

**Applications of Computer-Controlled Actuation in
Workbench Tangible User Interfaces**

by

Daniel Maynes-Aminzade

Submitted to the Department of Media Arts and Sciences,
in partial fulfillment of the requirements for the degree of

Master of Science in Media Arts and Sciences

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

May 16, 2003

© Massachusetts Institute of Technology, 2003. All Rights Reserved.

Author

Daniel Maynes-Aminzade
Program in Media Arts and Sciences
May 16, 2003

Certified by

Hiroshi Ishii
Associate Professor of Media Arts and Sciences
Thesis Supervisor

Accepted by

Andrew Lippman
Chair, Departmental Committee on Graduate Students
Program in Media Arts and Sciences

Applications of Computer-Controlled Actuation in Workbench Tangible User Interfaces

by

Daniel Maynes-Aminzade

Submitted to the Department of Media Arts and Sciences,
in partial fulfillment of the requirements for the degree of

Master of Science in Media Arts and Sciences

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

May 16, 2003

© Massachusetts Institute of Technology, 2003. All Rights Reserved.

Author

Daniel Maynes-Aminzade
Program in Media Arts and Sciences
May 16, 2003

Certified by

Hiroshi Ishii
Associate Professor of Media Arts and Sciences
Thesis Supervisor

Accepted by

Andrew Lippman
Chair, Departmental Committee on Graduate Students
Program in Media Arts and Sciences

Applications of Computer-Controlled Actuation in Workbench Tangible User Interfaces

by

Daniel Maynes-Aminzade

Submitted to the Department of Media Arts and Sciences on May 16, 2003, in partial fulfillment of the requirements for the degree of Master of Science in Media Arts and Sciences

Abstract

Tangible user interfaces give physical form to digital information, bridging the gap between the digital world of bits and the physical world of atoms. They offer many advantages over traditional graphical user interfaces, including two-handed manipulation, simplified multi-user collaboration, and the use of natural human spatial and tactile abilities. In the majority of existing tangible interfaces, different modalities are used for input and output; while the user provides input through the manipulation of physical objects, the computer produces only graphical output. This can lead to inconsistencies between the digital and physical representations of the same object. The goal of this thesis is to address this limitation by adding computer-controlled actuation to tangible interfaces. Actuation allows the computer to manipulate graspable objects the same way that a human user can. By developing and evaluating a variety of applications that use actuation, we demonstrate how it can best be employed in tangible interfaces.

Thesis Supervisor: Hiroshi Ishii

Title: Associate Professor of Media Arts and Sciences

Applications of Computer-Controlled Actuation in Workbench Tangible User Interfaces

by

Daniel Maynes-Aminzade

Submitted to the Department of Media Arts and Sciences on May 16, 2003, in partial fulfillment of the requirements for the degree of Master of Science in Media Arts and Sciences

Abstract

Tangible user interfaces give physical form to digital information, bridging the gap between the digital world of bits and the physical world of atoms. They offer many advantages over traditional graphical user interfaces, including two-handed manipulation, simplified multi-user collaboration, and the use of natural human spatial and tactile abilities. In the majority of existing tangible interfaces, different modalities are used for input and output; while the user provides input through the manipulation of physical objects, the computer produces only graphical output. This can lead to inconsistencies between the digital and physical representations of the same object. The goal of this thesis is to address this limitation by adding computer-controlled actuation to tangible interfaces. Actuation allows the computer to manipulate graspable objects the same way that a human user can. By developing and evaluating a variety of applications that use actuation, we demonstrate how it can best be employed in tangible interfaces.

Thesis Supervisor: Hiroshi Ishii

Title: Associate Professor of Media Arts and Sciences

Applications of Computer-Controlled Actuation in Workbench Tangible User Interfaces

by

Daniel Maynes-Aminzade

Thesis Supervisor

Hiroshi Ishii
Associate Professor of Media Arts and Sciences
Massachusetts Institute of Technology

Thesis Reader

Robert Jacob
Associate Professor of Computer Science
Tufts University

Thesis Reader

Chris Schmandt
Principal Research Scientist
Massachusetts Institute of Technology

Applications of Computer-Controlled Actuation in Workbench Tangible User Interfaces

by

Daniel Maynes-Aminzade

Thesis Supervisor

Hiroshi Ishii
Associate Professor of Media Arts and Sciences
Massachusetts Institute of Technology

Thesis Reader

Robert Jacob
Associate Professor of Computer Science
Tufts University

Thesis Reader

Chris Schmandt
Principal Research Scientist
Massachusetts Institute of Technology

Table of Contents

1	Introduction.....	8
1.1	Tangible User Interfaces.....	8
1.2	Interactive Workbenches.....	9
1.3	Inconsistencies in Tangible Interfaces.....	10
1.4	Preventing Inconsistency.....	11
1.5	Research Approach and Thesis Overview.....	12
2	Background and Related Work.....	14
2.1	Interactive Surfaces.....	14
2.2	Two-Dimensional Actuation Technologies.....	16
2.3	Haptic and Force Feedback Interfaces.....	18
2.4	Kinesthesia, Proprioception, Memory, and Perceptual Psychology.....	19
3	Hardware and Software Design.....	22
3.1	Design Parameters.....	22
3.2	Mechanical Design.....	23
3.3	Circuit Design and Hardware-Software Interface.....	24
3.4	Puck Design.....	25
3.5	Object Tracking and Position Sensing.....	26
3.6	Motion Control and Interpolation.....	29
3.7	Anti-Aliasing Techniques.....	32
4	Applications.....	35
4.1	Types of Applications.....	35
4.2	Extending GUI Functions to the Physical Domain.....	35
4.3	Navigation.....	36
4.4	Remote Synchronization.....	37
4.5	Scientific Visualization.....	39
4.6	Entertainment.....	39
4.7	Specifying and Maintaining Constraints.....	40
4.8	Computer Movement vs. User Movement.....	43
4.9	Remote Collaboration using Jigs.....	44

4.10 Resolving Conflicts in Remote Collaboration	44
5 Conclusion	46
5.1 Summary	46
5.2 Maintaining Constraints.....	46
5.3 Haptic Feedback	47
5.4 Spatial Thinking.....	47
5.5 Remote Collaboration	48
5.6 Future Vision	49
Appendix A Mathematical Derivations	50
A.1 Motion Control Equations	50
Appendix B Engineering Improvements	53
B.1 Tiling of Actuation Surfaces and Scale	53
B.2 Puck Modifications	53
B.3 New Types of Motion	53
Appendix C Software Improvements	55
C.1 Coordinating Multiple Objects	55
C.2 Unified Software Architecture.....	55
Appendix D New Directions for Actuation	57
D.1 Types of Actuated Interfaces	57
D.2 Purpose of Actuated Feedback	57
D.3 Spatial Complexity and Degrees of Freedom	58
Bibliography	60

Table of Contents

1	Introduction.....	8
1.1	Tangible User Interfaces.....	8
1.2	Interactive Workbenches.....	9
1.3	Inconsistencies in Tangible Interfaces.....	10
1.4	Preventing Inconsistency.....	11
1.5	Research Approach and Thesis Overview.....	12
2	Background and Related Work.....	14
2.1	Interactive Surfaces.....	14
2.2	Two-Dimensional Actuation Technologies.....	16
2.3	Haptic and Force Feedback Interfaces.....	18
2.4	Kinesthesia, Proprioception, Memory, and Perceptual Psychology.....	19
3	Hardware and Software Design.....	22
3.1	Design Parameters.....	22
3.2	Mechanical Design.....	23
3.3	Circuit Design and Hardware-Software Interface.....	24
3.4	Puck Design.....	25
3.5	Object Tracking and Position Sensing.....	26
3.6	Motion Control and Interpolation.....	29
3.7	Anti-Aliasing Techniques.....	32
4	Applications.....	35
4.1	Types of Applications.....	35
4.2	Extending GUI Functions to the Physical Domain.....	35
4.3	Navigation.....	36
4.4	Remote Synchronization.....	37
4.5	Scientific Visualization.....	39
4.6	Entertainment.....	39
4.7	Specifying and Maintaining Constraints.....	40
4.8	Computer Movement vs. User Movement.....	43
4.9	Remote Collaboration using Jigs.....	44

4.10 Resolving Conflicts in Remote Collaboration	44
5 Conclusion	46
5.1 Summary	46
5.2 Maintaining Constraints.....	46
5.3 Haptic Feedback	47
5.4 Spatial Thinking.....	47
5.5 Remote Collaboration	48
5.6 Future Vision	49
Appendix A Mathematical Derivations	50
A.1 Motion Control Equations	50
Appendix B Engineering Improvements	53
B.1 Tiling of Actuation Surfaces and Scale	53
B.2 Puck Modifications	53
B.3 New Types of Motion	53
Appendix C Software Improvements	55
C.1 Coordinating Multiple Objects	55
C.2 Unified Software Architecture.....	55
Appendix D New Directions for Actuation	57
D.1 Types of Actuated Interfaces	57
D.2 Purpose of Actuated Feedback	57
D.3 Spatial Complexity and Degrees of Freedom	58
Bibliography	60

List of Figures

Figure 1.1: Traditional interactive workbench systems provide feedback through video projection alone. The Actuated Workbench adds an additional feedback loop using physical movement of the tracked objects.	10
Figure 1.2: Addressing the nulling problem.	12
Figure 2.3: Conceptual sketch of Bricks system.....	14
Figure 2.1: Digital Desk (calculator application)	14
Figure 2.6: The BUILD-IT system for floor planning combined 2D and 3D views.	15
Figure 2.2: The GraspDraw application (left) used a tool palette and inkwell (right)....	15
Figure 2.4: The metaDESK system used physical icons and lenses to navigate and explore geographical spaces.	15
Figure 2.5: The “Real Reality” system for assembly line planning.....	15
Figure 2.7: Urp, a tangible workbench for urban planning.	16
Figure 2.8: The Sensetable platform has been used for supply chain visualization (left) and musical performance (right).....	16
Figure 3.1: Our basic actuation platform contains a grid of 64 computer-controlled electromagnets.	22
Figure 3.2: Overhead view of electromagnet array.	23
Figure 3.3: Custom-wound electromagnets produce broad, uncontained magnetic fields.	23
Figure 3.4: Custom-fabricated circuit board containing flip-flops and H-bridge transistor arrays.....	24
Figure 3.5: Our initial puck design included a permanent magnet and an infrared LED for vision tracking.....	25
Figure 3.6: Improved puck design with momentary pushbutton switch (top); exploded view of puck showing permanent magnet and LC tag (bottom).	26
Figure 3.7: Overhead view of the Actuated Workbench from vision camera without IR filter (left) and with IR filter (right).....	27
Figure 3.8: Tracking antenna coils and grid of electromagnets.....	28
Figure 3.9: “Manhattan” motion between points.....	29
Figure 3.10: Sweeping to a desired point from an unknown origin.	29
Figure 3.11: The electromagnets (lower left) exert forces on the puck (upper right).....	30

Figure 3.12: Magnetic field interactions between electromagnets. The top images show magnetic flux lines and the bottom images map flux density to brightness. The three image pairs show the fields resulting from a single center magnet turned on (left), the left and center magnets turned on (center), and all three magnets turned on (right). The effect of this field-shifting behavior can be modeled approximately using force summation. These images were generated with the VisiMag software package [5].31

Figure 3.13: Four electromagnets with different duty cycles combine to produce a force with a new direction and magnitude.....33

Figure 3.14: Anti-aliasing methods. The squares represent magnet cells, and intensity of each shaded square corresponds to the duty cycle of the magnet. The “dot” technique (left) results in slower travel than the “jet” technique (right).33

Figure 3.15: “Dot” and “Jet” equivalents in computer graphics. The three dots use different falloff metrics34

Figure 3.16: “Dot” force field (left) versus “Jet” force field (right).....34

Figure 4.1: Translate and rotate/zoom operations. The puck that the user is holding has been locked to a particular map location.37

Figure 4.2: Two actuated tangible workbenches running a distributed application for cellphone tower placement. The tables maintain a synchronized state over an Internet connection. Translucent hand silhouettes show remote user activity.....38

Figure 4.3: Collar and ring constraint jigs.41

Figure 4.4: An inconsistent set of constraints.43

Figure 4.5: Hands and jigs and their projected silhouettes.44

Figure C.1: Data flow diagram showing one iteration of the Actuated Workbench control loop.56

Figure D.1: Comparison of degrees of control in various actuation systems.58

List of Figures

Figure 1.1: Traditional interactive workbench systems provide feedback through video projection alone. The Actuated Workbench adds an additional feedback loop using physical movement of the tracked objects.	10
Figure 1.2: Addressing the nulling problem.	12
Figure 2.3: Conceptual sketch of Bricks system.....	14
Figure 2.1: Digital Desk (calculator application)	14
Figure 2.6: The BUILD-IT system for floor planning combined 2D and 3D views.	15
Figure 2.2: The GraspDraw application (left) used a tool palette and inkwell (right)....	15
Figure 2.4: The metaDESK system used physical icons and lenses to navigate and explore geographical spaces.	15
Figure 2.5: The “Real Reality” system for assembly line planning.....	15
Figure 2.7: Urp, a tangible workbench for urban planning.	16
Figure 2.8: The Sensetable platform has been used for supply chain visualization (left) and musical performance (right).....	16
Figure 3.1: Our basic actuation platform contains a grid of 64 computer-controlled electromagnets.	22
Figure 3.2: Overhead view of electromagnet array.	23
Figure 3.3: Custom-wound electromagnets produce broad, uncontained magnetic fields.	23
Figure 3.4: Custom-fabricated circuit board containing flip-flops and H-bridge transistor arrays.....	24
Figure 3.5: Our initial puck design included a permanent magnet and an infrared LED for vision tracking.....	25
Figure 3.6: Improved puck design with momentary pushbutton switch (top); exploded view of puck showing permanent magnet and LC tag (bottom).	26
Figure 3.7: Overhead view of the Actuated Workbench from vision camera without IR filter (left) and with IR filter (right).....	27
Figure 3.8: Tracking antenna coils and grid of electromagnets.....	28
Figure 3.9: “Manhattan” motion between points.....	29
Figure 3.10: Sweeping to a desired point from an unknown origin.	29
Figure 3.11: The electromagnets (lower left) exert forces on the puck (upper right).....	30

Figure 3.12: Magnetic field interactions between electromagnets. The top images show magnetic flux lines and the bottom images map flux density to brightness. The three image pairs show the fields resulting from a single center magnet turned on (left), the left and center magnets turned on (center), and all three magnets turned on (right). The effect of this field-shifting behavior can be modeled approximately using force summation. These images were generated with the VisiMag software package [5].31

Figure 3.13: Four electromagnets with different duty cycles combine to produce a force with a new direction and magnitude.....33

Figure 3.14: Anti-aliasing methods. The squares represent magnet cells, and intensity of each shaded square corresponds to the duty cycle of the magnet. The “dot” technique (left) results in slower travel than the “jet” technique (right).33

Figure 3.15: “Dot” and “Jet” equivalents in computer graphics. The three dots use different falloff metrics34

Figure 3.16: “Dot” force field (left) versus “Jet” force field (right).....34

Figure 4.1: Translate and rotate/zoom operations. The puck that the user is holding has been locked to a particular map location.37

Figure 4.2: Two actuated tangible workbenches running a distributed application for cellphone tower placement. The tables maintain a synchronized state over an Internet connection. Translucent hand silhouettes show remote user activity.....38

Figure 4.3: Collar and ring constraint jigs.41

Figure 4.4: An inconsistent set of constraints.43

Figure 4.5: Hands and jigs and their projected silhouettes.44

Figure C.1: Data flow diagram showing one iteration of the Actuated Workbench control loop.56

Figure D.1: Comparison of degrees of control in various actuation systems.58

Chapter 1

Introduction

1.1 Tangible User Interfaces

For most of the history of computing, the primary means of representing digital information has been screen-based text and graphics. The WIMP (Windows, Icons, Menus, and Pointer) interface metaphor, which grew out of the computational capabilities available in the late 1970's, has become highly pervasive. Although it represents a vast improvement over the batch processing and command line interfaces found in the earliest days of computing, it is unlikely to meet the demanding user interface needs of modern, real-time, high-performance applications. The WIMP paradigm has an inherent asymmetry between input and output: while it uses millions of pixels of graphical output, it uses a single cursor-driven point input, with no physical or kinesthetic affordances and no way to engage multiple users, multiple hands, or multiple senses. It is a predominantly visual paradigm that does not take advantage of the natural abilities humans have developed through a lifetime of interaction with the physical world.

The limitations of graphical user interfaces have resulted in many new branches of research that expand our vision of human-computer interaction beyond the limited modes available via a traditional keyboard and mouse. The new research areas of virtual reality, wearable computing, ubiquitous computing, and augmented reality take advantage of new input and output technologies such as large, high-resolution displays, high fidelity audio, haptic feedback devices, voice and gesture input, and pen computing.

Of particular relevance to this thesis is a type of interface that attempts to bridge the gap between cyberspace and the physical world by giving physical form to digital information. These “graspable” [18] or “tangible” [23] user interfaces use physical objects to

represent both digital information and computational operations, shifting computation from the onscreen desktop to our bodies and the physical spaces we inhabit. The tangible user interface (TUI) makes use of natural physical affordances to achieve a more seamless interaction between people and information. TUIs allow two-handed manipulation, simplify collocated multi-user collaboration, offer kinesthetic memory cues, leverage existing physical skills and work practices, and take advantage of our natural ability to spatially organize objects to solve problems.

1.2 Interactive Workbenches

Recently, many tangible user interfaces have adopted a “workbench” metaphor, in which the user manipulates tracked objects on an interactive surface. The desktops, walls, and windows of an architectural space are thereby transformed into active interfaces between the physical and virtual worlds. A computer senses the positions and states of the graspable objects on the surfaces and responds to users’ physical input with graphical output, projected on and around the objects on the table. These interfaces have been used for a wide variety of applications, from musical performance [40] to video logging [12]. Systems such as the DigitalDesk [53], Bricks [17], Sensetable [39], and Urp [51] offer many advantages over purely graphical interfaces, including the ability for users to organize objects spatially to aid problem solving, the potential for two-handed interaction, and ease of collaboration between multiple collocated users.

However, current interactive workbench systems share a common weakness. While input occurs through the physical manipulation of tangible objects, output is displayed only through sound or graphical projection on and around the objects. As a result, the objects can feel like loosely coupled handles to digital information rather than physical manifestations of the information itself.

In addition, the user must sometimes compensate for inconsistencies when links between the digital data and the physical objects are broken. Such broken links can arise when a change occurs in the computer model that is not reflected in a physical change of its associated object. With the computer system unable to move the objects on the table surface, it cannot undo physical input, correct physical inconsistencies in the layouts of the objects, or guide the user in the physical manipulation of the objects. In short, the physical interaction between human and computer remains one-sided.

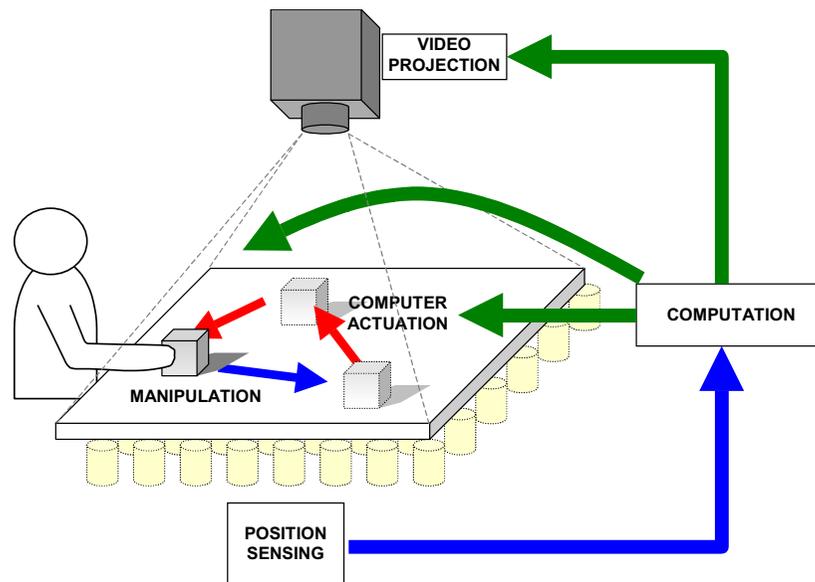


Figure 1.1: Traditional interactive workbench systems provide feedback through video projection alone. The Actuated Workbench adds an additional feedback loop using physical movement of the tracked objects.

1.3 Inconsistencies in Tangible Interfaces

Inconsistencies can occur in various types of interactions:

- **Remote Collaboration.** When there are multiple instantiations of a work table on which physical objects represent digital information, the physical state of each table can become inconsistent whenever a user moves objects on one table, but the remote user does not move the corresponding objects on the other table. Though the graphical projections

can be synchronized, the discrepancies between the physical positions of objects on the tables will remain.

- **Simulation.** A software simulation running in real-time may compute that the value associated with an object on the table has changed. The object's position or orientation may be inconsistent with the new value of its corresponding software parameter.

- **Constraints.** An application may have constraints on the spatial arrangement of objects on the table, such as zoning laws in an urban planning application. If the user moves an object to the wrong part of the table, some of these constraints may be violated. Existing systems can provide graphical feedback to inform the user that a constraint has been violated, but cannot fix the problem in both the digital and physical representations.

- **Navigation.** In an application with navigation features, such as one that uses a map [49], it is useful to rotate, translate or scale the map by moving the physical objects on top of it. However, if there are several physical objects representing fixed landmarks on the map, their positions on the table will no longer be correct once the underlying map is moved.

1.4 Preventing Inconsistency

There are several design approaches to dealing with the problem of inconsistency. One is to structure the interaction so that the physical objects can never become inconsistent with their digital counterparts. For example, in the Urp system [51], the user moves building models around on the tabletop, and the system displays graphical information in response. In this case the software has no control over the position or orientation of the buildings, so building placement cannot cause inconsistency. For some applications this approach works well, but it tends to limit the system's practicality in real-world situations.

Another approach to inconsistency is to use abstract physical objects, relying on graphics to represent system state. An example of this technique would be to represent the position of a dial with a plain circular “puck” and a projected arrow (Figure 1.2), rather than an arrow that is physically part of the puck. In this way the system can

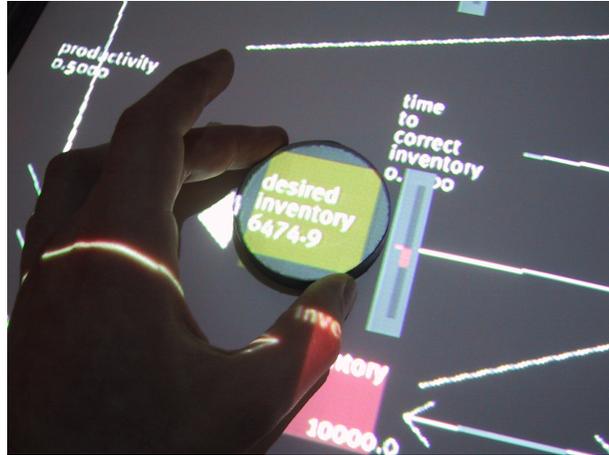


Figure 1.2: Addressing the nulling problem.

rotate the arrow independently of the physical puck if necessary. If the arrow were physically part of the puck, the user would have to set it to the correct position associated with the digital value of the parameter, a difficulty Buxton has referred to as “the nulling problem” [9]. This use of abstract objects for input works well for dealing with the rotation of pucks, but it cannot be applied to cases where the pucks must be translated.

Yet another approach is to provide graphical feedback to alert the user when an inconsistency arises. A projected graphical “shadow” of an object can represent that the physical object is not where the computer thinks it should be. The computer can then ignore the physical object until the user has “reattached” it to its shadow. In practice, this method can break the flow of interaction with the system and confuse users. It runs contrary to the design goal that the objects should be embodiments of digital information rather than just handles attached to that information.

1.5 Research Approach and Thesis Overview

By adding computer-controlled actuation to tabletop tangible interfaces, we provided new solutions to many of these inconsistency problems. We designed and built a system called the Actuated Workbench, a hardware and software infrastructure enabling a com-

puter to smoothly move objects on a table surface in two dimensions. This thesis describes the underlying technology of the Actuated Workbench and discusses the hardware and software design decisions involved in its construction. It then introduces a variety of techniques newly enabled by actuation and explains how they can improve upon tangible interfaces and lead to new applications.

Chapter 2

Background and Related Work

2.1 Interactive Surfaces

Interactive surfaces are a common variety of TUI in which physical objects are manipulated by users on an augmented planar surface. The presence, identity, and configuration of these objects are electronically tracked, and the computer interprets and processes this information to provide graphical feedback. The two primary varieties of interactive surfaces are (1) interactive workbenches, which use a horizontal work surface; and (2) interactive walls, which use a vertical work surface to which objects are affixed using magnets, sticky notes, or thumbtacks.

This thesis focuses on interactive workbench systems, of which there are many examples. One of the earliest such systems was the DigitalDesk [53], which supported augmented interaction with paper documents on a physical desktop. The paper documents were identified with overhead cameras, and information was projected around them (Figure 2.1).

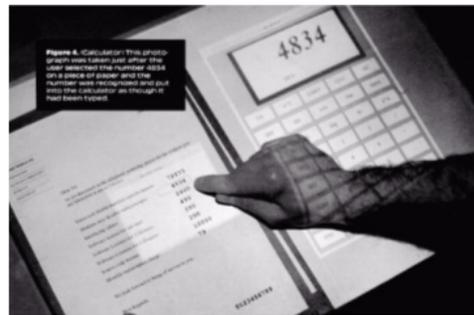


Figure 2.1: Digital Desk (calculator application)

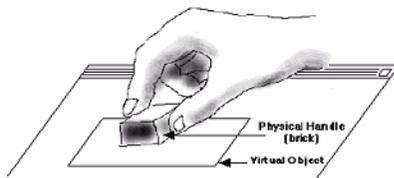


Figure 2.3: Conceptual sketch of Bricks system.

Bricks [17] was another early workbench system, in which users placed multiple bricks onto various screen-based virtual objects. The bricks were tracked with six degrees of freedom, and could be used to physically rotate, translate, scale, and deform the virtual entities to which they were attached (Figure 2.3). In the “GraspDraw” application

(Figure 2.2), bricks could also be bound to tools and attributes, making them operators rather than handles.

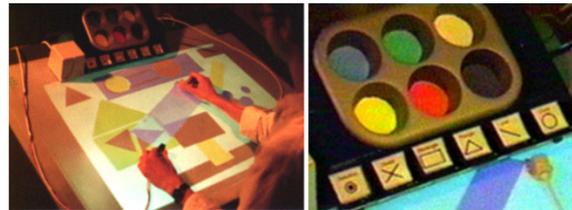


Figure 2.2: The GraspDraw application (left) used a tool palette and inkwell (right).

The metaDESK system [49], shown in Figure 2.4, supported interaction with a geographical space through the manipulation of physical tokens described as physical icons or “phicons.” This system was limited in that it used highly representational tokens that were permanently bound to their geographical “contents,” without mechanisms for rebinding.

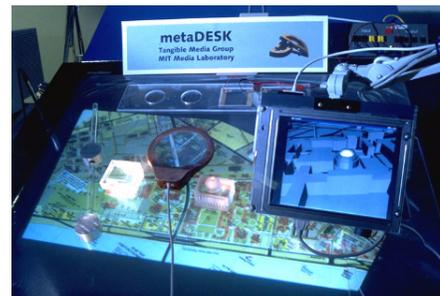


Figure 2.4: The metaDESK system used physical icons and lenses to navigate and explore geographical spaces.



Figure 2.5: The “Real Reality” system for assembly line planning.

Another workbench approach was the “Real Reality” system for assembly line planning in industrial contexts [44]. The system used a novel grasp-tracking approach to manipulate literal physical models of the assembly line in conjunction with physical representations of logical flows (Figure 2.5).

The BUILD-IT platform [19] used a special workbench layout that facilitated the binding of brick elements to data and func-

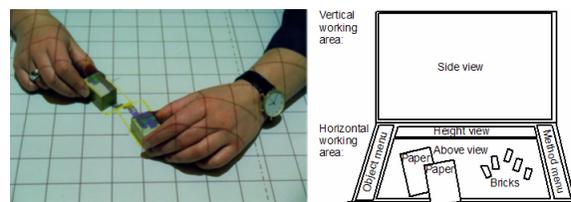


Figure 2.6: The BUILD-IT system for floor planning combined 2D and 3D views.

tional elements (Figure 2.6). It was used to create and test an application for floor planning that combined 2D and 3D views.

The Urp urban planning system [51] was one of the most fully developed interactive workbench systems. Combining physical building models with projected graphical simulations, it supported shadow studies, reflection studies, wind simulations, zoning metrics, and many other features useful for making urban planning decisions.

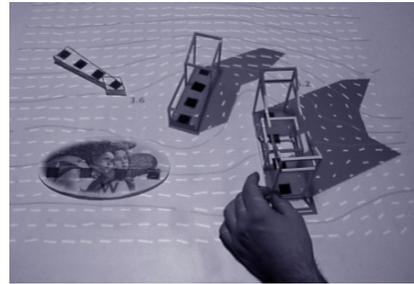


Figure 2.7: Urp, a tangible workbench for urban planning.



Figure 2.8: The Sensetable platform has been used for supply chain visualization (left) and musical performance (right).

Interactive workbench systems have also been applied to more abstract problem domains for which inherently geometrical representations do not exist. For example, the Sensetable system has been used for supply chain visualization [39] and musical performance [40].

2.2 Two-Dimensional Actuation Technologies

The computer-controlled configuration of objects on a flat surface has been studied in both the HCI domain and in the realm of industrial mechanics. Some early systems such as Seek [37] used robotic arms to arrange parts or objects on a table. Though an effective and dexterous method for computer control, the use of robotic arms would likely be distracting for interactive workbench systems. Moreover, it would be complicated and expensive to implement the multiple arms required to move multiple objects simultaneously. Recently, researchers in HCI and robotics have developed systems attempting to move objects with-

out the use of robotic arms. We examine some of these for their applicability to interactive workbench systems.

The PsyBench [7] prototype was built using parts from a computerized chess set that moved magnetic pieces using an electromagnet mounted on an x-y plotter under the table. This allowed the position of objects in the two workspaces to be synchronized. Though similar to the Actuated Workbench in its use of magnetism to grab objects, the PsyBench prototype had a variety of implementation limitations. It was only capable of inaccurate, teetering movements of the objects, and it was limited to straight-line motion. Furthermore, it was unable to control the orientation of the moving objects, and it could only move one object at a time.

Some recent robotics research targets actuation problems such as part feeding in factories, parcel sorting in distribution warehouses, and luggage sorting in airports. The Universal Planar Manipulator (UPM) [42] uses the horizontal vibration of a flat surface to move multiple objects at a time. Complex movements of specific objects on the surface are achieved using interference patterns of the vibration waves as they propagate across the surface. This system presents an effective way to manipulate many small parts without the need for motors or magnets, and its designers successfully used it in a closed-loop vision-tracking system. However, several aspects of the UPM's design detract from its usefulness in interactive workbench interfaces. First, in its present state, it is only capable of slow object translations and rotations; feed rates are on the order of millimeters per second. Second, the mechanism for vibrating the surface occupies space around the edges, preventing the easy tiling of multiple surfaces. Third, the system is noisy due to the mechanism needed to vibrate the flat surface and the sound of the vibrating objects. While not a problem in a factory assembly-line setting, this noise might be distracting for HCI.

Another system, the Modular Distributed Manipulator System (MDMS) [29] consists of an array of orthogonally oriented wheels that support and move objects through combined vector forces created by the rotating wheels. This actuation method presents a clever solution to the problem of friction: instead of dragging or sliding objects, they are rolled along the tops of the wheels. Like the Actuated Workbench, the MDMS is scalable to larger areas, requiring only that more actuators be set up next to the existing array. The MDMS differs from our work in that it is intended for manipulating large parcels, factory materials, or pieces of luggage in a conveyor belt type situation. Moreover, the surface upon which the objects rest is neither flat nor continuous (because it is made up of many small wheels), making it unsuitable for the projection often used in interactive workbench interfaces.

2.3 Haptic and Force Feedback Interfaces

Researchers in the field of haptics have devised many ways of employing the sense of touch in computer interfaces. Haptic feedback devices fall into two general categories: *force feedback* devices that interact with human muscles and tendons to give the sensation of a force being applied, and *tactile feedback* devices that interact with the nerve endings of the skin to indicate heat, pressure, or texture.

The most common application of haptic feedback devices is to provide touch stimuli to virtual reality environments, allowing computers to simulate the feel of a virtual object, but studies have also shown that augmenting any visual system with haptic feedback can prove useful. By adding an additional information channel to a visual interface, haptic feedback can increase the amount of information that is simultaneously processed by the brain. This extra information can reduce error, lower energy consumption, and accelerate task completion [45][46]. In addition, this extra information can compensate for ambiguities or errors in the visual information, making a visual display with haptic feedback per-

form more effectively than a stereoscopic display or a display with multiple viewpoints [8]. An experiment by Batter and Brooks suggests that haptic interfaces are effective educational aids; in their experiment, physics students who explored electrostatic fields using a touch interface developed a better understanding than students without access to haptic devices in their lab work [4].

A wide variety of haptic feedback technologies have been developed, some of which are commercially available. The most common devices are standard input devices such as joysticks, mice [21], or steering wheels, augmented with motors or vibrators to provide touch or force feedback along one or two dimensions. More complicated devices, such as the PHANTOM [33] and the HapticMaster [16], provide complex three-dimensional force control through robotic arm joints or magnetic levitation [6]. Other devices, like the CyberGrasp [52], are attached to the arm or hand as a robotic exoskeleton. The FEELEX system [22] can simulate texture and the contour of a landscape using an array of linear actuators.

2.4 Kinesthesia, Proprioception, Memory, and Perceptual Psychology

A large body of psychology work examines the ways in which humans use the spatial arrangement of objects to aid in problem solving. Kirsh showed that people frequently manipulate their environment to enhance memory or simplify their choices [26][27]. For example, people often sort a large group of objects into smaller groups to help them remember which objects share similar properties. A study by Zhang [54] demonstrated that the kind of objects used in a problem-solving task can dramatically effect how people think about a task and how long the task takes to solve. He compared the time required to solve a puzzle using two types of physical objects, and found that objects which afforded stacking allowed people to complete the task in half the time and with substantially fewer errors. Patten presented a study [38] in which participants using a tangible interface to

organize news articles performed better at location recall than participants using a graphical interface, often using the spatial relationship between physical objects and parts of the environment to help them remember the content of the physical objects.

There is also a variety of work on how people encode and use spatial information about their environment. A study by Malone [31] suggested that office workers with more sophisticated organizational schemes were better at locating their documents. However, formal experiments on this question suggest that it may be difficult to rely on spatial organization alone for recall. For example, Dumas and Jones found that retrieving documents by name was more effective than using spatial information for retrieval [13], and Lansdale argues that memory of location can be quite poor in cases where documents are not organized according to some logical structure. In cases where a structure is imposed, however, subjects can use it to help determine the location of documents, resulting in better recall [28].

Other studies have attempted to determine the extent to which spatial information is automatically encoded in the absence of a particular organizational scheme. Mandler et al. concluded that a great deal of object location information is encoded automatically, after a study showing only a small decrease in recall performance when subjects were not told to remember object location [32]. However, work by Naveh-Benjamin [36] suggested that this location information is encoded automatically only when subjects modify a spatial configuration of objects, and not when they simply observe such a configuration. A study by Tan et al. showed that using a touchscreen rather than a mouse results in better performance on a spatial recall task, confirming that kinesthetic cues are an important component of spatial memory encoding [48]. Despite disagreements in the literature as to the process of spatial memory encoding, it remains clear that spacial memory can be reliably used in practice to improve task performance. An evaluation of the Data Mountain system

by Robertson et al. demonstrated an effective application of spatial memory to a task involving the retrieval of web documents [43].

Chapter 3

Hardware and Software Design

3.1 Design Parameters

The Actuated Workbench’s design reflects several concerns of compatibility with current interactive workbench systems. First, the tagging and tracking technologies in these interfaces have begun to decrease in size, allowing the objects or “pucks” that hold them to be quite small. Zowie/LEGO demonstrated an example of such technology in a toy [20] which tracked objects with passive tags only 1.5cm in diameter and 2mm in height. While we considered designing motorized pucks that drive themselves around the tabletop on wheels, we felt these would tend to be relatively large compared to the tags. Motorized pucks would also require batteries that might need to be changed or recharged frequently due to the motors’ power requirements. Since many tagging technologies used today are passive devices, we sought to keep the actuation technology passive as well.

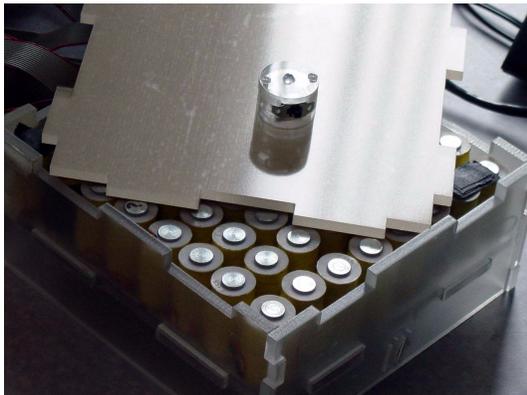


Figure 3.1: Our basic actuation platform contains a grid of 64 computer-controlled electromagnets.

A key interaction technique in most interactive workbench interfaces is the ability to manipulate multiple objects at the same time using both hands. Therefore, we wanted the computer actuation technology to be able to move multiple objects at the same time, preferably recreating users’ gestures with the objects. We also wanted the actuation system to be scalable to accom-

modate a variety of sensing areas. Finally, our ideal system would be silent, so as not to unintentionally distract the user when an object is moved on the surface.

3.2 Mechanical Design

Our basic actuation system consists of a 16.5cm fixed array of 64 electromagnets arranged in an 8 x 8 grid under a layer of 0.63cm acrylic (Figure 3.1). Though this provides only a limited area for actuation, we tile these arrays together to create larger actuation surfaces, the only limitations on scalability being the complexity



Figure 3.2: Overhead view of electromagnet array.

of electronically addressing the arrays, and the power requirements of running such a large number of electromagnets. We built the system using custom made electromagnets, each measuring 1.9cm in diameter and 3.8cm in length. They are wound with 32 gauge copper wire with a total length resistance of 120-122 ohms.

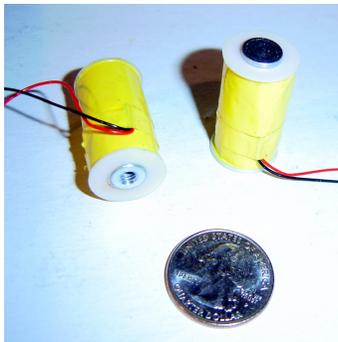


Figure 3.3: Custom-wound electromagnets produce broad, uncontained magnetic fields.

Using these custom-wound magnets proved an advantage over most commercially available electromagnets, which are often designed with metal housings intended to focus the magnetic field within a small area around the electromagnet. The uncontained fields of our electromagnets made it easier to create combinational flux patterns between individual electromagnets, the importance of which will be discussed later. Each electromagnet is driven with 27 DC volts and draws about 250mA. In our current applications, each electromagnet is only active for a few milliseconds at a time, and significant heating of the electromagnets does not occur. However, if many electromagnets were activated for a long period of time, cooling of the array might be necessary.

3.3 Circuit Design and Hardware-Software Interface

We designed custom electronics to drive each electromagnet in the array bidirectionally, making it possible to set the polarity of each magnet's field, as well as turn individual magnets on and off. Our electronics are designed to set the state of each electromagnet in the array at the same time. This makes moving multiple objects simultaneously a simple matter of setting up separate magnetic fields in dif-

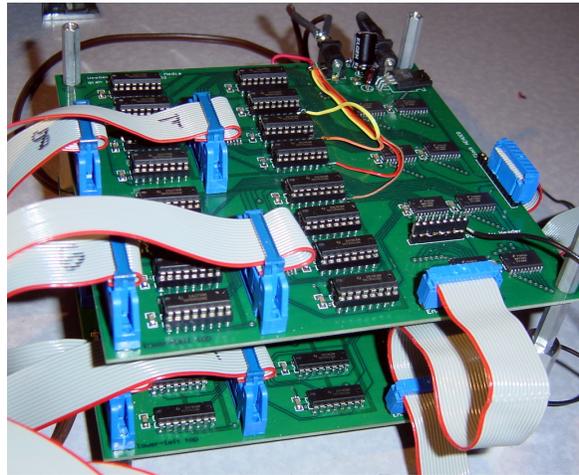


Figure 3.4: Custom-fabricated circuit board containing flip-flops and H-bridge transistor arrays.

ferent areas of the array. Of course we must take care that these magnetic fields do not overlap, and this consideration limits the number of objects that can be moved simultaneously.

An Ethernet-equipped microcontroller board, the Systronix SaJe board, natively runs a Java program that receives UDP packets sent via Ethernet from a control computer. It processes these packets and converts the data for output on two parallel 8-bit data buses. Every 15 microseconds, the microcontroller board clocks each magnet's polarity and enable status (off or on) into a set of octal flip-flops that connect to motor driver chips (containing the H-bridge transistor configuration frequently used for driving electric motors), which then connect to the electromagnets via ribbon cable.

The 15 microsecond refresh rate allows us to vary the strength of each electromagnet's field through pulse-width-modulation (PWM), a common technique for driving electric motors at different speeds by sending them pulses of various duty cycles. We can move objects between individual electromagnets by combining the magnetic fields of several

adjacent electromagnets, each set to a different strength through PWM, so that the object is attracted to a point somewhere in between the electromagnets.

3.4 Puck Design

Though all of the pucks that we use with the system contain permanent magnets, the system is capable of moving any lightweight ferromagnetic object, such as a paperclip or steel bolt. Our acrylic pucks are built to hold powerful 1.1 Tesla neodymium magnets, each 1.26cm x 1.26cm x 0.63cm, in order to provide the strong attractive forces needed to drag the 14g pucks around on the Active Workbench's acrylic surface.

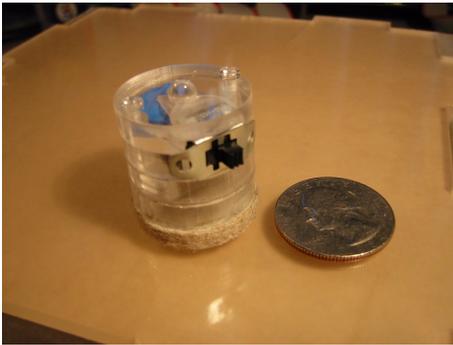


Figure 3.5: Our initial puck design included a permanent magnet and an infrared LED for vision tracking.

Our initial puck design had dimensions 2.54cm diameter x 2.54cm length. It included a battery, an IR LED for vision tracking, and a switch (to save the battery when not in use). Since the inclusion of a battery violated one of our design goals, we later switched to a passive radio frequency tag for object tracking. This allowed us to use slightly smaller and completely passive pucks

with an LC tag in place of an LED and battery. Our improved pucks measured 3cm diameter x 1.25cm height and held a permanent neodymium magnet and an LC radio frequency tag. Each puck also contained a small momentary pushbutton switch that shorted out the LC tag when pressed. This allowed users to perform mode toggling operations.

We attached felt pads to the bottom of each puck, providing the necessary kinetic friction to keep the object from sliding around uncontrollably on the table's surface; bare acrylic-on-acrylic is too slippery, resulting in oscillations as the puck slides past its goal and is then attracted back to it. The 0.63cm

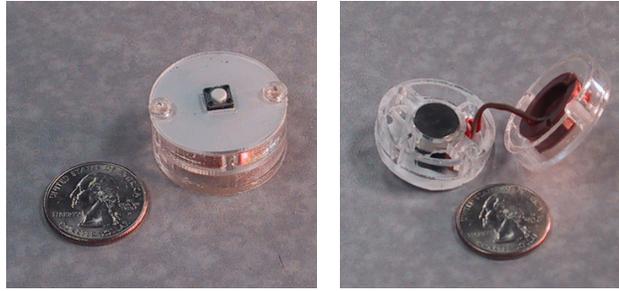


Figure 3.6: Improved puck design with momentary pushbutton switch (top); exploded view of puck showing permanent magnet and LC tag (bottom).

thickness of the felt pad, combined with the 0.63cm bottommost acrylic layer of the puck, results in the permanent magnet being about 1.26cm from the surface of the table, which is itself a piece of 0.63cm acrylic. This positions the permanent magnet about 1.89cm above the tops of the electromagnets. The height of the permanent magnet in the puck has significant effects on the performance of the system, since the neodymium magnet is strong enough to be attracted to the ferrous cores of the underlying electromagnets even when they are not activated. This attraction increases friction on the object, which affects the puck's ability to slide on the surface. We found the amount of friction between the pucks and the table to be a critical element in the system's ability to create smooth 2D motion. In general, we observed that static friction (the friction between the object and the surface when the object is at rest) inhibited smooth motion of the pucks, while kinetic friction facilitated smooth motion by controlling oscillations. After trying a variety of materials, we found that felt on acrylic gave adequate frictional characteristics, but other materials may yield better results in the future.

3.5 Object Tracking and Position Sensing

Electromagnetic radio frequency sensing technology is evolving rapidly to provide robust, low-latency object tracking on table surfaces [20][39]. Though this technology is used

often in interactive workbench systems, we encountered preliminary difficulties using electromagnetic sensing in conjunction with our magnetic actuation system because of distortions created by the strong magnetic fields of our electromagnets. We eventually overcame this problem through careful calibration of the tracking system, but to avoid these difficulties in the short term, we chose vision tracking for our first system prototype.

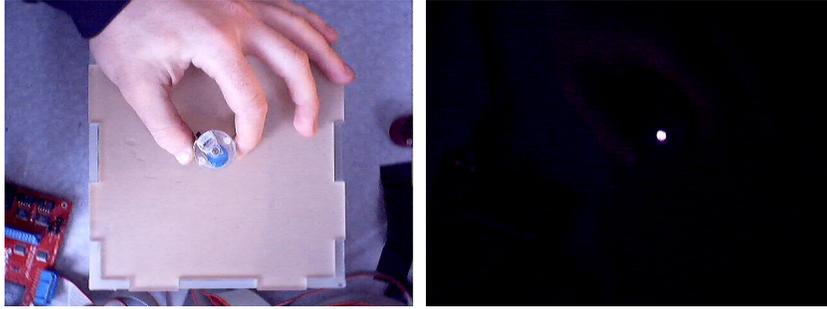


Figure 3.7: Overhead view of the Actuated Workbench from vision camera without IR filter (left) and with IR filter (right).

We embedded each puck with a small battery and an infrared LED, and suspended a camera directly above the Actuated Workbench. Adding an infrared filter to the camera blocked out ambient fluorescent light, making the video signal easy to process (Figure 3.7). We used an inexpensive Intel PC Camera Pro USB CCD camera and were able to achieve a tracking rate of 30 updates per second. This frame rate, though high from a human interaction standpoint, is somewhat slow from a control systems perspective.

Puck tracking was accomplished by detecting bright regions within the image. We used the image histogram to compute a threshold value on startup, and the threshold was used to divide the grayscale image into zeros and ones. We then employed standard blob-analysis techniques [25] to determine the longest horizontal segments. We could track multiple pucks simultaneously in real-time using an association method [3] to distinguish the pucks between frames. In every frame, we associated each observed location with the closest puck location in the previous frame. This association method is not wholly reli-

able, since puck paths that cross each other can interchange identities, but since the permanent magnets inside of the pucks tend to repel each other, the pucks rarely get close enough for this method to break down.

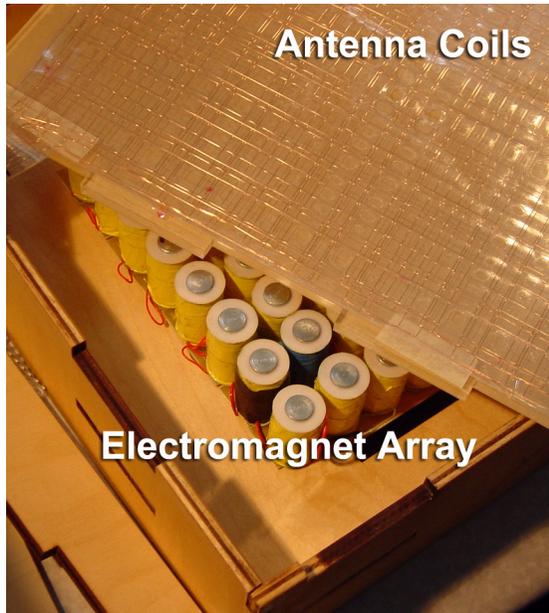


Figure 3.8: Tracking antenna coils and grid of electromagnets.

Our final system successfully employed electromagnetic tracking, which proved faster and more robust. The pucks contained passive radio frequency (RF) LC tags, each resonant on a unique frequency. We determined the position of each RF tag on the table surface using a modified version of the sensing apparatus found in the Zowie™ playsets [20]. We measured the amplitude of the tag resonances with several specially shaped antennas. The amplitude of the tag's resonance

with each antenna varies as a function of its position on top of the antenna array. This method gives very stable 2D position data accurate to within 2mm. Since each tag on the table resonates at a different frequency, their positions can be determined independently. This eliminates the need for the complex association algorithms required to track multiple objects in a vision-based system, and also provides substantially faster update rates than computer vision. Although the presence of dynamic magnetic fields on the table does create interference with our electromagnetic tracking system, our software calibrates for the presence of these magnetic fields, and they do not pose a problem when the system is in use.

3.6 Motion Control and Interpolation

Moving the puck across the table in a linear “Manhattan” fashion (in straight lines at right angles to each other) is a straightforward process. The puck can be moved to any grid cell on the table by consecutively activating the electromagnets in neighboring cells at full strength, as shown in Figure 3.9. Using Manhattan

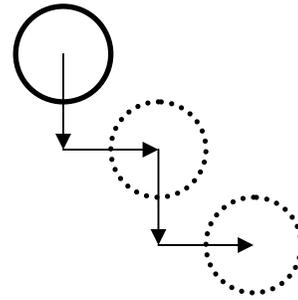


Figure 3.9: “Manhattan” motion

motion, objects can be moved across the table at rates on the order of 25cm/sec.

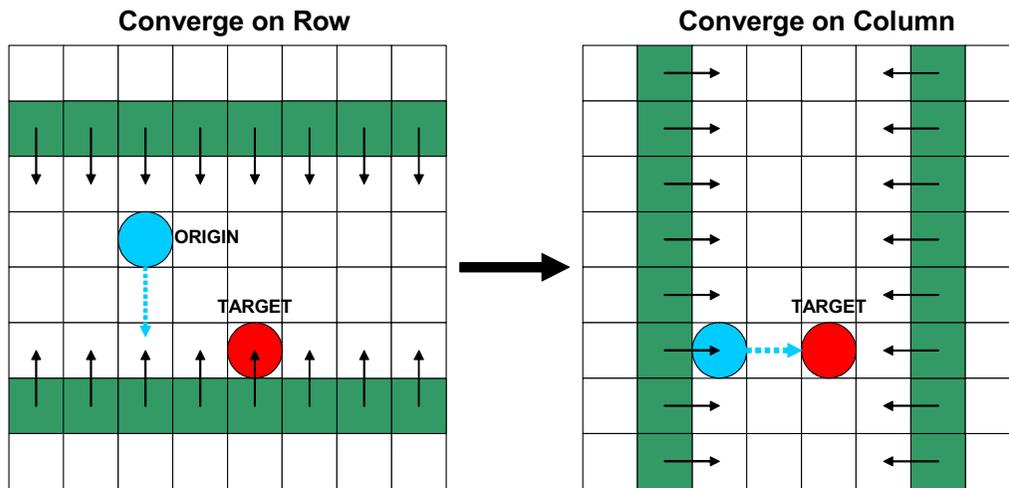


Figure 3.10: Sweeping to a desired point from an unknown origin.

If the board is operating in an “open loop” mode, in which we do not know the current position of the puck, we can still move it to any point on the table using a sweeping algorithm (Figure 3.10). To move the puck to the point (x,y) we begin by activating the outermost rows and then sweeping inward until the target row y is reached. Next, we begin with the outermost columns, and sweep inward in a similar fashion until we reach column x .

This method was useful for moving the puck to the far corners of the table to calibrate the tracking system.

Though Manhattan motion can move the pucks rapidly across the table, it is not very useful for recreating the smooth motions with which a user moves objects on an interactive workbench's surface. Since we can control the strength of individual electromagnets through PWM, we can perform a sort of physical anti-aliasing to create smooth travel paths along the table between the discrete positions of the electromagnets. In this section we describe our mathematical model of the Actuated Workbench and present the equations we used in our software to produce smooth motion along arbitrary paths. For a detailed derivation of these equations, refer to Appendix A.

Figure 3.11 is a vector diagram showing our force model. A single puck on the surface of the Actuated Workbench is subject to gravitational force, frictional force, the magnetic forces of attraction between the puck and the activated electromagnets, and the force of attraction between the permanent magnet in the puck and the iron cores of the electromagnets beneath.

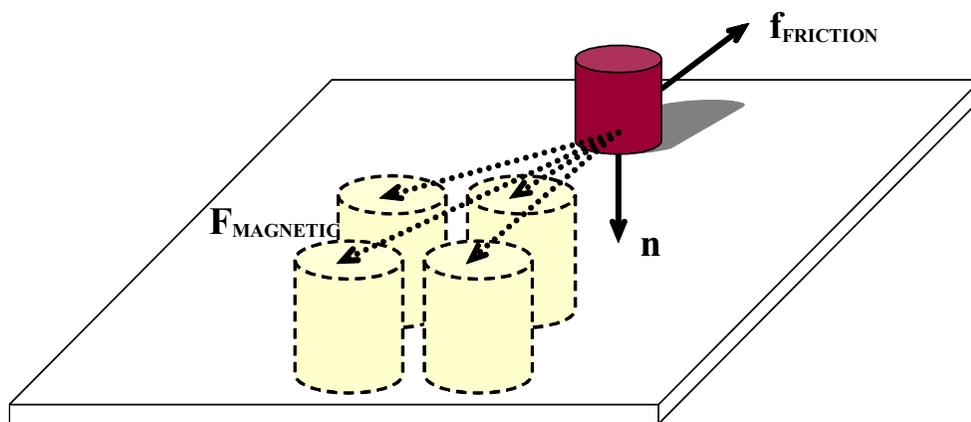


Figure 3.11: The electromagnets (lower left) exert forces on the puck (upper right).

We add these forces to arrive at an equation for the total force on the puck in terms of $f_{\text{MAG-NET}}$, the total force of magnetic attraction, and $f_{\text{FRICTION-NET}}$, the net friction:

$$f_{\text{TOTAL}} = f_{\text{MAG-NET}} \cdot \hat{x} + f_{\text{MAG-NET}} \cdot \hat{y} + f_{\text{FRICTION-NET}} \quad (3.1)$$

$$f_{\text{MAG-NET}} = \sum_i \frac{\alpha_i f_E}{(x - x_i)^2 + (y - y_i)^2 + z_S^2} \quad (3.2)$$

$$\|f_{\text{FRICTION-NET}}\| = \mu \left(mg + \frac{\|f_p\|}{z_S^2} + \|f_{\text{MAG-NET}} \cdot \hat{z}\| \right) \quad (3.3)$$

Here the puck is positioned at (x, y) and each electromagnet i is positioned at (x_i, y_i) with duty cycle α_i . f_E and f_p are constant-magnitude forces of attraction proportional to the strengths of the electromagnet and the permanent magnet in the puck, z_S is the vertical separation between the puck and the electromagnets, m is the mass of the puck, g is the acceleration due to gravity, μ is a dimensionless coefficient of friction, and \hat{x} , \hat{y} , and \hat{z} are the standard unit vectors.

In reality, the magnetic fields of the activated electromagnets interact in a somewhat more complex manner (Figure 3.12). Nonetheless, the force-summing model just described, in which electromagnets are treated independently of one another, is a reasonable method of approximating the more complicated underlying physics, since the summation of multiple forces due to individual magnets parallels the summation of multiple magnetic fields to produce a single force.

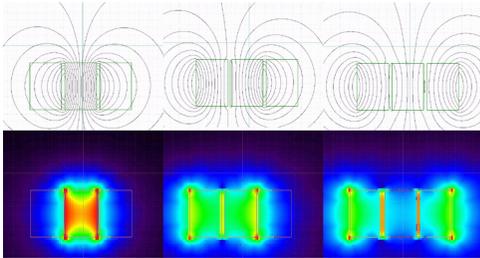


Figure 3.12: Magnetic field interactions between electromagnets. The top images show magnetic flux lines and the bottom images map flux density to brightness. The three image pairs show the fields resulting from a single center magnet turned on (left), the left and center magnets turned on (center), and all three magnets turned on (right). The effect of this field-shifting behavior can be modeled approximately using force summation. These images were generated with the VisiMag software package [5].

To produce a puck displacement Δx during a loop interval Δt , we activate the electromagnets with duty cycles α_i such that

$$f_{\text{TOTAL}} = 2m \frac{\Delta x - v_0 \Delta t}{\Delta t^2} \quad (3.4)$$

This equation assumes we are keeping track of the puck's instantaneous velocity v_0 . If we are using the Actuated Workbench in an “open-loop” mode in which we do not track the instantaneous position or velocity of the puck, we can still compute a reasonable estimate of f_{TOTAL} using a dead reckoning approach based on assumptions about how the previous electromagnet settings have affected the position and velocity of our puck according to our force model.

There are many ways in which we could activate the electromagnets so that the resulting forces summed to the desired value of f_{TOTAL} . In the next section, we describe several different methods for choosing the magnet values.

3.7 Anti-Aliasing Techniques

In computer graphics, the mathematical model of an image is a continuous analog signal that is sampled at discrete points called pixels. Aliasing occurs when the sampling frequency is too low for the signal frequency, resulting in a coarse image in which smooth curves are converted to steps and jagged outcrops. The anti-aliasing technique of prefiltering combats this problem by treating each pixel as an area, and computing pixel color based on the overlap of the scene's objects with a pixel's area.

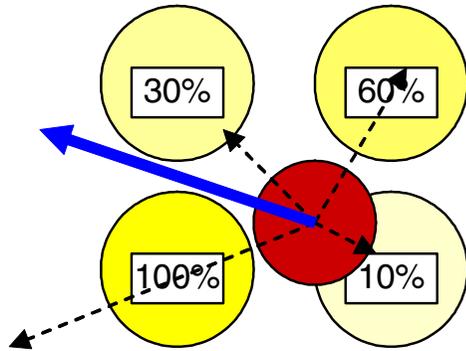


Figure 3.13: Four electromagnets with different duty cycles combine to produce a force with a new direction and magnitude.

With the Actuated Workbench, we are faced with a similar problem: we wish to render an analog signal (in this case, a force of a particular direction and magnitude) using a discrete array of cells (variable-duty electromagnets). To do so, we can employ a similar technique: the strength of each electromagnet is determined by the “overlap” of its magnetic flux lines with the location of the point

force. Figure 3.13 shows a configuration in which the forces of four neighboring electromagnets of different duty cycles combine to create a single force of a new magnitude and direction.

The simplest algorithm for anti-aliasing draws the computer graphics equivalent of a smoothed dot centered at the location of desired travel. Given a desired force vector with head at point (x,y) , we compute the distance from each electromagnet to (x,y) , and set its duty cycle in inverse proportion to this distance. As in computer graphics, we can choose any number of falloff metrics. We experimented with Gaussian falloff, but found that in practice it was no better than a simple linear falloff metric.

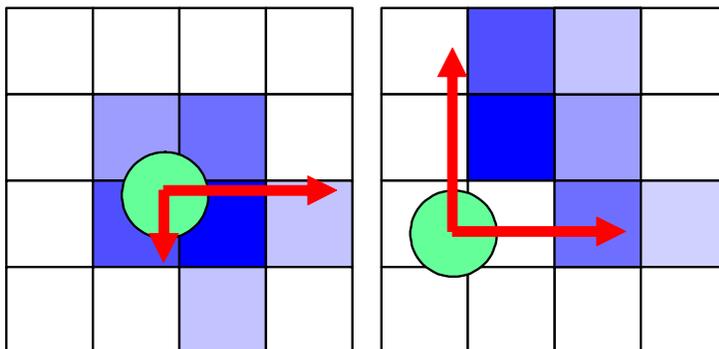


Figure 3.14: Anti-aliasing methods. The squares represent magnet cells, and intensity of each shaded square corresponds to the duty cycle of the magnet. The “dot” technique (left) results in slower travel than the “jet” technique (right).

A drawback of the dot-based method is that it limits the puck’s top speed of travel to about 15cm/sec. In order to produce enough force to move the puck, the center of the dot

must be positioned close to the puck, and the forces produced by some of the activated electromagnets will pull the puck backwards against the desired direction of travel (Figure 3.14).

If we know the position of the puck and the direction of travel that we hope to produce, we can pull the puck using only the electromagnets located in this direction relative to the puck. To do so, we first compute the vector from each electromagnet to the target (x,y) , and then compute the scalar projection of this vector onto the direction-of-travel vector. Taking the set of vectors of positive magnitude produces a collection of forces resembling a “jet” in

fluid mechanics (Figure 3.16). Jet-based movement can move pucks across the table almost as fast as Manhattan motion.

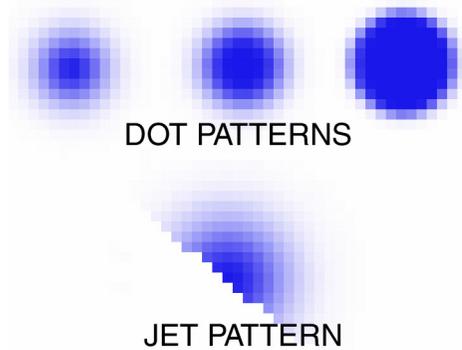


Figure 3.15: “Dot” and “Jet” equivalents in computer graphics. The three dots use different falloff metrics.

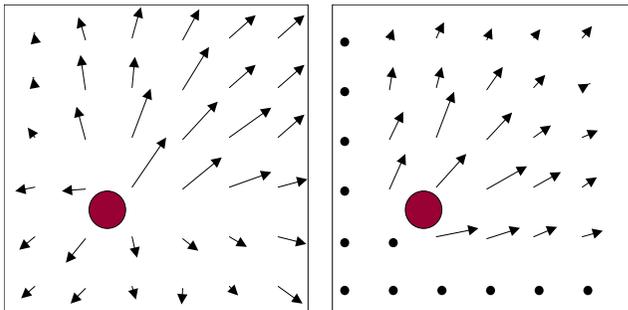


Figure 3.16: “Dot” force field (left) versus “Jet” force field (right).

Chapter 4

Applications

4.1 Types of Applications

Having developed a system meeting our design criteria for an interactive workbench actuation system, we began to imagine the new interaction techniques and applications that our system could support. This chapter begins by describing extensions of basic GUI functions into the physical domain, and then goes on to describe higher level applications, including some solutions to classic problems in interactive workbench interfaces. We built prototypes of many of these applications; other application ideas would require further development of the Actuated Workbench to address its limitations in speed, magnetic strength, scale, and resolution.

4.2 Extending GUI Functions to the Physical Domain

- **Search and retrieve.** As the number of pucks increases in an interactive workbench system, it becomes more difficult for a user to keep track of every item on the table, just as it is difficult to keep track of many graphical icons on a computer desktop. A search and retrieve function could respond to a user query by finding matching items and either moving them to another place on the tabletop or wiggling them to get the user's attention. Note that the Actuated Workbench would assist only in displaying the results of search queries; the input of the query expression would require a separate interface, such as Ullmer's token-based tangible query interface [50].

- **Sort.** A more powerful function would be one in which the computer could physically sort and arrange pucks on the table according to user-specified parameters. This could help the user organize a large number of data items before manually interacting with them.

- **History and Undo.** As a user makes changes to data through physical input, she may wish to undo some changes. A physical undo in this system could move the pucks back to their positions before the last change. It could also show the user the exact sequence of movements she had performed. In this sense, both “undo” and “rewind” commands are possible.

- **Teaching and Guiding.** Because the Actuated Workbench gives the computer the ability to recreate users’ gestures with the pucks, it becomes possible for the computer to teach the user something about interacting with the system through physical gestures. If specific gestures are used in the interface to trigger certain commands (such as a shaking gesture to unbind a puck from a data item), the computer can show a novice or a forgetful user how to make that gesture with the puck. This way, many of an application designer’s commands can be taught to users without the need for intensive human coaching. In addition, if a user is uncertain how to proceed while using a problem-solving or simulation system, the computer could suggest a physical configuration of the pucks.

4.3 Navigation

In an overhead map browsing interface such as metaDESK [49], it is often useful to rotate, translate, or scale the map. However, if there are physical objects on the map surface that correspond to map landmarks, their positions on the table will no longer be correct once the underlying map is moved. We built a map browsing interface that allows users to navigate around a map by moving physical objects and automatically corrects the positions of physical objects on the map surface using actuation.

Pushing the button on a puck locks down the position of its corresponding map landmark. The user can then move the puck to scroll the overhead map. Users can rotate or zoom the map using a two-handed technique; first a landmark is locked down, and then

another landmark is moved to specify a scaling and rotation transformation. Since these transformation operations change the position of each landmark on the table, actuation is necessary to preserve consistency between the physical and digital states of the system. As the user scrolls, rotates, or zooms the graphical display, the pucks automatically move on the surface of the table to keep up with the viewing transformation.

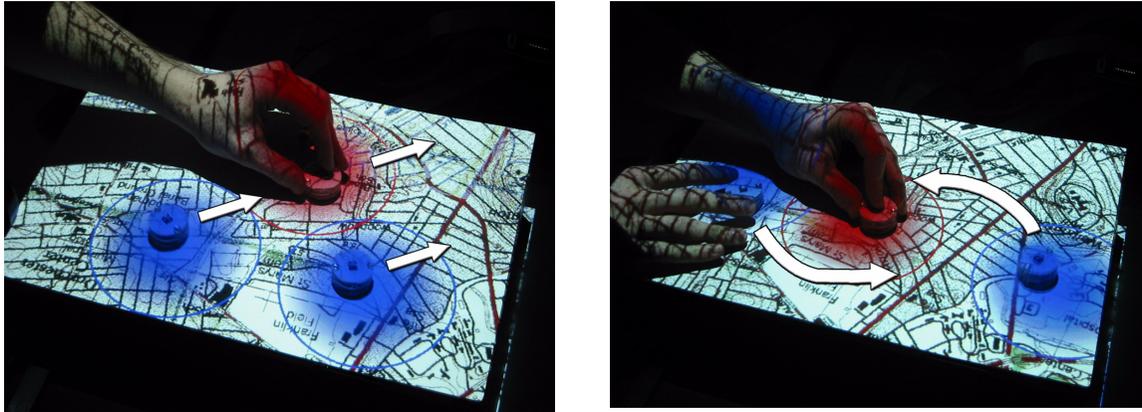


Figure 4.1: Translate and rotate/zoom operations. The puck that the user is holding has been locked to a particular map location.

4.4 Remote Synchronization

One advantage that interactive workbench interfaces offer is the ease with which multiple users can make simultaneous changes to the system. Users can observe each other's changes, and any user can reach out and physically change the shared layout without having to grab a mouse or other pointing device. This is not the case, however, when users are collaborating remotely. In this scenario, a mechanism for physical actuation of the pucks becomes valuable for synchronizing multiple physically separated workbench stations. Without such a mechanism, real-time physical synchronization of the two tables would not be possible, and inconsistencies could arise between the graphical projection and the physical state of the pucks on the table.

One example of a system that could benefit from physical synchronization is Urp [51]. In the Urp system, users manipulate physical models of buildings on a table and the com-

puter displays simulation information in the form of projected “digital shadows” around the buildings. “Distributed Urp” (Durp) later attempted to create distributed workspaces between multiple remote users. Identical Urp systems were set up in two separate locations, and the two systems were synchronized through identical graphical projections onto the workbench. However, if a user in one location moved a building, only the “digital shadow” of the building, and not the physical model, would move in the remote location. In addition to facilitating the simple synchronization of these models, the Actuated workbench could recreate remote users’ actual gestures with objects on the table, adding greatly to the “ghostly presence” [7] sought in remote collaboration interfaces.

We built a remote collaboration demonstration in which two remote tangible workbenches are kept synchronized. When the motion of a puck is sensed on one table, the remote table uses magnetic actuation to update the position of the corresponding remote puck, keeping the states of the two tables the same. If both users simultaneously attempt to move the same puck, each user will feel a force pulling against the puck, indicating the direction in which the remote user is attempting the move the object. In this situation, the puck becomes a conduit for interpersonal haptic communication. Actuation can therefore serve two purposes during remote collaboration: (1) synchronizing a shared workspace; and (2) providing a haptic communication link.

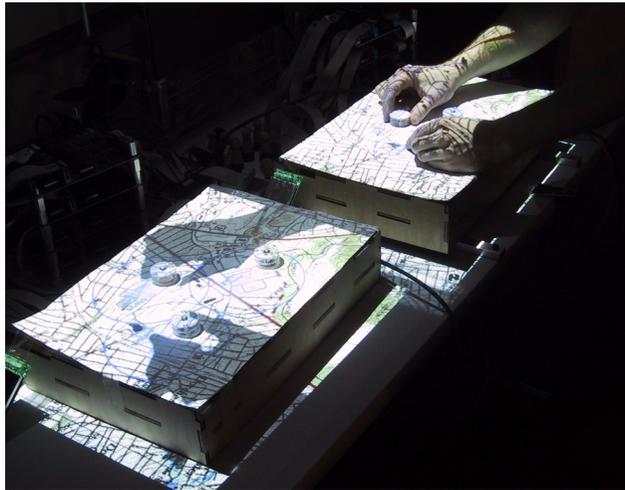


Figure 4.2: Two actuated tangible workbenches running a distributed application for cellphone tower placement. The tables maintain a synchronized state over an Internet connection. Translucent hand silhouettes show remote user activity.

4.5 Scientific Visualization

The Actuated Workbench could be helpful in the scientific visualization of complex mechanical systems. For example, a solar system model in the manner of an orrery could be created on an interactive interface with full actuation of the planetary orbits. The user could change the physical properties of the planets or teach the computer new orbital paths, and then watch the resulting motions of the planets.

Similarly, the Actuated Workbench could be used to teach students about physics by demonstrating the attraction and repulsion of charged particles represented by pucks on the table. As a student moved the pucks around on the table, the system could make them rush together or fly apart to illustrate forces between the objects.

4.6 Entertainment

In addition to these more practical applications, the Actuated Workbench could be used to add a physical dimension to computer entertainment. Though motorized chess sets have existed for many years, they operate using a single electromagnet mounted on an x-y plotter mechanism, limiting them to moving one object at a time. The Actuated Workbench could provide a significant improvement to these devices, making them more flexible for a variety of games. Classic computer games like Pong [2] could now be played using a physical puck and two physical paddles manipulated by the users. Distributed Pong could be played with a local user moving one paddle and the computer moving a remote user's paddle on the table. In addition, the Actuated Workbench can be used to flip over thin, polarized magnetic pucks by rapidly reversing the polarity of the electromagnets. This could be used to play a physical game of Reversi with the computer. Finally, one could create painting or drawing programs in which a pen or brush was attached to the puck. Various plotter-based computer-controlled drawing systems have been developed [11], but the Actuated Workbench could expand on these systems by using multiple pens

or sharing control between the computer and the user. The computer's movement of the puck could be used to teach certain artistic gestures or handwriting movements.

4.7 Specifying and Maintaining Constraints

Tabletop tangible interfaces are well suited to spatial layout of items in an application containing layout constraints. Examples of such constraints can be found in urban planning [51], circuit routing, and task scheduling [24]. Some of these constraints can be easily expressed as mathematical formulae, e.g. "two buildings may never be less than 20 meters apart." Making the computer responsible for monitoring these constraints frees the user to focus on the more salient aspects of the task at hand, instead of constantly having to check whether a proposed design violates any constraints.

We built a constraint solving system that supports both programmatic and physical constraints. Programmatic constraints are more appropriate for constraints that can be easily expressed mathematically, and do not change often in an application. Physical constraints are more appropriate for situations where a user would like to experiment with a constraint briefly, modify a constraint over time, or add new constraints as their design evolves. A physical constraint can simply be a user's hand on the table, holding a puck in place or preventing it from moving to a certain part of the table. A physical constraint may also be a physical "jig" placed around one or more pucks. We have used several types of jigs to represent common constraints:

- A “collar” can be placed around a puck to constrain its proximity to other pucks. This collar is simply a circular disk of plastic that bumps into other pucks to keep them a minimum distance away. The collar can be flipped over to raise the height of the plastic disc above the table. In this case, other pucks with raised collars will be kept farther away than those with lowered collars or no collars. In this way, minimum distance constraints can be applied selectively to different pucks.

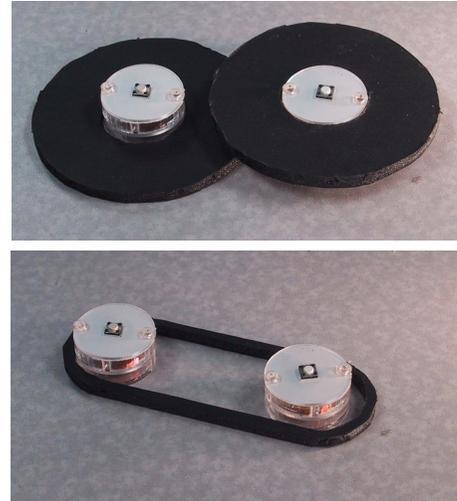


Figure 4.3: Collar and ring constraint jigs.

- A “ring” can ensure that pucks remain within a certain distance of each other. This constraint is an oval shaped piece of plastic that surrounds two or more pucks.
- Finally, a nonmagnetic object can be placed on the table and used to ensure that certain pucks stay inside or outside of a particular region on the board.

When developing software to support constraints with our actuation mechanism, we began by incorporating the GUI constraint solver Cassowary [1]. However, it was difficult to implement some parts of our system using Cassowary for two reasons. The first was Cassowary’s method of dealing with conflicting constraints: Cassowary will fully satisfy a higher priority constraint at the expense of a lower priority one. While this approach makes sense for problems like window placement in a GUI, it is not appropriate for many types of constraints on a TUI platform. In our actuated system, the constraint solver is run each time the system receives new data about the position of the pucks. With the Cassowary-style solution, slight differences in puck positions can lead to large discontinuities in the solution to the constraints. These discontinuities can cause pucks on the table to seem unresponsive to changes the user is making, and then suddenly fly across the table

unexpectedly. A better approach for TUI in these circumstances is to partially solve each of the conflicting constraints, thus providing less surprising motions on the table.

Another reason that traditional constraint solving toolkits like Cassowary are not well-suited to this task is that often these toolkits are not designed to handle constraints involving physical objects. In a constraint oriented GUI problem such as window placement, the constraint engine can safely assume that the window manager will be able to place a window at any position deemed optimal by the solver. However, when dealing with a physical system, there are issues of inertia, friction, and inter-object collision that must be considered when resolving constraints. In addition, there may be jigs or hands blocking the travel of pucks to the positions considered optimal by the solver.

We have developed an iterative solver for dealing with constraints involving physical objects. The system takes as input the current position of all pucks and the constraints to be resolved. For each constraint, it checks if the constraint is already resolved, and if not, it computes a motion vector for each puck that would resolve the constraint. The motion vectors for each puck for all constraints are fed into a weighted sum, based on the priority of each constraint. The resultant vector is scaled, based on how quickly the system should converge on a solution. This parameter controls the “viscosity” of the system, or how quickly the computer-controlled pucks respond to the user’s actions. The actuation mechanism then attempts to move each puck according to its corresponding motion vector. This attempt may or may not be successful depending on whether the puck is physically constrained. The system then reads the puck positions again and repeats the resolution process.

The process the actuation mechanism uses to resolve constraints has several interesting properties. First is that the physical properties of objects on the table become parameters in the constraint resolution process. The system will resolve constraints differently depending on whether a user is holding a puck, or a puck is otherwise physically con-

strained. The friction and mass of the pucks will also affect the results of the computation. For example, a puck with greater mass or friction will tend to stay close to its original position as the system moves pucks around it in an attempt to resolve constraints. This yields a tight coupling between the physical properties of objects on the table and the computational behavior of things they represent.

This constraint resolution approach also affords the use of jigs to represent and enforce constraints. One interesting property of these jigs is that they can prevent the user from establishing fundamentally inconsistent constraints, as shown in Figure 4.4; here the user is trying



Figure 4.4: An inconsistent set of constraints.

to constrain the system such that two pucks must be far apart and close together at the same time.

One issue with jigs is that they can sometimes interfere with each other in unusual ways. For example, the ring constraint between two pucks may inadvertently push a third puck out of the way. Minimum distance jigs affect all types of objects, making it difficult to use them for constraints that should only affect certain pucks. In practice, however, unexpected movements of objects make it clear that the system is interpreting a jig in an unintended way. The user can then temporarily remove the jig and constrain the pucks by hand.

4.8 Computer Movement vs. User Movement

In a remote collaboration scenario, confusion can arise as to whether an object was moved by the computer because of a constraint in the system, or by the remote user. To reduce this confusion, we displayed overlaid translucent silhouettes of the remote user's hands (Figure 4.2), captured from a camera mounted above the remote table. The silhouettes are

extracted from the captured image using background subtraction followed by thresholding and dilation operations. The projected silhouettes reinforce the ghostly presence of the remote user and indicate the focus of his attention.

4.9 Remote Collaboration using Jigs

One design trade-off for this system is whether or not to electromagnetically track the jigs. Our constraint system could potentially use knowledge about the positions of jigs to improve synchronization between the tables. In addition, in remote collaboration scenarios, all users should know what jigs are being used on the table. However, we believe an important property of this system is the ability to use any object to constrain a puck, not just a pre-made, tagged jig. To preserve this property we decided not to track the jigs. We use the same silhouetting technique as for displaying hands to keep users aware of all physical constraints in distributed collaboration scenarios. A physical constraint appears as a shadow on remote workbenches, as shown in Figure 4.5.

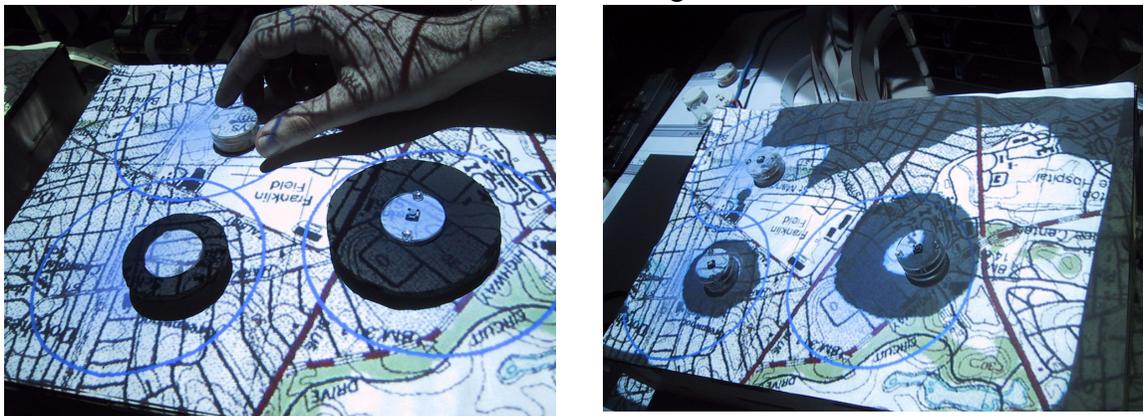


Figure 4.5: Hands and jigs and their projected silhouettes.

4.10 Resolving Conflicts in Remote Collaboration

When the system is being used in distributed collaboration mode, it uses a set of constraints to keep the tables consistent with each other. If two pucks are bound to the same object on different tables, the system constrains each puck to be in the same place relative to the board.

This approach assumes an actuation system with rapid response, since it does not distinguish between pucks that differ in position because they are being manipulated by both users and pucks that differ in position simply because one side is experiencing mechanical latency. If an object is quickly moved and then released, it will snap partially back, as it moves toward the average of its tracked positions on the local and remote tables. In the future, we could address this issue by adding capacitive touch sensors to our pucks to distinguish between pucks being manipulated by the users and pucks being manipulated by the actuation system. Unfortunately, this would require adding batteries to the pucks. Another possibility would be to introduce vision-based hand tracking, using the same camera that we currently use to extract the hand silhouettes.

Chapter 5

Conclusion

5.1 Summary

Tangible interfaces can benefit from the addition of computer-controlled actuation in a variety of ways. In addition to maintaining consistency between the physical and computational states of the system, actuation also enables new types of interactions, such as remote collaboration mediated through physical objects. This chapter describes the various benefits of computer-controlled actuation in tabletop tangible interfaces, and discusses some design parameters for the integration of actuation into these interfaces. It then goes on to suggest future research directions for actuation in tangible interfaces.

5.2 Maintaining Constraints

By maintaining constraints in tangible interfaces, actuation allows the computer to correct inconsistencies between digital and physical states. If the user attempts to adjust a parameter outside of a valid range, the computer can impose a physical limit on the value of the parameter. If an object on the table represents a dependent variable (i.e., its state depends on the state of another object), the computer can automatically update its state when the independent variable changes. Actuation also addresses the “nulling problem” [9]: the system can automatically attach physical pucks to digital parameters and adjust them to the appropriate value. In addition to making interaction more efficient, these techniques can make complex mathematical systems programmed in software behave more like transparent mechanical systems.

The usefulness of tangible physical constraints depends largely on the nature of the interaction task. If the task is highly structured and involves rigorous computation, constraints can quickly guide the user toward an optimal solution. However, the task may be

so highly structured (e.g. graph coloring) that human intervention is unnecessary, since the computer could probably determine the best solution on its own. If the task is free-form or artistic (calligraphy, sculpture), the importance of the human is emphasized, but the computer may have little to provide in the way of guidance. The types of tasks that make the best use of physical constraints are those that combine rigid mathematical constraints with aesthetic considerations of which the computer is unaware (as in architecture and urban planning) or with a negotiation component (as in business strategy planning) in which multiple people, each with a different objective in mind, must reach a compromise.

5.3 Haptic Feedback

In a system like the Actuated Workbench, information can be conveyed not only in the motion of pucks on the table, but also in the forces a user feels while holding a puck on the table surface. Users can explore electromagnetic or gravitational force fields, and feel forces that would otherwise be on a scale too large or too small to experience physically. By simulating notched sliders or knob endpoints, haptic feedback can also help users to adjust parameters by touch alone, allowing them to concentrate elsewhere in the interface [30][47].

Haptic feedback is particularly useful in situations in which visual attention is divided between the objects being manipulated and other information being displayed. In this situation, the additional physical I/O channel increases the amount of information that the user can simultaneously process. Even when the input and output spaces are coincident (a general design principle for tabletop tangible interfaces), the user's attention may be divided when multiple outputs change as the result of a single change in input.

5.4 Spatial Thinking

Tangible interfaces provide stronger spatial and kinesthetic memory cues than graphical

user interfaces, taking advantage of our natural tendency to remember information spatially and to use spatial organization to simplify problem solving. However, without the ability to restore saved physical configurations or to impose organizational constraints, the extent to which these spatial cues could be exploited in traditional tangible interfaces was limited. Adding actuation to a tangible interface can improve its capacity to leverage spatial thinking.

When a user creates a spatial configuration and ascribes a meaning to that configuration based on his own organizational criteria, the system can later load and restore the configuration he has previously established. In addition, the system has the power to impose an organizational scheme on the user, automatically sorting objects or snapping them to particular locations as the user arranges them. Studies have shown that encouraging the use of organizational strategies can lead to improved recall [38].

5.5 Remote Collaboration

Traditional tabletop tangible interfaces readily facilitate collocated collaboration, but are less effective in supporting remote collaboration, since they provide no means of maintaining a consistent physical state between remote systems. Actuation technology introduces this capability, eliminating the overhead of manual synchronization necessary in many tangible interfaces for remote collaboration [15]. In addition, the motion of physical objects rather than graphical ones may strengthen the sense of presence of the remote user, though this effect is much more difficult to evaluate quantitatively. In remote collaboration situations, actuation can be coupled with graphical feedback to help users understand which objects are being moved by the computer, and which are being moved by remote collaborators. Finally, when remote users hold the same object, each can feel the force exerted by the other, and the object becomes a conduit for haptic communication, much in the manner of ComTouch [10]. The actuation then serves a dual purpose, both synchroniz-

ing physical state and mediating interpersonal communication.

5.6 Future Vision

Although the system we constructed proved to be an effective platform for prototyping and demonstrating many applications of actuation in tangible user interfaces, not all of our ideas could be realized due to its shortcomings in speed, scale, and resolution. With some additional mechanical and electrical engineering, many of these limitations could be overcome, giving rise to a variety of new interaction possibilities. These possibilities are discussed in Appendix B.

In addition to these mechanical and electrical improvements, simplifying the software API would greatly improve the overall packaging. With some software redesign, the Actuated Workbench could provide a single, stable platform for integrated object tracking, graphical display, and actuation. This unified software architecture would allow for rapid prototyping of actuated tangible interfaces. A discussion of these considerations can be found in Appendix C.

Though this thesis focused on two-dimensional actuation on tabletop surfaces, many other kinds of actuation are possible in tangible interfaces, and many of the same concepts would apply to these alternative actuation systems. We discuss some future possibilities for alternative actuated interfaces in Appendix D.

Appendix A

Mathematical Derivations

A.1 Motion Control Equations

The magnitude of the frictional force on the moving puck is given by the equation

$$\|f_{\text{FRICTION}}\| = \mu_k \|n\| \quad (\text{A.1})$$

where μ_k is the dimensionless coefficient of kinetic friction and n is the normal force on the puck. The value of μ_k can range from 0.05 to 1.5, depending on the choice of materials for the puck bottom and the table surface. When the puck is stationary, we replace μ_k with the coefficient of static friction μ_s . In general, $\mu_s > \mu_k$, but we simplified matters by trying to choose materials for which μ_s and μ_k were nearly identical.

When none of the electromagnets are activated, the normal force on the puck is the sum of the gravitational force on the puck and the attractive force between the permanent magnet in the puck and the iron cores of the electromagnets below. Since the iron cores are spaced at intervals, this attractive force varies with the position of the puck on the table, but our simplified model assumes that this variation is slight enough to be negligible. Substituting these values, the equation for friction becomes

$$\|f_{\text{FRICTION}}\| = \mu \left(mg + \frac{\|f_p\|}{z_s^2} \right) \quad (\text{A.2})$$

where m is the mass of the puck, g is the acceleration due to gravity, f_p is a constant force of attraction proportional to the strength of the permanent magnet in the puck, and z_s is the vertical separation between the puck and the electromagnets. The z_s term is squared because the magnetic force between two objects attenuates in proportion to the square of the distance between the objects. In practice, we chose a value for z_s large enough to make the contribution of negligible, but small enough that the attractive forces of the activated

electromagnets are able to move the puck. With our current materials, we empirically found a z_S of 1.89cm gave good results.

We modeled the magnetic force between the puck and an activated electromagnet using the equation

$$f_{\text{MAG}} = \frac{\alpha f_E}{x_S^2 + y_S^2 + z_S^2} \quad (\text{A.3})$$

where α is the duty cycle of the pulse-width-modulated electromagnet (0-100%), f_E is a constant-magnitude force of attraction proportional to the strengths of the electromagnet and the permanent magnet in the puck, and x_S and y_S are the separation distances between the puck and the activated electromagnet along the horizontal axes. The direction of f_E is from the center of the puck to the center of the upper end of the electromagnet. Note that the z component of f_{MAG} will contribute to the normal force, increasing the magnitude of f_{FRICTION} . This can actually be desirable: as the puck approaches its target, the z -component of f_{MAG} increases, increasing the friction and preventing the puck from overshooting its goal.

We can sum the contributions of each activated electromagnet to compute the net force on the puck due to the electromagnets:

$$f_{\text{MAG-NET}} = \sum_i \frac{\alpha_i f_E}{(x - x_i)^2 + (y - y_i)^2 + z_S^2} \quad (\text{A.4})$$

In this equation, the puck is positioned at (x, y) and each electromagnet i is positioned at (x_i, y_i) with duty cycle α_i .

Adding the z component of $f_{\text{MAG-NET}}$ to the normal force in our friction equation, we reach the final equation for net friction:

$$\|f_{\text{FRICTION-NET}}\| = \mu \left(mg + \frac{\|f_P\|}{z_S^2} + \|f_{\text{MAG-NET}} \cdot \hat{z}\| \right) \quad (\text{A.5})$$

where \hat{z} is the unit vector in the direction of the positive z -axis. The direction of the friction vector $f_{\text{FRICTION-NET}}$ is opposite the direction of the applied force, in this case the horizontal components of $f_{\text{MAG-NET}}$. We are now ready to write an equation for the total horizontal force on the puck:

$$f_{\text{TOTAL}} = f_{\text{MAG-NET}} \cdot \hat{x} + f_{\text{MAG-NET}} \cdot \hat{y} + f_{\text{FRICTION-NET}} \quad (\text{A.6})$$

where \hat{x} and \hat{y} are the positive unit vectors along the horizontal axes. The acceleration of the puck is proportional to this total force:

$$a_{\text{PUCK}} = f_{\text{TOTAL}}/m \quad (\text{A.7})$$

The resulting velocity of the puck is given by the standard differential equation

$$\frac{\partial x}{\partial t} = a_{\text{PUCK}}t + v_0 \quad (\text{A.8})$$

which can be reduced to the position equation

$$x(t) = \frac{1}{2}a_{\text{PUCK}}t^2 + v_0t + x_0 \quad (\text{A.9})$$

where v_0 and x_0 are the instantaneous velocity and position of the puck, respectively. This means that if we keep track of the puck's velocity and position, we can produce any desired displacement Δx of the puck during loop interval Δt by solving this equation for a_{PUCK} and in turn for f_{TOTAL} :

$$f_{\text{TOTAL}} = 2m \frac{\Delta x - v_0 \Delta t}{\Delta t^2} \quad (\text{A.10})$$

We then find a combination of electromagnet settings α_i that produce this net force f_{TOTAL} , as described in Section 3.7.

Appendix B

Engineering Improvements

B.1 Tiling of Actuation Surfaces and Scale

Though each 64-magnet actuation grid is only 16.5cm square, we have successfully tiled two of them together to form a single table with twice the actuation area. Tiling four or even six grids together would produce an even larger surface, increasing the number of pucks that could simultaneously fit on the table and broadening the potential types of interaction. In addition, it would be interesting to explore the use of different sizes of electromagnets. Smaller electromagnets may yield higher resolution of object movement on the table, while larger or more powerful electromagnets may provide more force for moving objects, making it possible to provide stronger force feedback.

B.2 Puck Modifications

Though we focused primarily on translational movements of the pucks, computer-controlled rotation could be useful as well, since there are many applications in which the orientation of objects is significant. Puck rotation could be accomplished by designing new, slightly larger pucks in which permanent magnets are placed in the puck with opposite polarities facing downward. The two sides of the puck could then be pulled using opposite magnet polarities to control the puck's rotation. Alternately, one could achieve a similar effect by designing pucks with a single magnet oriented sideways, with the north pole on one side of the puck and the south pole on the other. This would make it easy to spin the puck about its center, but would make smooth translational motion more difficult.

B.3 New Types of Motion

In addition to controlling orientation, the Actuated Workbench is also capable of flipping

over magnetic objects or launching them into the air by reversing the polarity of the electromagnet underneath the object. If the polarities of the electromagnet and the permanent magnet are the same, a strong repulsion results. This repulsion could be used to flip over a double-sided object, so that the opposite side was attracted downward. A brief negative polarity burst could also provide a vibrational haptic cue, indicating to the user that he has crossed a boundary or threshold [35].

Since the strength of the magnetic field can be quickly controlled in any part of the table, the Actuated Workbench is theoretically capable of levitating magnetic objects above the table. The Hover system [34] levitated objects on a column of air to attract attention, but this levitation system was noisy and its scalability was limited. Magnetic levitation could silently hover multiple objects, but would require constant object monitoring and rapid adjustments in field configuration; a stable configuration of static magnetic forces is incapable of maintaining levitation, as stated by Earnshaw's Theorem [14].

Even if full levitation is not possible (or useful) in the future, small repulsive forces could be used to provide greater control over the friction between the pucks and the table surface. Giving the pucks a small "kick" to help them overcome static friction, or using repulsion as well as attraction to create a push-pull actuation system, could result in new motion possibilities.

Appendix C

Software Improvements

C.1 Coordinating Multiple Objects

Though our current system can manipulate and track multiple objects on the table, our software contains no mechanisms for preventing collisions between pucks. In the future, it may be useful to design path-planning algorithms to coordinate the simultaneous motion of multiple pucks. This would enable us to set the positions of multiple pucks to any configuration without the danger of destabilizing the system due to magnetic interactions between pucks.

C.2 Unified Software Architecture

All of our applications were programmed using three separate APIs: one for position tracking, one for graphical display, and one for actuation. The complexity that this entailed suggests a need for a single high-level API to support the design of actuated tabletop tangible interfaces.

Figure C.1 shows the data flow during a single iteration of the Actuated Workbench control loop. After sensing the positions of the objects on the table and receiving the positions of the objects on the remote workbenches, the system computes new target positions based on constraints and simulation rules. It then sets the state of the local magnet array, renders graphical simulations, and transmits the local object positions to the remote workbenches.

A well-designed software interface to this system would require the programmer to specify only a few parameters: (1) addresses of remote workbenches; (2) the ID of each physically instantiated object, local and remote; (3) equations relating object positions that specify the simulation parameters and constraints; and (4) graphical visualizations. The

system would then automatically handle the data exchange between the tracking system, the remote workbenches, and the array of magnets. This would eliminate a large portion of the overhead currently associated with designing a new application for the Actuated Workbench.

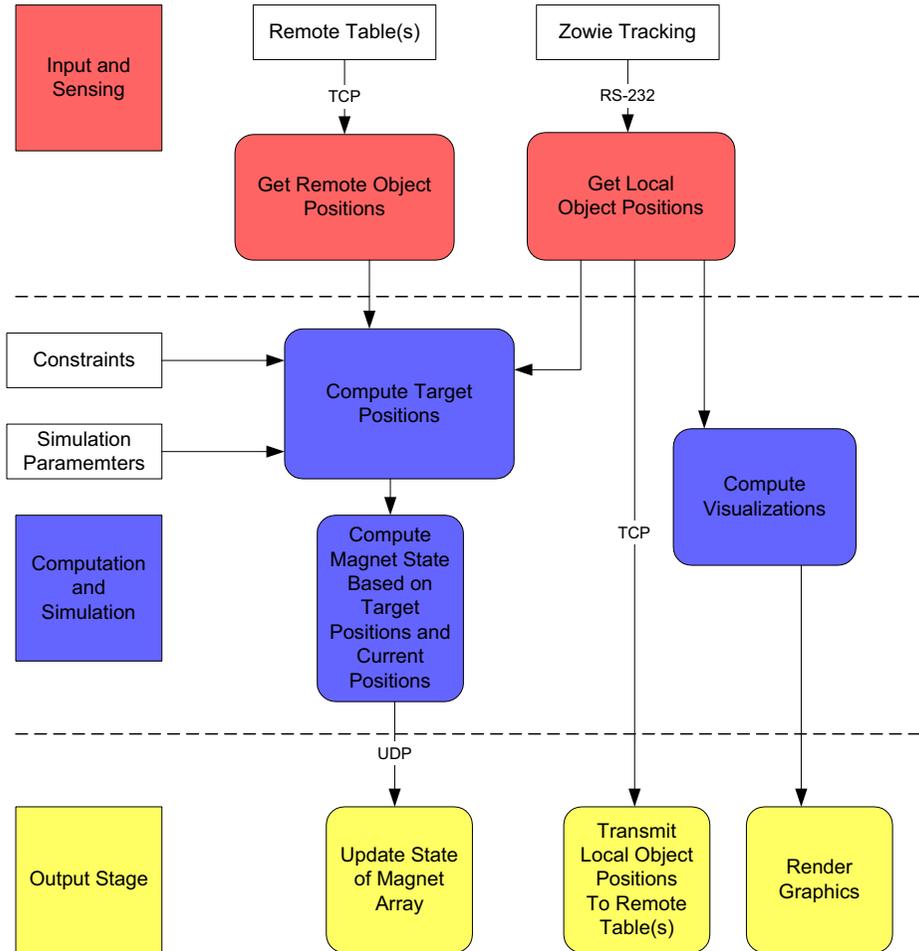


Figure C.1: Data flow diagram showing one iteration of the Actuated Workbench control loop.

Appendix D

New Directions for Actuation

D.1 Types of Actuated Interfaces

Researchers have developed a huge variety of computer-controlled actuation systems for human-computer interaction. This section describes some criteria for classifying these systems, with an eye towards identifying promising new areas of the design space.

D.2 Purpose of Actuated Feedback

In general, actuation systems in computer interfaces serve one of two objectives: providing haptic feedback, or changing physical layout. Haptic feedback is traditionally used in graphical user interfaces, in which the onscreen information is controlled remotely with an input device like a mouse or joystick. Since attention is divided between input and output in these interfaces, this touch feedback is particularly useful.

In most tabletop tangible interfaces, the input and output spaces are coincident, so haptic feedback is less important. However, since the physical layout of the objects is significant, it becomes useful for the computer to control the layout in the same way the user can. Tabletop tangible interfaces can then take advantage of spatial multiplexing without losing the flexibility of the GUI.

The way in which the computer interprets the configuration of physical objects in a tabletop tangible interface can range from the literal to the abstract. In the majority of tangible workbench systems, the configuration of objects specifies a fairly concrete geometric relationship. However, a new class of tangible interfaces is emerging that interprets geometric layout in a more abstract way, mapping physical layouts to ideas and relationships with no direct parallels in the physical world.

D.3 Spatial Complexity and Degrees of Freedom

In general, the trend in actuation systems has been towards higher spatial complexity. The earliest technologies that could be considered actuation systems for HCI were devices such as force feedback knobs, sliders, and steering wheels, and mice with small vibrating motors. Two-axis force feedback devices emerged soon afterward, including joysticks and mice on fixed pads with two-dimensional position control.

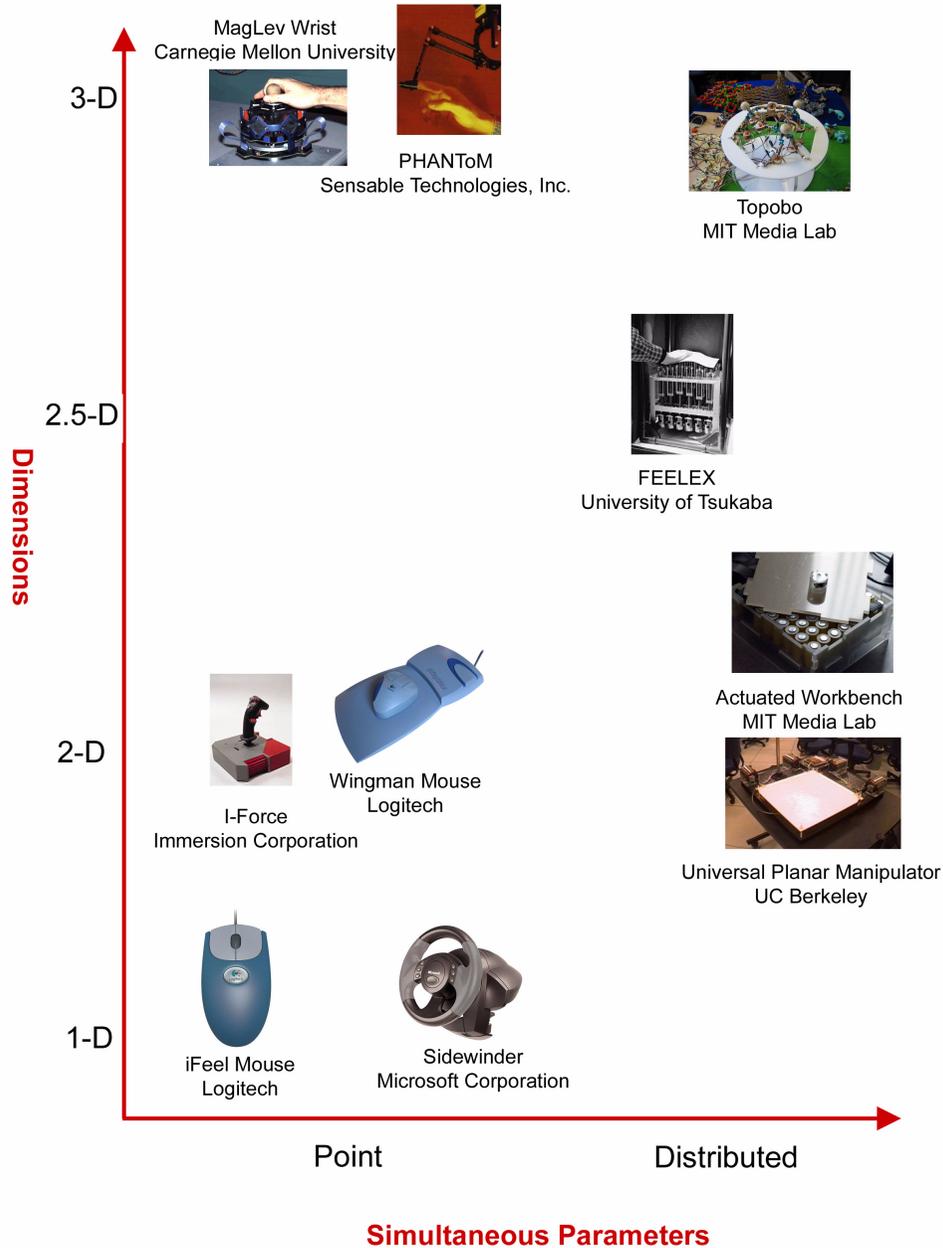


Figure D.1: Comparison of degrees of control in various actuation systems.

Newer technologies have brought even higher dimensionalities to actuation, as shown in Figure D.1. There is a distinction, however, between actuation dimensions and degrees of freedom. A system like the PHANToM [33] provides three dimensions of actuation, but only outputs forces at a single point. The Actuated Workbench operates only in two dimensions, but it is a distributed system capable of coordinating the simultaneous motion of multiple objects. As both the dimensionality and the simultaneous number of parameters in the interface increase, so does the potential expressiveness of the interface, but the difficulty becomes preventing the complexity from becoming overwhelming. As Stu Card wrote, “a major challenge of the post-WIMP interface is to find and characterize appropriate mappings from high degree-of-freedom input devices to high degree-of-freedom tasks.”

References

- [1] Badros, G. and Borning, A. "The Cassowary Linear Arithmetic Constraint Solving Algorithm: Interface and Implementation," Technical Report. UW-CSE-98-06-04, June 1998.
- [2] Baer, R., Television Gaming and Training Apparatus. US Patent #3728480, 1973.
- [3] Bal-shalom, Y. and Fortmann, T.E. Tracking and Data Association. Academic Press, 1988.
- [4] Batter, J.J. and Brooks, F P., Jr. Grope-1: A computer display to the sense of feel. In *Proceedings of IFIP 1972*, pp. 759-763.
- [5] Beetson, J.S. Visualizing Magnetic Fields: Numerical Equation Solvers in Action. Academic Press, 2001.
- [6] Berkelman, P., Hollis, R., and Baraff, D. Interaction with a Realtime Dynamic Environment Simulation using a Magnetic Levitation Haptic Interface Device. IEEE International Conference on Robotics and Automation, May 1999, pp. 3261 - 3266.
- [7] Brave, S., Ishii, H. and Dahley, A. Tangible Interfaces for Remote Collaboration and Communication. In *Proceedings of CSCW '98*, ACM Press, pp. 169-178.
- [8] Brooks, F. P., Jr. The computer scientist as toolsmith: studies in interactive computer graphics. In *Proceedings of IFIP 1977*, pp. 625-634.
- [9] Buxton, W. There's More to Interaction than Meets the Eye: Some Issues in Manual Input. In Norman, D.A. and Draper, S.W. (Eds.), (1986), *User Centered System Design: New Perspectives on Human-Computer Interaction*. Lawrence Erlbaum Associates, Hillsdale, New Jersey, 319-337.
- [10] Chang, A., O'Modhrain, S., Jacob, R., Gunther, E., Ishii, H. ComTouch: Design of a Vibrotactile Communication Device. In *Proceedings of Design of Interactive Systems 2002*.
- [11] Cohen, H. The Further Exploits of AARON, Painter. *Stanford Humanities Review*, Volume 4, Issue 2: Constructions of the Mind, 1995.
- [12] Cohen, J., Withgott, M., and Piernot, P. 1999. "Logjam: A Tangible Multi-Person Interface for Video Logging." In *Proceedings of CHI 1999*, ACM Press, pp. 128-135.
- [13] Dumas, S.T. and Jones, W. P. A Comparison of Symbolic and Spatial Filing. In *Proceedings of CHI 1998*, ACM Press, pp. 127-130.
- [14] Earnshaw, S. "On the Nature of the Molecular Forces which Regulate the Constitution of the Luminiferous Ether," *Transactions of the Cambridge Philosophical Society*, v. 7, 1842, pp. 97-112.
- [15] Everitt, K., Klemmer, S., Lee, R., and Landay, J. Two Worlds Apart: Bridging the Gap Between Physical and Virtual Media for Distributed Design Collaboration. In *Proceedings of CHI 2003*, ACM Press, pp. 553-560.
- [16] FCS Control Systems. 2001. FCS HapticMASTER. <http://www.fcs-robotics.com>.
- [17] Fitzmaurice, G.W., Ishii, H., and Buxton, W. Bricks: Laying the Foundations for Graspable User Interfaces. In *Proceedings of CHI 1995*, ACM Press, pp. 442-449.
- [18] Fitzmaurice, G. Graspable User Interfaces. Ph.D. Thesis, University of Toronto, 1996.
- [19] Fjeld, M., Bichsel, M., and Rauterberg, M. BUILD-IT: An Intuitive Design Tool Based on Direct Object Manipulation. In *Gesture and Sign Language in Human-Com-*

- puter Interaction, Lecture Notes in Artificial Intelligence*, v.1371, Wachsmut and Fröhlich, eds. Berlin: Springer-Verlag, pp. 297-308.
- [20] Francetic, A. and Shwe, H. Smart Toys: Benefits of Technology-Enhanced Play. In *Proceedings of Game Design Conference 2000*.
- [21] Immersion Corporation, "FEELit Mouse." Technical Document, San Jose, CA, October 1, 1997. Electronic version: <http://www.immerse.com>.
- [22] Iwata, H., Yano, H., Nakaizumi, F., and Kawamura, R. Project FEELEX: Adding Haptic Surface to Graphics. In *Proceedings of SIGGRAPH 2001*, ACM Press, pp. 469-476.
- [23] Ishii, H. and Ullmer, B., Tangible Bits: Towards Seamless Interfaces between People, Bits and Atoms. In *Proceedings of CHI 1997*, ACM Press, pp. 234-241.
- [24] Jacob, R., Ishii, H., et al. A Tangible Interface for Organizing Information Using a Grid." In *Proceedings of CHI 2002*, pp 339-346.
- [25] Jain, R. et al. Machine Vision. McGraw-Hill, 1995.
- [26] Kirsh, D. Complementary Strategies: Why We Use Our Hands When We Think. In *Proceedings of 17th Annual Conference of the Cognitive Science Society 1995*.
- [27] Kirsh, D. The Intelligent Use of Space. *Journal of Artificial Intelligence*, 72(1-2), 31-68, 1995.
- [28] Lansdale, M.W. Remembering About Documents: Memory for Appearance, Format, and Location. *Ergonomics*, 34, 8 (1991), 1161-1178.
- [29] Luntz, J., Messner, W., and Choset, H. Virtual Vehicle: Parcel Manipulation and Dynamics with a Distributed Actuator Array. In *Proceedings of the SPIE International Symposium on Intelligent Systems and Advanced Manufacturing*, Volume SPIE 3201, 1997.
- [30] MacLean, K., Snibbe, S., and Levin, G. 2000. Tagged Handles: Merging Discrete and Continuous Manual Control. In *Proceedings of CHI 2000*, ACM Press, pp. 225-232.
- [31] Malone, T.W. How do People Organize their Desks? Implications for the Design of Office Information Systems. *ACM Transactions on Office Information Systems* 1, 1 (January 1983), pp. 99-112.
- [32] Mandler, J. M., Seefmiller, D., Day, J. On the Coding of Spatial Information. *Memory & Cognition* 5, 1 (1997), 10-16.
- [33] Massie, T., and Salisbury, K., The PHANToM Haptic Interface: A Device for Probing Virtual Objects, *ASME Winter Annual Meeting*, Volume 55-1, pp. 295-300, 1994.
- [34] Maynes-Aminzade, D., Tan, B., Goulding, K., and Vaucelle, C. Hover: Conveying Remote Presence. Technical Sketch, SIGGRAPH 2002.
- [35] Nashel, A. and Razzaque, S. Tactile Virtual Buttons for Mobile Devices. *Extended Abstracts of CHI 2003*, pp. 854-855.
- [36] Naveh-Benjamin, M. Coding of Spatial Location Information: An Automatic Process? *Journal of Experimental Psychology: Learning, Memory, and Cognition* 13, 4 (1987), 595-605.
- [37] Negroponte, Nicholas and the Architecture Machine Group, MIT. "Seek." Originally shown at Software, Information Technology: Its New Meaning for Art Exhibition, Jewish Museum, New York, 1970.

- [38] Patten, J., and Ishii, H., A Comparison of Spatial Organization Strategies in Graphical and Tangible User Interfaces. In *Proceedings of DARE 2000*, pp. 41-50.
- [39] Patten, J., Ishii, H., Hines, J., Pangaro, G. Sensetable: A Wireless Object Tracking Platform for Tangible User Interfaces. In *Proceedings of CHI 2001*, ACM Press, pp. 253-260.
- [40] Patten, J., Recht, B., and Ishii, H. Audiopad: A Tag-based Interface for Musical Performance. In *Proceedings of Conference on New Interface for Musical Expression 2002*, pp. 11-16.
- [41] Piper, B., Ratti, C., and Ishii, H. Illuminating Clay: A 3-D Tangible Interface for Landscape Analysis. In *Proceedings of CHI 2002*, ACM Press, pp. 355-362.
- [42] Reznick, D. and Canny, J. C'mon Part, Do the Local Motion. In *Proceedings of the IEEE International Conference on Robotics & Automation 2001*.
- [43] Robertson, G., Czerwinski, M., Larson, K., Robbins, D.C., Thiel, D., van Dantzich, M. Data Mountain: Using Spatial Memory for Document Management. In *Proceedings of UIST 1998*, ACM Press, pp. 153-162.
- [44] Schäfer, K., Brauer, V., and Bruns, W. A New Approach to Human-Computer Interaction: Synchronous Modelling in Real and Virtual Spaces. In *Proceedings of DIS 1997*, pp.335-344.
- [45] Shimoga, K. B. A Survey of Perceptual Feedback Issues in Dexterous Telemanipulation: Part I. Finger Force Feedback. *IEEE Annual Virtual Reality International Symposium 1993*, pp. 263-270.
- [46] Shimoga, K. B. A Survey of Perceptual Feedback Issues in Dexterous Telemanipulation: Part II. Finger Touch Feedback. *IEEE Annual Virtual Reality International Symposium 1993*, pp. 271-279.
- [47] Snibbe, S. et al. Haptic Techniques for Media Control. In *Proceedings of UIST 2001*, ACM Press, pp. 199-208.
- [48] Tan, D.S., Stefanucci, J.K., Proffitt, D.R., Pausch, R. Kinesthetic Cues Aid Spatial Memory. In *Extended Abstracts of CHI 2002*, pp. 806-807.
- [49] Ullmer, B. and Ishii, H. The metaDESK: Models and Prototypes for Tangible User Interfaces, in *Proceedings of UIST 1997*, pp. 223-232.
- [50] Ullmer, B., Ishii, H., and Jacob, R. Tangible Query Interfaces. In *Proceedings of Interact 2003*.
- [51] Underkoffler, J. and Ishii, H. Urp: A Luminous-Tangible Workbench for Urban Planning and Design. In *Proceedings of CHI 1999*, ACM Press, pp. 386-393.
- [52] Virtual Technologies, "CyberTouch." Company brochure, Palo Alto, CA, 1998. Electronic version: <http://www.virtex.com>.
- [53] Wellner, P. Interacting with paper on the Digital Desk. *Communications of the ACM*, 36(7), 86-96, 1993.
- [54] Zhang, J. and Norman, D. A. Representations in Distributed Cognitive Tasks. *Cognitive Science* 18, 1 (1994), 87-122.

Chapter 1

Introduction

1.1 Tangible User Interfaces

For most of the history of computing, the primary means of representing digital information has been screen-based text and graphics. The WIMP (Windows, Icons, Menus, and Pointer) interface metaphor, which grew out of the computational capabilities available in the late 1970's, has become highly pervasive. Although it represents a vast improvement over the batch processing and command line interfaces found in the earliest days of computing, it is unlikely to meet the demanding user interface needs of modern, real-time, high-performance applications. The WIMP paradigm has an inherent asymmetry between input and output: while it uses millions of pixels of graphical output, it uses a single cursor-driven point input, with no physical or kinesthetic affordances and no way to engage multiple users, multiple hands, or multiple senses. It is a predominantly visual paradigm that does not take advantage of the natural abilities humans have developed through a lifetime of interaction with the physical world.

The limitations of graphical user interfaces have resulted in many new branches of research that expand our vision of human-computer interaction beyond the limited modes available via a traditional keyboard and mouse. The new research areas of virtual reality, wearable computing, ubiquitous computing, and augmented reality take advantage of new input and output technologies such as large, high-resolution displays, high fidelity audio, haptic feedback devices, voice and gesture input, and pen computing.

Of particular relevance to this thesis is a type of interface that attempts to bridge the gap between cyberspace and the physical world by giving physical form to digital information. These “graspable” [18] or “tangible” [23] user interfaces use physical objects to

represent both digital information and computational operations, shifting computation from the onscreen desktop to our bodies and the physical spaces we inhabit. The tangible user interface (TUI) makes use of natural physical affordances to achieve a more seamless interaction between people and information. TUIs allow two-handed manipulation, simplify collocated multi-user collaboration, offer kinesthetic memory cues, leverage existing physical skills and work practices, and take advantage of our natural ability to spatially organize objects to solve problems.

1.2 Interactive Workbenches

Recently, many tangible user interfaces have adopted a “workbench” metaphor, in which the user manipulates tracked objects on an interactive surface. The desktops, walls, and windows of an architectural space are thereby transformed into active interfaces between the physical and virtual worlds. A computer senses the positions and states of the graspable objects on the surfaces and responds to users’ physical input with graphical output, projected on and around the objects on the table. These interfaces have been used for a wide variety of applications, from musical performance [40] to video logging [12]. Systems such as the DigitalDesk [53], Bricks [17], Sensetable [39], and Urp [51] offer many advantages over purely graphical interfaces, including the ability for users to organize objects spatially to aid problem solving, the potential for two-handed interaction, and ease of collaboration between multiple collocated users.

However, current interactive workbench systems share a common weakness. While input occurs through the physical manipulation of tangible objects, output is displayed only through sound or graphical projection on and around the objects. As a result, the objects can feel like loosely coupled handles to digital information rather than physical manifestations of the information itself.

In addition, the user must sometimes compensate for inconsistencies when links between the digital data and the physical objects are broken. Such broken links can arise when a change occurs in the computer model that is not reflected in a physical change of its associated object. With the computer system unable to move the objects on the table surface, it cannot undo physical input, correct physical inconsistencies in the layouts of the objects, or guide the user in the physical manipulation of the objects. In short, the physical interaction between human and computer remains one-sided.

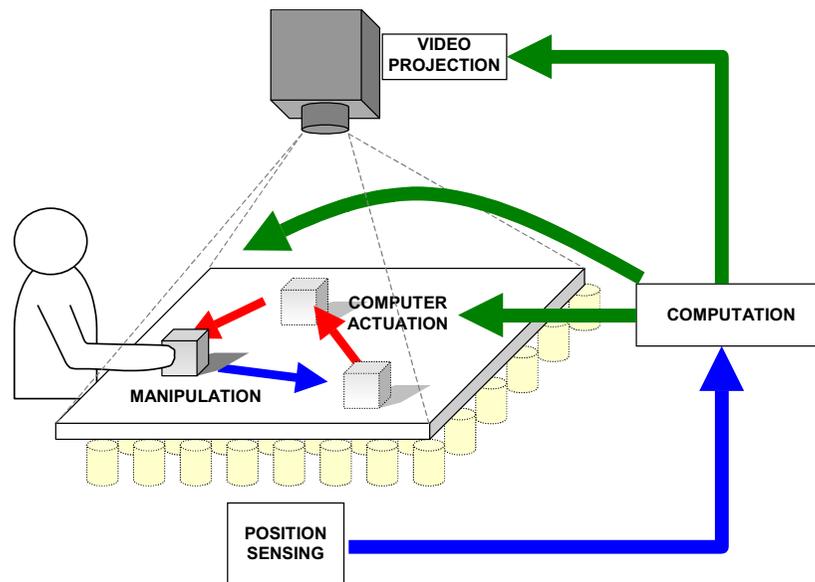


Figure 1.1: Traditional interactive workbench systems provide feedback through video projection alone. The Actuated Workbench adds an additional feedback loop using physical movement of the tracked objects.

1.3 Inconsistencies in Tangible Interfaces

Inconsistencies can occur in various types of interactions:

- **Remote Collaboration.** When there are multiple instantiations of a work table on which physical objects represent digital information, the physical state of each table can become inconsistent whenever a user moves objects on one table, but the remote user does not move the corresponding objects on the other table. Though the graphical projections

can be synchronized, the discrepancies between the physical positions of objects on the tables will remain.

- **Simulation.** A software simulation running in real-time may compute that the value associated with an object on the table has changed. The object's position or orientation may be inconsistent with the new value of its corresponding software parameter.

- **Constraints.** An application may have constraints on the spatial arrangement of objects on the table, such as zoning laws in an urban planning application. If the user moves an object to the wrong part of the table, some of these constraints may be violated. Existing systems can provide graphical feedback to inform the user that a constraint has been violated, but cannot fix the problem in both the digital and physical representations.

- **Navigation.** In an application with navigation features, such as one that uses a map [49], it is useful to rotate, translate or scale the map by moving the physical objects on top of it. However, if there are several physical objects representing fixed landmarks on the map, their positions on the table will no longer be correct once the underlying map is moved.

1.4 Preventing Inconsistency

There are several design approaches to dealing with the problem of inconsistency. One is to structure the interaction so that the physical objects can never become inconsistent with their digital counterparts. For example, in the Urp system [51], the user moves building models around on the tabletop, and the system displays graphical information in response. In this case the software has no control over the position or orientation of the buildings, so building placement cannot cause inconsistency. For some applications this approach works well, but it tends to limit the system's practicality in real-world situations.

Another approach to inconsistency is to use abstract physical objects, relying on graphics to represent system state. An example of this technique would be to represent the position of a dial with a plain circular “puck” and a projected arrow (Figure 1.2), rather than an arrow that is physically part of the puck. In this way the system can

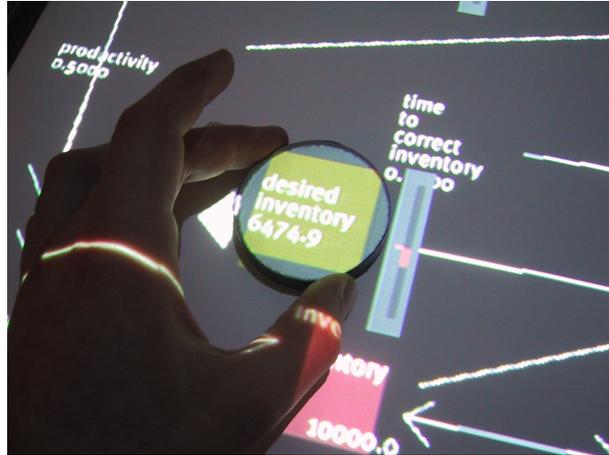


Figure 1.2: Addressing the nulling problem.

rotate the arrow independently of the physical puck if necessary. If the arrow were physically part of the puck, the user would have to set it to the correct position associated with the digital value of the parameter, a difficulty Buxton has referred to as “the nulling problem” [9]. This use of abstract objects for input works well for dealing with the rotation of pucks, but it cannot be applied to cases where the pucks must be translated.

Yet another approach is to provide graphical feedback to alert the user when an inconsistency arises. A projected graphical “shadow” of an object can represent that the physical object is not where the computer thinks it should be. The computer can then ignore the physical object until the user has “reattached” it to its shadow. In practice, this method can break the flow of interaction with the system and confuse users. It runs contrary to the design goal that the objects should be embodiments of digital information rather than just handles attached to that information.

1.5 Research Approach and Thesis Overview

By adding computer-controlled actuation to tabletop tangible interfaces, we provided new solutions to many of these inconsistency problems. We designed and built a system called the Actuated Workbench, a hardware and software infrastructure enabling a com-

puter to smoothly move objects on a table surface in two dimensions. This thesis describes the underlying technology of the Actuated Workbench and discusses the hardware and software design decisions involved in its construction. It then introduces a variety of techniques newly enabled by actuation and explains how they can improve upon tangible interfaces and lead to new applications.

Chapter 2

Background and Related Work

2.1 Interactive Surfaces

Interactive surfaces are a common variety of TUI in which physical objects are manipulated by users on an augmented planar surface. The presence, identity, and configuration of these objects are electronically tracked, and the computer interprets and processes this information to provide graphical feedback. The two primary varieties of interactive surfaces are (1) interactive workbenches, which use a horizontal work surface; and (2) interactive walls, which use a vertical work surface to which objects are affixed using magnets, sticky notes, or thumbtacks.

This thesis focuses on interactive workbench systems, of which there are many examples. One of the earliest such systems was the DigitalDesk [53], which supported augmented interaction with paper documents on a physical desktop. The paper documents were identified with overhead cameras, and information was projected around them (Figure 2.1).

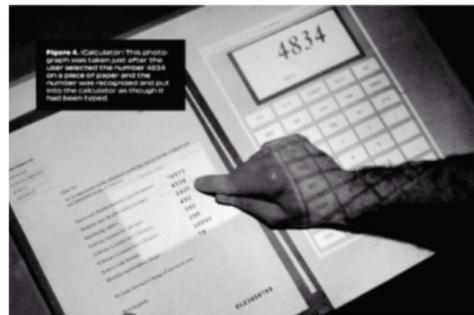


Figure 2.1: Digital Desk (calculator application)

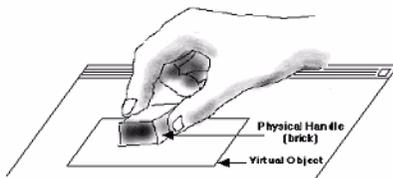


Figure 2.3: Conceptual sketch of Bricks system.

Bricks [17] was another early workbench system, in which users placed multiple bricks onto various screen-based virtual objects. The bricks were tracked with six degrees of freedom, and could be used to physically rotate, translate, scale, and deform the virtual entities to which they were attached (Figure 2.3). In the “GraspDraw” application

(Figure 2.2), bricks could also be bound to tools and attributes, making them operators rather than handles.

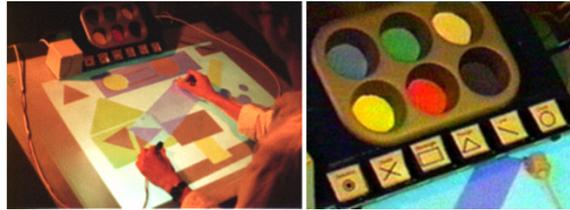


Figure 2.2: The GraspDraw application (left) used a tool palette and inkwell (right).

The metaDESK system [49], shown in Figure 2.4, supported interaction with a geographical space through the manipulation of physical tokens described as physical icons or “phicons.” This system was limited in that it used highly representational tokens that were permanently bound to their geographical “contents,” without mechanisms for rebinding.

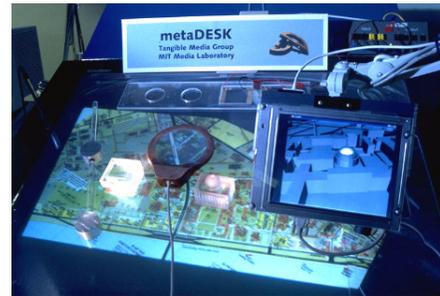


Figure 2.4: The metaDESK system used physical icons and lenses to navigate and explore geographical spaces.



Figure 2.5: The “Real Reality” system for assembly line planning.

Another workbench approach was the “Real Reality” system for assembly line planning in industrial contexts [44]. The system used a novel grasp-tracking approach to manipulate literal physical models of the assembly line in conjunction with physical representations of logical flows (Figure 2.5).

The BUILD-IT platform [19] used a special workbench layout that facilitated the binding of brick elements to data and func-

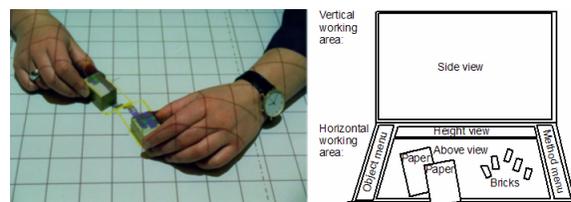


Figure 2.6: The BUILD-IT system for floor planning combined 2D and 3D views.

tional elements (Figure 2.6). It was used to create and test an application for floor planning that combined 2D and 3D views.

The Urp urban planning system [51] was one of the most fully developed interactive workbench systems. Combining physical building models with projected graphical simulations, it supported shadow studies, reflection studies, wind simulations, zoning metrics, and many other features useful for making urban planning decisions.

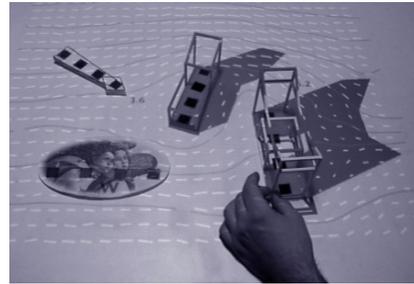


Figure 2.7: Urp, a tangible workbench for urban planning.



Figure 2.8: The Sensetable platform has been used for supply chain visualization (left) and musical performance (right).

Interactive workbench systems have also been applied to more abstract problem domains for which inherently geometrical representations do not exist. For example, the Sensetable system has been used for supply chain visualization [39] and musical performance [40].

2.2 Two-Dimensional Actuation Technologies

The computer-controlled configuration of objects on a flat surface has been studied in both the HCI domain and in the realm of industrial mechanics. Some early systems such as Seek [37] used robotic arms to arrange parts or objects on a table. Though an effective and dexterous method for computer control, the use of robotic arms would likely be distracting for interactive workbench systems. Moreover, it would be complicated and expensive to implement the multiple arms required to move multiple objects simultaneously. Recently, researchers in HCI and robotics have developed systems attempting to move objects with-

out the use of robotic arms. We examine some of these for their applicability to interactive workbench systems.

The PsyBench [7] prototype was built using parts from a computerized chess set that moved magnetic pieces using an electromagnet mounted on an x-y plotter under the table. This allowed the position of objects in the two workspaces to be synchronized. Though similar to the Actuated Workbench in its use of magnetism to grab objects, the PsyBench prototype had a variety of implementation limitations. It was only capable of inaccurate, teetering movements of the objects, and it was limited to straight-line motion. Furthermore, it was unable to control the orientation of the moving objects, and it could only move one object at a time.

Some recent robotics research targets actuation problems such as part feeding in factories, parcel sorting in distribution warehouses, and luggage sorting in airports. The Universal Planar Manipulator (UPM) [42] uses the horizontal vibration of a flat surface to move multiple objects at a time. Complex movements of specific objects on the surface are achieved using interference patterns of the vibration waves as they propagate across the surface. This system presents an effective way to manipulate many small parts without the need for motors or magnets, and its designers successfully used it in a closed-loop vision-tracking system. However, several aspects of the UPM's design detract from its usefulness in interactive workbench interfaces. First, in its present state, it is only capable of slow object translations and rotations; feed rates are on the order of millimeters per second. Second, the mechanism for vibrating the surface occupies space around the edges, preventing the easy tiling of multiple surfaces. Third, the system is noisy due to the mechanism needed to vibrate the flat surface and the sound of the vibrating objects. While not a problem in a factory assembly-line setting, this noise might be distracting for HCI.

Another system, the Modular Distributed Manipulator System (MDMS) [29] consists of an array of orthogonally oriented wheels that support and move objects through combined vector forces created by the rotating wheels. This actuation method presents a clever solution to the problem of friction: instead of dragging or sliding objects, they are rolled along the tops of the wheels. Like the Actuated Workbench, the MDMS is scalable to larger areas, requiring only that more actuators be set up next to the existing array. The MDMS differs from our work in that it is intended for manipulating large parcels, factory materials, or pieces of luggage in a conveyor belt type situation. Moreover, the surface upon which the objects rest is neither flat nor continuous (because it is made up of many small wheels), making it unsuitable for the projection often used in interactive workbench interfaces.

2.3 Haptic and Force Feedback Interfaces

Researchers in the field of haptics have devised many ways of employing the sense of touch in computer interfaces. Haptic feedback devices fall into two general categories: *force feedback* devices that interact with human muscles and tendons to give the sensation of a force being applied, and *tactile feedback* devices that interact with the nerve endings of the skin to indicate heat, pressure, or texture.

The most common application of haptic feedback devices is to provide touch stimuli to virtual reality environments, allowing computers to simulate the feel of a virtual object, but studies have also shown that augmenting any visual system with haptic feedback can prove useful. By adding an additional information channel to a visual interface, haptic feedback can increase the amount of information that is simultaneously processed by the brain. This extra information can reduce error, lower energy consumption, and accelerate task completion [45][46]. In addition, this extra information can compensate for ambiguities or errors in the visual information, making a visual display with haptic feedback per-

form more effectively than a stereoscopic display or a display with multiple viewpoints [8]. An experiment by Batter and Brooks suggests that haptic interfaces are effective educational aids; in their experiment, physics students who explored electrostatic fields using a touch interface developed a better understanding than students without access to haptic devices in their lab work [4].

A wide variety of haptic feedback technologies have been developed, some of which are commercially available. The most common devices are standard input devices such as joysticks, mice [21], or steering wheels, augmented with motors or vibrators to provide touch or force feedback along one or two dimensions. More complicated devices, such as the PHANTOM [33] and the HapticMaster [16], provide complex three-dimensional force control through robotic arm joints or magnetic levitation [6]. Other devices, like the CyberGrasp [52], are attached to the arm or hand as an robotic exoskeleton. The FEELEX system [22] can simulate texture and the contour of a landscape using an array of linear actuators.

2.4 Kinesthesia, Proprioception, Memory, and Perceptual Psychology

A large body of psychology work examines the ways in which humans use the spatial arrangement of objects to aid in problem solving. Kirsh showed that people frequently manipulate their environment to enhance memory or simplify their choices [26][27]. For example, people often sort a large group of objects into smaller groups to help them remember which objects share similar properties. A study by Zhang [54] demonstrated that the kind of objects used in a problem-solving task can dramatically effect how people think about a task and how long the task takes to solve. He compared the time required to solve a puzzle using two types of physical objects, and found that objects which afforded stacking allowed people to complete the task in half the time and with substantially fewer errors. Patten presented a study [38] in which participants using a tangible interface to

organize news articles performed better at location recall than participants using a graphical interface, often using the spatial relationship between physical objects and parts of the environment to help them remember the content of the physical objects.

There is also a variety of work on how people encode and use spatial information about their environment. A study by Malone [31] suggested that office workers with more sophisticated organizational schemes were better at locating their documents. However, formal experiments on this question suggest that it may be difficult to rely on spatial organization alone for recall. For example, Dumas and Jones found that retrieving documents by name was more effective than using spatial information for retrieval [13], and Lansdale argues that memory of location can be quite poor in cases where documents are not organized according to some logical structure. In cases where a structure is imposed, however, subjects can use it to help determine the location of documents, resulting in better recall [28].

Other studies have attempted to determine the extent to which spatial information is automatically encoded in the absence of a particular organizational scheme. Mandler et al. concluded that a great deal of object location information is encoded automatically, after a study showing only a small decrease in recall performance when subjects were not told to remember object location [32]. However, work by Naveh-Benjamin [36] suggested that this location information is encoded automatically only when subjects modify a spatial configuration of objects, and not when they simply observe such a configuration. A study by Tan et al. showed that using a touchscreen rather than a mouse results in better performance on a spatial recall task, confirming that kinesthetic cues are an important component of spatial memory encoding [48]. Despite disagreements in the literature as to the process of spatial memory encoding, it remains clear that spacial memory can be reliably used in practice to improve task performance. An evaluation of the Data Mountain system

by Robertson et al. demonstrated an effective application of spatial memory to a task involving the retrieval of web documents [43].

Chapter 3

Hardware and Software Design

3.1 Design Parameters

The Actuated Workbench’s design reflects several concerns of compatibility with current interactive workbench systems. First, the tagging and tracking technologies in these interfaces have begun to decrease in size, allowing the objects or “pucks” that hold them to be quite small. Zowie/LEGO demonstrated an example of such technology in a toy [20] which tracked objects with passive tags only 1.5cm in diameter and 2mm in height. While we considered designing motorized pucks that drive themselves around the tabletop on wheels, we felt these would tend to be relatively large compared to the tags. Motorized pucks would also require batteries that might need to be changed or recharged frequently due to the motors’ power requirements. Since many tagging technologies used today are passive devices, we sought to keep the actuation technology passive as well.



Figure 3.1: Our basic actuation platform contains a grid of 64 computer-controlled electromagnets.

A key interaction technique in most interactive workbench interfaces is the ability to manipulate multiple objects at the same time using both hands. Therefore, we wanted the computer actuation technology to be able to move multiple objects at the same time, preferably recreating users’ gestures with the objects. We also wanted the actuation system to be scalable to accom-

modate a variety of sensing areas. Finally, our ideal system would be silent, so as not to unintentionally distract the user when an object is moved on the surface.

3.2 Mechanical Design

Our basic actuation system consists of a 16.5cm fixed array of 64 electromagnets arranged in an 8 x 8 grid under a layer of 0.63cm acrylic (Figure 3.1). Though this provides only a limited area for actuation, we tile these arrays together to create larger actuation surfaces, the only limitations on scalability being the complexity



Figure 3.2: Overhead view of electromagnet array.

of electronically addressing the arrays, and the power requirements of running such a large number of electromagnets. We built the system using custom made electromagnets, each measuring 1.9cm in diameter and 3.8cm in length. They are wound with 32 gauge copper wire with a total length resistance of 120-122 ohms.

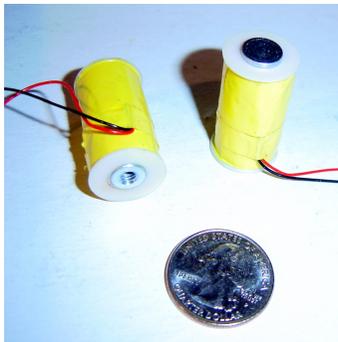


Figure 3.3: Custom-wound electromagnets produce broad, uncontained magnetic fields.

Using these custom-wound magnets proved an advantage over most commercially available electromagnets, which are often designed with metal housings intended to focus the magnetic field within a small area around the electromagnet. The uncontained fields of our electromagnets made it easier to create combinational flux patterns between individual electromagnets, the importance of which will be discussed later. Each electromagnet is driven with 27 DC volts and draws about 250mA. In our current applications, each electromagnet is only active for a few milliseconds at a time, and significant heating of the electromagnets does not occur. However, if many electromagnets were activated for a long period of time, cooling of the array might be necessary.

3.3 Circuit Design and Hardware-Software Interface

We designed custom electronics to drive each electromagnet in the array bidirectionally, making it possible to set the polarity of each magnet's field, as well as turn individual magnets on and off. Our electronics are designed to set the state of each electromagnet in the array at the same time. This makes moving multiple objects simultaneously a simple matter of setting up separate magnetic fields in dif-

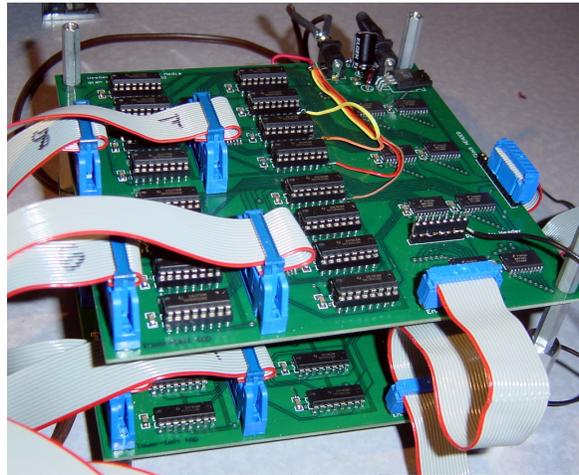


Figure 3.4: Custom-fabricated circuit board containing flip-flops and H-bridge transistor arrays.

ferent areas of the array. Of course we must take care that these magnetic fields do not overlap, and this consideration limits the number of objects that can be moved simultaneously.

An Ethernet-equipped microcontroller board, the Systronix SaJe board, natively runs a Java program that receives UDP packets sent via Ethernet from a control computer. It processes these packets and converts the data for output on two parallel 8-bit data buses. Every 15 microseconds, the microcontroller board clocks each magnet's polarity and enable status (off or on) into a set of octal flip-flops that connect to motor driver chips (containing the H-bridge transistor configuration frequently used for driving electric motors), which then connect to the electromagnets via ribbon cable.

The 15 microsecond refresh rate allows us to vary the strength of each electromagnet's field through pulse-width-modulation (PWM), a common technique for driving electric motors at different speeds by sending them pulses of various duty cycles. We can move objects between individual electromagnets by combining the magnetic fields of several

adjacent electromagnets, each set to a different strength through PWM, so that the object is attracted to a point somewhere in between the electromagnets.

3.4 Puck Design

Though all of the pucks that we use with the system contain permanent magnets, the system is capable of moving any lightweight ferromagnetic object, such as a paperclip or steel bolt. Our acrylic pucks are built to hold powerful 1.1 Tesla neodymium magnets, each 1.26cm x 1.26cm x 0.63cm, in order to provide the strong attractive forces needed to drag the 14g pucks around on the Active Workbench's acrylic surface.

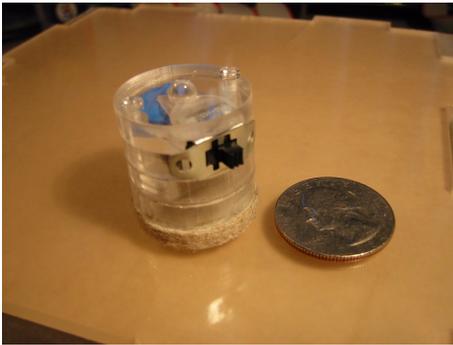


Figure 3.5: Our initial puck design included a permanent magnet and an infrared LED for vision tracking.

Our initial puck design had dimensions 2.54cm diameter x 2.54cm length. It included a battery, an IR LED for vision tracking, and a switch (to save the battery when not in use). Since the inclusion of a battery violated one of our design goals, we later switched to a passive radio frequency tag for object tracking. This allowed us to use slightly smaller and completely passive pucks

with an LC tag in place of an LED and battery. Our improved pucks measured 3cm diameter x 1.25cm height and held a permanent neodymium magnet and an LC radio frequency tag. Each puck also contained a small momentary pushbutton switch that shorted out the LC tag when pressed. This allowed users to perform mode toggling operations.

We attached felt pads to the bottom of each puck, providing the necessary kinetic friction to keep the object from sliding around uncontrollably on the table's surface; bare acrylic-on-acrylic is too slippery, resulting in oscillations as the puck slides past its goal and is then attracted back to it. The 0.63cm

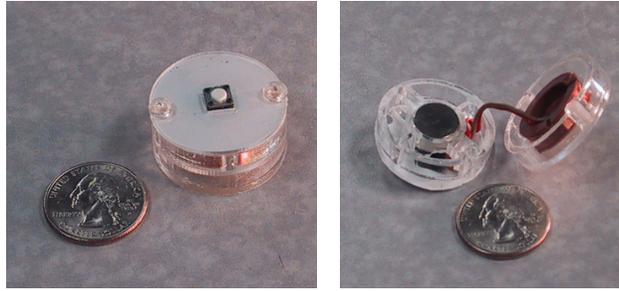


Figure 3.6: Improved puck design with momentary pushbutton switch (top); exploded view of puck showing permanent magnet and LC tag (bottom).

thickness of the felt pad, combined with the 0.63cm bottommost acrylic layer of the puck, results in the permanent magnet being about 1.26cm from the surface of the table, which is itself a piece of 0.63cm acrylic. This positions the permanent magnet about 1.89cm above the tops of the electromagnets. The height of the permanent magnet in the puck has significant effects on the performance of the system, since the neodymium magnet is strong enough to be attracted to the ferrous cores of the underlying electromagnets even when they are not activated. This attraction increases friction on the object, which affects the puck's ability to slide on the surface. We found the amount of friction between the pucks and the table to be a critical element in the system's ability to create smooth 2D motion. In general, we observed that static friction (the friction between the object and the surface when the object is at rest) inhibited smooth motion of the pucks, while kinetic friction facilitated smooth motion by controlling oscillations. After trying a variety of materials, we found that felt on acrylic gave adequate frictional characteristics, but other materials may yield better results in the future.

3.5 Object Tracking and Position Sensing

Electromagnetic radio frequency sensing technology is evolving rapidly to provide robust, low-latency object tracking on table surfaces [20][39]. Though this technology is used

often in interactive workbench systems, we encountered preliminary difficulties using electromagnetic sensing in conjunction with our magnetic actuation system because of distortions created by the strong magnetic fields of our electromagnets. We eventually overcame this problem through careful calibration of the tracking system, but to avoid these difficulties in the short term, we chose vision tracking for our first system prototype.



Figure 3.7: Overhead view of the Actuated Workbench from vision camera without IR filter (left) and with IR filter (right).

We embedded each puck with a small battery and an infrared LED, and suspended a camera directly above the Actuated Workbench. Adding an infrared filter to the camera blocked out ambient fluorescent light, making the video signal easy to process (Figure 3.7). We used an inexpensive Intel PC Camera Pro USB CCD camera and were able to achieve a tracking rate of 30 updates per second. This frame rate, though high from a human interaction standpoint, is somewhat slow from a control systems perspective.

Puck tracking was accomplished by detecting bright regions within the image. We used the image histogram to compute a threshold value on startup, and the threshold was used to divide the grayscale image into zeros and ones. We then employed standard blob-analysis techniques [25] to determine the longest horizontal segments. We could track multiple pucks simultaneously in real-time using an association method [3] to distinguish the pucks between frames. In every frame, we associated each observed location with the closest puck location in the previous frame. This association method is not wholly reli-

able, since puck paths that cross each other can interchange identities, but since the permanent magnets inside of the pucks tend to repel each other, the pucks rarely get close enough for this method to break down.

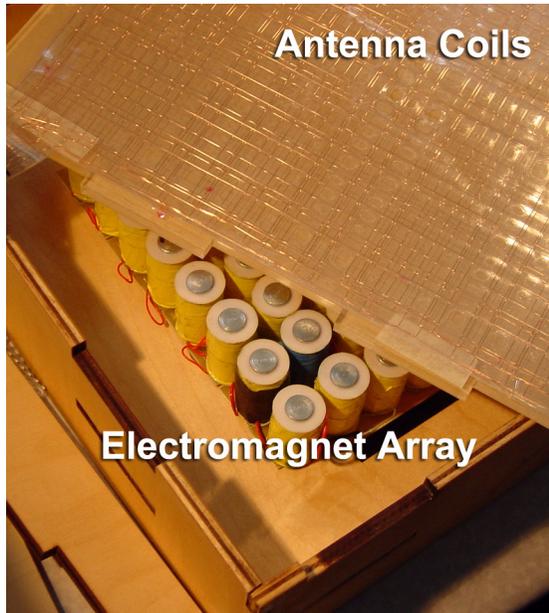


Figure 3.8: Tracking antenna coils and grid of electromagnets.

Our final system successfully employed electromagnetic tracking, which proved faster and more robust. The pucks contained passive radio frequency (RF) LC tags, each resonant on a unique frequency. We determined the position of each RF tag on the table surface using a modified version of the sensing apparatus found in the Zowie™ playsets [20]. We measured the amplitude of the tag resonances with several specially shaped antennas. The amplitude of the tag's resonance

with each antenna varies as a function of its position on top of the antenna array. This method gives very stable 2D position data accurate to within 2mm. Since each tag on the table resonates at a different frequency, their positions can be determined independently. This eliminates the need for the complex association algorithms required to track multiple objects in a vision-based system, and also provides substantially faster update rates than computer vision. Although the presence of dynamic magnetic fields on the table does create interference with our electromagnetic tracking system, our software calibrates for the presence of these magnetic fields, and they do not pose a problem when the system is in use.

3.6 Motion Control and Interpolation

Moving the puck across the table in a linear “Manhattan” fashion (in straight lines at right angles to each other) is a straightforward process. The puck can be moved to any grid cell on the table by consecutively activating the electromagnets in neighboring cells at full strength, as shown in Figure 3.9. Using Manhattan

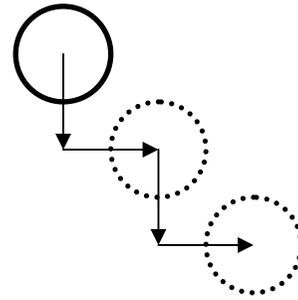


Figure 3.9: “Manhattan” motion

motion, objects can be moved across the table at rates on the order of 25cm/sec.

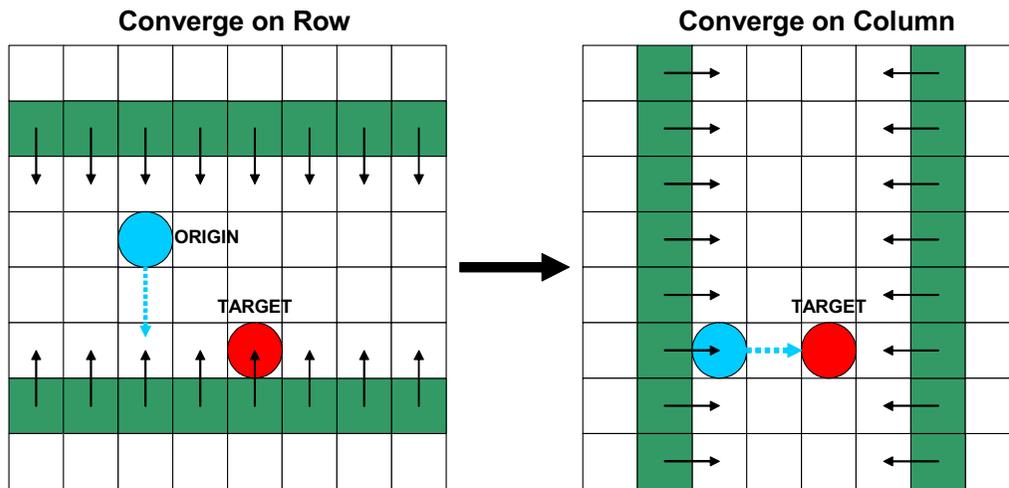


Figure 3.10: Sweeping to a desired point from an unknown origin.

If the board is operating in an “open loop” mode, in which we do not know the current position of the puck, we can still move it to any point on the table using a sweeping algorithm (Figure 3.10). To move the puck to the point (x,y) we begin by activating the outermost rows and then sweeping inward until the target row y is reached. Next, we begin with the outermost columns, and sweep inward in a similar fashion until we reach column x .

This method was useful for moving the puck to the far corners of the table to calibrate the tracking system.

Though Manhattan motion can move the pucks rapidly across the table, it is not very useful for recreating the smooth motions with which a user moves objects on an interactive workbench's surface. Since we can control the strength of individual electromagnets through PWM, we can perform a sort of physical anti-aliasing to create smooth travel paths along the table between the discrete positions of the electromagnets. In this section we describe our mathematical model of the Actuated Workbench and present the equations we used in our software to produce smooth motion along arbitrary paths. For a detailed derivation of these equations, refer to Appendix A.

Figure 3.11 is a vector diagram showing our force model. A single puck on the surface of the Actuated Workbench is subject to gravitational force, frictional force, the magnetic forces of attraction between the puck and the activated electromagnets, and the force of attraction between the permanent magnet in the puck and the iron cores of the electromagnets beneath.

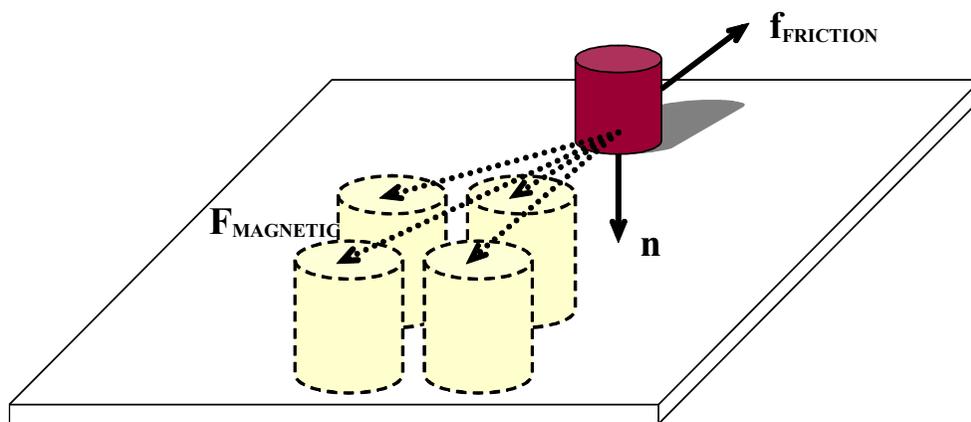


Figure 3.11: The electromagnets (lower left) exert forces on the puck (upper right).

We add these forces to arrive at an equation for the total force on the puck in terms of $f_{\text{MAG-NET}}$, the total force of magnetic attraction, and $f_{\text{FRICTION-NET}}$, the net friction:

$$f_{\text{TOTAL}} = f_{\text{MAG-NET}} \cdot \hat{x} + f_{\text{MAG-NET}} \cdot \hat{y} + f_{\text{FRICTION-NET}} \quad (3.1)$$

$$f_{\text{MAG-NET}} = \sum_i \frac{\alpha_i f_E}{(x - x_i)^2 + (y - y_i)^2 + z_S^2} \quad (3.2)$$

$$\|f_{\text{FRICTION-NET}}\| = \mu \left(mg + \frac{\|f_p\|}{z_S^2} + \|f_{\text{MAG-NET}} \cdot \hat{z}\| \right) \quad (3.3)$$

Here the puck is positioned at (x, y) and each electromagnet i is positioned at (x_i, y_i) with duty cycle α_i . f_E and f_p are constant-magnitude forces of attraction proportional to the strengths of the electromagnet and the permanent magnet in the puck, z_S is the vertical separation between the puck and the electromagnets, m is the mass of the puck, g is the acceleration due to gravity, μ is a dimensionless coefficient of friction, and \hat{x} , \hat{y} , and \hat{z} are the standard unit vectors.

In reality, the magnetic fields of the activated electromagnets interact in a somewhat more complex manner (Figure 3.12). Nonetheless, the force-summing model just described, in which electromagnets are treated independently of one another, is a reasonable method of approximating the more complicated underlying physics, since the summation of multiple forces due to individual magnets parallels the summation of multiple magnetic fields to produce a single force.

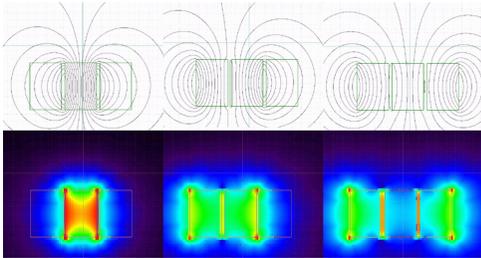


Figure 3.12: Magnetic field interactions between electromagnets. The top images show magnetic flux lines and the bottom images map flux density to brightness. The three image pairs show the fields resulting from a single center magnet turned on (left), the left and center magnets turned on (center), and all three magnets turned on (right). The effect of this field-shifting behavior can be modeled approximately using force summation. These images were generated with the VisiMag software package [5].

To produce a puck displacement Δx during a loop interval Δt , we activate the electromagnets with duty cycles α_i such that

$$f_{\text{TOTAL}} = 2m \frac{\Delta x - v_0 \Delta t}{\Delta t^2} \quad (3.4)$$

This equation assumes we are keeping track of the puck's instantaneous velocity v_0 . If we are using the Actuated Workbench in an “open-loop” mode in which we do not track the instantaneous position or velocity of the puck, we can still compute a reasonable estimate of f_{TOTAL} using a dead reckoning approach based on assumptions about how the previous electromagnet settings have affected the position and velocity of our puck according to our force model.

There are many ways in which we could activate the electromagnets so that the resulting forces summed to the desired value of f_{TOTAL} . In the next section, we describe several different methods for choosing the magnet values.

3.7 Anti-Aliasing Techniques

In computer graphics, the mathematical model of an image is a continuous analog signal that is sampled at discrete points called pixels. Aliasing occurs when the sampling frequency is too low for the signal frequency, resulting in a coarse image in which smooth curves are converted to steps and jagged outcrops. The anti-aliasing technique of prefiltering combats this problem by treating each pixel as an area, and computing pixel color based on the overlap of the scene's objects with a pixel's area.

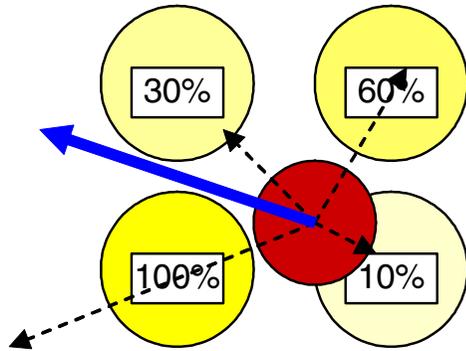


Figure 3.13: Four electromagnets with different duty cycles combine to produce a force with a new direction and magnitude.

With the Actuated Workbench, we are faced with a similar problem: we wish to render an analog signal (in this case, a force of a particular direction and magnitude) using a discrete array of cells (variable-duty electromagnets). To do so, we can employ a similar technique: the strength of each electromagnet is determined by the “overlap” of its magnetic flux lines with the location of the point

force. Figure 3.13 shows a configuration in which the forces of four neighboring electromagnets of different duty cycles combine to create a single force of a new magnitude and direction.

The simplest algorithm for anti-aliasing draws the computer graphics equivalent of a smoothed dot centered at the location of desired travel. Given a desired force vector with head at point (x,y) , we compute the distance from each electromagnet to (x,y) , and set its duty cycle in inverse proportion to this distance. As in computer graphics, we can choose any number of falloff metrics. We experimented with Gaussian falloff, but found that in practice it was no better than a simple linear falloff metric.

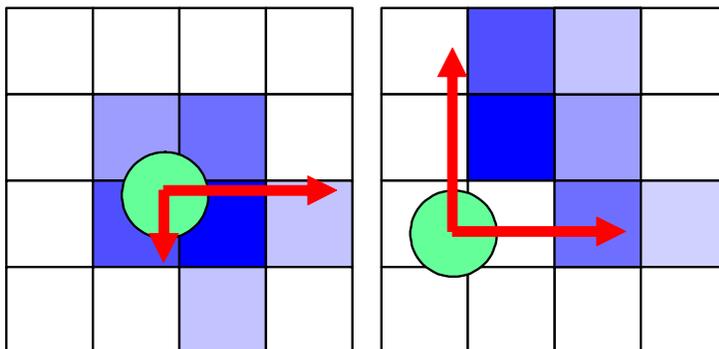


Figure 3.14: Anti-aliasing methods. The squares represent magnet cells, and intensity of each shaded square corresponds to the duty cycle of the magnet. The “dot” technique (left) results in slower travel than the “jet” technique (right).

A drawback of the dot-based method is that it limits the puck’s top speed of travel to about 15cm/sec. In order to produce enough force to move the puck, the center of the dot

must be positioned close to the puck, and the forces produced by some of the activated electromagnets will pull the puck backwards against the desired direction of travel (Figure 3.14).

If we know the position of the puck and the direction of travel that we hope to produce, we can pull the puck using only the electromagnets located in this direction relative to the puck. To do so, we first compute the vector from each electromagnet to the target (x,y) , and then compute the scalar projection of this vector onto the direction-of-travel vector. Taking the set of vectors of positive magnitude produces a collection of forces resembling a “jet” in

fluid mechanics (Figure 3.16). Jet-based movement can move pucks across the table almost as fast as Manhattan motion.

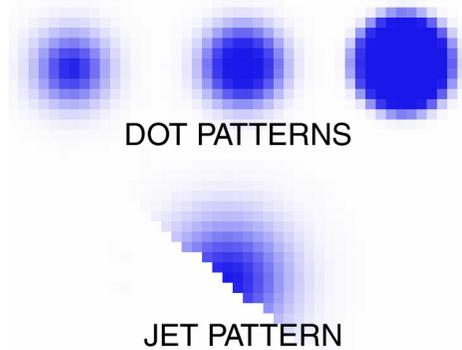


Figure 3.15: “Dot” and “Jet” equivalents in computer graphics. The three dots use different falloff metrics.

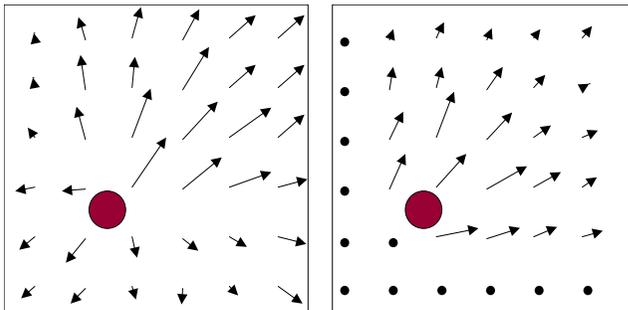


Figure 3.16: “Dot” force field (left) versus “Jet” force field (right).

Chapter 4

Applications

4.1 Types of Applications

Having developed a system meeting our design criteria for an interactive workbench actuation system, we began to imagine the new interaction techniques and applications that our system could support. This chapter begins by describing extensions of basic GUI functions into the physical domain, and then goes on to describe higher level applications, including some solutions to classic problems in interactive workbench interfaces. We built prototypes of many of these applications; other application ideas would require further development of the Actuated Workbench to address its limitations in speed, magnetic strength, scale, and resolution.

4.2 Extending GUI Functions to the Physical Domain

- **Search and retrieve.** As the number of pucks increases in an interactive workbench system, it becomes more difficult for a user to keep track of every item on the table, just as it is difficult to keep track of many graphical icons on a computer desktop. A search and retrieve function could respond to a user query by finding matching items and either moving them to another place on the tabletop or wiggling them to get the user's attention. Note that the Actuated Workbench would assist only in displaying the results of search queries; the input of the query expression would require a separate interface, such as Ullmer's token-based tangible query interface [50].

- **Sort.** A more powerful function would be one in which the computer could physically sort and arrange pucks on the table according to user-specified parameters. This could help the user organize a large number of data items before manually interacting with them.

- **History and Undo.** As a user makes changes to data through physical input, she may wish to undo some changes. A physical undo in this system could move the pucks back to their positions before the last change. It could also show the user the exact sequence of movements she had performed. In this sense, both “undo” and “rewind” commands are possible.

- **Teaching and Guiding.** Because the Actuated Workbench gives the computer the ability to recreate users’ gestures with the pucks, it becomes possible for the computer to teach the user something about interacting with the system through physical gestures. If specific gestures are used in the interface to trigger certain commands (such as a shaking gesture to unbind a puck from a data item), the computer can show a novice or a forgetful user how to make that gesture with the puck. This way, many of an application designer’s commands can be taught to users without the need for intensive human coaching. In addition, if a user is uncertain how to proceed while using a problem-solving or simulation system, the computer could suggest a physical configuration of the pucks.

4.3 Navigation

In an overhead map browsing interface such as metaDESK [49], it is often useful to rotate, translate, or scale the map. However, if there are physical objects on the map surface that correspond to map landmarks, their positions on the table will no longer be correct once the underlying map is moved. We built a map browsing interface that allows users to navigate around a map by moving physical objects and automatically corrects the positions of physical objects on the map surface using actuation.

Pushing the button on a puck locks down the position of its corresponding map landmark. The user can then move the puck to scroll the overhead map. Users can rotate or zoom the map using a two-handed technique; first a landmark is locked down, and then

another landmark is moved to specify a scaling and rotation transformation. Since these transformation operations change the position of each landmark on the table, actuation is necessary to preserve consistency between the physical and digital states of the system. As the user scrolls, rotates, or zooms the graphical display, the pucks automatically move on the surface of the table to keep up with the viewing transformation.

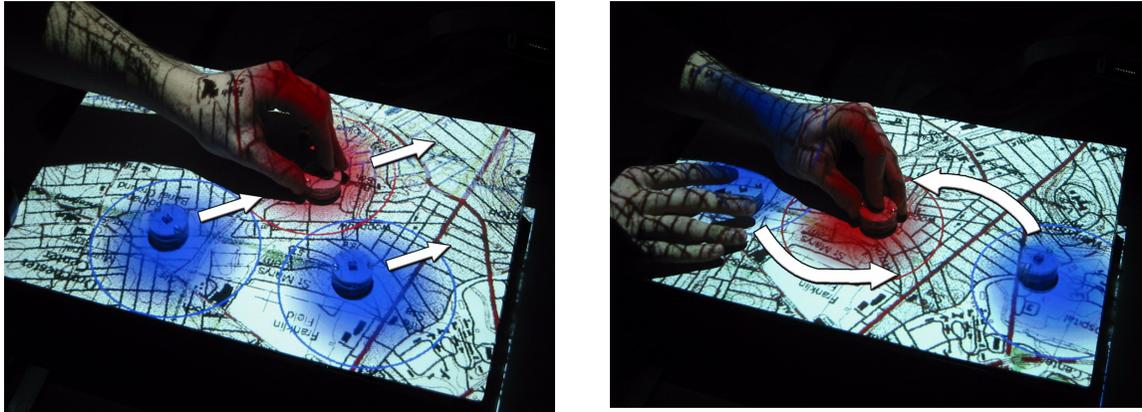


Figure 4.1: Translate and rotate/zoom operations. The puck that the user is holding has been locked to a particular map location.

4.4 Remote Synchronization

One advantage that interactive workbench interfaces offer is the ease with which multiple users can make simultaneous changes to the system. Users can observe each other's changes, and any user can reach out and physically change the shared layout without having to grab a mouse or other pointing device. This is not the case, however, when users are collaborating remotely. In this scenario, a mechanism for physical actuation of the pucks becomes valuable for synchronizing multiple physically separated workbench stations. Without such a mechanism, real-time physical synchronization of the two tables would not be possible, and inconsistencies could arise between the graphical projection and the physical state of the pucks on the table.

One example of a system that could benefit from physical synchronization is Urp [51]. In the Urp system, users manipulate physical models of buildings on a table and the com-

puter displays simulation information in the form of projected “digital shadows” around the buildings. “Distributed Urp” (Durp) later attempted to create distributed workspaces between multiple remote users. Identical Urp systems were set up in two separate locations, and the two systems were synchronized through identical graphical projections onto the workbench. However, if a user in one location moved a building, only the “digital shadow” of the building, and not the physical model, would move in the remote location. In addition to facilitating the simple synchronization of these models, the Actuated workbench could recreate remote users’ actual gestures with objects on the table, adding greatly to the “ghostly presence” [7] sought in remote collaboration interfaces.

We built a remote collaboration demonstration in which two remote tangible workbenches are kept synchronized. When the motion of a puck is sensed on one table, the remote table uses magnetic actuation to update the position of the corresponding remote puck, keeping the states of the two tables the same. If both users simultaneously attempt to move the same puck, each user will feel a force pulling against the puck, indicating the direction in which the remote user is attempting the move the object. In this situation, the puck becomes a conduit for interpersonal haptic communication. Actuation can therefore serve two purposes during remote collaboration: (1) synchronizing a shared workspace; and (2) providing a haptic communication link.

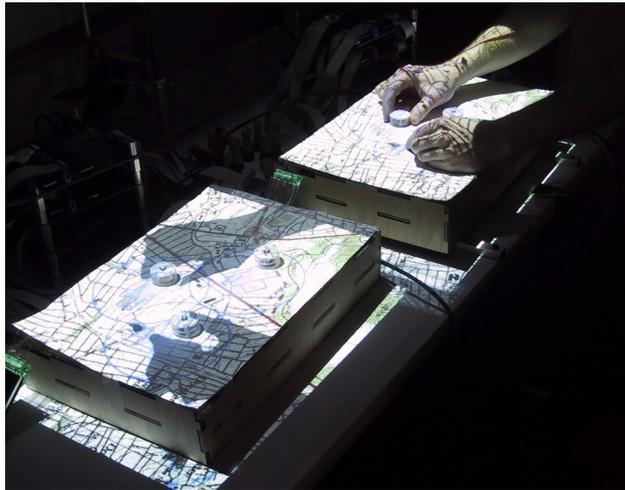


Figure 4.2: Two actuated tangible workbenches running a distributed application for cellphone tower placement. The tables maintain a synchronized state over an Internet connection. Translucent hand silhouettes show remote user activity.

4.5 Scientific Visualization

The Actuated Workbench could be helpful in the scientific visualization of complex mechanical systems. For example, a solar system model in the manner of an orrery could be created on an interactive interface with full actuation of the planetary orbits. The user could change the physical properties of the planets or teach the computer new orbital paths, and then watch the resulting motions of the planets.

Similarly, the Actuated Workbench could be used to teach students about physics by demonstrating the attraction and repulsion of charged particles represented by pucks on the table. As a student moved the pucks around on the table, the system could make them rush together or fly apart to illustrate forces between the objects.

4.6 Entertainment

In addition to these more practical applications, the Actuated Workbench could be used to add a physical dimension to computer entertainment. Though motorized chess sets have existed for many years, they operate using a single electromagnet mounted on an x-y plotter mechanism, limiting them to moving one object at a time. The Actuated Workbench could provide a significant improvement to these devices, making them more flexible for a variety of games. Classic computer games like Pong [2] could now be played using a physical puck and two physical paddles manipulated by the users. Distributed Pong could be played with a local user moving one paddle and the computer moving a remote user's paddle on the table. In addition, the Actuated Workbench can be used to flip over thin, polarized magnetic pucks by rapidly reversing the polarity of the electromagnets. This could be used to play a physical game of Reversi with the computer. Finally, one could create painting or drawing programs in which a pen or brush was attached to the puck. Various plotter-based computer-controlled drawing systems have been developed [11], but the Actuated Workbench could expand on these systems by using multiple pens

or sharing control between the computer and the user. The computer's movement of the puck could be used to teach certain artistic gestures or handwriting movements.

4.7 Specifying and Maintaining Constraints

Tabletop tangible interfaces are well suited to spatial layout of items in an application containing layout constraints. Examples of such constraints can be found in urban planning [51], circuit routing, and task scheduling [24]. Some of these constraints can be easily expressed as mathematical formulae, e.g. "two buildings may never be less than 20 meters apart." Making the computer responsible for monitoring these constraints frees the user to focus on the more salient aspects of the task at hand, instead of constantly having to check whether a proposed design violates any constraints.

We built a constraint solving system that supports both programmatic and physical constraints. Programmatic constraints are more appropriate for constraints that can be easily expressed mathematically, and do not change often in an application. Physical constraints are more appropriate for situations where a user would like to experiment with a constraint briefly, modify a constraint over time, or add new constraints as their design evolves. A physical constraint can simply be a user's hand on the table, holding a puck in place or preventing it from moving to a certain part of the table. A physical constraint may also be a physical "jig" placed around one or more pucks. We have used several types of jigs to represent common constraints:

- A “collar” can be placed around a puck to constrain its proximity to other pucks. This collar is simply a circular disk of plastic that bumps into other pucks to keep them a minimum distance away. The collar can be flipped over to raise the height of the plastic disc above the table. In this case, other pucks with raised collars will be kept farther away than those with lowered collars or no collars. In this way, minimum distance constraints can be applied selectively to different pucks.

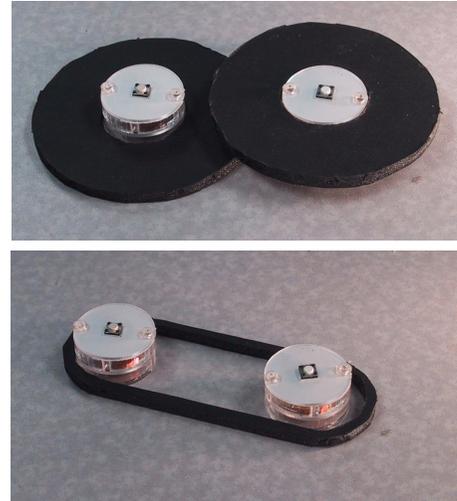


Figure 4.3: Collar and ring constraint jigs.

- A “ring” can ensure that pucks remain within a certain distance of each other. This constraint is an oval shaped piece of plastic that surrounds two or more pucks.
- Finally, a nonmagnetic object can be placed on the table and used to ensure that certain pucks stay inside or outside of a particular region on the board.

When developing software to support constraints with our actuation mechanism, we began by incorporating the GUI constraint solver Cassowary [1]. However, it was difficult to implement some parts of our system using Cassowary for two reasons. The first was Cassowary’s method of dealing with conflicting constraints: Cassowary will fully satisfy a higher priority constraint at the expense of a lower priority one. While this approach makes sense for problems like window placement in a GUI, it is not appropriate for many types of constraints on a TUI platform. In our actuated system, the constraint solver is run each time the system receives new data about the position of the pucks. With the Cassowary-style solution, slight differences in puck positions can lead to large discontinuities in the solution to the constraints. These discontinuities can cause pucks on the table to seem unresponsive to changes the user is making, and then suddenly fly across the table

unexpectedly. A better approach for TUI in these circumstances is to partially solve each of the conflicting constraints, thus providing less surprising motions on the table.

Another reason that traditional constraint solving toolkits like Cassowary are not well-suited to this task is that often these toolkits are not designed to handle constraints involving physical objects. In a constraint oriented GUI problem such as window placement, the constraint engine can safely assume that the window manager will be able to place a window at any position deemed optimal by the solver. However, when dealing with a physical system, there are issues of inertia, friction, and inter-object collision that must be considered when resolving constraints. In addition, there may be jigs or hands blocking the travel of pucks to the positions considered optimal by the solver.

We have developed an iterative solver for dealing with constraints involving physical objects. The system takes as input the current position of all pucks and the constraints to be resolved. For each constraint, it checks if the constraint is already resolved, and if not, it computes a motion vector for each puck that would resolve the constraint. The motion vectors for each puck for all constraints are fed into a weighted sum, based on the priority of each constraint. The resultant vector is scaled, based on how quickly the system should converge on a solution. This parameter controls the “viscosity” of the system, or how quickly the computer-controlled pucks respond to the user’s actions. The actuation mechanism then attempts to move each puck according to its corresponding motion vector. This attempt may or may not be successful depending on whether the puck is physically constrained. The system then reads the puck positions again and repeats the resolution process.

The process the actuation mechanism uses to resolve constraints has several interesting properties. First is that the physical properties of objects on the table become parameters in the constraint resolution process. The system will resolve constraints differently depending on whether a user is holding a puck, or a puck is otherwise physically con-

strained. The friction and mass of the pucks will also affect the results of the computation. For example, a puck with greater mass or friction will tend to stay close to its original position as the system moves pucks around it in an attempt to resolve constraints. This yields a tight coupling between the physical properties of objects on the table and the computational behavior of things they represent.

This constraint resolution approach also affords the use of jigs to represent and enforce constraints. One interesting property of these jigs is that they can prevent the user from establishing fundamentally inconsistent constraints, as shown in Figure 4.4; here the user is trying



Figure 4.4: An inconsistent set of constraints.

to constrain the system such that two pucks must be far apart and close together at the same time.

One issue with jigs is that they can sometimes interfere with each other in unusual ways. For example, the ring constraint between two pucks may inadvertently push a third puck out of the way. Minimum distance jigs affect all types of objects, making it difficult to use them for constraints that should only affect certain pucks. In practice, however, unexpected movements of objects make it clear that the system is interpreting a jig in an unintended way. The user can then temporarily remove the jig and constrain the pucks by hand.

4.8 Computer Movement vs. User Movement

In a remote collaboration scenario, confusion can arise as to whether an object was moved by the computer because of a constraint in the system, or by the remote user. To reduce this confusion, we displayed overlaid translucent silhouettes of the remote user's hands (Figure 4.2), captured from a camera mounted above the remote table. The silhouettes are

extracted from the captured image using background subtraction followed by thresholding and dilation operations. The projected silhouettes reinforce the ghostly presence of the remote user and indicate the focus of his attention.

4.9 Remote Collaboration using Jigs

One design trade-off for this system is whether or not to electromagnetically track the jigs. Our constraint system could potentially use knowledge about the positions of jigs to improve synchronization between the tables. In addition, in remote collaboration scenarios, all users should know what jigs are being used on the table. However, we believe an important property of this system is the ability to use any object to constrain a puck, not just a pre-made, tagged jig. To preserve this property we decided not to track the jigs. We use the same silhouetting technique as for displaying hands to keep users aware of all physical constraints in distributed collaboration scenarios. A physical constraint appears as a shadow on remote workbenches, as shown in Figure 4.5.

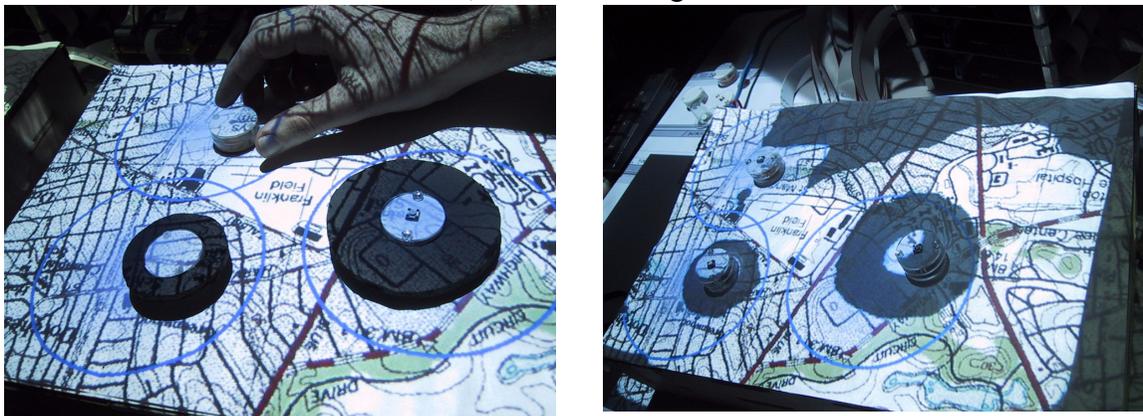


Figure 4.5: Hands and jigs and their projected silhouettes.

4.10 Resolving Conflicts in Remote Collaboration

When the system is being used in distributed collaboration mode, it uses a set of constraints to keep the tables consistent with each other. If two pucks are bound to the same object on different tables, the system constrains each puck to be in the same place relative to the board.

This approach assumes an actuation system with rapid response, since it does not distinguish between pucks that differ in position because they are being manipulated by both users and pucks that differ in position simply because one side is experiencing mechanical latency. If an object is quickly moved and then released, it will snap partially back, as it moves toward the average of its tracked positions on the local and remote tables. In the future, we could address this issue by adding capacitive touch sensors to our pucks to distinguish between pucks being manipulated by the users and pucks being manipulated by the actuation system. Unfortunately, this would require adding batteries to the pucks. Another possibility would be to introduce vision-based hand tracking, using the same camera that we currently use to extract the hand silhouettes.

Chapter 5

Conclusion

5.1 Summary

Tangible interfaces can benefit from the addition of computer-controlled actuation in a variety of ways. In addition to maintaining consistency between the physical and computational states of the system, actuation also enables new types of interactions, such as remote collaboration mediated through physical objects. This chapter describes the various benefits of computer-controlled actuation in tabletop tangible interfaces, and discusses some design parameters for the integration of actuation into these interfaces. It then goes on to suggest future research directions for actuation in tangible interfaces.

5.2 Maintaining Constraints

By maintaining constraints in tangible interfaces, actuation allows the computer to correct inconsistencies between digital and physical states. If the user attempts to adjust a parameter outside of a valid range, the computer can impose a physical limit on the value of the parameter. If an object on the table represents a dependent variable (i.e., its state depends on the state of another object), the computer can automatically update its state when the independent variable changes. Actuation also addresses the “nulling problem” [9]: the system can automatically attach physical pucks to digital parameters and adjust them to the appropriate value. In addition to making interaction more efficient, these techniques can make complex mathematical systems programmed in software behave more like transparent mechanical systems.

The usefulness of tangible physical constraints depends largely on the nature of the interaction task. If the task is highly structured and involves rigorous computation, constraints can quickly guide the user toward an optimal solution. However, the task may be

so highly structured (e.g. graph coloring) that human intervention is unnecessary, since the computer could probably determine the best solution on its own. If the task is free-form or artistic (calligraphy, sculpture), the importance of the human is emphasized, but the computer may have little to provide in the way of guidance. The types of tasks that make the best use of physical constraints are those that combine rigid mathematical constraints with aesthetic considerations of which the computer is unaware (as in architecture and urban planning) or with a negotiation component (as in business strategy planning) in which multiple people, each with a different objective in mind, must reach a compromise.

5.3 Haptic Feedback

In a system like the Actuated Workbench, information can be conveyed not only in the motion of pucks on the table, but also in the forces a user feels while holding a puck on the table surface. Users can explore electromagnetic or gravitational force fields, and feel forces that would otherwise be on a scale too large or too small to experience physically. By simulating notched sliders or knob endpoints, haptic feedback can also help users to adjust parameters by touch alone, allowing them to concentrate elsewhere in the interface [30][47].

Haptic feedback is particularly useful in situations in which visual attention is divided between the objects being manipulated and other information being displayed. In this situation, the additional physical I/O channel increases the amount of information that the user can simultaneously process. Even when the input and output spaces are coincident (a general design principle for tabletop tangible interfaces), the user's attention may be divided when multiple outputs change as the result of a single change in input.

5.4 Spatial Thinking

Tangible interfaces provide stronger spatial and kinesthetic memory cues than graphical

user interfaces, taking advantage of our natural tendency to remember information spatially and to use spatial organization to simplify problem solving. However, without the ability to restore saved physical configurations or to impose organizational constraints, the extent to which these spatial cues could be exploited in traditional tangible interfaces was limited. Adding actuation to a tangible interface can improve its capacity to leverage spatial thinking.

When a user creates a spatial configuration and ascribes a meaning to that configuration based on his own organizational criteria, the system can later load and restore the configuration he has previously established. In addition, the system has the power to impose an organizational scheme on the user, automatically sorting objects or snapping them to particular locations as the user arranges them. Studies have shown that encouraging the use of organizational strategies can lead to improved recall [38].

5.5 Remote Collaboration

Traditional tabletop tangible interfaces readily facilitate collocated collaboration, but are less effective in supporting remote collaboration, since they provide no means of maintaining a consistent physical state between remote systems. Actuation technology introduces this capability, eliminating the overhead of manual synchronization necessary in many tangible interfaces for remote collaboration [15]. In addition, the motion of physical objects rather than graphical ones may strengthen the sense of presence of the remote user, though this effect is much more difficult to evaluate quantitatively. In remote collaboration situations, actuation can be coupled with graphical feedback to help users understand which objects are being moved by the computer, and which are being moved by remote collaborators. Finally, when remote users hold the same object, each can feel the force exerted by the other, and the object becomes a conduit for haptic communication, much in the manner of ComTouch [10]. The actuation then serves a dual purpose, both synchroniz-

ing physical state and mediating interpersonal communication.

5.6 Future Vision

Although the system we constructed proved to be an effective platform for prototyping and demonstrating many applications of actuation in tangible user interfaces, not all of our ideas could be realized due to its shortcomings in speed, scale, and resolution. With some additional mechanical and electrical engineering, many of these limitations could be overcome, giving rise to a variety of new interaction possibilities. These possibilities are discussed in Appendix B.

In addition to these mechanical and electrical improvements, simplifying the software API would greatly improve the overall packaging. With some software redesign, the Actuated Workbench could provide a single, stable platform for integrated object tracking, graphical display, and actuation. This unified software architecture would allow for rapid prototyping of actuated tangible interfaces. A discussion of these considerations can be found in Appendix C.

Though this thesis focused on two-dimensional actuation on tabletop surfaces, many other kinds of actuation are possible in tangible interfaces, and many of the same concepts would apply to these alternative actuation systems. We discuss some future possibilities for alternative actuated interfaces in Appendix D.

Appendix A

Mathematical Derivations

A.1 Motion Control Equations

The magnitude of the frictional force on the moving puck is given by the equation

$$\|f_{\text{FRICTION}}\| = \mu_k \|n\| \quad (\text{A.1})$$

where μ_k is the dimensionless coefficient of kinetic friction and n is the normal force on the puck. The value of μ_k can range from 0.05 to 1.5, depending on the choice of materials for the puck bottom and the table surface. When the puck is stationary, we replace μ_k with the coefficient of static friction μ_s . In general, $\mu_s > \mu_k$, but we simplified matters by trying to choose materials for which μ_s and μ_k were nearly identical.

When none of the electromagnets are activated, the normal force on the puck is the sum of the gravitational force on the puck and the attractive force between the permanent magnet in the puck and the iron cores of the electromagnets below. Since the iron cores are spaced at intervals, this attractive force varies with the position of the puck on the table, but our simplified model assumes that this variation is slight enough to be negligible. Substituting these values, the equation for friction becomes

$$\|f_{\text{FRICTION}}\| = \mu \left(mg + \frac{\|f_p\|}{z_s^2} \right) \quad (\text{A.2})$$

where m is the mass of the puck, g is the acceleration due to gravity, f_p is a constant force of attraction proportional to the strength of the permanent magnet in the puck, and z_s is the vertical separation between the puck and the electromagnets. The z_s term is squared because the magnetic force between two objects attenuates in proportion to the square of the distance between the objects. In practice, we chose a value for z_s large enough to make the contribution of negligible, but small enough that the attractive forces of the activated

electromagnets are able to move the puck. With our current materials, we empirically found a z_S of 1.89cm gave good results.

We modeled the magnetic force between the puck and an activated electromagnet using the equation

$$f_{\text{MAG}} = \frac{\alpha f_E}{x_S^2 + y_S^2 + z_S^2} \quad (\text{A.3})$$

where α is the duty cycle of the pulse-width-modulated electromagnet (0-100%), f_E is a constant-magnitude force of attraction proportional to the strengths of the electromagnet and the permanent magnet in the puck, and x_S and y_S are the separation distances between the puck and the activated electromagnet along the horizontal axes. The direction of f_E is from the center of the puck to the center of the upper end of the electromagnet. Note that the z component of f_{MAG} will contribute to the normal force, increasing the magnitude of f_{FRICTION} . This can actually be desirable: as the puck approaches its target, the z -component of f_{MAG} increases, increasing the friction and preventing the puck from overshooting its goal.

We can sum the contributions of each activated electromagnet to compute the net force on the puck due to the electromagnets:

$$f_{\text{MAG-NET}} = \sum_i \frac{\alpha_i f_E}{(x - x_i)^2 + (y - y_i)^2 + z_S^2} \quad (\text{A.4})$$

In this equation, the puck is positioned at (x, y) and each electromagnet i is positioned at (x_i, y_i) with duty cycle α_i .

Adding the z component of $f_{\text{MAG-NET}}$ to the normal force in our friction equation, we reach the final equation for net friction:

$$\|f_{\text{FRICTION-NET}}\| = \mu \left(mg + \frac{\|f_P\|}{z_S^2} + \|f_{\text{MAG-NET}} \cdot \hat{z}\| \right) \quad (\text{A.5})$$

where \hat{z} is the unit vector in the direction of the positive z -axis. The direction of the friction vector $f_{\text{FRICTION-NET}}$ is opposite the direction of the applied force, in this case the horizontal components of $f_{\text{MAG-NET}}$. We are now ready to write an equation for the total horizontal force on the puck:

$$f_{\text{TOTAL}} = f_{\text{MAG-NET}} \cdot \hat{x} + f_{\text{MAG-NET}} \cdot \hat{y} + f_{\text{FRICTION-NET}} \quad (\text{A.6})$$

where \hat{x} and \hat{y} are the positive unit vectors along the horizontal axes. The acceleration of the puck is proportional to this total force:

$$a_{\text{PUCK}} = f_{\text{TOTAL}}/m \quad (\text{A.7})$$

The resulting velocity of the puck is given by the standard differential equation

$$\frac{\partial x}{\partial t} = a_{\text{PUCK}}t + v_0 \quad (\text{A.8})$$

which can be reduced to the position equation

$$x(t) = \frac{1}{2}a_{\text{PUCK}}t^2 + v_0t + x_0 \quad (\text{A.9})$$

where v_0 and x_0 are the instantaneous velocity and position of the puck, respectively. This means that if we keep track of the puck's velocity and position, we can produce any desired displacement Δx of the puck during loop interval Δt by solving this equation for a_{PUCK} and in turn for f_{TOTAL} :

$$f_{\text{TOTAL}} = 2m \frac{\Delta x - v_0 \Delta t}{\Delta t^2} \quad (\text{A.10})$$

We then find a combination of electromagnet settings α_i that produce this net force f_{TOTAL} , as described in Section 3.7.

Appendix B

Engineering Improvements

B.1 Tiling of Actuation Surfaces and Scale

Though each 64-magnet actuation grid is only 16.5cm square, we have successfully tiled two of them together to form a single table with twice the actuation area. Tiling four or even six grids together would produce an even larger surface, increasing the number of pucks that could simultaneously fit on the table and broadening the potential types of interaction. In addition, it would be interesting to explore the use of different sizes of electromagnets. Smaller electromagnets may yield higher resolution of object movement on the table, while larger or more powerful electromagnets may provide more force for moving objects, making it possible to provide stronger force feedback.

B.2 Puck Modifications

Though we focused primarily on translational movements of the pucks, computer-controlled rotation could be useful as well, since there are many applications in which the orientation of objects is significant. Puck rotation could be accomplished by designing new, slightly larger pucks in which permanent magnets are placed in the puck with opposite polarities facing downward. The two sides of the puck could then be pulled using opposite magnet polarities to control the puck's rotation. Alternately, one could achieve a similar effect by designing pucks with a single magnet oriented sideways, with the north pole on one side of the puck and the south pole on the other. This would make it easy to spin the puck about its center, but would make smooth translational motion more difficult.

B.3 New Types of Motion

In addition to controlling orientation, the Actuated Workbench is also capable of flipping

over magnetic objects or launching them into the air by reversing the polarity of the electromagnet underneath the object. If the polarities of the electromagnet and the permanent magnet are the same, a strong repulsion results. This repulsion could be used to flip over a double-sided object, so that the opposite side was attracted downward. A brief negative polarity burst could also provide a vibrational haptic cue, indicating to the user that he has crossed a boundary or threshold [35].

Since the strength of the magnetic field can be quickly controlled in any part of the table, the Actuated Workbench is theoretically capable of levitating magnetic objects above the table. The Hover system [34] levitated objects on a column of air to attract attention, but this levitation system was noisy and its scalability was limited. Magnetic levitation could silently hover multiple objects, but would require constant object monitoring and rapid adjustments in field configuration; a stable configuration of static magnetic forces is incapable of maintaining levitation, as stated by Earnshaw's Theorem [14].

Even if full levitation is not possible (or useful) in the future, small repulsive forces could be used to provide greater control over the friction between the pucks and the table surface. Giving the pucks a small "kick" to help them overcome static friction, or using repulsion as well as attraction to create a push-pull actuation system, could result in new motion possibilities.

Appendix C

Software Improvements

C.1 Coordinating Multiple Objects

Though our current system can manipulate and track multiple objects on the table, our software contains no mechanisms for preventing collisions between pucks. In the future, it may be useful to design path-planning algorithms to coordinate the simultaneous motion of multiple pucks. This would enable us to set the positions of multiple pucks to any configuration without the danger of destabilizing the system due to magnetic interactions between pucks.

C.2 Unified Software Architecture

All of our applications were programmed using three separate APIs: one for position tracking, one for graphical display, and one for actuation. The complexity that this entailed suggests a need for a single high-level API to support the design of actuated tabletop tangible interfaces.

Figure C.1 shows the data flow during a single iteration of the Actuated Workbench control loop. After sensing the positions of the objects on the table and receiving the positions of the objects on the remote workbenches, the system computes new target positions based on constraints and simulation rules. It then sets the state of the local magnet array, renders graphical simulations, and transmits the local object positions to the remote workbenches.

A well-designed software interface to this system would require the programmer to specify only a few parameters: (1) addresses of remote workbenches; (2) the ID of each physically instantiated object, local and remote; (3) equations relating object positions that specify the simulation parameters and constraints; and (4) graphical visualizations. The

system would then automatically handle the data exchange between the tracking system, the remote workbenches, and the array of magnets. This would eliminate a large portion of the overhead currently associated with designing a new application for the Actuated Workbench.

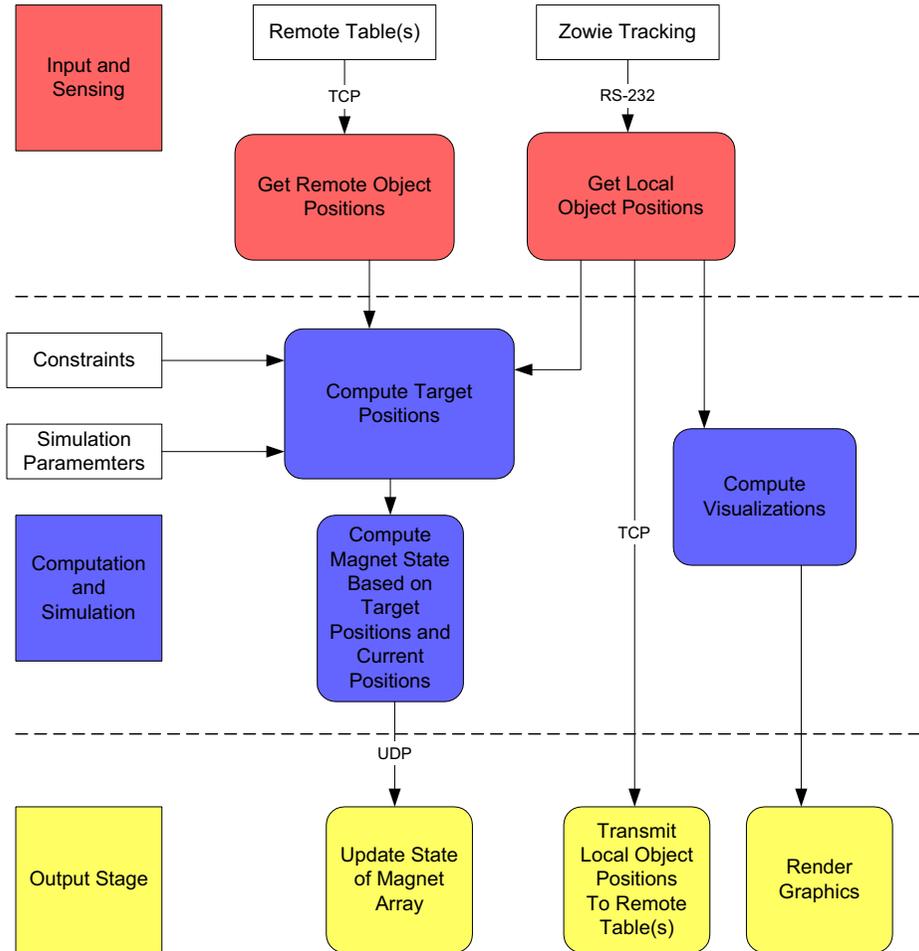


Figure C.1: Data flow diagram showing one iteration of the Actuated Workbench control loop.

Appendix D

New Directions for Actuation

D.1 Types of Actuated Interfaces

Researchers have developed a huge variety of computer-controlled actuation systems for human-computer interaction. This section describes some criteria for classifying these systems, with an eye towards identifying promising new areas of the design space.

D.2 Purpose of Actuated Feedback

In general, actuation systems in computer interfaces serve one of two objectives: providing haptic feedback, or changing physical layout. Haptic feedback is traditionally used in graphical user interfaces, in which the onscreen information is controlled remotely with an input device like a mouse or joystick. Since attention is divided between input and output in these interfaces, this touch feedback is particularly useful.

In most tabletop tangible interfaces, the input and output spaces are coincident, so haptic feedback is less important. However, since the physical layout of the objects is significant, it becomes useful for the computer to control the layout in the same way the user can. Tabletop tangible interfaces can then take advantage of spatial multiplexing without losing the flexibility of the GUI.

The way in which the computer interprets the configuration of physical objects in a tabletop tangible interface can range from the literal to the abstract. In the majority of tangible workbench systems, the configuration of objects specifies a fairly concrete geometric relationship. However, a new class of tangible interfaces is emerging that interprets geometric layout in a more abstract way, mapping physical layouts to ideas and relationships with no direct parallels in the physical world.

D.3 Spatial Complexity and Degrees of Freedom

In general, the trend in actuation systems has been towards higher spatial complexity. The earliest technologies that could be considered actuation systems for HCI were devices such as force feedback knobs, sliders, and steering wheels, and mice with small vibrating motors. Two-axis force feedback devices emerged soon afterward, including joysticks and mice on fixed pads with two-dimensional position control.

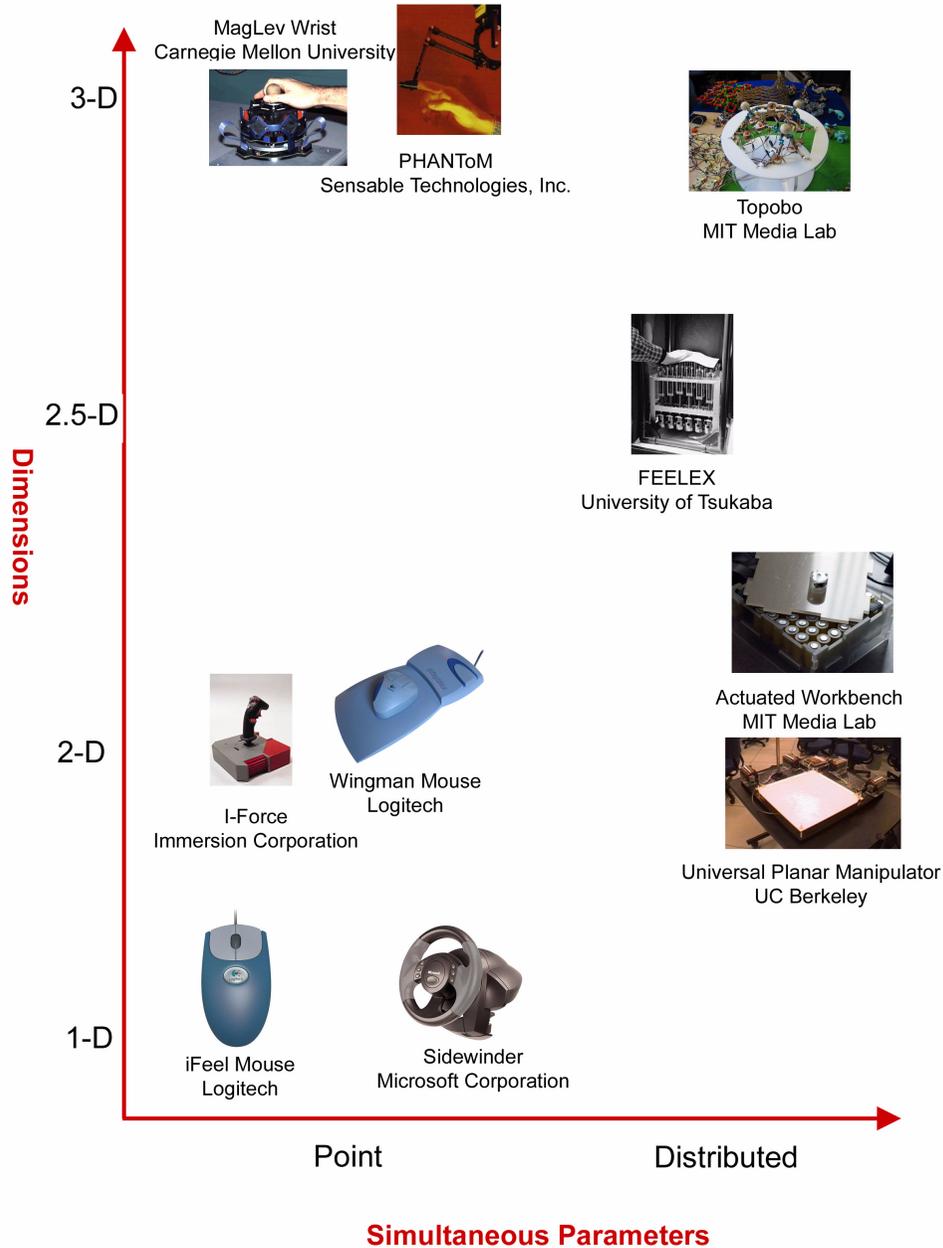


Figure D.1: Comparison of degrees of control in various actuation systems.

Newer technologies have brought even higher dimensionalities to actuation, as shown in Figure D.1. There is a distinction, however, between actuation dimensions and degrees of freedom. A system like the PHANToM [33] provides three dimensions of actuation, but only outputs forces at a single point. The Actuated Workbench operates only in two dimensions, but it is a distributed system capable of coordinating the simultaneous motion of multiple objects. As both the dimensionality and the simultaneous number of parameters in the interface increase, so does the potential expressiveness of the interface, but the difficulty becomes preventing the complexity from becoming overwhelming. As Stu Card wrote, “a major challenge of the post-WIMP interface is to find and characterize appropriate mappings from high degree-of-freedom input devices to high degree-of-freedom tasks.”

References

- [1] Badros, G. and Borning, A. "The Cassowary Linear Arithmetic Constraint Solving Algorithm: Interface and Implementation," Technical Report. UW-CSE-98-06-04, June 1998.
- [2] Baer, R., Television Gaming and Training Apparatus. US Patent #3728480, 1973.
- [3] Bal-shalom, Y. and Fortmann, T.E. Tracking and Data Association. Academic Press, 1988.
- [4] Batter, J.J. and Brooks, F P., Jr. Grope-1: A computer display to the sense of feel. In *Proceedings of IFIP 1972*, pp. 759-763.
- [5] Beetson, J.S. Visualizing Magnetic Fields: Numerical Equation Solvers in Action. Academic Press, 2001.
- [6] Berkelman, P., Hollis, R., and Baraff, D. Interaction with a Realtime Dynamic Environment Simulation using a Magnetic Levitation Haptic Interface Device. IEEE International Conference on Robotics and Automation, May 1999, pp. 3261 - 3266.
- [7] Brave, S., Ishii, H. and Dahley, A. Tangible Interfaces for Remote Collaboration and Communication. In *Proceedings of CSCW '98*, ACM Press, pp. 169-178.
- [8] Brooks, F. P., Jr. The computer scientist as toolsmith: studies in interactive computer graphics. In *Proceedings of IFIP 1977*, pp. 625-634.
- [9] Buxton, W. There's More to Interaction than Meets the Eye: Some Issues in Manual Input. In Norman, D.A. and Draper, S.W. (Eds.), (1986), *User Centered System Design: New Perspectives on Human-Computer Interaction*. Lawrence Erlbaum Associates, Hillsdale, New Jersey, 319-337.
- [10] Chang, A., O'Modhrain, S., Jacob, R., Gunther, E., Ishii, H. ComTouch: Design of a Vibrotactile Communication Device. In *Proceedings of Design of Interactive Systems 2002*.
- [11] Cohen, H. The Further Exploits of AARON, Painter. *Stanford Humanities Review*, Volume 4, Issue 2: Constructions of the Mind, 1995.
- [12] Cohen, J., Withgott, M., and Piernot, P. 1999. "Logjam: A Tangible Multi-Person Interface for Video Logging." In *Proceedings of CHI 1999*, ACM Press, pp. 128-135.
- [13] Dumas, S.T. and Jones, W. P. A Comparison of Symbolic and Spatial Filing. In *Proceedings of CHI 1998*, ACM Press, pp. 127-130.
- [14] Earnshaw, S. "On the Nature of the Molecular Forces which Regulate the Constitution of the Luminiferous Ether," *Transactions of the Cambridge Philosophical Society*, v. 7, 1842, pp. 97-112.
- [15] Everitt, K., Klemmer, S., Lee, R., and Landay, J. Two Worlds Apart: Bridging the Gap Between Physical and Virtual Media for Distributed Design Collaboration. In *Proceedings of CHI 2003*, ACM Press, pp. 553-560.
- [16] FCS Control Systems. 2001. FCS HapticMASTER. <http://www.fcs-robotics.com>.
- [17] Fitzmaurice, G.W., Ishii, H., and Buxton, W. Bricks: Laying the Foundations for Graspable User Interfaces. In *Proceedings of CHI 1995*, ACM Press, pp. 442-449.
- [18] Fitzmaurice, G. Graspable User Interfaces. Ph.D. Thesis, University of Toronto, 1996.
- [19] Fjeld, M., Bichsel, M., and Rauterberg, M. BUILD-IT: An Intuitive Design Tool Based on Direct Object Manipulation. In *Gesture and Sign Language in Human-Com-*

- puter Interaction, Lecture Notes in Artificial Intelligence*, v.1371, Wachsmut and Fröhlich, eds. Berlin: Springer-Verlag, pp. 297-308.
- [20] Francetic, A. and Shwe, H. Smart Toys: Benefits of Technology-Enhanced Play. In *Proceedings of Game Design Conference 2000*.
- [21] Immersion Corporation, "FEELit Mouse." Technical Document, San Jose, CA, October 1, 1997. Electronic version: <http://www.immerse.com>.
- [22] Iwata, H., Yano, H., Nakaizumi, F., and Kawamura, R. Project FEELEX: Adding Haptic Surface to Graphics. In *Proceedings of SIGGRAPH 2001*, ACM Press, pp. 469-476.
- [23] Ishii, H. and Ullmer, B., Tangible Bits: Towards Seamless Interfaces between People, Bits and Atoms. In *Proceedings of CHI 1997*, ACM Press, pp. 234-241.
- [24] Jacob, R., Ishii, H., et al. A Tangible Interface for Organizing Information Using a Grid." In *Proceedings of CHI 2002*, pp 339-346.
- [25] Jain, R. et al. Machine Vision. McGraw-Hill, 1995.
- [26] Kirsh, D. Complementary Strategies: Why We Use Our Hands When We Think. In *Proceedings of 17th Annual Conference of the Cognitive Science Society 1995*.
- [27] Kirsh, D. The Intelligent Use of Space. *Journal of Artificial Intelligence*, 72(1-2), 31-68, 1995.
- [28] Lansdale, M.W. Remembering About Documents: Memory for Appearance, Format, and Location. *Ergonomics*, 34, 8 (1991), 1161-1178.
- [29] Luntz, J., Messner, W., and Choset, H. Virtual Vehicle: Parcel Manipulation and Dynamics with a Distributed Actuator Array. In *Proceedings of the SPIE International Symposium on Intelligent Systems and Advanced Manufacturing*, Volume SPIE 3201, 1997.
- [30] MacLean, K., Snibbe, S., and Levin, G. 2000. Tagged Handles: Merging Discrete and Continuous Manual Control. In *Proceedings of CHI 2000*, ACM Press, pp. 225-232.
- [31] Malone, T.W. How do People Organize their Desks? Implications for the Design of Office Information Systems. *ACM Transactions on Office Information Systems* 1, 1 (January 1983), pp. 99-112.
- [32] Mandler, J. M., Seefmiller, D., Day, J. On the Coding of Spatial Information. *Memory & Cognition* 5, 1 (1997), 10-16.
- [33] Massie, T., and Salisbury, K., The PHANToM Haptic Interface: A Device for Probing Virtual Objects, *ASME Winter Annual Meeting*, Volume 55-1, pp. 295-300, 1994.
- [34] Maynes-Aminzade, D., Tan, B., Goulding, K., and Vaucelle, C. Hover: Conveying Remote Presence. Technical Sketch, SIGGRAPH 2002.
- [35] Nashel, A. and Razzaque, S. Tactile Virtual Buttons for Mobile Devices. *Extended Abstracts of CHI 2003*, pp. 854-855.
- [36] Naveh-Benjamin, M. Coding of Spatial Location Information: An Automatic Process? *Journal of Experimental Psychology: Learning, Memory, and Cognition* 13, 4 (1987), 595-605.
- [37] Negroponte, Nicholas and the Architecture Machine Group, MIT. "Seek." Originally shown at Software, Information Technology: Its New Meaning for Art Exhibition, Jewish Museum, New York, 1970.

- [38] Patten, J., and Ishii, H., A Comparison of Spatial Organization Strategies in Graphical and Tangible User Interfaces. In *Proceedings of DARE 2000*, pp. 41-50.
- [39] Patten, J., Ishii, H., Hines, J., Pangaro, G. Sensetable: A Wireless Object Tracking Platform for Tangible User Interfaces. In *Proceedings of CHI 2001*, ACM Press, pp. 253-260.
- [40] Patten, J., Recht, B., and Ishii, H. Audiopad: A Tag-based Interface for Musical Performance. In *Proceedings of Conference on New Interface for Musical Expression 2002*, pp. 11-16.
- [41] Piper, B., Ratti, C., and Ishii, H. Illuminating Clay: A 3-D Tangible Interface for Landscape Analysis. In *Proceedings of CHI 2002*, ACM Press, pp. 355-362.
- [42] Reznick, D. and Canny, J. C'mon Part, Do the Local Motion. In *Proceedings of the IEEE International Conference on Robotics & Automation 2001*.
- [43] Robertson, G., Czerwinski, M., Larson, K., Robbins, D.C., Thiel, D., van Dantzich, M. Data Mountain: Using Spatial Memory for Document Management. In *Proceedings of UIST 1998*, ACM Press, pp. 153-162.
- [44] Schäfer, K., Brauer, V., and Bruns, W. A New Approach to Human-Computer Interaction: Synchronous Modelling in Real and Virtual Spaces. In *Proceedings of DIS 1997*, pp.335-344.
- [45] Shimoga, K. B. A Survey of Perceptual Feedback Issues in Dexterous Telemanipulation: Part I. Finger Force Feedback. *IEEE Annual Virtual Reality International Symposium 1993*, pp. 263-270.
- [46] Shimoga, K. B. A Survey of Perceptual Feedback Issues in Dexterous Telemanipulation: Part II. Finger Touch Feedback. *IEEE Annual Virtual Reality International Symposium 1993*, pp. 271-279.
- [47] Snibbe, S. et al. Haptic Techniques for Media Control. In *Proceedings of UIST 2001*, ACM Press, pp. 199-208.
- [48] Tan, D.S., Stefanucci, J.K., Proffitt, D.R., Pausch, R. Kinesthetic Cues Aid Spatial Memory. In *Extended Abstracts of CHI 2002*, pp. 806-807.
- [49] Ullmer, B. and Ishii, H. The metaDESK: Models and Prototypes for Tangible User Interfaces, in *Proceedings of UIST 1997*, pp. 223-232.
- [50] Ullmer, B., Ishii, H., and Jacob, R. Tangible Query Interfaces. In *Proceedings of Interact 2003*.
- [51] Underkoffler, J. and Ishii, H. Urp: A Luminous-Tangible Workbench for Urban Planning and Design. In *Proceedings of CHI 1999*, ACM Press, pp. 386-393.
- [52] Virtual Technologies, "CyberTouch." Company brochure, Palo Alto, CA, 1998. Electronic version: <http://www.virtex.com>.
- [53] Wellner, P. Interacting with paper on the Digital Desk. *Communications of the ACM*, 36(7), 86-96, 1993.
- [54] Zhang, J. and Norman, D. A. Representations in Distributed Cognitive Tasks. *Cognitive Science* 18, 1 (1994), 87-122.

