

Trackmate: Large-Scale Accessibility of Tangible User Interfaces

by

Adam Kumpf

B.S., Massachusetts Institute of Technology (2005)
M.Eng., Massachusetts Institute of Technology (2007)

Submitted to the Program in Media Arts and Sciences,
School of Architecture and Planning
in partial fulfillment of the requirements for the degree of

Master of Science

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

June 2009

© Massachusetts Institute of Technology 2009. All rights reserved.

Author
Program in Media Arts and Sciences,
School of Architecture and Planning
May 8, 2009

Certified by
Hiroshi Ishii
Muriel R. Cooper Professor of Media Arts and Sciences
Thesis Supervisor

Accepted by
Deb Roy
Chairperson, Departmental Committee on Graduate Students
Program in Media Arts and Sciences

Trackmate: Large-Scale Accessibility of Tangible User Interfaces

by

Adam Kumpf

Submitted to the Program in Media Arts and Sciences,
School of Architecture and Planning
on May 8, 2009, in partial fulfillment of the
requirements for the degree of
Master of Science

Abstract

There is a long history of Tangible User Interfaces (TUI) in the community of human-computer interaction, but surprisingly few of these interfaces have made it beyond lab and gallery spaces. This thesis explores how the research community may begin to remedy the disconnect between modern TUIs and the everyday computing experience via the creation and dissemination of Trackmate, an accessible (both ubiquitous and enabling) tabletop tangible user interface that scales to a large number of users with minimal hardware and configuration overhead. Trackmate is entirely open source and designed: to be community-centric; to leverage common objects and infrastructure; to provide a low floor, high ceiling, and wide walls for development; to allow user modifications and improvisation; to be shared easily via the web; and to work alongside a broad range of existing applications and new research interface prototypes.

Thesis Supervisor: Hiroshi Ishii

Title: Muriel R. Cooper Professor of Media Arts and Sciences

**Trackmate: Large-Scale Accessibility
of Tangible User Interfaces**

by
Adam Kumpf

The following people served as readers for this thesis:

Thesis Reader

Hiroshi Ishii

Muriel R. Cooper Professor of Media Arts and Sciences

MIT Media Lab

Thesis Reader

Mitchel Resnick

LEGO Papert Professor of Learning Research

MIT Media Lab

Thesis Reader

V. Michael Bove, Jr.

Principal Research Scientist

MIT Media Lab

Acknowledgments

I would like to thank my advisor, Hiroshi Ishii, for his guidance and keen critical perspective throughout the research process. I would also like to thank my readers, Mitchel Resnick and Michael Bove, for their time and feedback regarding this thesis.

Special thanks to Dr. Jean-Baptiste Labrune, who supplied countless creative counterpoints, frequently offering both pointed technical suggestions and high-level conceptual ideas to the project. I am appreciative of all the students in the Tangible Media Group (Jamie Zigelbaum, Daniel Leithinger, Keywon Chung, Cati Vaucelle, Leonardo Bonanni, and Amanda Parkes) for their suggestions, constructive criticism, and help with prototyping over the past two years.

I am indebted to the entire community of Trackmate users and user-developers, a strong motivating force for this research. Their questions and innovations shaped the direction of the Trackmate project.

To my loving family, I am thankful for unwavering support and encouragement of this project, and of all my other endeavors, past, present, and future.

And finally, I am grateful beyond words to Maura Atwater for helping to edit this thesis for days on end and for excitedly journeying with me as this exploration unfolded.

Contents

1	What's the problem?	15
1.1	Still the same ol' mouse and keyboard	15
1.2	The interface bottleneck	16
1.3	Barriers to new ideas and learning	18
1.4	The importance of community	20
2	What's been tried before?	23
2.1	Sketchpad (1963)	23
2.2	Engelbart's Mouse (1968)	25
2.3	Put-That-There (1982)	26
2.4	Knowledge Navigator (1987)	27
2.5	TeamWorkStation (1990)	29
2.6	Wellner's Digital Desk (1991)	30
2.7	ClearBoard (1992)	31
2.8	Bricks (1995)	32
2.9	Tangible Bits (1997)	33
2.10	Augmented Surfaces (1999)	34
2.11	Audiopad (2002)	35
2.12	Reactable (2005)	36
2.13	Mixed Reality Interface (2006)	37
2.14	Mir:ror (2008)	38
2.15	DIY for CHI Workshop (2009)	39

3	What's the big idea?	41
3.1	Looking for the right mix	41
3.2	Lowering the floor	43
3.3	Raising the ceiling	43
3.4	Widening the walls	44
3.5	What Trackmate aims to address	45
4	How does Trackmate work?	47
4.1	Overview of the Trackmate project	47
4.2	Easy-to-track tag design	49
4.3	Tag creator software	51
4.4	Tracking software	52
4.5	The LusidOSC protocol	54
4.6	Protocol libraries	56
4.7	Spatial applications	56
4.8	Ways to build the system	60
4.9	Support for the community	61
5	What happened?	63
5.1	The first release	63
5.2	Effects of press and new web content	64
5.3	Usage trends so far	65
5.4	Users as innovators	68
5.5	Ideas for hire	69
5.6	What has been unexpected?	71
6	Conclusion. What's next?	73
6.1	Conclusion	73
6.2	What to work on in the next 10 years?	74
6.3	What to work on after the next 10 years?	75
6.4	A note about complexity	76

6.5 A final challenge	77
A Mechanical Turk User Data	79
B Trackmate Tag Creation Code	91
Bibliography	95

List of Figures

1-1	A typical computer in 2009	16
1-2	The evolution of the airplane's user interface.	17
1-3	A cartoon illustrating the problem of local maxima.	19
2-1	Sutherland's Sketchpad	23
2-2	Engelbart's Mouse	25
2-3	Put-That-There	27
2-4	Apple's Knowledge Navigator	28
2-5	TeamWorkStation	29
2-6	Wellner's Digital Desk	30
2-7	ClearBoard	31
2-8	Bricks	32
2-9	Tangible Bits	33
2-10	Augmented Surface with files on a tabletop	34
2-11	Audiopad	35
2-12	Reactable	37
2-13	Mixed Reality Interface	38
2-14	Mir:ror RFID Reader	39
2-15	DIY for CHI Workshop	40
3-1	Illustration of emergence	41
3-2	Low floor, high ceiling, wide walls	42
4-1	Trackmate logo	47

4-2	Developing Trackmate; Post-it layout	48
4-3	The Trackmate tag design	49
4-4	Screenshot from the Trackmate Tagger	51
4-5	Screenshot from the Trackmate Tracker	52
4-6	Illustration of the tag-finding algorithm.	53
4-7	LusidOSC Logo	54
4-8	LusidOSC Library code in Java/Processing	57
4-9	Six examples of spatial applications	58
4-10	Five ways to build Trackmate	60
4-11	Screenshot of the Trackmate website	61
5-1	Trackmate online press	64
5-2	Graph of visitors to the Trackmate website	66
5-3	Demographic of visitors to the Trackmate website	67
5-4	Traffic sources leading people to the Trackmate website	67
5-5	Number of ideas by tag from Mechanical Turk	70

Chapter 1

What's the problem?

1.1 Still the same ol' mouse and keyboard

Over the past 40 years of computer interface research, the fundamental input devices and interaction techniques have remained relatively unchanged. The dominance of the keyboard and mouse is both a strength and a weakness for the ways we currently use our computers.

While Moore's law has held true – computation, memory, and connectivity continue to increase at an exponential rate[25] – we still use the same basic interfaces developed in the late 1960s. The keyboard and mouse are both highly generic interfaces; this property allows them to be used across all computing tasks, but at the cost of not having any specialized functionality or feedback that may aid in a particular task.

Furthermore, there is a large risk that any general purpose interface, no matter how well designed, may not be optimal for any one task or person. This is a common problem among designers who work on interfaces that relate to the human body. Designing a chair that fits the average body size perfectly may only be comfortable for a small fraction of the intended audience. Allowing for adjustment and passive conformability (such as padded seating) broadens the range of users that may comfortably sit in the chair, but the use of a chair will still require that it be differentiated for each major context (e.g., a car seat should fit differently than a dining room chair

or a recliner).

The prevalence of a generic interface, such as the keyboard and mouse, is an exception, not the rule, in comparison to most other interfaces designed for complex systems.

1.2 The interface bottleneck

Humans are multi-modal, providing redundancy in our observations and allowing us to experience something as simple as biting an apple with all five of our senses. Our experiences with computers, while significantly more complex, are relatively impoverished. To interact with a computer using a single pointing device (such as the mouse) or a group of two-state momentary push-button switches (such as the keyboard) is both an inefficient approach to complex tasks and an insufficient use of our multi-modal abilities to make sense of the world.



Figure 1-1: A typical computer in 2009; image from Apple's website[9].

When we use the computer as a mediator to communicate to other people, our human-to-human communication is limited by these same generic interfaces. The bottleneck when using applications or communicating via the web with colleagues, friends, and family is no longer the bandwidth or processing power of the computer –

it's the interface. Figure 1-1 shows a typical computer interface today with a keyboard and mouse. Notice the relative size of the interface to the overall system as well as the total functionality of the input devices.



Figure 1-2: The evolution of the airplane's user interface (1903 Wright Flyer[36], Boeing 747-206BM[23]).

Compare this to another complex system: the airplane. Figure 1-2 shows the first airplane, the Wright Flyer from 1903, with its user interface highlighted. The user interface is understandably minimal, since the Wright brothers' main focus was on the technical aspects of propulsion and wing design to achieve flight. Fast forward 100 years, and the user interface of a Boeing 747 looks quite different. The pilot is given precise control over almost every aspect of the machine. Even though many settings are never touched during a flight, in the event that something goes wrong, quick access to controls can be extremely important. The modern airplane's interface allows for direct mappings between knobs, sliders, buttons, dials, and joysticks to hundreds of unique parameters; all of which can be manipulated with both hands, used by multiple users simultaneously, and easily read at a glance.

Large, directly-mapped interfaces are common to practically every complex system built. It's almost impossible to envision a power plant, subway control station, recording studio, or factory floor without carefully constructed user interfaces, and instead controlled entirely by a single mouse and keyboard.

From Shneiderman's 1983 paper on Direct Manipulation[35] to Wellner's Digital Desk[42] in 1993, researchers in the field of human-computer interaction have been

looking at ways to make computers more intuitive by relating them back to our everyday lives. Shneiderman suggested that directly interacting with elements on the screen as if they were mapped to reality produced in users an understanding and an enthusiasm not found in other contexts. Wellner took these ideas a step further by bringing onto the computing surface real documents which could then be augmented and manipulated digitally. Soon after Wellner's work, Fitzmaurice, Ishii, and Buxton published on the advantages of small graspable objects, called Bricks[4], and ideas began to emerge in the CHI research community about how far the physical side of computers could be taken.

Hiroshi Ishii and Brygg Ullmer's 1997 paper, Tangible Bits[14], was the first major publication to clearly define Tangible User Interfaces (TUI) as a new area of research which seamlessly coupled digital information with physical objects. CHI Researchers around the world showed great enthusiasm for this new field thereafter, citing Ishii more than twice as much as any other author[28]. In 2007, just 10 years after Tangible Bits, the international conference on Tangible and Embedded Interaction (TEI) was created to bring together the many researchers already working on tangible interfaces.

With such support from the academic community, it is surprising that so few tangible interfaces have trickled out into the everyday, non-academic world. What are the missing ingredients that have kept tangible user interfaces under the warming lamp for so long?

1.3 Barriers to new ideas and learning

One likely explanation for the hesitation in adopting new computer interfaces is that users are sitting nicely atop a local maximum within a much larger space of possibilities (as shown on the left of Figure 1-3). Deviating slightly in any direction is either more difficult, inefficient, uncomfortable, or expensive; there is not enough incentive to cause users to gradually shift elsewhere. Rather, since users will likely require a process of learning and adaptation to fully utilize a new interface, they must see an extremely compelling reason for venturing off of the keyboard-mouse hilltop to the

promise of an even taller peak.

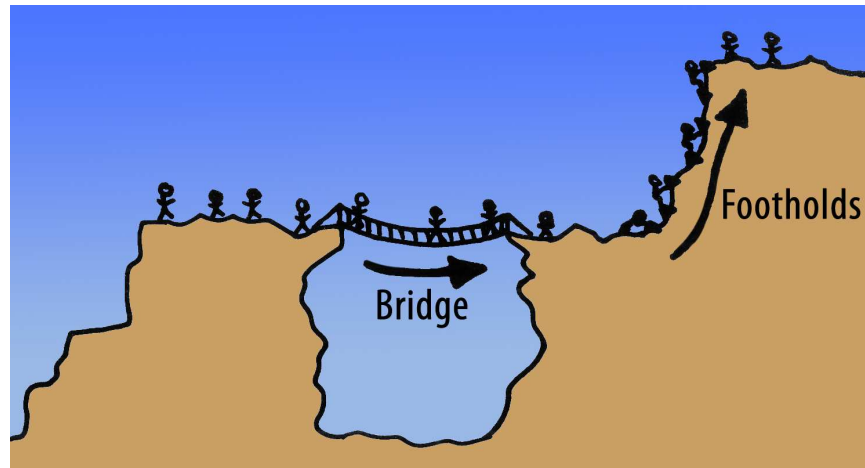


Figure 1-3: A cartoon illustrating the problem of local maxima. Bridges allow users to cross gaps, and footholds enable climbing up slopes that would otherwise be too steep.

Two approaches help with the problem of being stuck at a local maximum. First, if there is no easy path to the next peak (i.e., it is too different from what users already know), a metaphorical bridge or walkway can be built to ease the transition. For example, instead of trying to replace the keyboard and mouse, we may start by exploring interfaces that work in concert with existing technology; eventually users may gravitate towards the non-traditional interfaces if they do in fact support additional functionality and improve the overall computing experience (both productivity as well as the user's enjoyment).

Second, if the new peak is too steep (i.e., it is too difficult for users to learn), a metaphorical rope or footholds can be added to support the climbers' journey to the top. Just as game designers use levels of increasing difficulty and teachers use lessons that sequentially build in complexity, a scaffolding to help learning of the new interface or paradigm may allow users to gradually ascend with confidence rather than facing the entire magnitude of the challenge from the beginning. To rise above today's keyboard and mouse, strategic footholds may make for an easier climb toward exciting new interfaces that lie ahead.

1.4 The importance of community

Having a better idea isn't always reason enough for people to be willing to accept it. Ideas are cheap (free to conceive and numerous in quantity), but implementing them or adapting to their new constraints can be hard work. Communities of both users and developers should not be an afterthought to a new idea, but rather, integral throughout the idea's conception.

Many successful open source initiatives have been centered on a given community of users from a very early stage. Even projects that begin with a small group of visionaries must quickly learn to incorporate outside perspectives and be open to conflicting viewpoints as discussions unfold in mailing lists and forums.

Still, it is unrealistic to design any large, complex system by committee since people often disagree on fundamental decisions and the project can become stalled before ever getting off the ground. For this reason, many new projects are started with a small group of people who take a first pass at constructing their idea, focused on a short list of things they believe to be important. Members from the larger community can then work together to find bugs, spot inefficiencies, and suggest new directions, often making concrete suggestions on how to pursue them via online discourse.

To gain support from the project's community and grow beyond its initial iteration, developers must be open (though not indifferent) to new ideas and allow much of the project to rest in the hands of other users and developers. Unique to open source software, if the community does not agree with the implementation decisions and overall vision of a project, any user may choose to start a new project leveraging all of the existing development of the current code-base, but with their perspective accounted for. Even though "forking" code is extremely rare in practice, it provides strong motivation for developers to not disregard the response of the community.

It is important to understand the history of user interfaces before working to build a community initiative around improving them. The next chapter walks through some of the most significant human-computer interface research pursued over the past 55 years in an effort to lay the foundation for Trackmate, a project that combines many

of the most salient ideas and serves as a catalyst for bringing tangible user interfaces beyond the research lab.

Chapter 2

What's been tried before?

2.1 Sketchpad (1963)



Figure 2-1: Sutherland's Sketchpad CAD application; still image from a demonstration video recorded at MIT Lincoln Labs in 1963.

In 1963, Ivan Sutherland demonstrated Sketchpad, a system in which users could directly interact with a CRT screen via a pen-like pointing device[37]. Sketchpad was a big leap beyond even the most sophisticated analog computers (such as the Astrolabe[26] used by astronomers and navigators or the Librascope[27], which physically transformed the positions of dials and knobs into a display that indicated an

aircraft’s balance). Using the Sketchpad’s “light pen,” for the first time users could draw objects on the screen, copy them to other locations, and even specify constraints which the computer would then solve (for example, forcing lines to be parallel or perpendicular). These ideas were then extended to include 3D coordinates to draw and manipulate wire-frame objects. Additionally, Sketchpad was the first computer system to use a window that the user could move on the screen as if it were a sheet of paper, as well as dynamically zoom in or out from the drawing in real-time.

Sketchpad was also the first object-oriented software system; it enabled the user to work with various components as instances of a larger class. In his demonstration, Sutherland was able to specify the shape and constraints of a single rivet, and then proceed to insert multiple copies of that rivet into another drawing. When the original rivet was later changed, each place where the rivet had been used before was updated as well.

Focusing on Sketchpad’s user interface, the light pen was highly acclaimed for its ability to allow the user to interact directly with the screen, but also criticized for the strain that it put on the user when working with the system for a long period of time. In a video discussing Sketchpad 25 years after its debut, Alan Kay mentions that although the light pen “was definitely discovered to be a very bad input device, because the blood runs out of your hand in about 20 seconds and leaves it numb, [...] it’s been reinvented at least 90 times in the last 25 years.[18]” Intuitive functional mappings are often sacrificed for physical ease-of-use, but human-computer interaction research has often revisited this dilemma in hopes of bringing mental ease-of-use to the table as well.

Ivan Sutherland’s thesis was astonishing on a number of levels; it redefined the way users both interacted with and perceived computers. Manipulation of virtual objects could be directly coincident with the screen and the computer could solve problems that required high levels of precision, mathematics, constraint satisfaction, or redundancy.

2.2 Engelbart's Mouse (1968)



Figure 2-2: A still image from Doug Engelbart's demonstration video at SRI in 1968; this is the first time the mouse was shown to the public (in Engelbart's right hand).

In 1968, Doug Engelbart presented his team's work from Stanford Research Institute, demonstrating another significant step forward in the way that users perceive and interact with computers[3]. Most notably, the mouse was shown to the public for the first time as a way to indirectly interact with content on the screen via a small black dot. Next the mouse was a standard keyboard, and next to that, a chord keyboard: a five-finger input device that allowed one hand to represent up to 32 different actions by pressing combinations of fingers at the same time. Using the mouse and chord keyboard together, the user was able to navigate large information spaces (such as lists of hierarchically organized tasks) without needing to reposition his or her hands.

The NLS (an abbreviation for oNLine System) that Doug and his team created also allowed for networked collaboration with multiple video inputs, multiple mouse cursors, and inter-person messaging (similar to email, but centralized on one machine). NLS was the first system to implement hypertext, allowing users to jump between documents via specially assigned words or phrases (or links within drawings). This functionality was inspired by Vannevar Bush's paper, "As We May Think[2],"

which laid out a groundbreaking vision of the future including “associative trails” to visualize and navigate between related content.

Engelbart’s 1968 demonstration has been nicknamed the “mother of all demos” due to its incredible impact on computing interfaces and the incredible depth to which his team executed their vision to present it to the world. The complete demonstration was captured on video and lasted well over an hour as Engelbart and his colleague, Bill English, walked through every component of the system and employed it in real-time.

Engelbart and his team made an enormous contribution to the evolution of the computer interface, but even more intriguing is the methodology that he adamantly pursued to develop the system. Since the NLS was intended to aid in productivity and thinking, they decided to follow a “bootstrapping strategy” where the system itself was used to further develop and study the NLS. In the 1968 demonstration, Engelbart briefly discusses bootstrapping and emphasizes that the best way to develop a system is to be working closely with the study group; bootstrapping suggests that the developers pull themselves up by their own bootstraps by becoming dual citizens and simultaneously being a part of the study group. Engelbart’s bootstrapping approach required a significant initial investment of time and resources to make the system usable enough, after which it had the potential to increase abilities of all developers exponentially.

2.3 Put-That-There (1982)

In 1982 Chris Schmandt demonstrated Put-That-There, a speech and gesture interface system that allowed the user to control spatial information with voice commands, creating objects, placing them on a map with simple pointing gestures, and then asking the system questions about the objects’ locations[33]. Even though the user was not touching the information directly, being able to use natural gestures of the human body to point at a location which the computer could understand allowed for seamless and intuitive interaction.

Put-That-There also supported multiple users at the same time. Not only could



Figure 2-3: Chris Schmandt adds boats to a map with pointing gestures and voice commands; image from Put-That-There video.

users interact with the space via pointing, but the computer also kept track of who had performed which actions. This enabled users to have ownership over their actions and specify whether an object should be locked. If another user tried to manipulate a locked object, they would first need permission from the person who originally locked the object. Here, the computer facilitated human-to-human interaction attempting to fully mediate the communication via graphical dialogs.

By working in a larger space, more users could be involved with the computer simultaneously. No single person blocked the screen or had absolute control via a keyboard and mouse. And with the use of gestures and voice, all actions performed in the space were highly legible by other users – something often overlooked in the design of computer interfaces as detailed later in Zigelbaum’s work with External Legibility[43].

2.4 Knowledge Navigator (1987)

In 1987, Apple released a conceptual prototype video to demonstrate their vision for the future of computer interaction using natural conversation, portable data types, and mass networking (detailed in the epilogue of a book by former Apple CEO, John

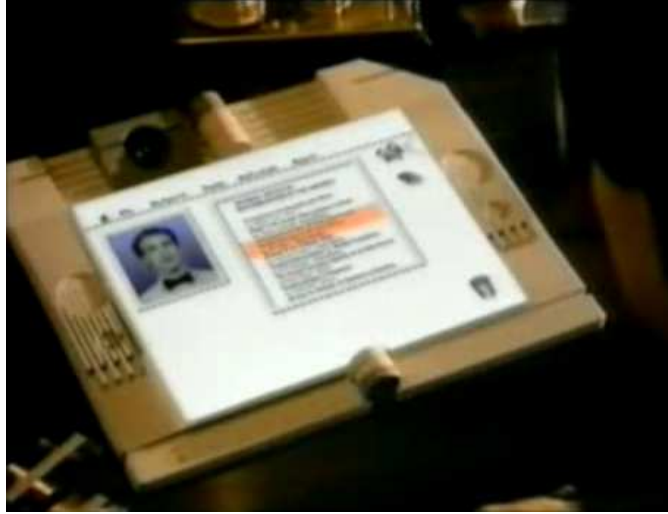


Figure 2-4: Image from Apple’s Knowledge Navigator concept video.

Sculley[34]). In the video, the Knowledge Navigator acts as a personal assistant, able to hold a conversation with the user, notify the user of important announcements and information, and dynamically check schedules, search data, and run simulations.

The form factor chosen for the Knowledge Navigator is a familiar two-page book layout. When open, the screen seamlessly covers both pages to create one display. Surprisingly, there is no keyboard or mouse in sight – with an intelligent digital assistant, voice can be used to interact with the computer on every level. While the video was intended purely as a conceptual mock-up of future possibilities, its limitations are clear: it would not allow for complex physical interaction, such as drawing or quickly manipulating spatial objects.

Twenty-two years later, staying true to Apple’s original focus, it would be interesting to remake the video with significant modifications to reflect upon the current state of human-computer interaction research. For example, perhaps drawing could be input directly onscreen, or even indirectly by using paper and sending the image to the computer as the drawing takes place. Additionally, tangible user interfaces could help to resolve the issue of opaque abstraction by utilizing different pages to segment content or using separate books to aid with particular tasks.

2.5 TeamWorkStation (1990)



Figure 2-5: Two collaborators share physical desktops via translucent video overlay; image from TeamWorkStation explanation video.

In 1990, TeamWorkStation was developed to create a seamless shared workspace and allow collaborative annotation of physical documents via translucent overlays and live video conferencing[11]. Using an overhead camera to capture the user's physical desktop, collaborators could draw on real paper and share it with each other digitally. Since anything could be placed on the desktop, 3D objects and non-traditional media could be immediately incorporated without any additional infrastructure.

With video as the main communication medium, the results could not be shared directly (each user's desktop reflected only the parts of the composite translucent overlay that they have contributed), but could easily be captured and printed out for future reference. This physical screenshot could then be used with the system recursively, enabling meta dialog about an idea or previous session.

TeamWorkStation mainly focused on collaborative discussions of technical material, but the same technology was also shown to be applicable to other scenarios. For example, a calligraphy teacher could make brush strokes on a piece of paper, and the remote student could follow the lesson via her or his own paper. Mistakes could be quickly visualized as the difference between strokes, and additional feedback could be

given either by the instructor’s voice or with subsequent markings on the paper.

2.6 Wellner’s Digital Desk (1991)

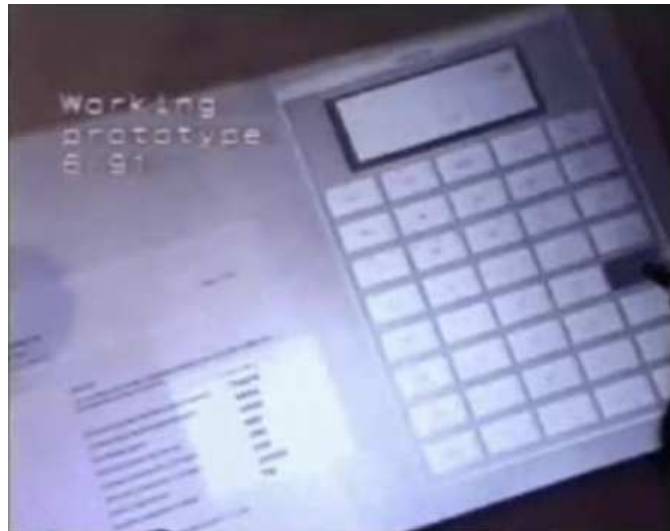


Figure 2-6: The Digital Desk is augmented with a digital calculator overlaid via projection onto a physical desk; image from a video of the working prototype in 1991.

In 1991, Pierre Wellner created the Digital Desk, mapping the desktop metaphor onto the physical paper-centric desk using a coincident video camera and projector (later published in 1993[42]). Using an LED-tipped pen, the user was free to interact with the surface of a real desk as if the digital workspace had been extended out onto the surrounding environment. With both a camera and a video projector, the desk became bidirectional, allowing the user to both recognize written information on a document (using optical character recognition) and graphically display data and interface elements in the same space.

In the Digital Desk video demonstration, Wellner asks in response to the graphical desktop metaphor, “But, what if we took the opposite approach? What if, instead of making the workstation more like the desk, we made the real desk more like the workstation?” This simple reversal of the question led the research community to reconsider fundamental assumptions about the user’s computer experience and to

dream about how it could someday exist beyond the constraints of a desktop box.

An envisionment was also included in the demonstration video to suggest other areas for which this digitally augmented desktop could be useful. When working with large amounts of financial data (for example, organizing receipts), the Digital Desk could allow the user to quickly copy and paste data, as well as execute calculations (such as summing a column). Another envisionment suggests using the Digital Desk for illustration, where the artist could easily perform redundant operations (such as copying trees along a roadside or drawing shingles on a rooftop) by using a small set of gestures for selecting, copying, and scaling.

2.7 ClearBoard (1992)



Figure 2-7: A user collaborates by drawing information onto the ClearBoard.

In 1992, two years after TeamWorkStation and closely related to its conceptual framework, ClearBoard was demonstrated[12]. ClearBoard was designed to give the users a feeling of sharing a single piece of glass. The semi-transparent shared workspace was developed to enable collaborative sketching while maintaining awareness of eye gaze, gesture, and voice of the other user. By using video to capture the user's face and drawing input, the image could be flipped to give the other user the

impression that they were both on the same side of the glass (allowing text to be readable for both users, while still maintaining eye gaze legibility).

Another important distinction from TeamWorkStation is that ClearBoard was meant to take the place of the computer, not reside within the computer as a single application. This enabled the system to function as an always-on device, where users could immediately sit down and begin working with the system.

While there were drawbacks due to the technical implementation (such as hard-to-erase markers and low contrast video), ClearBoard made a strong statement that the future of computers could be both familiar yet radically different. With overlaid video and drawing, the focus of the user could remain on the task at hand and its associated dialog, without needing to explicitly use the computer's graphical interface or command structure.

2.8 Bricks (1995)



Figure 2-8: Bricks as part of the GraspDraw application using ActiveDesk[4].

In 1995, George Fitzmaurice, Hiroshi Ishii, and Bill Buxton created Bricks, a Graspable User Interface that used physical objects as tightly-coupled controllers of virtual objects[4]. Bricks encompasses a larger set of principles applied to graspable interfaces and was demonstrated with the GraspDraw application on top of the Ac-

tiveDesk platform.

Much of the Bricks research focused on the way in which users would interact with the physical pieces. When working with a single Brick, translating and rotating the object corresponded directly with the graphical element attached to it. Lifting the Brick vertically would detach the object from the graphical element, and placing it back down on another element would reassign the Brick. Using multiple objects together allowed for more complex action, enabling users to stretch, rotate, and define curved paths quickly on a tabletop.

Along with their particular implementation of Bricks, the authors also provided an extensive overview of the design space in which similar interfaces could be constructed. Bricks served as a call-to-action for the user interface community around a large and emerging space that needed more detailed research and broad exploration.

2.9 Tangible Bits (1997)

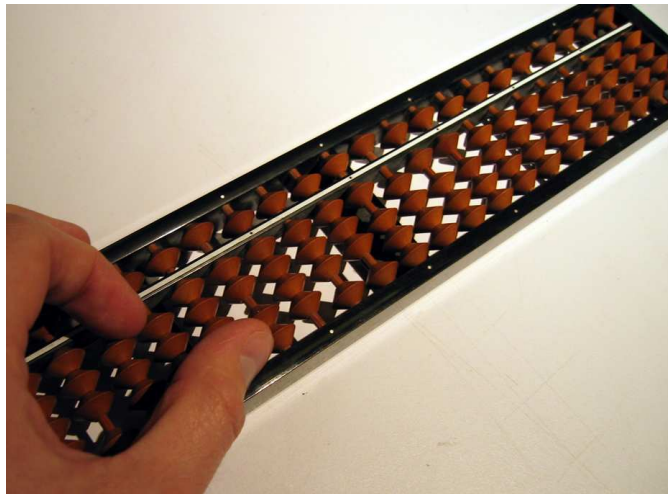


Figure 2-9: An abacus; the icon often used to represent the idea of Tangible Bits.

In 1997, Hiroshi Ishii and Brygg Ullmer published a paper in CHI on Tangible Bits[14], formalizing Tangible User Interfaces as a way of grasping and manipulating digital information in the center of the user's attention as if it were part of an everyday physical object. With Tangible Bits, the objects of an interface were designed to feel

as though they were physically iconic handles, or Phicons, into a much larger digital world.

Tangible Bits built upon on ideas from Bricks, but with a more refined scope and direction for fellow researchers to join the effort. The paper was presented as a visionary new space, but was also backed by numerous implemented tangible systems to ground the ideas, including the metaDESK, the Tangible Geospace, and the ambi-entROOM. The research community showed great enthusiasm for the new direction of physically coupled digital interfaces.

2.10 Augmented Surfaces (1999)

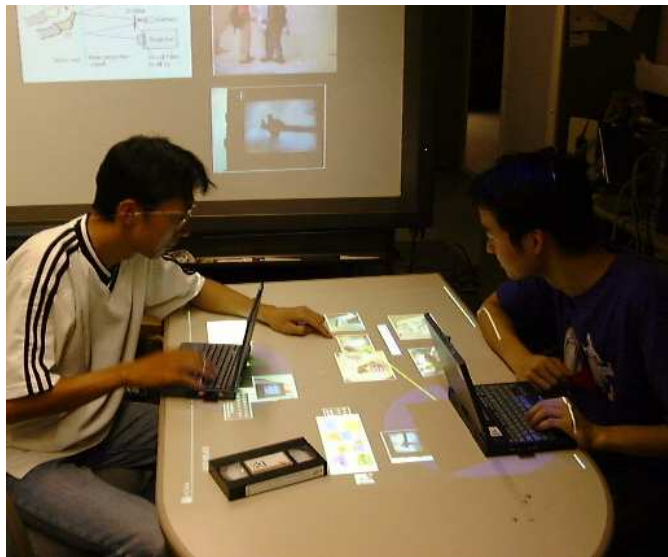


Figure 2-10: An Augmented Surface with laptops and files on a tabletop.

In 1999, Jun Rekimoto created a range of systems that allowed information to be dragged between electronic devices and displays[31]. By using overhead cameras to track objects, numerous projected displays, and an infrastructure to store objects with associated content, users could smoothly interchange digital data between devices in the same space. Rekimoto's systems used techniques such as HyperDragging and Pick-and-Drop by laser pointer to move virtual contents around in the physical

environment without needing to explicitly input the position information of computers or objects.

To uniquely identify and locate objects, visual markers were used to identify tagged items in the space. Users could simply open their laptop (which would be identified by an overhead camera) and see associated contents spill out onto the work surface. If collaborators wished to share items with one another, they could grab an object with a pointer and drag it over to another computer or large-screen display. Since any tagged item could act as a digital container, non-computational items, such as video tapes or documents, could still function within the augmented space.

Additionally, some tagged props were set up to map to specific actions and perspectives. For example, an object that looked like a camera was mapped to a perspective view within a 3D model. When a user moved the camera on the tabletop (showing a 2D overhead view, typical in an architectural drawing), he or she would then see a view from within the space projected on a large wall-mounted display, as if physically looking into the virtual camera.

2.11 Audiopad (2002)



Figure 2-11: James Patten's Audiopad.

In 2002, James Patten developed a music-based tangible interface on a tabletop

with physical tokens that linked to digital sounds[30]. Audiopad, an audio mixing application for electronic DJs, uses a set of round tokens on a tabletop to represent various sound samples, a microphone, and a tool for dynamically selecting different sounds, while a projector overlays information about the audio being played. Audiopad is built using the Sensetable platform, an analog electronic surface that can locate the position of specially tuned circuits when in close proximity.

Similar to Englebart’s bootstrapping design approach of being both a developer and a user of his new system, Patten both built and actively performed with Audiopad. The interface quickly evolved into something that was more of an instrument than a prototype as performance-necessary features (such as realtime effects, tempo, and grouping sounds in a hierarchical structure) emerged.

Besides the specific Audiopad application, the underlying Sensetable infrastructure was also used for a range of other tangible interface prototypes, including network simulation and analysis tools. Applications requiring both position and rotation information (such as an implementation of Urp, an urban planning workbench with architectural models) were able to obtain it by using two tokens attached to a single object, then solving for the angle between their locations.

2.12 Reactable (2005)

In 2005, Sergi Jord, Martin Kaltenburnner, Gnter Geiger, and Ross Bencina created Reactable, a tangible electro-acoustic musical instrument with multi-touch support[16]. In contrast to Audiopad, which required substantial sensing circuitry to function, Reactable used optical tracking to find objects on its surface. A projector was also used to overlay information, but since the light from the projector could interfere with sensing, the camera was filtered to only see infrared light.

Using an “amoeba” dot pattern of printed black and white fiducial markers, the Reactable software finds each token by looking for specific image contours. This information is then processed to uniquely identify each token (typically up to 180 different patterns, but the number can be increased by using a larger size tag), as



Figure 2-12: Reactable; image from the project’s website.

well as find its 2D position and orientation. While many projects before had used optical methods of various forms, the Reactable project used specially designed tags that were easy to process and also decided to make their tracking software open source.

The decision to make their tracking system open to the public for use and modifications had profound effects. Hackers who saw Reactable either online or in person began to look for ways to make their own similar system. Software was created to automate tag designs and other non-affiliated projects began to incorporate Reactable’s tracking system into their own. Most notably, NUI Group was formed and has since attracted hundreds of interested members, defining themselves as “[...] an interactive media community researching and creating open source machine sensing techniques to benefit artistic, commercial and educational applications.”

2.13 Mixed Reality Interface (2006)

In 2006, the Kommerz company showcased their Mixed Reality Interface and released it as a commercial product for navigating 3D spaces[39]. The MRI used a backlit tabletop with an embedded camera and sensed objects based on their particular color patterns. As shown in Figure 2-13, the system is split into two parts: the tabletop



Figure 2-13: Mixed Reality Interface used for viewing 3D models.

input and the vertical graphical display, showing the results of the user's actions.

The MRI not only opened the doors to tangible interfaces as a commercial product, but also offered new ideas about passively sensing an object's analog changes by recognizing differences in the object's color pattern. For example, when viewing a 3D scene, the user can move the camera around the space to change the view; they can also adjust the angle of the camera by physically tilting it and thus changing the colors of the bottom of the object.

Kommerz focuses mainly on letting users and designers navigate high-end graphical renderings with a simple interface. Although somewhat expensive, the MRI was a great step forward for tangible user interfaces, since Kommerz brought them outside the research laboratory as a commercial product.

2.14 Mir:ror (2008)

In 2008, Violet started producing a tangible computer accessory, the Mir:ror[40]. This USB-based device is a simple RFID scanner with edge lighting to act as a gateway between specially tagged physical objects and functions to be performed by the computer. Since the system uses RFID, the address space is extremely large (essentially every tagged object can be unique in the world) allowing applications

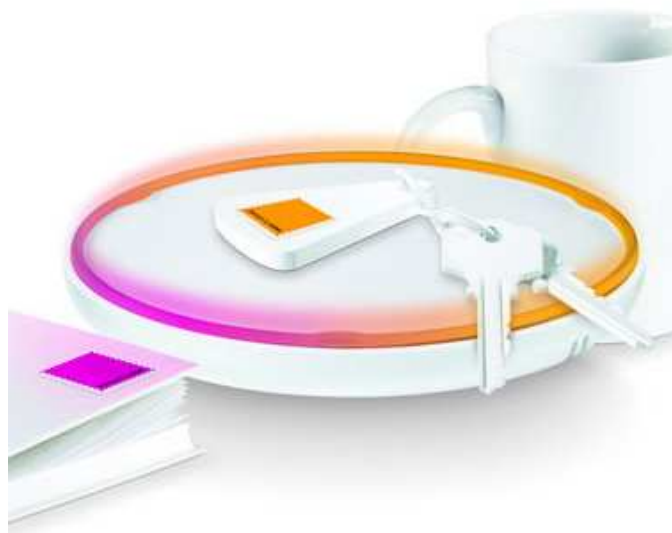


Figure 2-14: The Mir:ror RFID system for linking objects to actions.

that access network databases to perform similar actions on any system. The RFID tags are sold as stickers, known as “ztamps,” and can be affixed to most objects.

Unfortunately, the Mir:ror does not provide any position information about the sensed objects. It is designed to work very well as a binary on/off detector with any item that has been tagged, but does not sense any details about the location of the object on the surface or information (such as analog inputs from the user) about the tag beyond binary control.

Despite its limitations, the Mir:ror is an interesting venture into the space of computer peripherals as tangible interfaces. The price of the mirror is comparable to that of a wireless mouse, allowing it to potentially sit alongside the vast assortment of computer peripherals in an electronics store and catch the eye of everyday users.

2.15 DIY for CHI Workshop (2009)

In April of 2009, Leah Buechley, Eric Paulos, Daniela Rosner, and Amanda Williams led the DIY for CHI Workshop, bringing together builders from around the world who were exploring human-computer interaction through the lens of do-it-yourself culture and community[1]. Over 35 projects were featured, spanning a range of categories:







		
3D Motion Capture.....10	Fuse Beads I/O controller.....24	Command/Crap Chowder.....36
How to Make a Pinkeye Lens Attachment for a Handheld Camcorder.....16	Roomies Costume.....28	Striated Terebra on Earrings.....38
MacOS Footswitch from a Guitar Amp Pedal.....18	Hiigo Tablet.....30	
		
Make Skee-ball inspired Games Using Pizza Boxes, Party Favors, and a PC.....42	Change a Cloth Diaper!.....62	BiowatcherMap: Who Wants to Be a Molecular Meteorologist?.....76
Fabric Band Sensors.....46	Low-Cost Water Flow Sensor and Ambient Display.....65	Measure and Map Air Quality Pollution With Your Mobile Phone.....78
Gelatinic Ben-Response Computer Mouse.....50	Crap Packer Folding.....69	
Portabler from Old Newspapers.....52	A Simple Low-Cost Home VOC sensor.....70	5Mhz DNA Extraction in a Shot Glass.....83
HDD: Turning an old hard disk drive into a rotary input device.....56		

Figure 2-15: Some of the projects shown in the 2009 DIY for CHI Workshop; image from the workshop’s book of projects.

subversive technologies, working with constraints, on the cheap, and citizen science.

While the technical merits of step-by-step instructions, open source, and do-it-yourself projects are still heavily debated in the academic community, it is very clear that those involved feel empowered, impassioned, and enlightened; they are willing (sometimes demanding, even) to take on new ideas and challenge the limits of what is possible. The insights from their work are significant and, regardless of their individual academic contributions, there is much to be learned from their tenacity, creativity, and diverse perspectives on how to interact with technology and envision the future.

There is a long history of researchers exploring the complex range of possibilities for new human-computer interfaces, and recently, there has been increased excitement for researchers who take an open approach to designing and sharing new interface ideas. If an initiative can be constructed that continues building upon the strongest ideas from previous research while also taking cues from emerging open source projects, communities, and practices, the user experience could change drastically – for the better.

Chapter 3

What's the big idea?

3.1 Looking for the right mix

With a significant human-computer interaction problem – Why are tangible user interfaces still predominantly confined to the lab, even after 20 years of compelling research? – and a recombinant perspective, this research looks at new possibilities that could arise if tangible user interfaces were accessible (both ubiquitous and enabling) on a large scale via an open initiative called Trackmate.

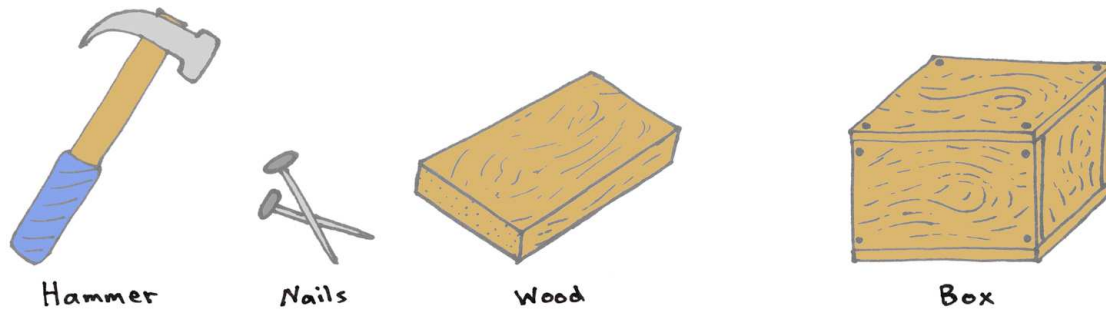


Figure 3-1: A box with the property of containment can be constructed from objects that do not themselves exhibit any aspect of containment; this is a form of emergence.

It is worth noting that the whole can be greater than the sum of its parts. For example, using a hammer, nails, and some wood together can create a box, giving rise to the property of ‘containment’ which none of the objects used to make the

box possesses on its own[24]. This kind of emergent behavior is, by definition, hard to predict, but there is much to gain from the careful study and experimentation of combining various materials, ideas, and methods to develop something that goes beyond the linear combination of the materials' properties alone.

Similarly, many projects and ideas have been woven together in Trackmate's research direction to explore new possibilities that may emerge when tangible interfaces can be made inexpensive, easy to build, and shared among diverse users on a large scale. Fitzmaurice's Bricks were the foundational building blocks to support ideas of mapping physical objects directly to virtual objects within the computer. Audiopad added another layer for which people of all levels of musical skill could enjoy using a tangible interface to combine abstract data (sound samples) together to form music. Reactable demonstrated a simple framework for other researchers to make multi-touch and object tracking tables, as well as making the tracking code (although not the music application) open source. Do-it-yourself (DIY) culture promotes ways to clearly convey step-by-step projects and disseminate new devices and technologies without needing centralized manufacturing. And finally, open source philosophies and initiatives spanning mechanical construction, electrical hardware, source code, and protocols have empowered users to become developers as a means of pushing the boundaries of existing systems.

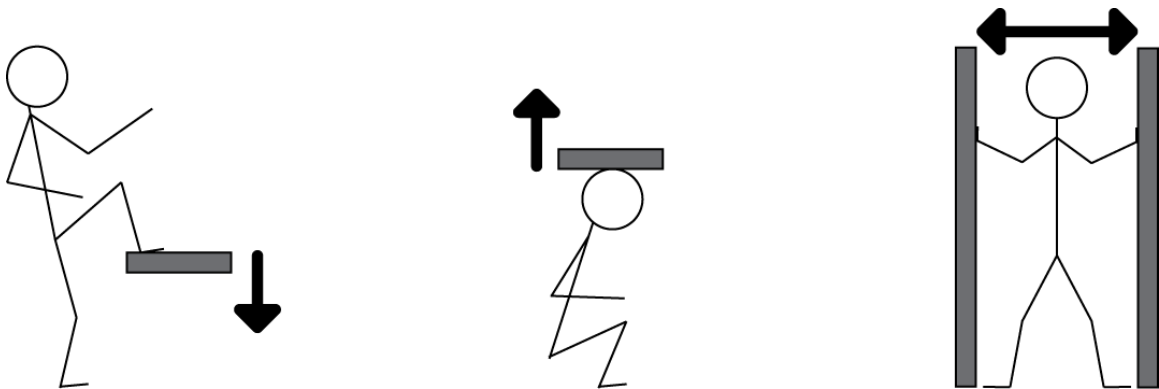


Figure 3-2: Lowering the floor, raising the ceiling, and widening the walls.

3.2 Lowering the floor

From ideas developed by Seymour Papert[29] and Mitchel Resnick[32], a low floor indicates that a system is easy for novices to get started using it. There are many aspects that can make a system difficult to start using; significant learning hurdles and expensive hardware have been common in past tangible interface research and have limited most people from being able to build and use the systems on their own.

To lower the floor further, we must make tangible user interfaces easier to build, install, and program. Step-by-step instructions, detailed documentation, and multiple entry points are all needed and correspond nicely with the DIY movement. Furthermore, the overall cost of new tangible systems should be drastically reduced such that anyone interested in getting started can do so. Additionally, some tools can be provided freely via software and webpages to allow curious, but uncertain, users a means to try before investing much time or money.

One good example of lowering the floor is the TORTIS system, a tangible programming tool developed by Radia Perlman in the mid 1970s[22]. TORTIS was comprised of cards and a set of large buttons for controlling a physical robotic turtle, enabling very young children to begin programming before they could type in a formal language on a computer.

Products such as Phidgets have also successfully pushed in this direction, allowing skilled computer programmers to easily incorporate hardware with their systems[6]. However, the floor for non-programmers to use the hardware is still quite high, and Phidgets main focus on generic USB sensing and control provides an explosion of possibilities that can be daunting to new users.

3.3 Raising the ceiling

Another important consideration for tangible interfaces is the height of the ceiling. A high ceiling indicates that a system scales with the user as they advance, offering increased functionality and complexity as projects become increasingly sophisticated.

Expert systems are rarely designed to be general purpose or simple to use. To keep the floor low while raising the height of the ceiling is non-trivial and requires careful thought to execute without substantial sacrifice for novices.

Without altering the floor, there are a few aspects that can be expanded upon to raise the ceiling for more advanced users. Tangible tabletop systems need to uniquely identify and detect objects. While simple TUIs require only a small number of objects (fewer than 10), designing the system to handle a much larger number of unique objects can allow it to expand beyond a single tabletop or location. On the physical side, allowing objects to communicate additional analog information is a common configuration for advanced user interfaces (such as knobs, sliders, or joysticks). Similarly on the software side, distributing all of the code as open source allows expert users and developers to modify aspects of the system; the computational side is then bounded only by the skills of the experts and the underlying computer itself.

3.4 Widening the walls

Finally, not only should a tool or system be easy to start using, as well as allow the user to grow toward more sophisticated tasks, but the range of ways the system can be used should also be diverse. Resnick describes these wide walls as “[...] technologies that support and suggest a wide range of different explorations.”[32] This is especially relevant for tangible user interface research, where most systems are vertically integrated (built to support one particular implementation from the ground up), and could greatly benefit from supporting a broader range of application domains.

By encouraging wide walls, new tangible user interfaces could allow users to improve more easily by using everyday objects and interactions with the system. Instead of producing a single beautiful demonstration interface for labs and museums, many applications should be developed to both illustrate and facilitate a wide range of uses for the system. The physical form of the system should also be broadened; there is no one perfect configuration, but rather, a spectrum of tradeoffs for different appli-

cations, environments, and users. For example, a small, portable configuration for a tangible user interface may be best for traveling and giving presentations, while a larger shared tabletop may be preferred in a classroom setting.

3.5 What Trackmate aims to address

Trackmate strives to lower the floor, raise the ceiling, and widen the walls of the tangible user interface by addressing the aforementioned issues facing current tabletop TUI technologies and research. However, as some tradeoffs are unavoidable, Trackmate focuses more on the floor and the walls than the ceiling, since HCI researchers are already working to raise the ceiling as high as they can with new technologies. Working towards new interfaces that can be simultaneously easy, powerful, and open-ended is likely to be a never-ending pursuit, but one this research aims to advance. The next chapter will discuss in detail the design decisions made for Trackmate and how they were subsequently implemented.

Chapter 4

How does Trackmate work?



Figure 4-1: Trackmate logo from the project’s website[21].

4.1 Overview of the Trackmate project

Trackmate is an open source initiative to create an inexpensive, do-it-yourself, tangible tracking system. The Trackmate Tracker application allows any computer to recognize tagged objects and their corresponding position, rotation, and color information when placed on a surface. All data is sent from the Tracker via LusiDOSC (a protocol layer for unique spatial input devices), allowing any LusiDOSC-based application to work with the system.

As shown in Figure 4-2, there are many components to the Trackmate system. The issues facing tangible interfaces are broad and complex, ranging from cost and system architecture to compelling application design and community involvement.

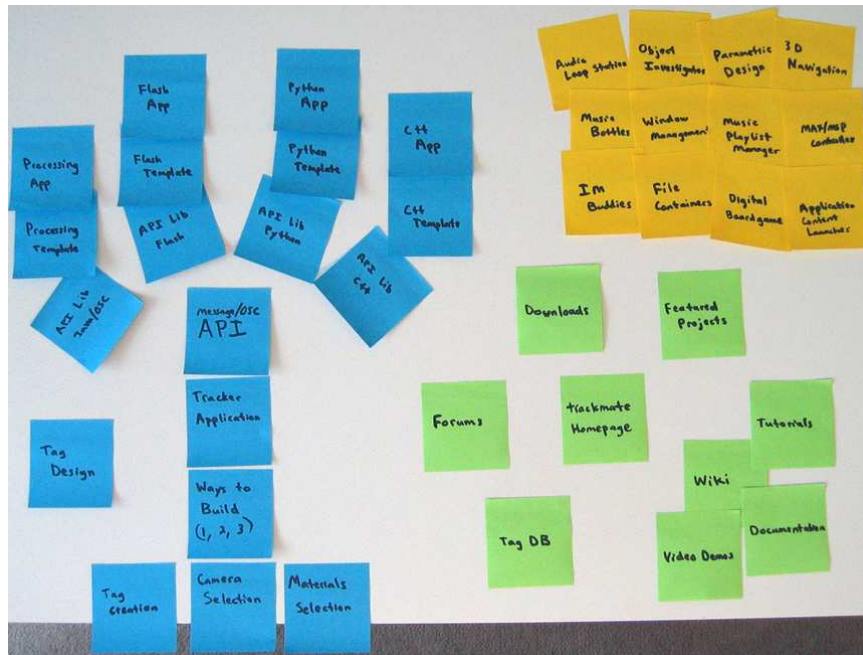


Figure 4-2: Developing Trackmate; Post-its were initially used to layout and organize the various components of the project. Technical infrastructure is shown in blue, user applications in yellow, and community tools in green.

Trackmate aims to take a large step forward by addressing as many of these issues as possible with a focus on learning from past projects and combining their various strengths via a unified large-scale initiative.

Tags for the Trackmate system allow for fast recognition, can be created from trillions of possible uniquely identifiable IDs, are small in size, and can be produced with any printer. Using any standard, low-cost webcam connected to a computer (Windows, Mac, and Linux system are all supported), the tracking software can easily be set up to scan for tags and transfer the information to desired applications.

To lower the floor for getting started, Trackmate does not require I/O coincidence, freeing the user from the constraints of expensive equipment (such as a projector or large display) and complicated setup procedures (such as aligning objects coincidentally with graphical representations or using infrared illumination). I/O coincidence can be achieved if desired, but it has been intentionally reserved for advanced users working on sophisticated projects who require a higher ceiling.

All code, content, and instructions for Trackmate are open source and posted to the internet for anyone to use, critique, and modify. The project’s website includes videos, documentation, step-by-step instructions, application downloads, and source code. There are also community tools such as forums and wikis to foster user communication and manage issues that arise as the project grows.

4.2 Easy-to-track tag design

Trackmate uses a small, specially designed circular barcode that stores information which can be quickly interpreted by the Trackmate Tracker application. The tag measures less than 2.5x2.5cm (0.95 inches square) and contains a six-byte unique ID (over 280 trillion unique IDs are possible), as well as a single-byte checksum for simple error detection. By using a circular shape for the tag, simple rotationally invariant algorithms can be used to find the center of each tag without requiring high-end processors or imaging equipment.

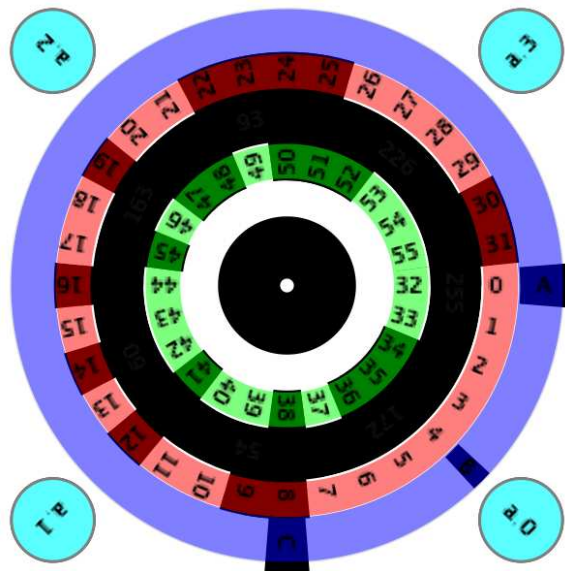


Figure 4-3: The Trackmate tag design, color coded for descriptive purposes.

Technically, the tag consists of a rotationally invariant pattern, three rings of data, and four analog color sensing zones. The outermost ring (shown in blue) is used to

to determine the tag’s orientation via three black segments within a white strip. The middle data ring (shown in red) contains the first four bytes (32 bits) of the tag’s address, encoded as short black and white segments. Similarly, the inner ring of data (shown in green) contains three bytes (24 bits) of information: the top two bytes of the tag’s address as well as a one-byte inverted checksum, used for error detection.

The four circular regions in the corners (shown in cyan) are optional and can be read as analog values (sensing the actual RGB color at the center of each region). For example, the color of a region can be mechanically changed when a user presses on the object, then read by the tracking software, and passed along to any application that uses the tag’s color values. By allowing for color inputs, tags can have increased functionality (including various buttons, knobs, and sliders) without the need for peripheral electronics.

The remaining black and white concentric circles are important for quickly finding the tag within an image as viewed by the Tracker. Since the circular rings appear the same regardless of the tag’s rotation on the surface, simple algorithms can be used to determine locations that are highly probable to be the center of each tag. The Trackmate tag is different from ShotCode[7], a popular circular database-backed barcode, since ShotCode uses a different graphical pattern, is larger in size, contains less information (256 times fewer IDs), does not allow for analog input, requires users to register tags (prohibiting experimentation with large-scale applications or alternate tag designs), and is specifically designed to work with mobile phones.

Trackmate tags are, however, not scale-invariant (they must be the correct size) or perspective-invariant. Despite the limitations, this choice was made for three reasons: the algorithm to detect the tags can be very simple and fast, it has good noise immunity (i.e., it will only detect tags that seem like a very good match), and it allows for high data-density (i.e., it can reliably read 48 bits of information in a small graphic). Further details about how the tags are actually processed and tracked can be found in Section 4.4.

4.3 Tag creator software

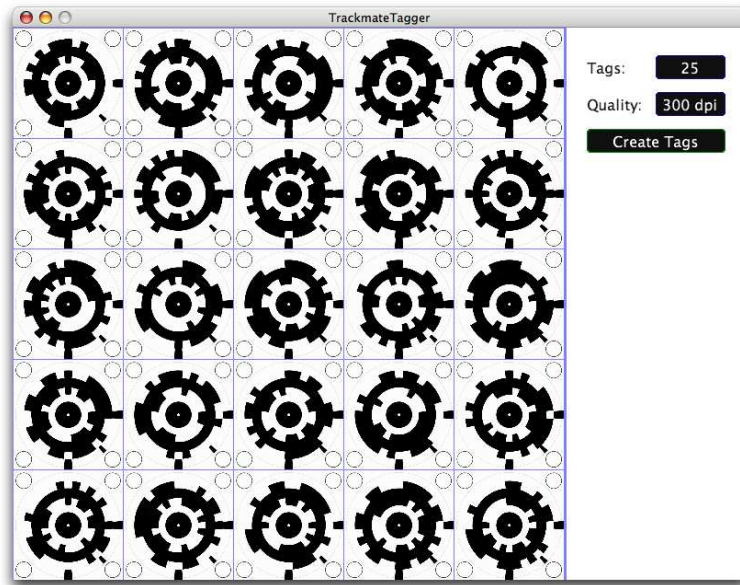


Figure 4-4: Screenshot from the Trackmate Tagger, a simple application that creates a random array of tags for printing.

The Trackmate Tagger application (written in Java/Processing) offers a simple way to randomly produce Trackmate tags in the form of PNG image which can then be printed. Tags can be created individually, or as a grid of 4, 9, 16, or 25 tags per image. Since the code is open source and easily modified in the Processing environment, developers requiring specific groups of IDs can produce them with little additional effort.

By making the tag creation available to everyone, users and developers can begin to work with large-scale applications that manage networked databases of information mapped to each tag's unique ID. Without enforcing segmentation of the address space, duplicate tags are likely to exist – this is an area that should be explored further in the future, but was intentionally left open (at this stage) so that users are free to develop the system as they choose. The Trackmate tag creation code is included for reference in Appendix B.

4.4 Tracking software

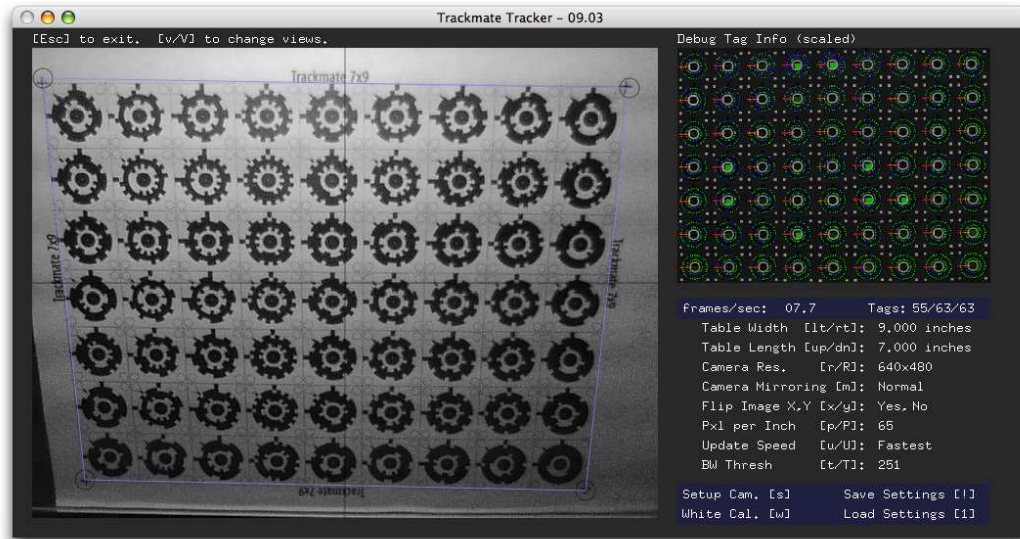


Figure 4-5: Screenshot from the Trackmate Tracker.

The Trackmate Tracker application processes images acquired from a webcam and finds all of the Trackmate tags visible within the specified frame. Each tag that is correctly sensed is then sent to other applications via LusidOSC, a protocol developed to work with a large range of tracking devices. The code is written in C++ and is designed to be simple (although lengthy), such that anyone interested in modifying or extending the application can easily do so. The complete tracking code can be viewed via the Trackmate website for reference.

Figure 4-6 shows an overview of the algorithm that processes each image frame of a webcam's video input and finds Trackmate tags.

First, a four-point homography is used to adjust the webcam's image for perspective warping. The homography corrects for distortion that occurs if the camera is not centered exactly under the center of the sensing surface, something difficult to avoid for most camera configurations. To calibrate the system and compute the homography matrix, the user places a sheet of tags on the sensing surface and clicks the onscreen image where the sheet's four corners appear. This perspective-corrected image is then used for the remainder of the image processing.

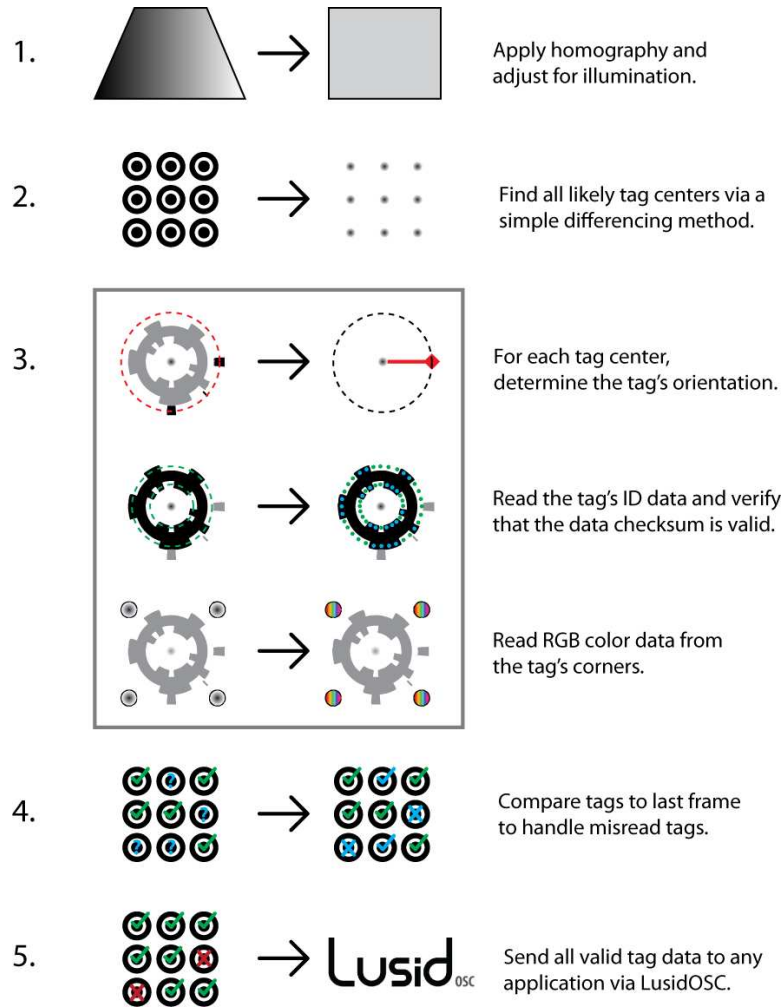


Figure 4-6: An illustrated overview of the Trackmate Tracker application's tag-finding algorithm.

Second, a simple differencing method is used to locate hot-spots within the image that are likely to represent tag centers. Since the tag contains rotationally invariant components (alternating black and white concentric circular rings), the image can be searched for a specific combination of pixel differences to find small regions that are very likely to represent the center of a tag. This is accomplished by subtracting the brightness value of pixels corresponding to locations on the white ring from the black rings, and then using the least of those differences (thus requiring a strong match to be considered a good point).

Third, each likely tag center is found by locating the brightest hot-spots computed

in the previous step. The algorithm then proceeds to read the pixels corresponding to the outer data ring (which is composed of a white strip with three small black segments) to determine the tag's orientation. If an orientation can be successfully determined, both inner rings of data are read next (consisting of six bytes for the unique ID and a single byte inverted checksum). If the ID checksum matches the tag's checksum data, the tag is assumed to be valid and the RGB color data is read from its corners. All valid tag information (including ID, position, rotation, and RGB color) is stored in an array for later use.

Fourth, the complete list of valid tags within the current image (a single frame from the webcam's incoming video) is compared to the previous frame. This enables misread tags in the current frame to be paired with valid tags in the previous frame (only considering likely matches that are within a specified distance). Since inexpensive webcams often produce noisy images in low lighting, and sometimes sporadically return invalid or blank images, previous-frame comparison caching helps to avoid false negatives (where an object is assumed to have been removed when it is actually still present).

Finally, the entire list of valid tags (including adjustments made in the previous step to allow for caching) are wrapped in an Open Sound Control (OSC) bundle and sent via the LusidOSC protocol to any application that may be listening for the object data.

4.5 The LusidOSC protocol



Figure 4-7: LusidOSC logo from the project's website[20].

LusidOSC is an open source initiative to define and support a layer for unique

spatial input devices using Open Sound Control. While inspired by TUIO (developed by Kaltenbrunner, Bovermann, Bencina, and Costanza and described in TUIO: A Protocol for Table-Top Tangible User Interfaces in 2005[17]), LusidOSC links many new and evolving spatial input devices (systems that uniquely identify objects such as markers, tags, regions, fingertips, etc. in physical space) with LusidOSC-based applications. The protocol layer is designed to support data from interfaces such as Trackmate, Reactable, Sensetable, and g-speak, and can easily be extended to support others.

LusidOSC aims to cleanly connect spatial interfaces with user-level applications via a simple, extensible protocol using a local or remote network connection. Additionally, LusidOSC can be used as a bridge to connect spatial input devices (such as Trackmate) to existing applications by mapping each object’s data to specified commands or actions. The protocol is currently in its first release and is designed to grow as new interfaces emerge and their underlying technologies evolve.

LusidOSC allows for flexibility within a single, standardized profile rather than requiring the creation of a new profile for each type of tracking system (as is the case with TUIO). This design choice enables LusidOSC libraries to remain entirely separate from the tracking system while also enabling certain functionality by default with any LusidOSC-compliant application. For example, tracking systems and applications are not required to have the same dimensionality to function (3D positions can be used in a 2D application, and 2D positions can be manipulated as 3D data on a surface). If, however, a large number of applications emerge that are clearly segmented based on particular tracking technologies or capabilities, future LusidOSC versions could support multiple profiles (as TUIO does), but only insofar as they are necessary, thus keeping library support and application integration as strong as possible.

Information about each sensed object is broadcast to applications via OSC (an abstract layer on top of UDP) and provides eight components that define it in the physical world: its uniqueID (u); position (x, y, z); rotation (a, b, c); and time (s). Regardless of the sensing platform’s capabilities, every object message has the

same fundamental data structure (plus additional space allocated for platform-specific data), allowing any LusidOSC-enabled application to function with any spatial sensing platform.

4.6 Protocol libraries

LusidOSC is built on top of Open Sound Control, which already supports a very wide range of libraries for most programming languages. With a small amount of additional structuring, OSC libraries can be extended to provide LusidOSC protocol support. By creating LusidOSC libraries, developers can quickly add functionality to their applications that map sensing and interaction to uniquely identified objects in 3D space.

A library for the Java/Processing development environment was chosen first because of its large community of programmers spanning a diverse set of backgrounds (graphic design, large-scale art exhibits, user interface research, prototyping with circuitry, networking between computers, 3D graphics, sound manipulation, and web applications)[5]. Furthermore, the LusidOSC Processing library makes it possible to add spatial input device functionality to existing applications with a small amount of additional code.

Figure 4-8 shows a simple example of how a program written in Java/Processing uses the LusidOSC library to listen for events that are triggered when object data is received. With just a few lines of code, the library can be imported, initialized, and mapped to desired functionality. For users who prefer not to use an event model, the current lists of objects can also be obtained directly (as shown in the `draw()` method, which paints a rectangle for each object based on its position).

4.7 Spatial applications

A “spatial application” is any program that uses information from a spatial input device (such as Trackmate). Each object’s 3D data is received by the application via


```

// we need to import the Lusid OSC library and declare a LusidClient variable
import lusidOSC.*;
LusidClient lusidClient;

void setup()
{
    // setup the processing display window
    size(320,240,P2D);
    // set the rectangle drawing mode to: center.
    rectMode(CENTER);

    // an instance of the LusidClient
    // since we add "this" class as an argument the LusidClient expects
    // an implementation of the Lusid OSC callback methods (see below).
    lusidClient = new LusidClient(this);
}

// within the draw method we retrieve an array of LusidObject
// from the LusidClient and then loop over both lists to draw the graphical feedback.
void draw()
{
    background(255);
    fill(0);
    // get the list of all objects that are currently present
    LusidObject[] lusidObjectList = lusidClient.getLusidObjects();
    for (int i=0;i<lusidObjectList.length;i++) {
        LusidObject lObj = lusidObjectList[i];
        // draw the object to the screen.
        rect(width/2 + lObj.getX(), height/2 - lObj.getY(), 10, 10);
    }
}

// -----
// these methods are called whenever a Lusid OSC event occurs.
// -----
// called when an object is added to the scene
void addLusidObject(LusidObject lObj) {
}
// called when an object is removed from the scene
void removeLusidObject(LusidObject lObj) {
}
// called when an object is moved
void updateLusidObject (LusidObject lObj) {
}

```

Figure 4-8: Example of simple Java/Processing code using the LusidOSC library.

LusidOSC and can be mapped to any functionality the developer chooses, including program interaction, abstract parameter control, navigation, data inputs, and spatial feedback.

To provide examples of spatial applications and mappings to incoming object data via LusidOSC, a set of simple programs was written to illustrate some of what is possible. Source code is available with each application, enabling users to derive their own code from the examples if they choose. All of the example applications are provided within the “LusidOSC Processing Bundle” (available on the LusidOSC website); users can download one file and have many programs with which to experiment.

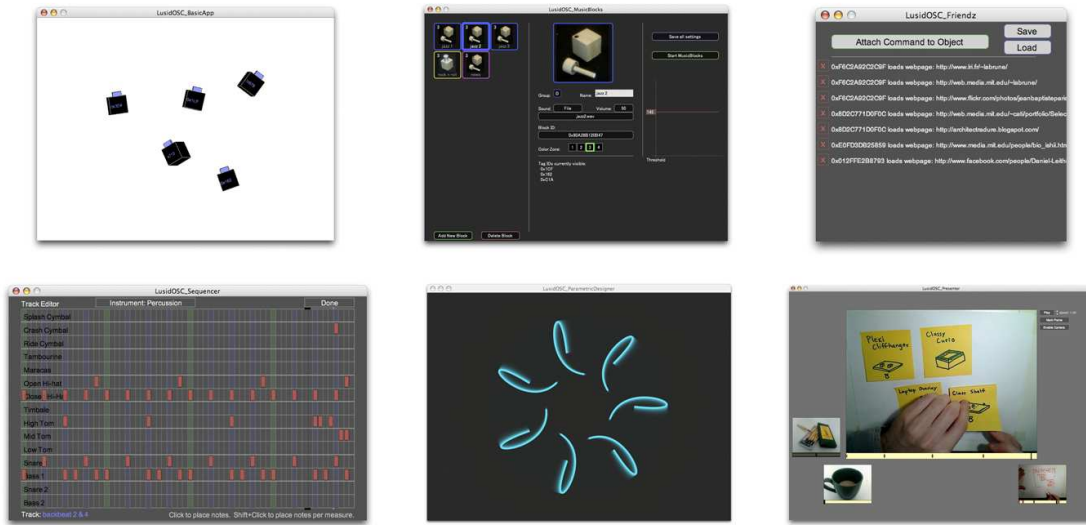


Figure 4-9: Six examples of spatial applications written in Java/Processing and included in the LusidOSC Processing Bundle.

Two of the applications (SimplestApp and BasicApp) are intended to serve as extremely basic templates for further development. Both applications paint rectangles to the screen based on the location of incoming LusidOSC object data. BasicApp goes one step further by also mapping the physical object's rotation to the virtual object, overlaying the object's unique ID, and drawing all graphics in 3D (shown in the top left of Figure 4-9).

For users interested in recreating past tangible interfaces as a way to learn or extend them in new directions, a version of musicBottles[13], called MusicBlocks, is also included (shown in the top middle of Figure 4-9). Using the color sensed at any one of the tag's four corners, a mechanical stopper or peg can be used to trigger music playback from the application. The threshold for triggering playback can be set by the user and includes adjustable hysteresis to compensate for minor fluctuations in color readings due to webcam image noise.

Friendz (shown in the top right of Figure 4-9) is an application for linking physical objects to digital actions and web content. An object's orientation can also be mapped to variations in an action, such as loading different web content based on which direction an object is turned. While mapping simple binary actions to objects does

not make for very rich tangible interaction, there is reason to believe that many users may appreciate it for simple, everyday activities (similar to Violet's *Mirror* product[40]).

Sequencer (shown in the bottom left of Figure 4-9) is another Processing application included in the bundle that allows for realtime music composition using both surface interfaces (such as Trackmate) and the existing keyboard and mouse interface for text entry and piano-roll note entry. Each object represents a sequence of notes, and the volume of each sequence can be mixed according to the object's position on the surface. Additionally, special objects can be selected to act as a tempo controller (altered by turning the object, as if it were a knob) or a proximity mixer (where each sequence's volume becomes inversely proportional to the distance from the mixing object).

Parametric control is often desirable for abstract data sets or tasks. The ParametricDesigner application (shown in the bottom middle of Figure 4-9) maps components of each object's position, rotation, and color information to parameters of a design space. The example shown maps the location (x,y) and rotation (angle relative to the z-axis) of three objects to a graphic design problem. This simple mapping allows the user to control nine parameters of a visual space with only three handheld objects and can be easily adapted to work with more complex parametric problems.

Integrating with common tasks that users already perform on their computer may provide a bridge to trying new spatial interfaces. The Presenter application (shown in the bottom right of Figure 4-9) was created to provide an alternative to traditional presentation tools via LusiDOSC. Any object can be mapped to a sequence of images; the object is then perceived by the user as a container of presentation content. Images within a sequence can be advanced or rewound by rotating the object clockwise or counterclockwise, respectively. Sequences can also be set to advance automatically until a stop point for showing stop-motion video.

The ensemble of programs provided in the LusiDOSC Processing Bundle is a first step toward giving users an idea of the forms that spatial applications may take. Users can immediately experiment with a wide range of example applications and

build their own ideas upon the examples when possible. The bundle should continue to expand as new mappings and interaction techniques are explored.

4.8 Ways to build the system



Figure 4-10: Five ways to build Trackmate. From left to right: Portable Plexi Cliffhanger, Classy Hardwood Curio, Overhead LCD Overture, Simple Floating Shelf, and the Basic Basswood Boxcar.

Figure 4-10 shows five different configurations of a Trackmate system, each with a sensing surface and a webcam. The left-most system places the camera on a portable tripod below a piece of frosted acrylic (used as the sensing surface) extending beyond the edge of a desk. The next configuration uses a picture frame as the surface, resting on a wooden box containing LED lighting, a mirror at the bottom, and a small downward-pointing camera. The third setup allows for coincident input and output (I/O) without requiring a projector by simply aiming the webcam down at objects that are placed on top of an LCD screen. The fourth composition is intended to blend in to a home or office environment by using a glass shelf as the surface and a desk lamp as the source of illumination. Finally, a larger version of the picture frame configuration was built with cost and simplicity in mind, using a compact fluorescent lamp and standard pre-cut sizes for all hardware. Trackmate has the capability to be both portable and easily personalized, encouraging users to make the system their own and integrate it within their everyday lives.

So why provide multiple ways to build the Trackmate system? Since the goal of this research is to bring tangible interfaces beyond the lab by making the technologies and ideas accessible to everyday computer users, there is no optimal single configuration. Instead, each user is likely to have different ideas about what she or

he wants to do with the system. Even though people are inherently creative and may conceive of new ways to build a system without several examples, demonstrating multiple configurations helps to catalyze the process.

It is also culturally important to providing alternatives for building a project. In particular, do-it-yourselfers often look at various ways to make a project before starting to build it themselves, frequently combining aspects of different projects depending on the materials they have on hand and what they are trying to accomplish. There is pride associated with finding a new way to construct an existing DIY project. Providing multiple ways to build the system from the start encourages remixing while also setting an initial standard upon which to improve.

4.9 Support for the community

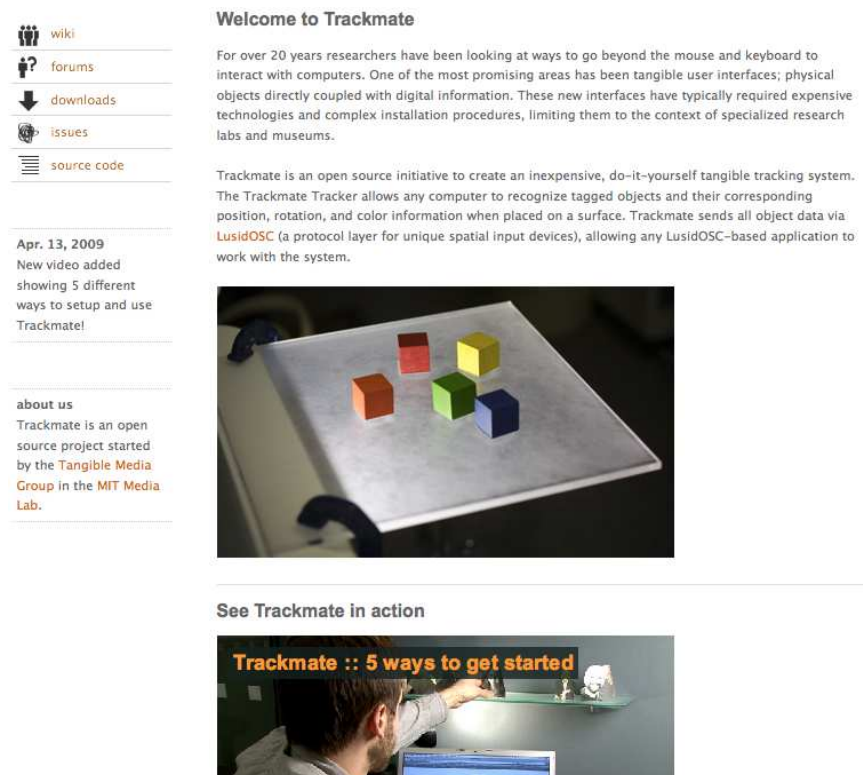


Figure 4-11: Screenshot of the Trackmate website.

The internet provides a powerful way to share ideas with very little cost to both

developers and users. However, simply posting content online does not guarantee that visitors to the site will be able to find what they are looking for or take the time to give the presented ideas serious consideration. For this reason, all of Trackmate's initial web presence was designed to look clean, easy to navigate, and provide compelling content (such as pictures, video, and screenshots).

Instead of trying to make the Trackmate website a portal which hosts all content locally, integration with existing communities was chosen wherever possible. Video content was posted on YouTube and Vimeo (both of which have large existing communities) to increase viewership and encourage responses or additions. Step-by-step building instructions were posted on Instructables, which not only opened the door for community exposure and input, but also provided a clean, well-developed format. While the website itself is hosted via Sourceforge (an established community of open source projects and developers), the page stands alone and remains Trackmate-centric.

The homepage also includes Trackmate-specific community tools, such as a wiki, forums, and bug tracker. These services are hosted locally on the Trackmate site since they are specific to the project and allow a new community of users to form. This is different from posting content within existing communities, because everything is centered on the project and enables targeted feedback within the group of users and developers. For example, the Trackmate wiki contains detailed documentation and guides for how to set up and use the Tagger and Tracker applications. Because anyone can change the content of pages within a wiki (regardless of the number of people who actually do), the wiki has grown to symbolize a spirit of openness and collaboration for online projects. Additionally, discussion forums provide a space to ask detailed questions, share new projects, and exchange ideas.

Chapter 5

What happened?

5.1 The first release

On February 5, 2009, Trackmate was first released to the public. The project's website was put online as well as the accompanying software, source code, forums, wiki, issue tracking, and setup guide. LusidOSC, the protocol that Trackmate uses to communicate with user-level applications, was also launched at the same time.

The goal was to be atomic in the first release, opening the entire initiative at once so that early-adopters would not be stranded between broken links or half-baked applications. Naturally, the publicly-constructed aspects of the site, such as the forums and the wiki, took more time to develop; it took many weeks for developers and users to start exploring the project and to begin building up an archive of questions and ideas via the public community tools.

The LusidOSC website was also important, since it contained the protocol specification and user-level applications capable of working with Trackmate. Additionally, a LusidOSC simulator was made available for download. This allowed spatial application developers to begin experimenting with code and debugging problems without building a physical tracking system.



Figure 5-1: Trackmate featured on Hack-a-Day, Create Digital Music, and Make Magazine’s blog.

5.2 Effects of press and new web content

The first major press that Trackmate received was on February 8th from Hack-a-Day, a technology blog where articles are posted about ways that people alter everyday technology to do new things. Writers from Hack-a-Day found the content via a step-by-step guide for setting up the “Portable Plexi Cliffhanger,” posted to Instructables the day before. While this sort of coverage was certainly welcome, it came a bit too soon; the tracking software was not yet sufficiently robust, the wiki had not been significantly populated, and the most compelling content (such as an in-depth video of various ways to setup and use the system) had not yet been added to the site . Although the article spurred on hundreds of users to visit the site, community tools (such as the Trackmate forums and wiki) were not highly utilized, and the overall involvement of the visitors with the project was very low.

The first video of the system fully working with an application – the LusidOSC Sequencer application was chosen because of its clear visual mappings and coincident sound – was posted via YouTube and Vimeo on February 16th. The video was then embedded in both the Trackmate homepage and the Instructables project to show users one possible use for the system. Overall, the response to posting the video was positive, with more people lingering on the site and starting to get involved with comments and questions.

On February 24th, Trackmate was featured on the front page of the Instructables website. This came as a surprise, but was well received since the majority of the

project was up and rolling with activity in the forums and a set of fairly comprehensive wiki pages to help users get started. With a video now part of the homepage, visitors spent more time on the site and seemed to engage with the possibilities of what a system like Trackmate could provide.

The next significant online press occurred on March 3rd from Create Digital Music, an active blog about projects by musician-hackers who use technology to create new kinds of digital interfaces and sounds. A writer from Make Magazine's blog saw the CDM article and posted a piece on Trackmate as well. Both writeups sparked quite a bit of activity on the website, and unlike the quick spike from Hack-a-Day, visitors coming from Create Digital Music and Make Magazine's blog navigated to the Trackmate site somewhat uniformly over a span of a few weeks.

Most recently, a high-quality video of Trackmate was added (on April 13th) to provide a more rounded perspective of the possibilities of the tracking system and the bundle of existing LusidOSC applications with which to get started. In all, the video shows five different configurations, five unique applications, and a demonstration of how to set up the system in less than three minutes' time. This was shown to numerous Media Lab sponsors and started many subsequent dialogs about potential applications and new directions.

5.3 Usage trends so far

Google Analytics[10] was used to study the overall demographic of users, traffic sources, and page statistics; the service is very reliable and attempts to eliminate double-counting of returning visitors. The first ten weeks after Trackmate's initial release (February 5th, 2009 to April 16th, 2009) were the focus of this preliminary analysis. This same methodology could be extended as the project continues, studying its impact and correlations with the release of videos, press coverage, and user innovations.

Figure 5-2 shows aggregate statistics of visitors to the Trackmate website over the past ten weeks from Google Analytics. All data was collected by adding a small

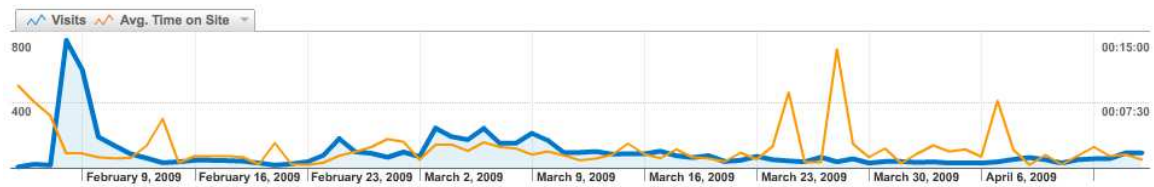


Figure 5-2: Graph of the number of visitors and their average time on the site for each day; source: Google Analytics.

piece of javascript to the Trackmate website that transparently collected non-personal information about each visitor. Two lines are shown: the number of unique visitors on each day is in blue, and the average time each user spent on the site is in orange.

Peaks in visitor traffic indicate significant events where many users checked out the website in a short period of time, often due to online press or new content. For example, the largest peak, on February 8th, was directly caused by an article posted on the Hack-a-Day blog. The effect of subsequent online press, such as Create Digital Music and Make Magazine's blog can also be seen as increased traffic in late February and early March.

The average time spent by each user on the site is important as an indirect way to visualize trends in visitor engagement with the site. Peaks can sometimes be misleading, often caused by users who leaving the page open in some form. However, some important information can still be derived from the average length of time users spend on the page. For example, users who visited the site in early February (before any videos had been posted) spent about half as much time as users in early March (who, incidentally, predominantly came from the DIY community and were able to view the videos).

Visitor information over the first ten weeks can also be broken down by country, as shown in Figure 5-3. The majority of visitors came from the United States; this is no surprise since the project was initiated in the U.S. and many of the descriptions, resources, and communities integrated with the site are also based there.

However, it is interesting to note that over 21% of visitors to the Trackmate website are from either Germany, the United Kingdom, or France. This is likely because of

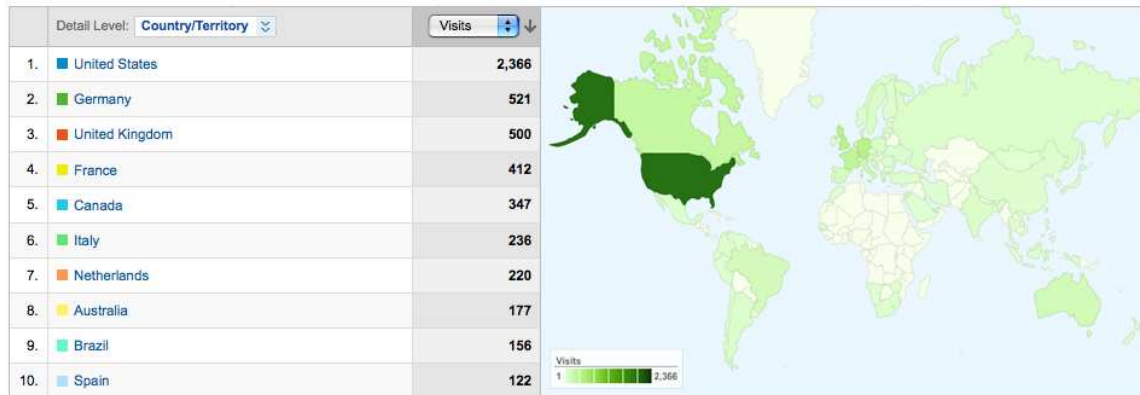


Figure 5-3: Demographic of visitors by country (top ten shown); source: Google Analytics.

the numerous design communities and hacker groups throughout Europe, as well as social connections of Media Lab colleagues to those groups. Having a diverse user-base is important to the development of new insights and ideas for Trackmate, and the worldwide spread of visitors thus far is encouraging.

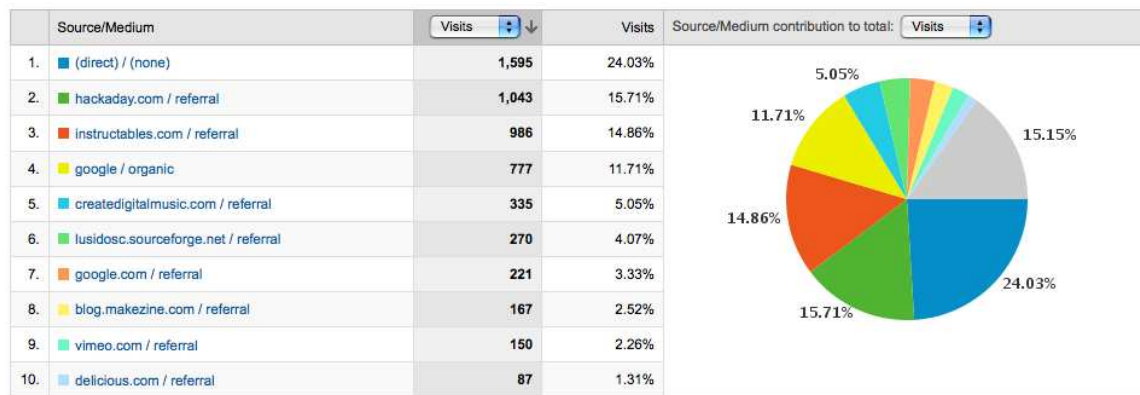


Figure 5-4: Traffic sources leading people to Trackmate's website (top ten shown); source: Google Analytics.

Figure 5-4 shows aggregate data about the traffic sources of visitors to the site. Almost one quarter of viewers found Trackmate directly (such as by bookmark, email, or typing in the URL); this occurs anytime the user navigates to the site without linking from another webpage. Hack-a-Day and Instructables each contributed about 15% of the visitors, although Hack-a-Day's traffic came in a short burst over a few

days, while Instructables led visitors to the site at a slower, but more consistent, pace.

The net result of searching via google.com also contributed 15% (derived by combining both the organic and the referral components of their service). This indicates that many people found the site by using a few keywords (such as “trackmate”) and did not arrive via articles from other sites. However, it is unclear if those users heard about the site from another person or if they were returning visitors who remembered the name after viewing an article.

Additionally, some broader perspectives can be gleaned from comments and feedback related to videos, instructional guides, and forum posts. In response to the first Trackmate video illustrating musical sequencing, users posted encouraging comments such as, “this is a perfect thing for me to make as a prototype for a much larger installation i’m working on.” Community members from Instructables mainly asked specific questions about how to build the system and suggested application ideas like, “Is it possible to integrate something like this with Audacity or something similar and use it as a sound system?”

Posts in the Trackmate forums mainly addressed problems with software such as, “I’m just using the built in webcam and holding up some printed out images to get a hang of the app, but am getting nothing in the tag center view [...] is there anything else i need to do?” Questions were answered as quickly as possible and always within 24 hours of posting; answers often received follow-up messages from the original posters with either further details of their problem or acknowledgment of resolution.

5.4 Users as innovators

User innovations can play an important role in the development of new technologies. Lead users – those who push the boundaries of what is currently possible to fulfill their own needs – are often the first to conceive of new ideas from the bottom up[41]. Two user innovations have already sprung up in the short time since Trackmate was first released. Seeing users expand the original project software and hardware to work

with their ideas is very encouraging.

One lead user, Jonathan Ward, extended Trackmate’s design to allow for switching tags on and off, and to no longer require a broad source of illumination on the surface. Instead of reflecting light off of black and white printed tags, Jonathan made a semi-transparent tag with internal lighting (thus illuminating the white regions of the tag when the user presses a button). In addition to the lighting, Jonathan also modified the size of the tags for his application, since he wanted to track objects in a larger space without requiring an expensive, high-resolution camera.

Another user, Oliol Pascual, developed a new form factor for Trackmate by embedding it into a glass desktop. Integrating Trackmate into an existing structure is an exciting possibility; spurred on by seeing photos of Pascual’s work via Flickr, the “Simple Floating Shelf” configuration was built (shown in Figure 4-10).

In addition to specific user contributions, many other ideas have surfaced through conversations and demonstrations with colleagues and visitors. One common idea was to include physical constraints that could guide objects (such as paths for sliders or circular rings to contain knobs), keying off of Ullmer’s Token+constraint paradigm[38]. Another interesting idea was to make the tags more physical by cutting them into a material; this could serve to add tactile feel to the tag’s ID as well as allow stamping of an ID onto other objects (such as documents), or even people. It is important that developers of Trackmate continue to watch lead users’ innovations and include them to help other users with similar constraints as the project moves forward.

5.5 Ideas for hire

A simple user test was conducted to see what first-time visitors to the Trackmate website thought about the project and its suggested uses. The study was held via Mechanical Turk[8][19], an online jobs-for-hire tool that allows workers to be paid a small amount for performing a simple task. Workers received \$0.25 for each suggestion (up to three suggestions per person maximum) and were asked to visit the Trackmate website, suggest what kinds of things they would like to use the system for if they

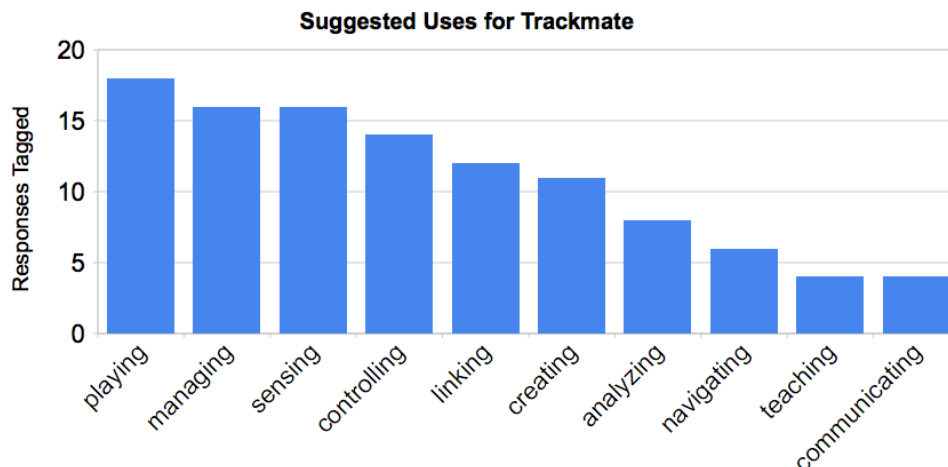


Figure 5-5: Number of ideas by tag from responses on Mechanical Turk.

had one themselves, and briefly describe how they would want the physical objects mapped to their ideas.

Over 70 suggestions were collected, and only a small number of responses (fewer than 5) were rejected, because a worker either did not follow directions or submitted the same suggestion twice. The results were then subjectively tagged as pertaining to different areas of interest to loosely categorize what visitors perceived to be the best domains of use. By using descriptive tags, each idea was able to fit into multiple categories, thus allowing categories to be freely chosen for the purposes of identification and not requiring them to be mutually exclusive. The complete list of valid data collected from workers via Mechanical Turk can be viewed in Appendix A.

As shown in Figure 5-5, most visitors imagined ideas for Trackmate that center on playing games, managing applications, sensing visual input, and controlling parameters within an application. Designing a game for Trackmate has surfaced in discussions many times, and this qualitative analysis clearly supports those suggestions. Also, since managing graphical windows is a common task with all modern operating systems, it comes as no surprise that visitors see potential in offloading interaction to physical objects. Other ideas included linking digital information to objects, facilitating the creative process, analyzing situations based on object locations, navigating virtual content, using Trackmate as a teaching tool, and communicating via objects

or their configuration.

5.6 What has been unexpected?

The general approach taken to foster an initiative for accessible tangible user interfaces has gone quite well. The rate at which people have visited the site, contributed to the community, and suggested new ideas has been slow, but omnipresent. There have been, however, a few unexpected twists in the project.

One major surprise was the impact of online press, such as blogs and forums, on the amount of traffic to the Trackmate website. Fluctuation was expected, but not nearly to the extent that occurred. Another unexpected observation was the importance of video on users' level of interaction with the site and how they perceived the project's capabilities. Combining these two aspects emphasizes how important it is to put up the most compelling content first. It is impossible to know for sure how the initial traffic from Hack-a-Day would have responded if a video showing many different uses for the system would had been included on the site, but there is reason to believe it could have yielded greater interest and deeper interaction. While the site was cleanly laid out and included links to applications, code, and forums, the lack of rich media likely deterred many who might otherwise have been interested in the project.

Another unexpected observation occurred when the project was shown to Media Lab sponsor companies in early April of 2009. Multiple versions of Trackmate were demonstrated and many of the LusidOSC applications were run, depending on visitors' interest. Surprisingly, the more complex and in-depth applications were not the ones that garnered attention. Rather, a very simplistic program, MusicBlocks, produced the most conversation and sparked intrigue among those who were seeing the project for the first time. Balancing the low floor for getting started with a high ceiling of potential interaction techniques and applications is a difficult endeavor and one that musn't be overlooked as Trackmate continues to grow in new directions.

Chapter 6

Conclusion. What's next?

6.1 Conclusion

Trackmate was released to the public in early of February 2009 as an initiative to take tangible user interfaces beyond the lab and into homes, offices, and tinkering spaces of everyday computer users. The project was designed to be an accessible (both ubiquitous and enabling) TUI, scaling to a large number of users with minimal hardware and configuration overhead. All aspects of Trackmate were made to be open source and designed to be community-centric; to leverage common objects and infrastructure; to provide a low floor, high ceiling, and wide walls for development; to allow user modifications and improvisation; to be shared easily via the web; and to work alongside a broad range of existing applications and new research interface prototypes.

The response to Trackmate suggests that users are interested in new possibilities for tangible user interfaces, but that the limiting factors may be beyond low-cost, ease-of-use, and application functionality alone. Trackmate's potential for success also hinges upon community involvement, perceived "coolness" or novelty (for press coverage), and lead users who can bring new ideas forward and show others that the system is dynamic and open-ended. The analysis thus far has only covered the project's first ten weeks, so it is too soon to tell if it will grow to a point that it can stand on its own without being cared for under the umbrella of a directed research

project. There is still hope, since even the most successful open source projects often take longer than a year to form a community and grow beyond niche users, given that they have a strong sense of purpose and diligent caretakers, both of which Trackmate strives to possess.

6.2 What to work on in the next 10 years?

Trackmate is intended to remain directly relevant for a somewhat short lifespan of three to ten years, allowing users to develop tangible user interface ideas before they are potentially produced on a large scale and made to be low-cost (like many other computer peripherals are today). The cost of electronic devices will almost certainly continue to fall (as it has done quite predictably for the past 30 years), bringing today's more expensive technologies within reach, while also further lowering the price of low-cost interfaces. These factors will soon draw more focus to two areas of research: bridges and footholds.

As described previously in Section 1.3, bridges create new metaphorical paths from one area to another that otherwise would have been difficult or impossible to traverse. Much work is still needed to make the transition from the current keyboard and mouse paradigm to a context where humans can more quickly and intuitively interact with digital information – and each other. Eventually, many of these bridges may be formed by companies who have a vested interest in pushing their new interface technologies to succeed in the marketplace, but do-it-yourself bridge builders have a unique opportunity to create their own structures that rest on non-commercial motives and think both more critically and more creatively about how to elevate users from their current computing experience to something substantially richer.

Footholds for allowing users to more easily climb up difficult technical or conceptual terrain should also be explored further as TUIs move forward over the coming decade. Whether it is sequencing along a progression of increasingly sophisticated applications, interfaces, or technologies, many users will greatly benefit from a scaffolding that enables step-by-step movement toward a higher goal. Even expert users

who do not otherwise require the scaffolding may appreciate it momentarily for help as they work their way to the top. In the coming years, researchers should be careful not to overlook the need for support structures (explicitly, but also implicitly throughout their designs) in addition to interface advancements and their underlying technologies.

6.3 What to work on after the next 10 years?

The technologies likely to exist in ten years are beyond anyone's guess and cannot be characterized in concrete terms. Breakthroughs in artificial intelligence, robotics, networked computing, 3D displays, projection devices, and object tracking could all profoundly impact the future of human-computer interface design. However, it is still worthwhile to predict how new technologies will change user interfaces, and more importantly, what will remain essentially the same regardless of future progress or discoveries.

The scale of objects and interfaces is predominantly dependent on the size of the human body, not the technology within them. Interfaces that fill rooms may someday fit in the palm of a hand, but the fundamental modes of interaction are unlikely to get much smaller than that. Thousands of knobs and sliders, each the size of a grain of sand, would be inherently difficult to use. Instead of merely shrinking, new interfaces may add dynamic mappings and functionality to the interaction. Since Trackmate works with objects on the scale of the human hand, lessons learned about interaction with such a system are applicable to future interfaces.

As much as people may immerse themselves in virtual environments, all humans share common ground in the physics of the real world. Reality-Based Interaction[15] is a meta-level observation indicating that the most compelling interfaces overlap significantly with intuition about the real world; they may build upon our environment in very abstract ways, but it is human nature to seek grounding. Interfaces of the distant future may be radically different from interfaces today, but it is likely that they will still be rooted in physical metaphors, utilize persistence, and draw upon

invariant aspects of the human experience, ends to which Trackmate could potentially offer valuable means.

Furthermore, humans are social, and interfaces will likely evolve to focus on communities, regardless of their particular technologies. This prediction is derived from two observations: users are becoming more involved with online communities as their primary focus; and the user interface (not the performance of the computer or network) is quickly becoming the bottleneck between the user and a world of friends, colleagues, jobs, recreation, information, and means for self-expression. Applications could be created for Trackmate that offer new interfaces for community interaction as a way to explore future communications tools.

Finally, the distinction between what is virtual and what is real is likely to become increasingly blurred. Telephones, music players, digital cameras, computers, and many other electronic devices continue to populate our physical space, as well as their own. Digital bits exist in the physical world, and it is only a matter of time until they are considered one with their physical counterparts. This has significant implications for interface design as integration occurs in both directions; the real world is recreated within virtual spaces (such as the new field of X-Reality) and virtual representations gradually become more real[42]. Tightly coupling the digital and the physical has long been the dream of tangible user interfaces, and the future looks bright for some form of TUI to virtually become our future reality.

6.4 A note about complexity

Much of interface design can be boiled down to managing complexity. Initially, complexity is managed by pruning away every aspect that is unnecessary to the system's most important functions. But eventually, users demand more. The simplest interface is not always the best when experienced users are in control. A pilot would not fly a plane with a keyboard and a mouse, even though it is completely possible (as made clear by numerous realistic flight simulator applications). More knobs, sliders, buttons, and displays can actually lower the overall complexity of an interface because

all of the parameters can be spatially mapped; instead of digging through menus to find particular controls, the physical location of a setting can be remembered. Additionally, many users can easily work together because all actions are visible and multiple unique functions are not combined into any single interface.

This trend is prevalent in almost every other complex system humans have constructed. From airplanes and power plants to recording studios and factory floors, making interfaces spatial and physical allows users to build intuition about each part's position and purpose. So why not the computer? Is the keyboard and mouse superior to the intuition that people have developed naturally? Or are computers still in their infancy and awaiting the next big user interface revolution?

Tangible user interfaces have the potential to be part of the computer's transformation from a basic, functional machine into an intuitive, expressive, complex system unlike anything seen before. Trackmate aims to provide both a bridge and footholds up to something much richer than the current binary-switch keyboard and remote-control mouse can offer. Trackmate is not the destination, but rather, a means to allow everyday users to explore beyond the confines of the current computer interface and open up new tangible possibilities.

6.5 A final challenge

In closing, I propose a challenge to future researchers of tangible user interfaces: To greatly reformulate the computer's user interface, why not use the very interfaces being researched in the pursuit of such a goal? Just as Doug Engelbart was able to create a revolution in the way users thought about and interacted with their computer by bootstrapping (using his team's system and its developers to both evaluate and improve the new system), taking the same approach with tangible user interfaces could enable a vast realm of new possibilities. History is full of inventions like Dvorak layouts and Betamax videos; there is no guarantee that users will adopt new technology just because it's technically superior. To truly advance the state of tangible user interfaces, the community needs to prove that such a system can achieve drastically

different results. Bootstrapping TUI may be the only way to exponentially advance the interface's abilities while also demonstrating first-hand its enormous potential.

Appendix A

Mechanical Turk User Data

WorkerId	Time(sec)	Idea	Mapping	Tagged As:
AZ7OF1IHH3E7W	56	Closing, Dragging, Dropping, Moving	The Trackmate Tracker reads Trackmate tags (by processing images from a webcam) and then sends the corresponding data to any spatial application via Lusi-dOSC. The Tracker is easy to setup and provides feedback helpful for debugging your system.	managing
AZ7OF1IHH3E7W	174	Navigating, Bookmarking, Multimedia button options	Trackmate is an open source initiative to create an inexpensive, do-it-yourself tangible tracking system. The Trackmate Tracker allows any computer to recognize tagged objects and their corresponding position, rotation, and color information when placed on a surface.	managing, navigating
AUHUK2MIJFUA	66	Have more than one mouse cursor	Like a mouse but with more cursors	controlling
AM5F8FEZAL0BV	65	teaching sign language	I would want it to monitor hand movements (perhaps via gloves) and interpret the data into sign language—to better learn it.	teaching
AM5F8FEZAL0BV	144	Composing music—also conducting it, using usual hand movements.	Perhaps through use of gloves attached with motion sensors—these would relay information about movements to a computer program playing music, and adapt accordingly.	creating, controlling
AM5F8FEZAL0BV	110	Charting troop positions and movements.	Various physical actions would be linked to troop commands/movements. Troops would be represented by objects.	analyzing
AJE8VXZUFKR4C	485	I think the ones that can benefit the most from this interface are, actually, children.	The same few (up to the 280 trillion) pieces could be used with several puzzles for kids.	playing, teaching

AG579RN4237EC	128	Bookmarking, Tracing, Opening any new Application	The mapping of the physical objects to desired actions within an application using Trackmate uses a small, specially designed circular barcode that stores information which can be easily decoded.	managing
AG579RN4237EC	232	Tracking, Viewing, Clicking, Dragging, Dropping, Maximizing, Minimizing, Restoring, Closing	We can map the physical object to desired actions by means of one to one mapping. For this we have to specify the superiority of each and every actions that are involved in the component.	managing
AG579RN4237EC	163	All type of GUI Application operations. New tab operation for browser. Navigation function.	We can map the physical objects to desired actions by means of lot of application functionality available in the language. It allows any computer to recognize tagged objects and their corresponding position, rotation, and color information when placed on a surface.	managing, navigating
AEY7WLB4OYZA0	595	I would like to play ping pong.	Trackmate Tracker allows any computer to recognize tagged objects and their corresponding position, rotation, and color information when placed on a surface. Trackmate sends all object data via LusidOSC (a protocol layer for unique spatial input devices), allowing any LusidOSC-based application to work with the system	playing
AEY7WLB4OYZA0	96	I would like to use it along with online games. where I could play games against other people using the interface.	Trackmate Tracker allows any computer to recognize tagged objects and their corresponding position, rotation, and color information when placed on a surface. Trackmate sends all object data via LusidOSC (a protocol layer for unique spatial input devices), allowing any LusidOSC-based application to work with the system	playing, communicating
AEY7WLB4OYZA0	228	To illustrate theories. It can show group dynamics and how the objects interact with one another.	Trackmate Tracker allows any computer to recognize tagged objects and their corresponding position, rotation, and color information when placed on a surface. Trackmate sends all object data via LusidOSC (a protocol layer for unique spatial input devices), allowing any LusidOSC-based application to work with the system	teaching
A7TZL6F8HAFUU	313	Play building with blocks.	Cable line maybe.	playing

A3P9TNE2UMUNAR	188	Tool setting. Function Button allocation. Scrolling	For over 20 years researchers have been looking at ways to go beyond the mouse and keyboard to interact with computers. One of the most promising areas has been tangible user interfaces; physical objects directly coupled with digital information. These new interfaces have typically required expensive technologies and complex installation procedures, limiting them to the context of specialized research labs and museums. Trackmate is an open source initiative to create an inexpensive, do it yourself tangible tracking system.	managing, navigating
A3P9TNE2UMUNAR	149	Clicking, Navigating, Bookmarking	Trackmate uses a small, specially designed circular barcode that stores information which can be easily decoded by the Trackmate Tracker.	managing, navigating
A3OXBUEYMV06FT	65	music recording/working within a DAW	blocks could be mapped to knobs in the software, turned to adjust parameters. the position of blocks could be also be used to control volume faders	creating
A3NUPM6XLEF9J9	671	it would be coll if you can use it as onebig touch pad capable of handling multitouch. it can also be used in games such as chess, some people like playing with real pieces but also like playing with people on the internet.	you can have a small infrared beam that shines across the trackpad and a mirror so that the ir light bounces into an reciever. when an object is place on the trackpad in will disrupt the beam letting the computer know where the object is on the trackpad. another way is to have a small electrical grid running on the surface of the pad(not enough to kill you). since the majority of the things your going to put on it is non conductive when an object is place on the pad it will break the current on a part of the pad which the computer can figure out where it is.	playing, communicating
A3J985EPA8I8J4	96	May be it should help us doing some work based on our instructions.	Based on the application and the required of the user to have the desired action, may be for the specific work.	analyzing
A3HGHDBNYPHXXI	528	Pick up patterns, colours, shape from external objects, increase or decrease brightness, speed of transition etc	By using pictures to create a movie or using some digital photo data to create a movie. The computer can process and record the transition.	sensing
A3HGHDBNYPHXXI	176	To play games involving placement and strategy with army and components or a chess or a snooker game.	The computer can build up the terrain , landscaping and advancement and the interface to the human can be tiny dolls which can be moved and the changes are reflected on the computer.	playing

A3HGHDBNYPHXXI	236	A work out program that tabulates calories burnt, muscle conditions, blood circulation, fatigue level, sugar level, etc with every move of the human body.	The human body can be attached with sensors which detects signals and transfers it to computer through wireless interface and the computer detects every move and collects and organizes data and suggests the ideal work out routine	sensing, analyzing
A3FLPOAD0RZJH2	210	Help me find my phone, wallet, keys, etc. among my pile of clutter.	The location of the object and its position will be shown on scale with the computer screen.	sensing, linking
A3EBVT79QHA20X	755	I design documents, it would be helpful to be able to physically move objects that correspond to the location of data on a dynamic document.	The trackmate could be used to design video games, that employ items of a given size interacting. The trackmate can produce better 3D emulation.	manipulating
A3DQBO9UDJOCYD	620	I would like objects to be able to quickly open programs that I use a lot. It would also be cool to have objects for bookmarked websites to open them fast. I'd really like it if I could use a pen like object to write and draw directly into the computer.	For opening programs or websites, it could be as simple as putting the object on the trackmate to open it, and removing it to close it. Moving the object to the top or bottom could maximize and minimize the current window. If video or audio is playing, moving an object from top to bottom could adjust the volume and side to side could adjust the position, to rewind or fast forward.	managing, navigating, sensing
A3CQDPRMBHZD48	191	I would imagine possibly some actions for games. Possibly like picking up objects or moving objects on screen. Also, maybe using it as a volume level adjuster in multi track recording. Other more simple ideas would be drawing pad or signature pad of sorts.	Up and down movement of objects could be volume level adjustments on the screen. Also pertaining to games, some sort of aim zeroing system. I can imagine probably some flash games taking advantage as mapped objects in real time get applied to the games. Maybe like a puzzle of sorts.	manipulating, controlling, creating, sensing
A393XVIDYIN3U9	178	I would want it to help with photo production. Not sure how it could be done, but a way for an artist to further expand on areas that have never been created.	I can see this being useful with creating three-dimensional objects into a two-demesional computer screen. If the object could have different types of forms that would allow the computer to read it in a three dimesional form.	controlling

A35JEX1CTHJ33Q	1530	<p>1. Spatial planning, such as room planning. That would be particularly useful here in Europe where everyone needs to plan and by a new kitchen each time they move.</p> <p>2. I'd like to be able to take photographs of objects and move them around in both 2- and 3-dimensional space for visual planning. I'd like to be able to attribute measurements or other non-visual specifications to those objects to determine fit. And I'd also like to be able to map the objects and their attributes to other data, such as specifications.</p> <p>3. I'd like to use it to create flow charts, diagrams, process maps, use case scenarios, decision trees, relationship maps and such. Again, I'd like to be able to link each element to additional data.</p> <p>4. I think it would be useful for planning things like traffic patterns. This is a vague notion in my head, but one that I think would be made easier with a movable interface like this.</p> <p>5. I can also see some application for things like choreography and stage direction.</p> <p>6. I would like something like this for quickly automating file structures on my computer. Specifically, something that would allow me to sort and clean up my desktop and files based on a desired information structure that might be unique to a particular set of files or a particular project.</p> <p>4. Obviously, it would be a great interface for multiplayer gaming where fast action or complex movement is required.</p>	<p>1. I would include an optional feature that allowed me to track my movement of the objects and draw those paths on the screen.</p> <p>2. I'd include a feature that would allow me to ask the computer to suggest more efficient placement of objects in certain kinds of applications.</p> <p>3. I'd want to be able to rewind to any point in a previous version and have the screen tell me how to reset the objects on the input pad.</p> <p>4. In certain scenarios I'd like to optionally set tolerance thresholds that would, for example, allow me to ask the computer to warn me when two objects were in conflict with each other for some reason. For example, in the above-mentioned space planning scenario, if I placed an electrical outlet too close to a water supply, I'd like to get an alert. I'm sure there are hundreds more desired actions I could think of more if I really thought each of the above scenarios through their entire process.</p>	analyzing, controlling, creating, linking, managing, playing
A339YTJ2ZQFH84	130	I work in tech support. I often have to work with multiple customers at the same time.	It would be great to be able to drop an object on the board that is prelabelled for each customer. Once placed on the board, it would start the billing clock. Once the work is finished I could lift the customers piece and it would stop the billing and automatically calculate the billable time.	managing, linking
A339YTJ2ZQFH84	126	It would be awesome if a game could be developed.	Where you move your pieces would change the battle on the screen. Maybe each piece is a warrior and you can move them on the battlefield? Awesome technology.	playing

A339YTJ2ZQFH84	40	I want to control all the devices in my house.	You could use a piece to adjust the lights in one room or all room. Turn on a TV in a room by placing a piece on the board. Assign a movie to that piece. Same idea with a radio. Turn on sprinklers or open a garage door just by dropping a piece on the board. So many possibilities.	controlling
A31Z34DH99FNPC	259	If I were to give a good idea for this... for a computer that needs to be secure you could make those items as some form of failsafe to make sure that the owner was the one accessing the computer... while if you look at the items arranged on the desk it would simple look like some form of decoration	I value privacy... so I would set certain applications and websites I might have open to a block or item... and all I would have to do is slide the block away to hide the application or so from eyes of anyone who walks into my room or workplace	controlling, sensing
A31G4O708OLU9D	907	Musical composition (obvious from the video), graphic art (data brushes), (Following are some ideas in general, not just for me), Making computers accessible to the handicapped, Creating taxonomies - this looks like a cross between Wacom and XML, Competitive games played within a room whose walls are trackmate interfaces, Web equivalent of these games (a la second life), (and now, for a walk on the wild side ...), make the interface two-way, so the computer could cause physical objects to move, make the interface three-dimensional, with light-nodes that could be manipulated by specially-made finger-pad interface units and positioned in three-space to indicate (frinstance) musical tonalities. Then, by combining these two, you could have computer-generated music played in a three-dimensional volume, much like the screen-savers look except in 3D, OK, my brain's fried now. I could probably come up with lots more but it would take a greater familiarity with trackmate first.	x- and y-axis variables assigned according to type of data	creating, controlling, playing, actuating, sensing

A31FSDLW6ZN5LC	61	Games, something as simple as chess up to modern turn based or real time strategy games.	Having game pieces with track pads on the bottom would bring an new interactive feel to on-line multi-player games. Moving pieces on the tracking board giving the sense of an over-head strategic map with close-up graphics on the screen. Turning the pieces to operate different functions.	playing
A30U18M7PKRCV0	845	Online games, particularly tabletop in nature (ie checkers, chess, table top strategy such as warhammer minis). Combine with a projector or LCD to display my opponent's pieces on the glass. Virtual turntable/audio mixing. Would allow me to set up my own control layout. Combine with IR/laser/whatever interface to make each block a mountable hard drive that will show up on my desktop in the same spot that I set it on the glass. Like a usb flash drive with an optical interface. Let me set an object with a UPC on the glass and have it get info on it. Use objects to open files and manipulate the image in a 3D modeling/printing program. Example: When I turn the army man on the glass the army man on the screen turns to match. I can then send that to my 3D printer and have a new army man.	The setup should be completely customizable to each persons' preference. Depending on the task, I might want to have just one object in the center which will be rotated. I may prefer rectangular jenga-like blocks set up like a piano keyboard for a virtual piano. Give me a stylus and i can use it like a drawing pad or for photoshop work.	playing, creating, managing, linking
A2ZAEPBIZTK1MB	333	Why not make them work also with the other non-LusidOSC based networking applications ?	Sensor can on the other hand be used to detect movements of such objects and thus can be used as a tracker too.. Same in case of your trackmate.	sensing
A2S0HI89KVC9ZQ	345	Business Cards –003E It would be great if Trackmate could recognize the data and store it in the computer.	Are we talking about variables such as size, shape, color? If that is the case, black font type over light background should probably be easily recognizable.	linking
A2N3V8S5LXTV21	35	i can't think of any. the problem i guess with the application is finding what it could really be used for. why would somebody need something physical if it could already be stored and/or customized datawise? i'm seeing that gamers could use this perhaps to simulate physical actions on a certain game.	cameras	playing
A2LZ9KG42SZ3I0	645	It would use it to help produce music and would be a good media to use is a drawing pad.	I would use bluetooth to connect them and the computer would respond to the trackmate in real time.	creating, sensing

A2GANZ92RS0E3D	724	The system could be used for playing games like chess or checkers against the computer or against another remote human player. The system could also simply read a chess board being used by two human players and suggest moves or predict who will win a game based on the pieces on the board and their relative positions. (A human would have to move pieces on the physical board based on the computer's or remote human's move choices unless there were a board that could take instructions from the computer and move pieces itself.)	Whenever a piece was moved on the physical board by a human, its relative position and whether it had captured another piece would be updated in the computer's memory.	playing, sensing, analyzing, actuating
A2DXIZ55K43DR2	262	Type the information like writing a letter much faster. Here using objects to recognize the writing style.	Based on movements of my hand for writing the words. Objects are linked based on their position and size of them.	sensing
A2DLNQGTALP2M8	392	hmm recommend beats or sounds that sound well together, or suggest an appropriate genre after you make a song..you put a circle	you put a circle on the Track mate and label it. That circle moves up and down to control volume and other circles can be placed to create more sounds at once, knobs that turn can increase or decrease the tempo as well.	analyzing
A2A4OZEX98MVZS	2665	- use it as an input device for gaming, therapy tool for developing hand-eye coordination	Gaming: for a flight simulator, objects could map to different controls of an airplane, or for a puzzle game, moving pieces. Therapy: use a simple interface to teach a child or a person who once had poor vision the connection between moving an object and the reaction of the computer	playing, teaching
A29U8EMOMRFV2N	154	Be able to type by moving an object, not just pressing keys.	As i said, i would like to use the objects to somehow become a replacement for the standard keyboard.	sensing
A27IINK80IAJ5W	410	I could see it useful to play a very realistic game of chess with someone else.	Each piece would correspond to the different pieces on a chess board and the surface of the tablet would be divided into the squares of a chess board. A knight would then move in the L shape of the Knight and a Rook would move across and down , etc.	playing, communicating
A248XNL650JMTB	429	One of the New task may be that it should help in getting the actual dimension of the objects along with other information.	Objects can be linked by using special sensors, image processing Techniques etc.	sensing
A21EIQJKROG050	86	Math, website building, and audio playback.	A simple system that would be a one touch button system so that everyone can use it.	analyzing, creating, controlling

A1YEXP2UXRMA2I	180	Inventory managment, tracking shipments of items (i.e.barcoding system), digital tracking to have on eyes of sensitive shipments	link Computer to the item with a tracking device or bar code reader similar to how the military tracks shipments	managing, linking
A1W6B93TS8A38R	312	It should help me to tag the documents and files on my computer.	The Trackmate Tracker uses a webcam to sense and make a valid relation/link between the object and the thing happening on the computer.	linking, managing
A1VOTOGS2SN3MU	589	I see no value in such a device unless I would use my finger as a mouse on my leg, cause that would be much more comfortable.	I would need a whole lot of objects... If I were to have a cd in a certain spot Media Player Classic would come up. If I was to move a tape to a spot it would bring up a record menu. I don't get why anyone would want to use this. RFID is the way to go to 'track' stuff imo.	controlling
A1VJKPQDRP838K	53	you can use it to build a model ship or car.	I would lookup the instructions on how to build a ship. And place each piece in the correct location.	creating
A1VJKPQDRP838K	168	I would use it to play games against the computer. Since it can locate position, rotation, and color information when placed on a surface. Like chess or checkers	I would put a chess board on the table that the computer can locate then set the pieces in proper positions.	playing
A1U0Y8ZMS7GYIE	272	Composing my own songs and then later printing sheet music for it.	It appears to be a special board that can pick up on the location of objects and assign tones. These tones go to the memory and can be manipulated by moving them around.	creating
A1S6DLZLD2K531	453	I would experiment with tracking insects or small animals (humanely of course) and their interaction with one another by possibly assigning sounds to different behaviors.	You would generate print tags using the Trackmate Tracker and put the tags on objects you want to track.	sensing, controlling
A1RY1AF3L1BGWW	281	I would like to have the objects appear on the computer screen as people that i select(photos or drawing), and have them fight, or dance. Make a video of that and share with friends, would be a much easier way than to do it with combustion or other post production software.	Maybe write and draw images that would appear on a page, and then i could simply save it and email it to friends. It would be like writing a personal letter.	linking, communicating, managing
A1M05NHIHZQOCH	284	AUTOMATE TASKS	AUTOMATE MOUSE MOVEMENTS	navigating
A1M05NHIHZQOCH	451	AUTOMATE KEYBOARD TYPING	IF I'M DOING A SURVEY AND I HAVE TO INPUT CERTAIN FEILD I COULD SAVE THOSE INPUTS AND HAVE IT FILL AUTOMATICALLY INSTEAD OF ME TYPING.	managing
A1M05NHIHZQOCH	556	remember and remind me of appointments for me	put in appointments and have a reminder automatically	linking

A1LHKQZIZWKRMQ	72	I would want to be able to play games such as battleship over the internet using this system	It would be a battleship game.	playing
A1JQLD26MO35V2	435	I would use it with my interactive home computer system. being able to place on different objects to represent parts of my house that i would like to control.	Through my computer I would be able to control the temperature, lighting, and music in my home. I would be able to do this with Trackmate by placing the object that reflects what I would like to control.	controlling
A1HHIEO107JPLB	1182	to manually manipulate objects on screen, read my handwriting and translate it into formatted, typed text	The link would have to be established via a medium that uses a USB connection like a webcam. After connecting the webcam, use a clear surface and shine a light from the bottom up. Put objects on top of the clear surface.	controlling, sensing
A1GQYLUJZCXRC3	413	I suppose something like telipresence(sp?) games. Chess for example. It could see where we have both moved and adjust where the opponents pieces should be. Kind of redundant but it is nice to use actual pieces.	The chess pieces would probably have to have some color shade or an rfid tag in order to be recognized. With that many pieces it would be a huge challenge. Maybe some kind of graph paper base to know the position. Although that is the base of a chessboard.	playing
A1FE8XKP4Q5JKD	416	Some sort of synthesizer would be very neat. It could also be used for gaming, maybe a game like chess, or any other board game. All the graphics and stuff play on the computer screen while all you do is move some chips on the Trackmate interface.	As mentioned above, a chess like game would work great, actions done on the Trackmate could be enhanced on the monitor by showing effects and animations on the screen.	creating
A1DELVFH0N35JO	412	refrigerator or pantry inventory system, heck, any inventory control system	I guess one would need to use multiple cameras in the fridge, barcode scanning what was coded and some type of user tag on the non-barcode stuff .. that would like to a DB that would track time-out-of-fridge (ie, took the milk out for 003E 2 hours must mean we're out of milk) .. something like a learning DB system to track your fridge contents and set up a buying guide for food shopping.	linking
A18B5QL328E02A	116	1. controls for nanny cam that I can manage via the website to supervise with without having to use keyboard control 2. security system interface to be able to look at settings and zoom in via webpage controls, not having to peck out commands on a keyboard or moving a mouse	selecting an area would immediately zoom to it, pulling off screen makes it zoom out, moving the mouse lets you move the gaze elsewhere	controlling, linking
A14OKIBGWHJE1F	312	Database design, with the objects being fields and grouping them into tables, even creating references by proximity.	Each time the object touches the screen, the application would give the user the choice of what to do with it, or simply recognize it if had been there already.	linking, creating

A11EY4NH8PG49J	1031	<p>I could see lots of uses for this device, especially when applied with a scanning function (not just location, tracking and color, but object recognition as well). For instance money counting (a blind person, or even just a busy cashier could now eliminate error), or a child's learning game (place fifty-five cents on the scanner, you have placed twenty-five cents, etc. as a learning game). It could also be used for mapping and charting, calculating distances, trip planning for travel guides etc. I could definitely see potential for a virtual DJ program - forget the turntables, you could mix with preloaded beats and sound effects. Also this type of technology is starting to lean towards what we all want - that nifty computer that Tom Cruise uses in his action movies, where you can layer screens - without needing more than one monitor (how about that for multi-tasking?) sifting through programs by dragging your finger through the air!</p>	<p>That would take some coding, or a program that could be loaded with Trackmate where one could customize the way they use the Trackmate. For instance, start the program, place something on the trackmate, it record the shape, color, position and any other identifiers its able to and then you assign variables to it through the program (ie. link sound file, or record placement a, or scan recognition of object to match preloaded images - with the money example, kind of like character recognition). Then once you have loaded the variables, you can determine how it interacts with the changes on the trackmate, in either placement, size, color, or object itself.</p>	<p>sensing, analyzing, teaching, creating, managing, navigating</p>
A105TUSJXTDWFZ	139	<p>This tactile interface would be awesome for laying out flowcharts/block diagrams/etc... (e.g.: like in Visio or Omnigraffle).</p>	<p>Each block in a block diagram or box in a flowchart would correspond to one of the physical objects.</p>	<p>creating</p>

Appendix B

Trackmate Tag Creation Code

Listing B.1: Trackmate Tagger Java/Processing code snippet used to create 1 inch square printable tags; only main drawing code is shown. Complete code can be found on the project's website[21].

```
stroke(128, 128, 255);
strokeWeight(5);
noFill();
rect(0, 0, 600, 600);
int data[] = new int[7];
pushMatrix();
{
  translate(TRANS_XY, TRANS_XY);
  scale(SCALE_XY); // 3.0 = 1" at 600dpi, 2.85 = 0.95" at 600dpi

  data[0] = 0x00FF & (int)random(256);
  data[1] = 0x00FF & (int)random(256);
  data[2] = 0x00FF & (int)random(256);
  data[3] = 0x00FF & (int)random(256);
  data[4] = 0x00FF & (int)random(256);
  data[5] = 0x00FF & (int)random(256);
  int sum = data[0]+data[1]+data[2]+data[3]+data[4]+data[5];
  data[6] = (0x0000FF & ~sum); // inverted checksum (inversion keeps allZero error case from
    happening).
  strokeWeight(1.0);
  noFill();
  stroke(220);
  ellipse(100, 100, 200, 200);

  fill(255);
  noStroke();
  smooth();
  ellipse(100, 100, 200, 200);
  fill(0);
  arc(100, 100, 208, 208, (-0.5)/40.0*TWO_PI, (0.5)/40.0*TWO_PI);
  arc(100, 100, 200, 200, (5-0.25)/40.0*TWO_PI, (5+0.25)/40.0*TWO_PI);
```

```

arc(100, 100, 208, 208, (10-0.5)/40.0*TWO_PI, (10+0.5)/40.0*TWO_PI);
ellipse(100, 100, 150, 150);
fill(255);
ellipse(100, 100, 80, 80);
fill(0);
ellipse(100, 100, 50, 50);
fill(255);
ellipse(100, 100, 5, 5);

// print on the info (dark), for users own reference.
for(int i=0; i<7; i++){
    fill(16);
    pushMatrix();
    translate(100, 100);
    rotate((i)/7.0*TWO_PI+PI/2);
    text(data[i], -5, -58);
    popMatrix();
    noFill();
}

// analog data corner spots
for(int i=0; i<4; i++){
    stroke(64);
    strokeWeight(1);
    pushMatrix();
    translate(100, 100);
    rotate((i)/4.0*TWO_PI+PI/4);
    ellipse(120, 0, 30, 30);
    popMatrix();
    noFill();
}

noFill();
strokeCap(SQUARE);
strokeWeight(14.0);
boolean lastWasOne = false;
boolean isOne = false;
for(int i=0; i<32; i++){
    int bit = i % 8;
    int byt = i / 8;
    isOne = ((data[byt] & (0x01<<bit)) != 0);
    if(isOne){
        stroke(255);
    }
    else{
        stroke(0);
    }
}
arc(100, 100, 155, 155, (i-0.5)/32.0*TWO_PI, (i+0.5)/32.0*TWO_PI);
if(isOne && lastWasOne){
    arc(100, 100, 155, 155, (i-1.0)/32.0*TWO_PI, (i)/32.0*TWO_PI);
}
if(!isOne && !lastWasOne){
    arc(100, 100, 155, 155, (i-1.0)/32.0*TWO_PI, (i)/32.0*TWO_PI);
}
lastWasOne = isOne;
}

for(int i=0; i<24; i++){
    int bit = i % 8;
    int byt = i / 8 + 4;

```

```

isOne = ((data[byt] & (0x01<<bit)) != 0);
if(isOne){
    stroke(255);
}
else{
    stroke(0);
}
arc(100, 100, 90, 90, (i-0.5)/24.0*TWO_PI, (i+0.5)/24.0*TWO_PI);
// now, make sure any seams are removed if adjacent data is the same.
if(isOne && lastWasOne){
    arc(100, 100, 90, 90, (i-1.0)/24.0*TWO_PI, (i)/24.0*TWO_PI);
}
if(!isOne && !lastWasOne){
    arc(100, 100, 90, 90, (i-1.0)/24.0*TWO_PI, (i)/24.0*TWO_PI);
}
lastWasOne = isOne;
}

```


Bibliography

- [1] Leah Buechley, Daniela K. Rosner, Eric Paulos, and Amanda Williams. Diy for chi: methods, communities, and values of reuse and customization. In *CHI EA '09: Proceedings of the 27th international conference extended abstracts on Human factors in computing systems*, pages 4823–4826, New York, NY, USA, 2009. ACM.
- [2] Vannevar Bush. As we may think. In *The Atlantic Monthly*, 176, 1, pages 641–649, July 1945.
- [3] Douglas Engelbart. Demonstration; a research center for augmenting human intellect. *Stanford Research Institute, Menlo Park, California*, December 1968.
- [4] George W. Fitzmaurice, Hiroshi Ishii, and William Buxton. Bricks: Laying the foundations for graspable user interfaces. In *Proceedings of CHI, ACM*, pages 442–449, May 1995.
- [5] Ben Fry and Casey Reas. Processing website, tools and community for simple Java development environment. <http://www.processing.org>, Released in 2001.
- [6] Saul Greenberg and Chester Fitchett. Phidgets: easy development of physical interfaces through physical widgets. In *UIST '01: Proceedings of the 14th annual ACM symposium on User interface software and technology*, pages 209–218, New York, NY, USA, 2001. ACM.
- [7] Dennis Hettema. ShotCode - Mobile tagging based interactive marketing. <http://www.shotcode.com>, May 2009.

- [8] Amazon.com Inc. Mechanical Turk: artificial intelligence, a marketplace for work. <http://www.mturk.com/>, 2009.
- [9] Apple Inc. Image of the new Apple iMac, April 2009, advertisement. <http://www.apple.com>.
- [10] Google Inc. Google Analytics: website keyword, ad, and referral tracking service. <http://www.google.com/analytics/>, 2009.
- [11] Hiroshi Ishii. Teamworkstation: towards a seamless shared workspace. In *CSCW '90: Proceedings of the 1990 ACM conference on Computer-supported cooperative work*, pages 13–26, New York, NY, USA, 1990. ACM.
- [12] Hiroshi Ishii and Minoru Kobayashi. Clearboard: a seamless medium for shared drawing and conversation with eye contact. In *CHI '92: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 525–532, New York, NY, USA, 1992. ACM.
- [13] Hiroshi Ishii, Ali Mazalek, and Jay Lee. Bottles as a minimal interface to access digital information. In *CHI '01: CHI '01 extended abstracts on Human factors in computing systems*, pages 187–188, New York, NY, USA, 2001. ACM.
- [14] Hiroshi Ishii and Brygg Ullmer. Tangible bits: towards seamless interfaces between people, bits and atoms. In *CHI '97: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 234–241, New York, NY, USA, 1997. ACM.
- [15] Robert J.K. Jacob, Audrey Girouard, Leanne M. Hirshfield, Michael S. Horn, Orit Shaer, Erin Treacy Solovey, and Jamie Zigelbaum. Reality-based interaction: a framework for post-wimp interfaces. In *CHI '08: Proceeding of the twenty-sixth annual SIGCHI conference on Human factors in computing systems*, pages 201–210, New York, NY, USA, 2008. ACM.

- [16] Sergi Jordà, Martin Kaltenbrunner, Günter Geiger, and Ross Bencina. The re-actable*. In *Proceedings of the International Computer Music Conference (ICMC 2005)*, Barcelona, Spain, 2005.
- [17] M. Kaltenbrunner, T. Bovermann, R. Bencina, and E. Costanza. Tuio - a protocol for table based tangible user interfaces. In *Proceedings of the 6th International Workshop on Gesture in Human-Computer Interaction and Simulation (GW 2005)*, Vannes, France, 2005.
- [18] Alan Kay. Doing with images makes symbols (archived video). <http://www.archive.org/details/AlanKeyD1987>, University Video Communications, 1987.
- [19] Aniket Kittur, Ed H. Chi, and Bongwon Suh. Crowdsourcing user studies with mechanical turk. In *CHI '08: Proceeding of the twenty-sixth annual SIGCHI conference on Human factors in computing systems*, pages 453–456, New York, NY, USA, 2008. ACM.
- [20] Adam Kumpf. LusidOSC website with protocol specification and example applications. <http://lusidosc.sourceforge.net>, 2009.
- [21] Adam Kumpf. Trackmate website with documentation, applications, forums, wiki, and source code. <http://trackmate.sourceforge.net>, Tangible Media Group, 2009.
- [22] Maria Cruz Leonel Morgado and Ken Kahn. Radia perلمان a pioneer of young children computer programming. In *m-ICTE 2006, Fourth International Conference on Multimedia and Information and Communication Technologies in Education*, 2006.
- [23] AirTeamImages Martin Boschhuizen. Boeing 747-206bm(sud) cockpit photograph. <http://www.airliners.net>.
- [24] Marvin Minsky. *Society of Mind*. Simon & Schuster, March 1988.

- [25] Gordon E. Moore. Cramming more components onto integrated circuits. *Electronics*, 38(8), April 1965.
- [26] James E. Morrison. *The Astrolabe*. Janus, November 2007.
- [27] Doug Coward’s Analog Computer Museum and History Center. Librascope Development Company, Type LC-1: WWII Navy PV-1 “balance computer”. <http://dcoward.best.vwh.net/analog/libra.htm>, May 2009.
- [28] A. Oulasvirta. Chi statistics, Helsinki Institute for Information Technology. <http://www.hiit.fi/node/290>, 2008.
- [29] Seymour Papert. *Mindstorms : Children, computers and powerful ideas*. Basic Books, New York, 1980.
- [30] James Patten, Ben Recht, and Hiroshi Ishii. Audiopad: a tag-based interface for musical performance. In *NIME ’02: Proceedings of the 2002 conference on New interfaces for musical expression*, pages 1–6, Singapore, Singapore, 2002. National University of Singapore.
- [31] Jun Rekimoto and Masanori Saitoh. Augmented surfaces: a spatially continuous work space for hybrid computing environments. In *CHI ’99: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 378–385, New York, NY, USA, 1999. ACM.
- [32] Mitchel Resnick and Brian Silverman. Some reflections on designing construction kits for kids. In *IDC ’05: Proceedings of the 2005 conference on Interaction design and children*, pages 117–122, New York, NY, USA, 2005. ACM.
- [33] Christopher Schmandt and Eric A. Hulteen. The intelligent voice-interactive interface. In *Proceedings of the 1982 conference on Human factors in computing systems*, pages 363–366, New York, NY, USA, 1982. ACM.
- [34] John Sculley and John A. Byrne. *Odyssey:Pepsi to Apple: A Journey of Adventure, Ideas, and the Future*. HarperCollins, October 1987.

- [35] Ben Shneiderman. Direct manipulation: A step beyond programming languages. *IEEE*, pages 57–69, August 1983.
- [36] Smithsonian. 1903 wright flyer photograph, from Smithsonian National Air and Space Museum. <http://www.nasm.si.edu>.
- [37] Ivan Edward Sutherland. Sketchpad, a man-machine graphical communication system. *Massachusetts Institute of Technology, Ph.D. Thesis*, January 1963.
- [38] Brygg Ullmer, Hiroshi Ishii, and Robert J. K. Jacob. Token+constraint systems for tangible interaction with digital information. *ACM Trans. Comput.-Hum. Interact.*, 12(1):81–118, 2005.
- [39] Peter Uray, DI Thomas Kienzl, and DI Ulf Marsche. MRI: a mixed reality interface for the masses. In *SIGGRAPH '06: ACM SIGGRAPH 2006 Emerging technologies*, page 24, New York, NY, USA, 2006. ACM.
- [40] Violet. Mir:ror rfid reader, zstamps, and nabaztag. <http://www.violet.net>.
- [41] Eric v. von Hippel. *Democratizing Innovation*. The MIT Press, April 2005.
- [42] Pierre Wellner. Interacting with paper on the digitaldesk. *Communications of the ACM*, 36(7), July 1993.
- [43] Jamie B Zigelbaum. Mending fractured spaces: External legibility and seamlessness in interface design. *Massachusetts Institute of Technology, M.S. Thesis*, September 2008.