(43/30-
25*t)+29/20*sin(67/26-24*t)+64/25*sin(136/29-23*t)+162/37*sin(59/34-
21*t)+871/435*sin(199/51-20*t)+61/42*sin(58/17-
19*t)+159/25*sin(77/31-17*t)+241/15*sin(94/31-
13*t)+259/18*sin(114/91-12*t)+356/57*sin(23/25-
11*t)+2283/137*sin(23/25-10*t)+1267/45*sin(139/42-
9*t)+613/26*sin(41/23-8*t)+189/16*sin(122/47-6*t)+385/6*sin(151/41-
5*t)+2551/38*sin(106/35-4*t)+1997/18*sin(6/5-2*t)+43357/47*sin(81/26-
t)-4699/35*sin(3*t+25/31)-1029/34*sin(7*t+20/21)-
250/17*sin(14*t+7/40)-140/17*sin(15*t+14/25)-194/29*sin(16*t+29/44)-
277/52*sin(18*t+37/53)-94/41*sin(22*t+33/31)-57/28*sin(26*t+44/45)-
128/61*sin(34*t+11/14)-111/95*sin(38*t+55/37)-85/71*sin(39*t+4/45)-
25/29*sin(43*t+129/103)-7/37*sin(46*t+9/20)-17/32*sin(57*t+11/28)-
5/16*sin(59*t+32/39);
q_z=5/11*sin(163/37-59*t)+7/22*sin(19/41-58*t)+30/41*sin(1-
57*t)+37/29*sin(137/57-56*t)+5/7*sin(17/6-55*t)+ 11/39*sin(46/45-
52*t)+25/28*sin(116/83-51*t)+25/34*sin(11/20-47*t)+8/27*sin(81/41-
46*t)+44/39*sin(78/37-45*t)+11/25*sin(107/37-44*t)+7/20*sin(7/16-
41*t)+30/31*sin(19/5-40*t)+37/27*sin(148/59-39*t)+44/39*sin(17/27-
38*t)+13/11*sin(7/11-37*t)+28/33*sin(119/39-36*t)+27/13*sin(244/81-
35*t)+13/23*sin(113/27-34*t)+47/38*sin(127/32-
33*t)+155/59*sin(173/45-29*t)+105/37*sin(22/43-
27*t)+106/27*sin(23/37-26*t)+97/41*sin(53/29-25*t)+83/45*sin(109/31-
24*t)+81/31*sin(96/29-23*t)+56/37*sin(29/10-22*t)+44/13*sin(29/19-
19*t)+18/5*sin(34/31-18*t)+163/51*sin(75/17-17*t)+152/31*sin(61/18-
16*t)+146/19*sin(47/20-15*t)+353/35*sin(55/48-
14*t)+355/28*sin(102/25-12*t)+1259/63*sin(71/18-
11*t)+17/35*sin(125/52-10*t)+786/23*sin(23/26-6*t)+2470/41*sin(77/30-
5*t)+2329/47*sin(47/21-4*t)+2527/33*sin(23/14-3*t)+9931/33*sin(51/35-
2*t)-11506/19*sin(t+56/67)-2081/42*sin(7*t+9/28)-
537/14*sin(8*t+3/25)-278/29*sin(9*t+23/33)-107/15*sin(13*t+35/26)-
56/19*sin(20*t+5/9)-5/9*sin(21*t+1/34)-17/24*sin(28*t+36/23)-
21/11*sin(30*t+27/37)-138/83*sin(31*t+1/7)-10/17*sin(32*t+29/48)-
31/63*sin(42*t+27/28)-4/27*sin(43*t+29/43)-13/24*sin(48*t+5/21)-
4/7*sin(49*t+29/23)-26/77*sin(50*t+29/27)-
19/14*sin(53*t+61/48)+34/25*sin(54*t+37/26);
q=[0*t; r*q_y; r*q_z+h_center];
end
```

```
function q = OTcat(theta,r)
h_center = 0.1194;

q_y=-(721*sin(theta))/4+196/3*sin(2*theta)-86/3*sin(3*theta)-
131/2*sin(4*theta)+477/14*sin(5*theta)+27*sin(6*theta)-
29/2*sin(7*theta)+68/5*sin(8*theta)+1/10*sin(9*theta)+23/4*sin(10*theta)-
19/2*sin(12*theta)-
85/21*sin(13*theta)+2/3*sin(14*theta)+27/5*sin(15*theta)+7/4*sin(16*theta
)+17/9*sin(17*theta)-4*sin(18*theta)-
1/2*sin(19*theta)+1/6*sin(20*theta)+6/7*sin(21*theta)-
1/8*sin(22*theta)+1/3*sin(23*theta)+3/2*sin(24*theta)+13/5*sin(25*theta)+
sin(26*theta)-2*sin(27*theta)+3/5*sin(28*theta)-
1/5*sin(29*theta)+1/5*sin(30*theta)+(2337*cos(theta))/8-
43/5*cos(2*theta)+322/5*cos(3*theta)-117/5*cos(4*theta)-
26/5*cos(5*theta)-23/3*cos(6*theta)+143/4*cos(7*theta)-11/4*cos(8*theta)-
31/3*cos(9*theta)-13/4*cos(10*theta)-
9/2*cos(11*theta)+41/20*cos(12*theta)+8*cos(13*theta)+2/3*cos(14*theta)+6
*cos(15*theta)+17/4*cos(16*theta)-3/2*cos(17*theta)-
29/10*cos(18*theta)+11/6*cos(19*theta)+12/5*cos(20*theta)+3/2*cos(21*thet
a)+11/12*cos(22*theta)-
4/5*cos(23*theta)+cos(24*theta)+17/8*cos(25*theta)-7/2*cos(26*theta)-
5/6*cos(27*theta)-11/10*cos(28*theta)+1/2*cos(29*theta)-
1/5*cos(30*theta);
q_z=-(637*sin(theta))/2-188/5*sin(2*theta)-11/7*sin(3*theta)-
12/5*sin(4*theta)+11/3*sin(5*theta)-
37/4*sin(6*theta)+8/3*sin(7*theta)+65/6*sin(8*theta)-32/5*sin(9*theta)-
41/4*sin(10*theta)-38/3*sin(11*theta)-
47/8*sin(12*theta)+5/4*sin(13*theta)-41/7*sin(14*theta)-
7/3*sin(15*theta)-13/7*sin(16*theta)+17/4*sin(17*theta)-
9/4*sin(18*theta)+8/9*sin(19*theta)+3/5*sin(20*theta)-
2/5*sin(21*theta)+4/3*sin(22*theta)+1/3*sin(23*theta)+3/5*sin(24*theta)-
3/5*sin(25*theta)+6/5*sin(26*theta)-
1/5*sin(27*theta)+10/9*sin(28*theta)+1/3*sin(29*theta)-3/4*sin(30*theta)-
(125*cos(theta))/2-521/9*cos(2*theta)-
359/3*cos(3*theta)+47/3*cos(4*theta)-33/2*cos(5*theta)-
5/4*cos(6*theta)+31/8*cos(7*theta)+9/10*cos(8*theta)-119/4*cos(9*theta)-
17/2*cos(10*theta)+22/3*cos(11*theta)+15/4*cos(12*theta)-
5/2*cos(13*theta)+19/6*cos(14*theta)+7/4*cos(15*theta)+31/4*cos(16*theta)
-cos(17*theta)+11/10*cos(18*theta)-2/3*cos(19*theta)+13/3*cos(20*theta)-
5/4*cos(21*theta)+2/3*cos(22*theta)+1/4*cos(23*theta)+5/6*cos(24*theta)+3
/4*cos(26*theta)-1/2*cos(27*theta)-1/10*cos(28*theta)-1/3*cos(29*theta)-
1/19*cos(30*theta);
q=[0*theta; r*q_y; r*q_z+h_center];
end
```

Publications of the Author

Peer-reviewed

[109] Paneva, V., Fleig, A., Plasencia, D. M., Faulwasser, T., and Müller, J. (2022). Optitrap: Optimal Trap Trajectories for Acoustic Levitation Displays. ACM Trans. Graph., 41(5).

[108] Paneva, V., Bachynskyi, M., and Müller, J. (2020a). Levitation Simulator: Prototyping for Ultrasonic Levitation Interfaces in Virtual Reality. In Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems, CHI '20, New York, NY, USA. ACM.

[110] Paneva, V., Seinfeld, S., Kraiczi, M., and Müller, J. (2020b). HaptiRead: Reading Braille as Mid-Air Haptic Information, pages 13–20, New York, NY, USA. ACM.

[9] Bachynskyi, M., Paneva, V., and Müller, J. (2018). Levicursor: Dexterous Interaction with a Levitating Object. In Proceedings of the 2018 ACM International Conference on Interactive Surfaces and Spaces, ISS '18, pages 253–262, New York, NY, USA. ACM.

Other

Fischer, F., Fleig, A., Klar, M., Paneva, V., & Müller, J. (2023). Towards a Deep (er) Understanding of Interaction through Modeling, Simulation, and Optimization. arXiv preprint arXiv:2302.11409.

Paneva, V., Fleig, A., Plasencia, D. M., Faulwasser, T. , & Müller, J. (2022). Content Rendering for Acoustic Levitation Displays via Optimal Path Following. In Proceedings of the 25th International Symposium on Mathematical Theory of Networks and Systems MTNS, pages 770-773.

Paneva, V. (2019). VR Prototyping Tool for Levitating Interfaces. In Extended Abstracts of the 2. ZD.B Symposium - Networking Congress of the ZD.B Research Incubator, pages 94-95.

Paneva, V. (2018). Dexterous Interaction with Levitating Interfaces. In Extended Abstracts of the 1. ZD.B Symposium - Networking Congress of the ZD.B Research Incubator, pages 62-63.

Paneva, V., Bachynskyi, M., & Müller, J. (2018). Towards Multimodal Mid-Air Interfaces using Acoustic Levitation. Workshop on Mid-Air Haptics for Control Interfaces at the 2018 CHI Conference on Human Factors in Computing Systems.

Eidesstattliche Versicherung

Hiermit versichere ich an Eides statt, dass ich die vorliegende Arbeit selbstständig verfasst und keine anderen als die von mir angegebenen Quellen und Hilfsmittel verwendet habe.

Weiterhin erkläre ich, dass ich die Hilfe von gewerblichen Promotionsberatern bzw. Promotionsvermittlern oder ähnlichen Dienstleistern weder bisher in Anspruch genommen habe, noch künftig in Anspruch nehmen werde.

Zusätzlich erkläre ich hiermit, dass ich keinerlei frühere Promotionsversuche unternommen habe.

Bayreuth, den

Viktorija Paneva