Spatial Computing

Simon Greenwold

B.S., English & Applied Math
Yale University, June 1995

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 in Media Arts and Sciences* at the Massachusetts Institute of Technology
June 2003

Author: Simon Greenwold
Program in Media Arts and Sciences
May 9, 2003

Certified by: John Maeda
Associate Professor of Design and Computation
Thesis Supervisor

Accepted by: Dr. Andew B. Lippman
Chair, Departmental Committee on Graduate Studies
Program in Media Arts and Sciences

Spatial Computing

Simon Greenwold

Submitted to the Program in Media Arts and Sciences, School of Architecture and Planning, on May 9, 2003, in partial fulfillment of the requirements for the degree of Master of Science in Media Arts and Sciences

Abstract
Not yet written.

John Maeda
Professor of Design and Computation
MIT Program in Media Arts and Sciences

thesis advisor

Spatial Computing

Simon Greenwold

Joseph A. Paradiso
Principal Research Scientist
Co-Director, Things That Think Consortium
MIT Media Laboratory

thesis reader

Chris Csikszentmihályi
Assistant Professor of Media Arts and Sciences
MIT Program in Media Arts and Sciences

thesis reader

John Maeda
Professor of Design and Computation
MIT Program in Media Arts and Sciences

thesis advisor

Acknowledgements

Blah, blah, blah.

Contents

1. Introduction
1.1 Thesis Overview
1.2 Definition
1.3 Themes Traced

2. Background
2.1 History
2.1.1 The Machine in Space
2.1.2 Space in the Machine
2.1.3 Networked Space
2.1.4 The Denial of Space
2.3 The Problems With Virtual Space
2.4 The Problems With Interactivity
2.5 Enter Spatial Computing

3. Methodology

4. Motivation

5. Precedents

6. Roadmap of Experiments
6.1 Installation
6.2 Internaut
6.3 Stomping Ground
6.4 Hotpants/LittleVision
6.5 Pointable Computing
6.6 EyeBox

7. Experiments
7.1 Installation
7.1.1 Introduction
7.1.2 System Description
7.1.3 Technical Details
7.1.4 Precedents
7.1.5 Evaluation and Critique

7.1.6 Future Work

7.2 Internaut
7.2.1 Introduction
7.2.2 System Description
7.2.3 Precedents
7.2.4 Evaluation and Critique
7.2.5 Future Work

7.3 Stomping Ground
7.3.1 Introduction
7.3.2 System Description
7.3.3 Precedents
7.3.4 Evaluation and Critique

7.4 Hotpants/LittleVision
7.4.1 Introduction
7.4.2 System Description
7.4.3 Technical Details
7.4.4 Precedents
7.4.5 Evaluation and Critique

7.5 Pointable Computing
7.5.1 Introduction
7.5.2 System Description
7.5.3 Technical Details
7.5.4 Precedents
7.5.5 Use Scenarios
7.5.5.1 Universal Remote
7.5.5.2 Active Tagging
7.5.5.3 Getting and Putting
7.5.5.4 Instant Wiring
7.5.5.5 Reactive Surfaces
7.5.6 Evaluation and Critique

7.6 EyeBox
7.6.1 Introduction
7.6.2 System Description
7.6.3 Motivation
7.6.4 Method
7.6.5 Precedents
7.6.6 Design and Operation
7.6.7 Results
7.6.8 Evaluation and Critique

8. Summary Conclusions
8.1 It Doesn't Take Much
8.2 Object Resonance
8.3 Feedback and Relativity
8.4 Consistency and Expectation
8.5 Literalness
8.6 Transparency

9. Sources

A. Associative Interface

1.1 Thesis Overview

This thesis presents the results of a two-year program of research in spatial computing. During that time there was no single project or experiment that was definitive or summary of my work in the field. Therefore I have chosen to present each of the major projects I completed rather than focus on one of them exclusively.

The structure of the thesis is recursive in that it has similar outer and interior structures. The broad structure of the thesis has a classical research format:

Background
Methodology
Motivation
Precedents
Experiments
Analysis & Conclusions

These sections pertain to spatial computation at large. The analysis and conclusions are synthetic of the analyses of the individual projects. These outer sections frame the shared concepts that run through each of the experiments.

In turn each of the six experiments described has a format similar to the top level structure:

Introduction
DescriptionPrecedents
Evaluation & Critique
Future Work

These sections stand alone as independent reports on each project. The arguement of the thesis is that taken together as a body of work, these projects say more than they do individually.

Figure #: This project, *Installation*, allows users to place virtual objects in real space. It is a good example of spatial computing. (Discussed in detail below.)

1.2 Definition

Spatial computing is human interaction with a machine in which the machine retains and manipulates referents to real objects and spaces. Ideally, these real objects and spaces have prior significance to the user. For instance, a system that allows a user to create virtual forms and install them into the actual space surrounding him is spatial computing. A system that allows a user to place objects from his environment into a machine for digitization is spatial computing. Spatial computing differs from related fields such as 3D modeling and digital design in that it requires the forms and spaces it deals with to pre-exist and have real-world valence. It is not enough that the screen be used to represent a virtual space—it must be meaningfully related to an actual place.

I use "virtual space" broadly here not just to refer to three-dimensional Cartesian worlds, but any space maintained by a computer and supposed to appeal to a human sense of space. By this definition, a "desktop" in a graphical user interface is a virtual space. Similarly, spatial computing does not necessarily take place in a three-dimensional representation. For many human purposes a piece of paper is better understood as a two-dimensional surface than a three-dimensional object. In fact, spatial computing may not present a space to the user at all. It necessarily maintains an internal representation of space, even if it is only implicit in collected data, but its interaction with a user need not be visual or spatial. The simplest example may be an auto-flushing toilet that senses the user's movement away to trigger a flush. This is trivial spatial computing, but it qualifies. The space of the system's engagement is a real human space.

Figure #: The desktop is a virtual space. Notice here shading and occlusion.

The criterion that the objects and places in spatial computing have physical instantiation is not an arbitrary or trivial distinction. There are specific characteristics that make the production and analysis of spatial computing systems different from purely synthetic virtual systems. This distinction does not imply a value judgment—virtual systems have their place. However there are many cases, some discussed below, in which the purposes currently served by virtual systems could be significantly benefited by the adoption of spatial computing.

It may seem that the category of computational systems that engage true space is too broad to tackle in a single thesis. That is likely true, and I wish to be careful with the generality of the claims I make. But I do not think that the diversity inside the topic defeats the purpose of considering it as a whole. Instead, I think it may be useful to do so in order to upset a traditional taxonomy, which would not allow the analysis of physical systems next to software systems. I cannot imagine anyone arguing that there is a single correct hierarchy of similarity of ideas. In presenting spatial computing as an organizing principle, I allow several systems I have engineered to be brought into analysis together closely enough that they can shed light on one another.

1.3 Themes Traced

In order to make the connections between projects more clear, it is helpful to outline some of the themes common to many of the projects. They will come up repeatedly in the individual project critiques. These are the same ideas that form the body of my global analysis and conclusion, and I present them here without justification so that the reader knows what to expect and attend to. They are primarily qualities of and guidelines for successful spatial computation systems.

It Doesn't Take Much

Simple suggestions of space are often more convicining than detailed renderings. This idea is not surprising to anyone who enjoys comics or impressionist paintings.

Object Resonance

The physical objects involved must be approachable and pleasing. They should not deny their physicality by trying to disappear, but use their form for all of its potential value.

Fullness

The ways a system appears to be usable are often called its "affordances." The affordances of a sucessful spatial computation system must be implemented so fully that there are no invisible barriers to its operation that disturb the illusion it is trying to create.

Relativity

Perception is relative. We carry very few absolute sensory benchmarks. Many systems can become much simpler by using this to their advantage. Perhaps there is no need for hard calibration to an external reality.

Feedback
Feedback is essential to human control. The levels and kinds of feedback offered by spatial systems dramatically influence their usability.

Consistency and Expectation
The second ingredient in accomodating human control is not frustrating expectation. A user's desire to control a system should require as little conscious effort to achieve as possible. This demands total consistency in operation and gratification of expectated behavior.

Literalness
Metaphor is the primary ingredient of "interface" as we commonly experience it. To the extent that it is possible I advocate the elimination of metaphor in interaction with environments. Objects should be themselves and should not have to be referenced by an icon or a name.

Transparency
Some systems should become transparent—essentially unnoticable to their users. Some should remain solid and visible. There is no hard rule, contrary to some opinions, that says all successful systems become transparent. Much depends on the intended focus of user attention. In many cases the system itself is part of what should be experienced. The extent to which a system should assert its presence must be considered and controlled closely by its designer.

2.1 History

We have arrived at a critical point in the history of the machine in space. Engineers are rapidly banishing the last moving parts in consumer electronics, allowing them finally to shrink into near invisibility. Bulky CRTs are yielding to flat panels, allowing us to embed them into the surfaces we use daily and to free up valuable "real estate" on our desks. The businesses of computer graphics and surveillance have pushed our abilities to recover spatial information from the world at large. The long-standing divide between the idealized spaces of computer science and the heavy, cluttered spaces of real-world engineering are wider than ever, polarizing research around the world. Now that computation's denial of physicality has gone about as far as it can, it is time for a reclamation of space as a computational medium. In order to understand how we got here it is necessary to examine a history of computation in physical space.

2.1.1 The Machine in Space

The earliest machines designed as engines for calculation did not try to deny their physicality. They wouldn't have because they were purely mechanical devices. The abacus, from about 600 BC, for example, encodes numbers entirely spatially. It is programmed as it is read, in position. Here there is absolutely no abstraction of space. Data space is physical space.

Fig 2.1.1. The abacus is a physical computer not only in its computation, but also in its input and output.

Early computers couldn't help but be spatial. They took up space, and they used the nature and qualities of the physical world to perform their work. This continued to be true as the calculating machines abstracted their input and output away from physical configuration to digital displays, as in Blaise Pascal's mechanical adder of 1640.

The critical shift did not occur until electrical logic became cheaper, smaller, faster, and more reliable than physical switching. Motors and gears gave way to tubes and wires. Suddenly physics, which had been a computational medium became an enemy to be conquered. Computers were too big and too heavy, and things needed to get denser. Initially, computers were made into furniture as in the IBMs in order to make their outlandish size more palatable. [Scan photo from COMPUTER book]

Transistors, of course, proved to be the vehicle for shrinkage. As they replaced tubes, computers became objects in space as opposed to defining their own spaces. The rest of this history is common knowledge, how the computer shrank and shrank until we began to fold them up and put them in our pockets. We are constantly asked to remember, as if it mattered, that four-ounce phones we carry around would have weighed ten tons forty years ago, or some such shocker. But what does this neutron-star-like compression imply?

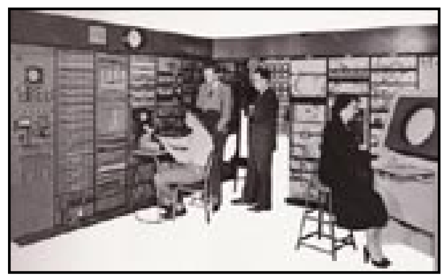


Figure #: Not so long ago computers made their own spaces.

First, it puts a clear value-system in place: for computation smaller is better. This seems obvious, but it is not the case for many things—houses and snack food, for instance. There is an obvious advantage to a computer that is small enough to carry. And physical space has emerged as perhaps world's primary limited resource. But we never seem to stop the furious miniaturizing, and that has to do with computing power. The outsides of electronics have on whole stopped getting smaller. We have already seen cellular phones hit an uncomfortable level of tinyness and bounce back somewhat in size. Things that are of the body must remain proportionate to it, but computational core of electronic objects are not bound to the body. If they are, it is only as added weight to be

Figure #: The SG2200 from Sewon claims to be the smallest cell phone.

Figure #: Current phones are larger than they were. Now they hide behind large color displays.

Figure #: Cygnal proudly offers us the C8051xxx microcontroller family. Good luck soldering that one. [http://www.cygnal.com/]

minimized. The parts of computation that are necessarily human-scale are the points at which the machine meets the user—input and output. So there is a tension introduced as the limits of human physiology keep computers from spiraling into nothingness, but at the same time we must keep making the insides smaller so that the objects themselves can become more powerful.

No one feels this tension more acutely than the electronics hobbyist. Traditionally integrated circuits, the bread and butter of any reasonably complicated electronics project, have been available in packages of sufficient size to allow them to be handled with fingers and soldered by hand—DIP "dual inline packages," for instance. But many of today's technologies such as BlueTooth are available for use only in packages with leads so many and so small that no human being could reasonably expect to manipulate it. These types of chips are designed for large companies who design circuits on a computer and then have them assembled by robots. This happens, of course, because the economics of serving a hobbyist population doesn't justify the expenditure. But there is the feeling that consumer electronics technologies are shrinking away from accessibility to human experimenters.

The physical shrinkage of the machine manifests itself as an embarrasment of the flesh. The thinner the notebook computer, the better. Electronics is an anorexic industry. As Niel Gershenfeld points out, there is no information without a physical medium. Spatial computing proposes to celebrate corporeality of data rather than trying to deny it.

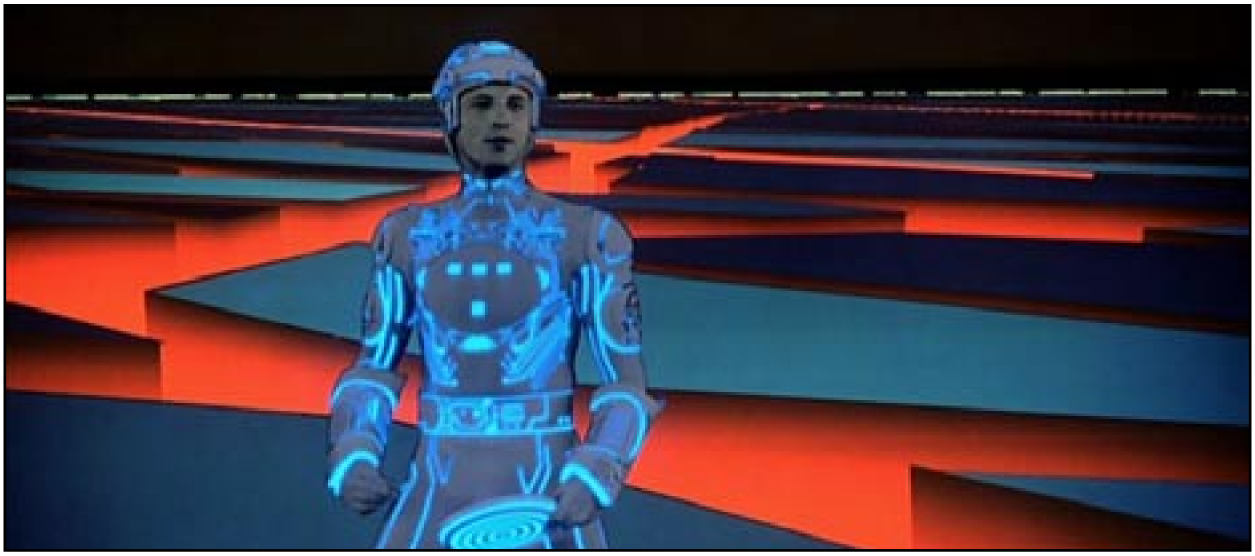


Figure #: The 1982 film Tron demonstrated the cultural fascination with and fear of being swallowed by the machine.

2.1.2 Space in the Machine

Our fascination with the space inside the machine is not new. The Aristotelian universe was essentially a mechanical system that described planetary motions as part of a giant machine. Describing life inside space stations and bubbles large enough to hold populations has been a staple of science fiction for as long as it's been around. And in 1964 Archigram reimagined the city as a huge walking robot that could dock with other cities.

Since at least the Renaissance, artists such as Durer, used machines to help them represent space. In the second half of the twentieth, however, the growing internal power of machines began to allow them to represent spaces and objects directly to our eyes. They turned out to be masters of perspective and simple shading, a few of the artist's simplest tricks for conveying depth. Suddenly there appeared to be whole open landscapes inside the machine.

And as the outsides of the machines shrank and the "space" of memory and storage inside exploded, it became possible to popularize the idea of moving ourselves wholesale out of messy old real space

Figure #: In 1959, the DAC-1 (Design Augmented by Computers), developed by General Motors and IBM, was the first interactive 3D computer graphics system.

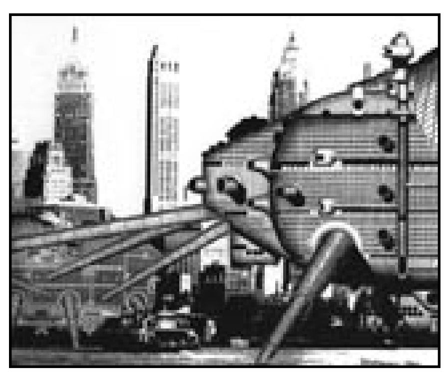


Figure #: Archigram's Walking City, 1964. [Herron, 1964]

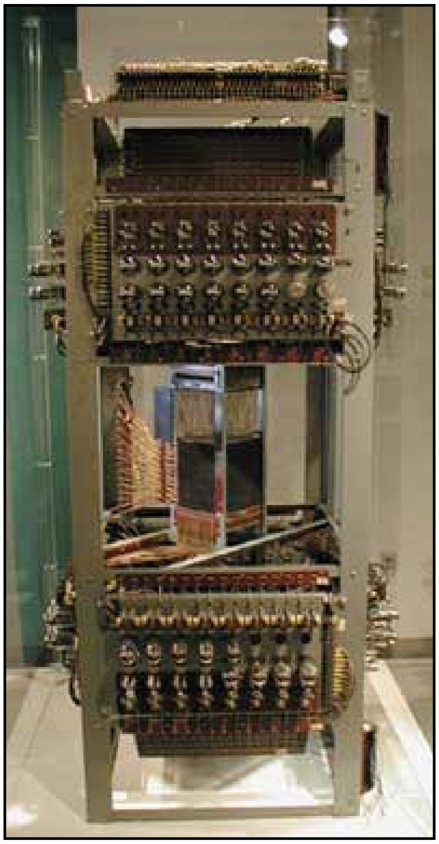


Figure #: 4,000 bytes of memory from the 1951 Whirlwind computer, standing roughly 9 feet tall. Today we put 1,000,000,000 bytes on a single chip.

and into virtual space. A magnetic core memory of 4,000 bits weighed tons in 1951, but now (April 9, 2003), we store a billion bits on a chip the size of a fingernail. The scarcity, expense, and imperfection of real land made the possibility of a boundless internal landscape too tempting to resist. This possiblility was also greeted with anxiety as demonstrated by movies such as *Tron* and *Lawnmower Man*, in which humans are sucked into and trapped inside a virtual environment.

Early computer-generated spaces tended to be (and still often are) rigidly planar expanses of exaggerated linear perspective. Lines are straight, corners are perfect, and ornamentation is minimal. Interestingly this represents something of a return to Modernist form. Mies van der Rohe's architecture, for instance, exhibits what he called "universal space" and the "open plan." It results in floating planes and broad gridded plazas. Interstingly Lev Manovich also finds a return to a kind of "soft modernism" in the aesthetics of the web [Manovich, 2002].

Le Corbusier, in many ways the father of Modernist architecture famously called the house, "a

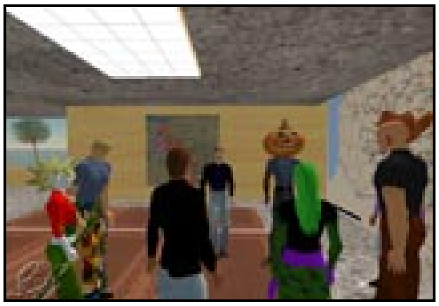


Figure #: A typical representation of a machine-generated "space" from the soon to be realeased online environment, "SecondLife."

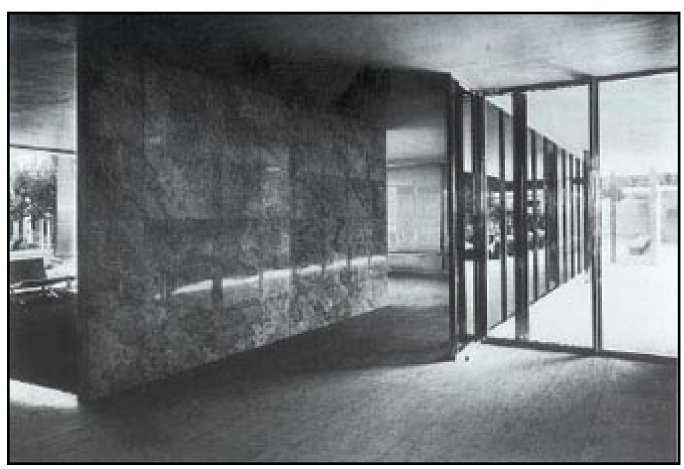


Figure #: The Barcelona Pavillion by Mies Van der Rohe. One of the few real spaces that looks virtual.

machine for living in." This paved the way for a functionalist rationalization of pure forms: "Form follows function." This was possible to espouse in the 1920's, but could not survive the critique of rationality that followed World War II and the failure of computer science to model human thought. These "functionalist" spaces were also extremely difficult to build and maintain. Interestingly it is exactly their ease of production and maintenance in machines that keeps them present as virtual architecture although they had faded from prominence in physcial architecture before the first computer graphics arrived.

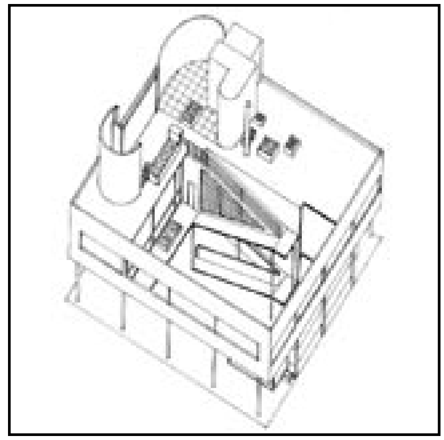


Figure #: Villa Savoye a Poissy by Le Corbusier, who famously called a house "a machine for living in."

What this really serves to demonstrate is that form follows economics of production. Computers make it cheap and easy to make clean corners, so that's what we see. Baseboards help cover up irregularities in physical meetings of wall and floor, so most real buildings have them. That virtual environments are becoming more detailed and more topographically complex is due to improved tools for their construction and deployment. There seems to be little ideology driving the development of a virtual "style" except for the quest to do whatever technology has made newly possible.

Figure #: It is obvious the floating object is artificial because its colors are too consistent, its lines and corners too sharp.

One of the hallmarks of the unreality of virtual spaces is their over-perfection. On a computer screen, things look like their ideals (or at least anything with corners and flat faces). A line is a line and a cube is a cube. These images are unconvincing because we know that there is no real substance that could be kept so perfect. Much time and attention in computer graphics nowadays goes toward making things look imperfect enough to be convincing [Dorsey, 1996]. It is a hard problem, and it isn't yet solved.

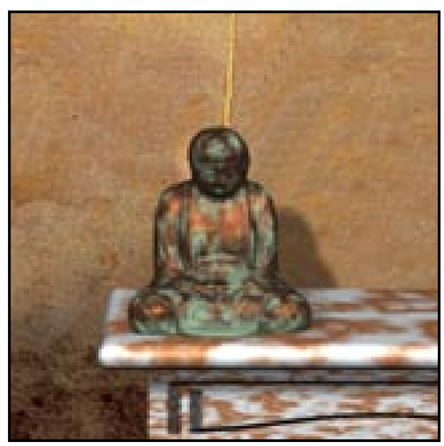


Figure #: A bronze Buddah rendered with a procedurally-generated patina. [Dorsey, 1996]

Figure #: Solutions such as this may have difficulty catching on.

Computer graphics' primitive appeal to an impossible purity makes the idea of virtual space feel somewhat immature and naive, and its throwback to long outgrown architectural ideologies doesn't help either. The proponents of virtual environments have suggested without irony that we use systems that make us look like cyborg monsters. There really isn't anything appealing about this vision to many important sectors of culture. All of this leads to some deserved ridicule surrounding the field of virtual reality.

Figure #: Robert Venturi and Denise Scott Brown see the world as a layering of signs and symbols. [Venturi, 2001]

Where computer graphics diverge completely from spare modern spaces is in matters of graphical interface. The collapse of Modernism brought forward the dominance of the symbol. Architecture proliferated with overt historical quotations and references. Robert Venturi and others recognized that there is no form that does not carry infinite layers of meaning. What is suggested is as real as what is physically present. This is the language of graphical user interface, where the icon reigns supreme, and language is larded over the top of everything.

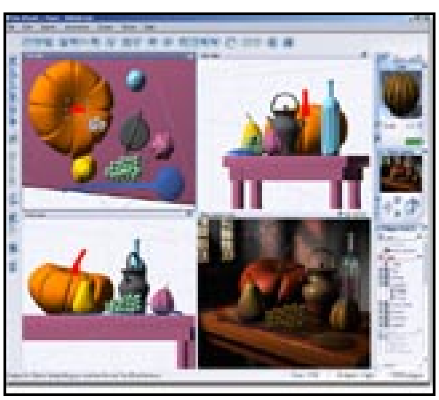


Figure #: Venturi and Scott Brown's vision of architecture makes the world look a lot like software interface.

This mess of signifiers is pretty much where software spaces remain today. Spatial computing proposes to do away with icons, lists, and menus as much as possible, to allow things to stand for themselves.

Networked Spaces

The advent of the Internet considerably complicated the relationship of computation to space. Suddenly connections made inside the machine had the potential actually to span half the globe. Every screen became a portal onto the same shared parallel virtual world.

The bright side was the promise of an end to solitary virtual existence, replaced by virtual networked communities. And it is true that much of Internet traffic consists of email and instant messages. However, one of the strange qualities of web space is that the user is always alone in it. No matter how many other people are looking at the same information one is, one does not see them. One has the feeling of having the entire vast Internet to oneself.

People saw the expansion of the World Wide Web as a kind of virtual space, and it did take root even if it didn't replace the physical world as many feared. It seemed that the Internet could act as a kind of spatial prosthesis, a vastly enhanced telephone. (Remeber AT&T used to encourage you to "Reach out and touch someone.") Everything on the web is a single address away—maybe a click, maybe two. (Advertisers consider three clicks to be an unacceptably large "distance.") But what is the product of total equidistance if not collapse into singularity? It's not a new spatiality, it's a non-spatiality. And what is the result of the devaluation of space to the point that it can be minted practically for free? Space becomes valueless. As soon as some space becomes ruined, we can just make another new one twice its size. Perhaps what makes the Internet valuable is that it is non-spatial and attempts to introduce space to it are fundamentally flawed. (I will have more to say on that in my analysis of my own attempt to do this, *Internaut*.)

The Denial of Space

The Internet is not the only agent of spatial denial in computer science. The dream of escaping the imperfect and unpredictable real world is the engineer's heaven. It is a denial of heaviness, friction, death, and decay. The memory spaces of computer science are the site of huge projects in idealized engineering—where programmers construct machines of astonishing complexity in the absence of gravity and corrosion. This absence of physical constraint in the digital world is precisely the reason that the replacement of analog and mechanical systems with digital systems will run its course until the only remaining analog circuits serve as interfaces to the digital and mechanics serve exclusively to sense and to actuate.

Escape from the uncontrollable and capricious real world into a perfect world of rules, where every consequence has a cause if one knows enough to discover it helps explain the motives of strange hackers and virus writers who measure their success by the quantity of their access and spread. These people, powerless in the real world, are masters of the machine. It is perfect in its willingness to do as it's told. The author can attest that this kind of power can be very compelling to a young person who longs for a logical world in which every problem eventually yields to explanation.
It also helps explain why virtual spaces have had only one resounding area of success—violent first-person games in which players shoot each other at will. These scenarios appeal to the same crowd of teenage boys.

Absurdity grows like a barnacle at sites of cultural tension. All it takes is a look at the size and complexity of the heatsinks that accompany any modern microprocessor to know that engineering is engaged in a fight with physics. We are poised at a point of extreme tension in the spatial relations of computation. I propse a computation that embraces the machine as a spatial object at the same time integrating it with the space inside itself.

Figure #: The size of this heatsink relative to its host is a sign of the heroic struggle of technology against the physical world.

The problems with virtual spaces

Something that tends to go unchallenged is the realism of virtual spaces. The increasing power of processors and graphics cards enables more and more accurate modeling of the physics of light and the mathematics of surfaces. As Lev Manovich understands it, realism has become a commodity we can pay more to buy more of [Manovich, 1996]. But there is a subtlety that is missing from all of the marketing and analysis of virtual systems.

There is a tremendous disconnect between screen-based representations of reality and experiential reality that makes increasingly accurate physical modeling somehow less engaging than it seems it ought to be. The computer graphics term for rendered realism is "photorealism," and that hints at the problem. The realism that computation tends to aspire toward is the realism of a photograph. A human being does not experience a photograph as an instantaneous and engaging reality in which he is part. He does not imagine the camera's eye to be his own. He remains firmly outside the image, and understands it usually as a captured moment of objective representation. It is undeniable that there is something compelling about the asymptotic approach to photorealism. Increasingly "accurate" renderings continue to inspire wonder even now that the game of chasing reality has grown old.

Figure #: The Cornell box is the benchmark for photorealism. Rendered images are compared against pictures taken inside a real box. (This one is rendered.) [http://www.graphics.cornell.edu/online/box/compare.html]

But the wonder masks an important distinction that virtual reality denies. The wonder is the wonder that the image was not produced by a camera, not the wonder that the viewer was not present as the perceiver of the scene. There hangs above the discipline this notion that we are just a breath away from producing representations that are sufficiently accurate to fool the viewer into total engagement. It can't happen that way.

Figure #: Sega's *NBA 2K3*. Widely touted as "the most realistic basketball game ever." [http://www.epinions.com/content_85992509060#]

This confusion of "realism" is apparent from looking at the use of the term "realistic" as it is applied to computer simulations such as games. Sega's basketball game NBA 2K3 is hailed all over the Internet as the most "realistic" basketball game ever to be produced. What this seems to mean is that the players bodies and faces are taken from real NBA players and the camera shots look like television coverage of basketball. The view is not first-person from a player in the game, and not even from a fan. Instead "realistic" here means creating television with your thumbs. This could hardly be farther from the reality of a player in the game.

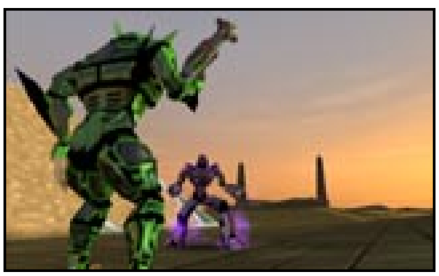


Figure #: The view from behind yourself in One Must Fall Battlegrounds. [http://thegamebd.tripod.com/previews/OMF/Omf.htm]

This is again evident in the popular, "behind your own back" view in fist-person games. It is often possible to switch the first-person viewpoint which is supposed to correspond to the player's eyesight to a view that is over the player's own shoulder or behind him. This is often more convenient for game-play because it shows the player in the context of the scene. But there is no disorientation involved in switching from the eye to outside the self. It is enough to indicate that the "eye" view does not really engage the player as if it were his eyesight.

This has everything to do with the nature of perception. The fundamental discovery of art and the physiology of perception since the Renaissance is that the eye is not a camera. Vision is a constructed sense. We have a tiny area of acuity with which we constantly and actively scan the world. Any notion of a photographic experience of a real scene is one constructed by the brain. This is different from the experience of a photograph, which appears as a small colored patch in our field of view. We can understand it as it relates to our experience of the visual world, but it does not mimic our experience of it.

There is nothing “natural” about a rendered perspective projection. It is intelligible, but it isn’t how we see things. In some cases, increasingly “realistic” representations only serve to alienate us from what we are seeing. For instance, in the Quake II engine from Id Software, as the protagonist walks, his eye bounces up and down. It is particularly noticeable when he is walking close and parallel to a textured wall. It is a bizarre sensation to watch the representation of space bob up and down the player moves forward. But if one looks closely at walls when he walks in the real world, it actually does the same thing. But we filter it out so we don’t even notice it. In our minds, walls don’t bounce. So which is the more “realistic” representation? There is a perfectly valid argument that whatever alienates the viewer less is the more realistic. Game players say that after a while one ceases to notice the bouncing, just as presumably, we cease to notice it in the world because it is always present. But I expect that learning to ignore this effect is the same kind of learning that allows players to meld their being with a paddle in Pong. They simply ignore the clear signals that tell them there is an other reality outside of this small area of focus, and proceed as if it were not the case.

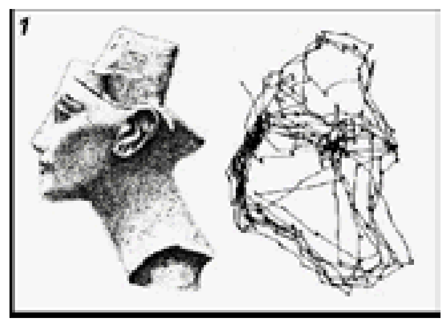


Figure #: Eye-movement traces while a subject explores a picture of the bust of Nefertiti. [Yarbus, 1967]

E. H. Gombrich points out that vision proceeds not as construction of image but as progressive hypothesis testing against actively acquired percepts [Gombrich, 1969]. We refine our understanding of the world by actively testing it with our eyes, which are attached to our heads. That means if there is an uncertain condition to our right, we may turn our heads. Any visual information we gather there is added to our mental image of the scene in front of us, but the image is as much constructed in reverse from beliefs and memories as it is from light hitting our retinas. A photograph does not allow for active perception other than simple surface scanning, pulling our faces close to discern detail or away to get a fuller view. There are many conditions that appear visually confusing in photographs that could never be in reality. Sometimes a tree appears to be growing out of a person's head. We almost never get that impression in reality. The active quality of perception will disambiguate the situation before it even becomes questionable in reality. For instance, there is always motion in the real world, and there will be differences in the relative speeds of motion of the tree and the head in the visual field. This effect, called head-motion parallax, is more important to our perception of depth than stereopsis [Arthur, 1993]. Our ability to perceive is distinctly limited in virtual realms becaue the system cannot possibly respond to all the techniques for active perception that we use. Some of them try to allow for it by using gaze or head-position tracking [ref]. But even these cannot yet offer the wearer touch.

Figure #: A VR "CAVE" projects 10' X 10' images on four sides. [http://www.eds.com/services_offerings/vr/centers_the_way.shtml]

Systems that use specialized hardware and sensing to try to replace as much of a subject's sensory input with synthetic information are called "immersive," and they all suffer the same unavoidable problem. We have no experience of "immersion" in our real existence. We are part of it, and it is part of us. There aren't even clear boundaries between the self and environment—it has been contested for centuries. When a subject is "immersed" in a virtual simulation such as a "CAVE," which projects images on 10' square walls all around a subject, he has an experience of immersion, which is distinctly not a quality of reality. Immersion is like diving into cold water. One of reality's hallmarks is that its experience is mundane, and any excitement it contains comes from the thing experienced, not the act of perception.

Paradoxically, the disconnect with reality become most apparent in the most "realistic" simulations. The more the viewer is supposed to be relieved of the effort of constructing a reality out of partial information, the more obvious it is in its modes of failure. This is why an artifact such as an aliased halo around an otherwise well-rendered character is so disturbing, or why the slightest anti-physical movement is so disruptive. This lies behind the movement toward "cartoon" renderings of virtual systems. [Kline, 1999] They have discovered that when the metaphor is not denied, but allowed to exist and even fostered, there is less tension in the perceiver. The action of making a narrative reality out of the image sequence has a consistent and pleasing quality. We are not disturbed by the obvious unreality.

Figure #: Screenshot from the AMP II game engine. [http://www.4drulers.com/amp.html]

This same failure of over-literalness is apparent in the virtual reality of telepresence, in which a non-present party is brought into "presence" by a virtualizing technology. In all of the telepresence systems I have witnessed, the most obvious quality of the remote parties is their non-presence. The technology that is supposed to bring them closer only serves to emphasize their distance from the goings-on. Having, experimented with webcams for personal connection to help maintain a long distance relationship, I can attest to their inadequacy. (We went back to telephone only.) Often a mentally-constructed reality is more compelling than a sloppily constructed representation of a fuller set of sensory information. Readers usually find this the case with film adaptations of books they love.

Figure #: Dobie T. Coyote from Bruce Blumberg's Synthetic Characters Group. [http://web.media.mit.edu/~bruce/whatsnew.html#Anchor_new1]

Figure #: Video conferencing facilities are available at the New Greenham Park Hotel. [www.greenham-common-trust.co.uk/ images/video.jpg]

The inadequacies of virtual environments make it worthwhile to look for alternative modes for dealing with the space inside the machine.

The problems with interactivity

Where the problems of virtuality are problems of space in the machine, the problems with "interactivity" are problems of the machine in space.

There is an irony in the use of the words "active," "interactive," and "reactive" to describe computational objects—both physical and virtual. It is a common practice, as though nothing had those qualities until the computer swooped down and started endowing ordinary objects with buttons and microphones. The truth is that non-computational objects are far more active, interactive, and reactive than any working computational version of the same thing. The reason is that in order to consider an object computationally, we must derive data from it, and that means outfitting it with sensors in some way. As soon as we do that, we chop away all of the interactions we have with that object that are not meaningful to the specific sensor we have chosen. No matter how many sensors we add, we are taking a huge variety of interactive modalities and reducing them to several. How could a simulation of a cup ever be as interactive as a cup?

Some argue that adding sensors to a physical object does not constrain its existing interactivity, but augments it electronically. I believe that is true as long as the object remains primarily itself with respect to the user and does not undergo some metaphoric transformation into a virtual representation of itself or into a semantic placeholder. That is difficult to achieve, and cannot be done as long as a user must consult a secondary source to determine the quality of his interaction. For a user to check a screen or even to listen to a tone to determine the pressure with which he is squeezing an object supercedes his own senses and reduces any squeezable object into a pressure sensor. In order for a physical object to be augmented rather than flattened by computation, the computation must occur (or appear to occur) inside the object and the consequences of the computation be registered by the object. The object must also not become fragile or restricted in its manipulability.

This challenges the claim of mouse-based Flash authoring to be "interactive design." It is interactive relative to a painting but it certainly isn't as interactive as an orange. In order for us to design objects that meet that level of interactivity we will have to concern ourselves with more than the screen. The physical body of the computational object is vital to its interactivity.

Enter Spatial Computing

Spatial computing proposes hybrid real/virtual computation that erodes the barriers between the physical and the ideal worlds. Wherever possible the machine in space and space in the machine should be allowed to bleed into each other. Sometimes this means bringing space into the computer, sometime this means injecting computation into objects. Mostly it means designing systems that push through the traditional boundaries of screen and keyboard without getting hung up there and melting into “interface” or meek simulation.

In order for our machines to become fuller partners in our work and play, they are going to need to join us in our physical world. They are going to have to operate on the same objects we do, and we are going to need to operate on them using our physical intuitions. Interface needs to be pried from the surfaces of the screen and keyboard and exploded into every constituent of our environment. If we are not already, we will become human beings embedded inside our connected machines. We will be the processors working within the giant spatial networks that surround us. How will we use space, place, and objects to direct that computation?

Methodology

Every project I have undertaken here at the Aesthetics + Computation Group has had a component of mixed-up spatiality. My goal has been to attack the boundaries between physical and virtual spaces with small incursions from all sides. Therefore my explorations have been many and various in media and scope. Some have been more about place, some more about objects. Each one has led me further in the direction of spatial computing. As I leave here I imagine each of the projects I developed as a component that could be integrated into future systems that more powerfully complicate the real and virtual than any of them taken singly.

Obviousness

In the course of my study my primary method has been to make things first, and ask questions later. This process privileges intuition over scientific inquiry because it does not produce artifacts designed to test hypotheses. It is an engineering methodology driven not by a functional brief but instead by demand that the product simply be of interest to its author. This could seem solipsistic and indulgent. (I think it is indulgent, but that doesn't denigrate the products.) Its value to a wider world lies in the faith that my concerns and interests are not so specialized, and my background is not so narrow, that things that I believe are interesting projects will be interesting also to some public. This has proven to be the case. And my greatest pleasure has been that the appeal of my best work has been obvious. I have not therefore felt the need to mine those specific projects further for their second-order hidden value, but have instead chosen always to move on to something new.

Obviousness is not an intrinsic quality of ideas. It is conditioned on the state of the observable world and the experience and intellect of the observer. There is a clear pejorative connotation to the word as applied to the products of research, but I would argue that the most important research results are always obvious. Perhaps they were not obvious before the program of research that uncovered them, but with the background of the researchers and the data and analysis they performed, strong conclusions become unavoidably clear—accessible to intuition. The same is true for good argument. By the end of it, to anyone who experiences it, its conclusions should have become obvious. Anything below the level of obviousness is a secondary truth that requires more digging. It is not a present reality. This argument is probably syllogistic—obvious. But by making it, I mean to defend my methodology against those who would say it is not suitably scientific. In fact, it is not at all scientific, and I never intended it to be. Where my methods came closest to scientific, my products were the least successful.

So my program has been to make things—as many as I possibly could—and see what they made obvious, singly and together, to me and to others. It is a faith in the value of what I have produced that allows me to do this, and it is justified only if there are readers who agree that these projects make certain things clear that would otherwise have been obscure. I have taken pains as I produced these projects to allow them to change as I made them, to take their best course. It is impossible to know what something ought best to be ahead of time. It becomes clear only in the process of making what a thing's most valuable form will be. This freedom to allow ideas to change as they became real has made my work better. Nothing leads to more tortured and awkward instantiations of ideas than rigidity of purpose.

It was not always clear to me as I worked what the connections between my projects were, and it has required a period of introspection, a reprieve from building, to hear what they have to tell me. The theory has arisen from the artifacts, not the other way around, and that is the only reason I have faith in it. As William Carlos Williams said, “No ideas but in things.”

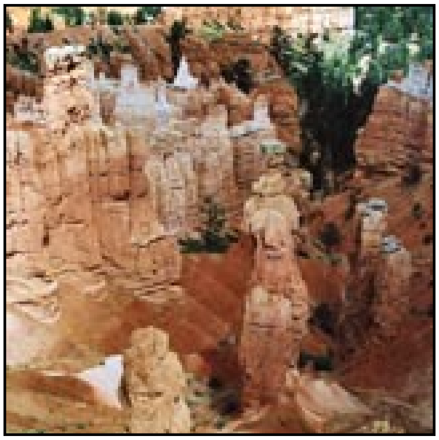


Figure #: I vividly remember Bryce Canyon in Utah. [http://globetr.bei.t-online.de]

Motivation

I have access to my motivation only through introspection. I do not think I have any special privilege to it. So I will construct what I believe is one plausible explanation of my interest in making the set of things that I have.

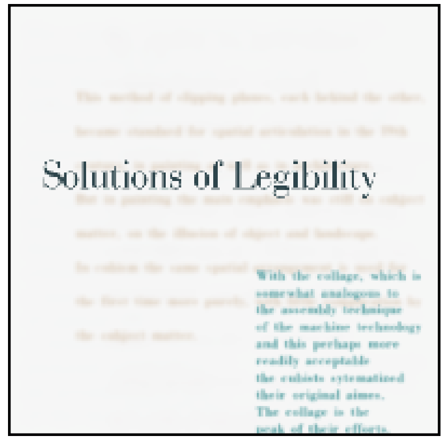
My family used to take trips to national parks. These were some of my favorite vacations because I liked to walk inside landscapes that were much larger than I was. I liked to be able to see things distantly and then gradually to approach them and find them to be even more richly detailed than I had imagined them. I was a computer child too, so I often thought about this in terms of resolution and quantization—how the strongest flavors of the real world were due to its infinite resolution. Every pinecone had something interesting to say under each one of its scales if you took the time to examine it with eyes and fingers. No simulation I had ever experienced had that power. They reached only as far as the attention of their creators. But I dreamed of making that simulation. My fantasy was to be able to take flight from where I stood and zoom in any direction to close in at high speed on anything that caught my interest. I would be able to experience it in all of its detail. That was a juvenile fantasy, but what hasn't left me is a love of the real. What I understand better now are the limits of computation. I no longer dream about producing such a system inside the machine. Instead I have turned my attention to an idea that I think hold more promise, the integration of the real and computed. Rather than try to simulate the qualities of the world I love, why not let the world stand and be present in all its complexity. I have been trying to make systems that engage the physical world rather than deny it.

Precedents

I am not the only researcher commited to bringing together human and machine space. There are many such programs around the world, each with a somewhat different focus. Many of these have been influences on my work or my thinking about it afterwards. Each of the individual projects I describe in this thesis had specific precedents, and those I will detail in their own sections. Here I will discuss only programs that were influential in my broad conception of spatial computing.

There are important precedents for me quite close to home. The Visible Language Workshop was the group at the MIT Media Lab that later became the Aesthetics + Computation Group, of which I am a member. They did much of the pioneering graphics work on integrating perceptual depth cues other than linear perspective into computer graphics. In particular some of their research dealt with layering, blur, and transparency [Colby, 1992].

Some recent and ongoing research at the Lab also shares much with spatial computing. In particular, Hiroshi Ishii's Tangible Media Group has an interest in physical manipulation of objects as a medium for computational control. The work of Brygg Ullmer such as his *metaDESK* [Ullmer, 1998], and *mediaBlocks* [Ullmer, 1997] provide a variety of ways to use physical objects and spaces to explore and manipulate digital information. One of the primary differences between what Brygg and the rest of Ishii's group have done and what I am have been doing is that their work focuses directly on interface. They are willing to use physical objects as icons "phicons." These are objects without previous valence to the user, often abstract blocks or disks. Their manipulation does provide control over a system, but it isn't fundamentally



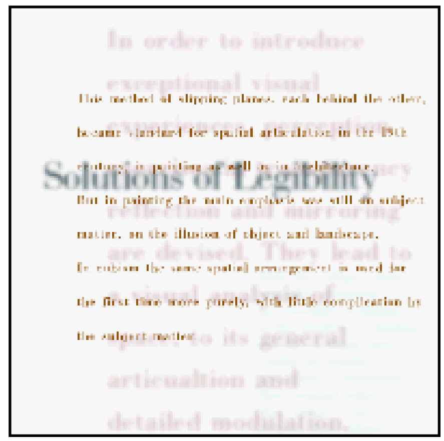


Figure #: The Visible Language Workshop explored layering, translucency, and blur as visual tools.

Figure #: Brygg Ullmer's metaDESK uses a variety of physical tools and metaphors to allow users to interact with geographical data.

different from software interface except that it exists outside the screen. They call these systems "TUIs" for Tangible User Interface. I think tangibility is important, but it is not my primary concern. Tangibility is a necessary byproduct of computational engagement with real objects in real spaces. I would not want to miss it, but I do think that reducing physical object to interface controls unnecessarily saps them of their own identity and autonomy. As Ullmer points out, they are symbolic, standing for something for something other than themselves [Ullmer, 2001].

Figure #: Brygg Ullmer's mediaBlocks lets users store and manipulate media clips as if they were stored in woden blocks.

Where Tangible Media deals with physical objects as interface, my aim is to obscure and distribute interface so that it becomes impossible to locate its surface. Interface itself is unavaoidable. It happens at the meeting of any two different media. But in our interactions with physical objects we are seldom aware of interface as such. Our attention extends beyond the interface to the object of our intention. I hope to allow for that push through interface in spatial computing.

A group at the Lab that has done significant work toward embedding computation in existing objects is Joe Paradiso's Responsive Environments group. They have placed sensors and computers in objects such as shoes for dance and gait analysis without making them fragile or limiting their use [Paradiso, 2000]. They are also interested in sensor networks, which effectively spread the locus of interface so widely that it may become invisible. Matt Laibowitz is currently defining a "Phenomenological Model for Distributed Systems," which deals explicitly with issues of active computational perception [Laibowitz, 2003]. These projects go a long way toward integrating the machine into human space.

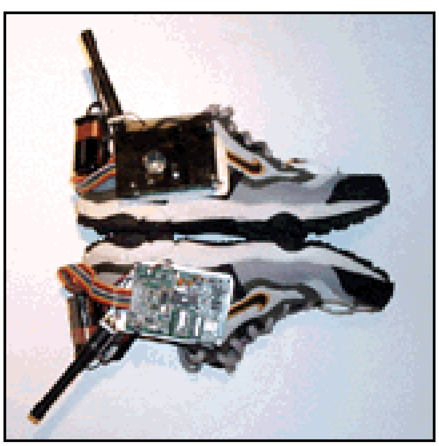


Figure #: Expressive footwear from the Responsive Environments group.

Bill Buxton has done a tremendous amount of work throughout his career on human physicality in interface design. Almost any topic in human-computer interaction has at least one Buxton paper on it including layering and transparency in 3D environments [Zai, 1996], and ubiquitous computing [Buxton, 1997]. In recent talks he has expressed concern over the difficulty of transducing objects. We have very few ways to get them into and out of our machines. This is a concern central to spatial computing.

On the humanist side of this research, Anthony Dunne and Fiona Raby have been looking at ways people react to objects with technological appendages. For instance they embedded a GPS receiver in a table and had people keep it in the homes for periods of time. They found people became attached to the object and its operation and were concerned when it lost its signal. Some were compelled to take the table outside where it could tell where it was. The attachment people make to active objects is of central importance to spatial computing. The qualities of design that establish that pseudo-empathic relationship are part of what I hoped to engage.

Spatial computing is such a broad umbrella that many disciplines and programs of research are valid precedents. Rather than try to be exhaustive here, I will consider precendents closely related to each project as I describe it.

Roadmap of Explorations

The six projects I describe in this thesis could be organized on several different axes. They could be ordered by their bias toward real or virtual space, or the amount they deal with objects versus the amount they deal with space. Instead I will present them as a chronology because it will give the reader some idea of the circumstances that lead to their conception and the forces that shaped their development.

Installation

Figure #: Installation.

I arrived at the Aesthetics + Computation group after two years studying Architecture at MIT. I was ready to think about computation and space, and eager to explore the resources the group had to offer. Among these was a set of inductive position and orientation sensors called a “Flock of Birds,” enough surplus flat panel CRT displays that I could be allowed to dismember one, and a miniature video camera. I quickly sketched out an idea for a system called Installation that would allow users to create and modify virtual sculptures that were visible only through a viewing screen. The viewing screen could be moved freely in space to see the virtual construct from any angle. This involved the use of two of the inductive sensors (one to use as a 3D stylus, and one to track the position and orientation of a viewing screen) one gutted flat panel; and the camera mounted on the back of the screen. The system took shape quickly and ended up surprisingly close to my original intention. In the end the system allowed users to sketch free-form blobs with the stylus and then install them permanently at any depth into the space of the room as seen through the view screen. When the user moved the view screen, the objects responded as if they were actually in the room. I later wrote an external client for the system, which I ran on several machines around the room. Whenever a user threw an object close enough to one of the clients, it would disappear from the viewing screen and appear on the screen of the client. This gave the strong impression that one had actually flung a virtual object through real space.

Internaut

After the success of Installation, I returned to an old idea that I had wanted to realize for some time—a mapping of the structures of web sites into three-dimensional spaces that could be navigated with a first-person game engine. I guessed that there would be qualities of three-dimensional space that would give some added richness to the experience of navigating the web. After finding a suitable open source game engine, Quake II, from ID Software, I modified it to use maps that I generated from the structure and content of web sites. I called the system Internaut. The resulting virtual spaces proved interesting in some regards but nearly impossible to navigate. Users of the system thought of many ways to improve the legibility of the spaces generated, but I think the fundamental flaw was the naïve injection of space into a medium that is fundamentally space-denying. Analysis of this project led me to understand the importance of retaining reference to real space.

Figure #: Internaut.

Stomping Ground

Shortly after this I got the opportunity to work with the Responsive Environments group on a richly spatial installation at the MIT Museum. An old project of theirs, the Magic Carpet, a carpet as musical instrument, was to be permanently installed in the MIT Museum, and they wanted to add a visual component to it. The carpet had a grid of piezoelectric sensor wires underneath it and two

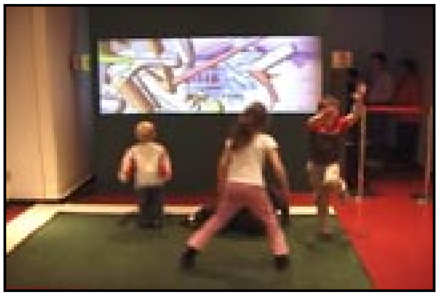


Figure #: Stomping ground.

Figure #: The NYLON microncontroller teaching platform.

Figure #: A proud workshop participant and his Hotpants.

Doppler radars to sense upper body movement. Users could control the music it made by where and how hard they stepped on the carpet and the overall speed and direction of their body movements. The system had been used in performance by dancers and had had a thorough tour of the world. It was my job to take the same sensor information that Kai-yuh Hsiao had made into music and make it visual. The resulting system, now renamed Stomping Ground, used rear-projection to present people on the carpet with greater than life size images of their own legs and feet with blobs rising out of the floor wherever they stepped. In the resulting piece, the space of the carpet was legibly translated into a virtual space in which people mingled with virtual forms.

Hotpants/LittleVision

After these experiments in screen-based virtuality, my work took a turn toward the hand-held object. I was part of a team that helped teach an undergraduate class in microcontroller design. Our advisor, John Maeda, had us create a development environment from the ground up. We called our system Nylon because we expected it to be extensible and connect to multiple hardware modules. A common problem in elementary hardware design classes is a frustrating bottleneck in actuation. No matter how interesting or exciting student designs are, they are limited in their range of actions: maybe spinning a motor or lighting a few LEDs. We decided to alleviate this problem by building for them a palm-size output device that had significant expressive range. We called the circuit Hotpants. It was a grid of 10 by 14 red LEDs each of which could be on, off, or half brightness. We wrote a display language that a microcontroller onboard the display interpreted so that students could send primitive graphics commands to the displays to do things like draw points, lines, rectangles, and circles.

For the purposes of the class, the device served as a display. But because of its size and shape, it was more than a screen. It was a physical entity to be handled and manipulated. Because each pixel was visible, it wasn't possible to forget the physicality of the device and become seduced by the image it produced. The image was always teetering on the edge of legibility, requiring the viewer to position himself in space at just the right distance to make it properly resolve.

After the class I became interested in developing the display as an independent object. It had its own processor and I supposed it could be used to store and play back small video sequences. I wrote software that allowed image sequences to be compressed and burned directly into the display. This use of the display we called LittleVision. Justin Manor wrote video software that allowed us to shoot movies with a webcam and downsample them to the resolution of the display. We ran several workshops in which participants filmed tiny movies using their bodies and large foamcore props. They got surprisingly good results. The most engaging characteristic about LittleVision was its size and weight, just large and heavy enough to feel good in the hand. It was a morsel of video, and object to which a person could become attached. Its thingness, its substance in the world was its most important quality.

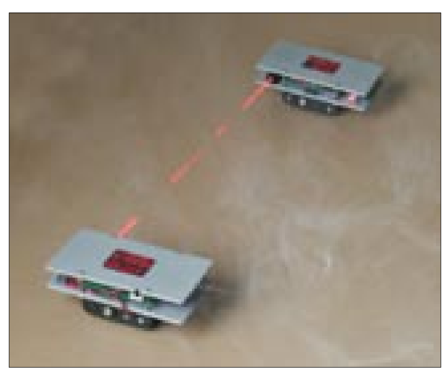


Figure #: The WordToss handhelds demonstrating pointable computing. Smoke provided by Justin Manor.

Pointable Computing

As I began to use LittleVisions, I started to think about the possibilities and implications of their communicating with each other, which led me to an analysis of the spatial qualities of different modes of wireless information transfer. It struck me that as the world moves away from wires and infra-red communication in favor of radio-frequency (RF) technologies such as 802.11 and BlueTooth, we are losing the specificity of address that a spatially directed connection offers. It is always possible to tell what a wired device is attached to—just follow the wire. And infra-red devices like remotes are aimable within a fairly narrow cone as is obvious when using a television remote. But RF communications extend almost spherically from their source, making directional intention impossible. We have to resort to selecting the objects of our intentions from a list of names or identifiers. My idea was to emphasize directionality and specificity of spatial communication over all other qualities, and therefore for my carrier of communication, I chose a laser beam, the apotheosis of directedness. I built a system for communication between devices that operates much like an infra-red transceiver, but since it is laser-bound, it is longer-range and totally pointable. This pointability and the feedback the aimer gets as a red spot on the object of control are an obvious example of the benefit of maintaining a spatial relationship with computational objects.

Figure #: EyeBox is a mini-fridge turned 3D scanner.

EyeBox

My last experiment, EyeBox, went further in the direction of integrating existing physical objects into computation than any of the previous projects. I made a simple 3D scanner out of a collection of inexpensive webcams. I used a technique called “visual hull” reconstruction, which determines the volume of an object based on the intersection of generalized cones of volume produced from silhouettes of the object taken at multiple angles around it. The technique is described more fully below. It is not capable of reproducing every topography, but it take surprisingly little sensing to produce a very good representation of many everyday objects.

As interesting as EyeBox was as a 3D scanner, it was at least as interesting as a model of a new spatial interaction with a computer. The screen in EyeBox is mounted on the door of the fridge, and the system is operated by opening up the computer and putting an object inside. The repurposing of the space inside the machine as an active space, not just the cavity containing the guts of the machine engages people. It makes intuitive sense to them that they should be able to open the machine and put things inside. It is a very pleasurable and complete interaction.

[Here I think it may be necessary to organize them visually along an axis or two.]

Chronologically, they generally proceed from most abstract and space-related to most concrete and object-based. In this order they are

Look at them as components vs. systems.

System / Components

Installation

Figure #: Installation allowed users to create virtual forms and install them permanently into real space.

Introduction

My first project, and in some ways the my most successful was Installation, a system for the creation of virtual forms and their permanent installation into real space. Installation consisted of a viewing window and stylus. A tiny camera on the back of the viewing window showed a live feed of the room behind the screen. The stylus and the viewing window were tracked in three dimensional position and orientation to calibrate virtual coordinates with real viewing position. Virtual objects created in the system responded as though they were physically in the space of the room. Once objects were placed in the environment, they stayed there in perpetuity, pulsing and growing over time.

System Description

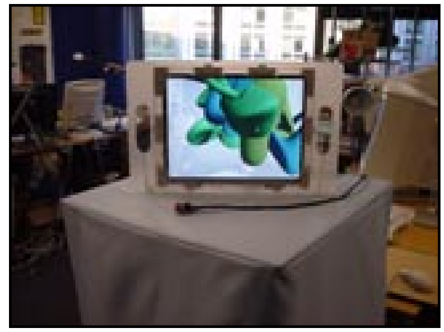


Figure #: The Installation setup in context.

Installation was an exploration in what is traditionally called "augmented reality," to indicate that rather than trying to replace an experienced reality with a virtual substitute, we are adding to an existing reality with virtual constructs. This certainly qualifies as spatial computing.

Installation presented itself as a cloth-draped chest-height table with a very light flat-screen panel resting on it, which had been liberated from its housing and placed in a translucent plexiglass frame with handles that allowed it to be held and moved in two hands. In the panel, the user could see the room behind the screen in a live video feed. This feed was coming from a tiny camera mounted in the middle of the back of the screen. The screen did not quite appear to be transparent, but somehow it was an obvious leap for a user to allow it to stand in place of his eye. Also resting on the table was the end of a long black cord with a half-inch red cube and a single button at its tip—the stylus. When a user picked up the stylus he noticed a pencil-like object that appeared onscreen and closely tracked the user's hand in space. There was no difficulty in understanding this mapping; it was a literal translation of real space to virtual space, and users spent no time disoriented by it or adjusting to it.

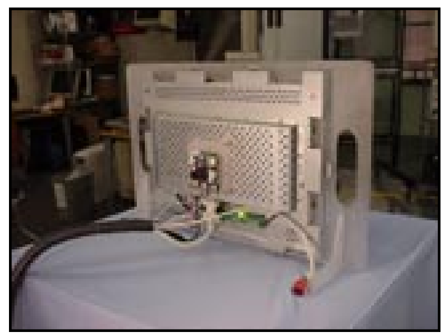


Figure #: The back of the system showing the camera.

When the user brought the stylus in front of the screen, a white haze settled over the video feed of the room as if it had become suddenly foggy. The fog cleared up if he moved the stylus behind the screen. The foggy and clear states represented the two operational states of the system, object

Figure #: The stylus.

creation, and object placement. In object creation mode, with the stylus in front of the window, when the user pressed the button, a blobby substance appeared to be squirted out from the end of the pencil-like cursor. If the user stopped moving, the blobby form continued to inflate. If the user moved the stylus quickly, the form was a thin track of his gesture, but if he moved slowly, the blob inflated in place, making a thicker form. In this way, a user had direct gestural control over virtual forms created in the system. It was easy to make pretzel-like knots or letters this way. Installation was not intended as a drafting tool, but a simple gestural sketcher for organic blobby forms. A user could add many separate blobs to a single form by stopping and starting his drawing.

Figure #: Object creation mode. The form tracks the user's gesture.

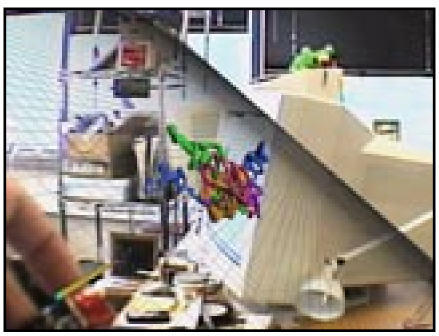


Figure #: In object placement mode, the user can throw the object into the space of the room.

Once a form had been created, if a user moved the stylus behind the screen, the pencil-cursor was shown emitting a ray of laser-like red light. This was object placement mode. The orientation of the stylus was tracked, so he could point the beam in any direction he pleased, even back toward himself. The object he created in creation mode appeared attached to the laser beam a few inches away from the tip of the pointer. Wherever the user pointed the beam, the object followed. When he pressed the button on the stylus, the object shot further down the beam. A grid appeared which helped to show the user how far he had cast the object into the scene. Otherwise it would have been very difficult to tell how far away it was, since the object was of purely invented form, and its relative size told him nothing. When the user had positioned the object in the space of the room where he wanted it, he could bring the stylus back to the front of the screen, and the blob was left floating in space wherever he put it. He could then use the stylus to create other forms to further populate the space of the room.

When a user pickde up the viewing window, the video feed moved in a predictable way because the camera moved. The virtual forms represented onscreen moved in exactly the way they would if they were truly positioned in space where they were placed. This allowed the user to move the viewing window to look at the objects he had made from any angle, even to cut through them by pushing the window through space they occupied. Through the viewscreen, the objects as seen through the window were fully fledged members of the space of the room. They floated wherever they had been put. In order to add some life to the system I gave the forms the ability to change shape and grow over time. If they were left too long, they grew out of control, filling the space of the room.

Figure #: Moving the viewscreen around causes the forms to react as if they were exactly where they were placed in the room.

The system had no representation of the actual geometry of the room. Therefore the only occlusion that occured to the objects came from other objects in the system. If a user wanted to place an object a mile away, he could, and at no point would it disappear behind the far wall of the room. This detracted somewhat from the completeness of the illusion. One of the very nice qualities of the system, however, was that it was entirely self-calibrated. That meant that it would work just as well in any space. I did, in fact, show it in a remote location, and it required no special calibration. That movable quality could be important to potential applications of the system, so it would not do to have it interact with a single pre-constructed 3D model of the scene in front of the screen. However, gathering real-time range data and integrating it into the system would be an interesting future effort.

Figure #: A client screen (outlined in blue tape) as seen through the views-creen.

I added a networked client feature to the system, by which objects could be “thrown” to other machines in the room—including the printer. To set up a client, I installed the client software, which in its default mode, simply displayed a blank white screen. I then entered a special mode on the master system (the flat panel), in which I placed a sphere into the room directly surrounding each client. I taped blue tape around the border of the monitor of each client, so that a user of the system could identify them. Whenever he was in placement mode, and he threw an object close enough to a client, it would disappear from the viewing window, and immediately show up on the client's screen, rotating slowly in space. I set up the printer as a client too, and when an object was sent there, it disappeared from the viewing window and got printed out. In this way, users actually had the sense that they were making objects and throwing them around the room.

Figure #: System diagram.

Technical details

Installation was a fairly simple piece of engineering. It had seven primary components, the PC, the client machines, the sensing system, the display, the stylus, the camera, and the software. The PC and the clients were totally ordinary Windows machines. The PC talked to the client machines over wired Ethernet. The camera was a small NTSC CMOS camera that went right to a capture board in the PC. The display was a flat-panel LCD monitor with all its housing and shielding removed. (Once such an operation is done, a flat panel monitor is a very light, wonderful thing.) It had a laser-cut plexiglass frame surrounding it that had handles for its manipulation. This frame went through two iterations, making it smaller and lighter.
The single button on the stylus, and the several control buttons on the back of the display were implemented as stolen key switches from a hacked-up keyboard—probably the easiest way to get a bunch of momentary switches into a PC.

Sensing System

The sensing system was a “Flock of Birds” from Ascension Technologies, an off-the-shelf inductive position and orientation sensing system. This system itself consisted of three separate types of unit—the signal-processing boxes, which talked to the PC via a serial connection, the base station, and the sensing coils. The base station was placed out of sight under the blue cloth. It was about as large and heavy as a brick. It emitted a magnetic field at a certain frequency. The two sensing coils, one for the display, and one for the stylus were just coils of wire wrapped in two different directions. [So how does it work? I need to ask you about this, Joe.]

Figure #: Flock of Birds diagram.

Software

All of the software was written in C++ using OpenGL for graphics. Software development fell into three categories. The first software layer processed and integrated data from the sensors, buttons and camera. The second layer acted to calibrate the virtual space to the real space to establish an appropriate projection for the viewing window. The third layer was for creating the forms themselves. I developed a method using spheres connected with Catmull-Rom splines, which provided a fast way to model and render complex organic-looking forms.

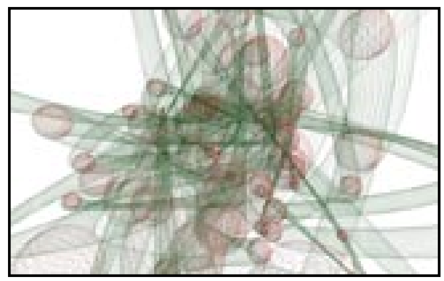


Figure #: The blobby forms were spherical nodes connected with Catmull-Rom splines.

Precedents

ARToolkit

Installation shares features with many augmented reality systems. Some, like AR Toolkit [Billinghurst, 2002], are purely vision-based. They spot known patterns in the world which a user prints out ahead of time. They infer the location and orientation of the pattern by vision algorithms, and then

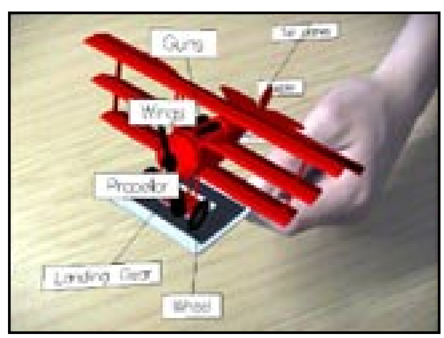


Figure #: The ARToolkit is used to composite a virtual plane into a video image. [http://www.equator.ecs.soton.ac.uk/projects/arproject/fokker-ar.jpg]

Figure #: The Diorama system [Karahalios, 1998]

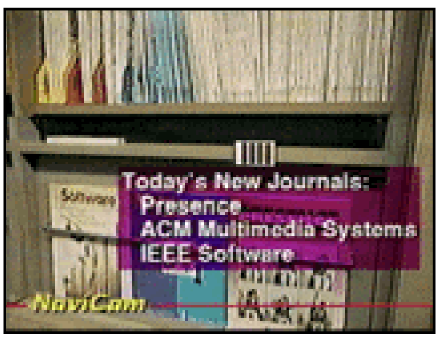


Figure #: Rekimoto's "Magnifying Glass" approach uses a handheld screen to superimpose information. [Rekimoto, 1995] [http://www.csl.sony.co.jp/person/rekimoto/navi.html]

composites a previously-defined object into the scene at the same point. These systems typically act to annote prepared scenes with prepared overlays. They do not easily allow for creation of new forms or positioning them in arbitrary places in space.

Overlay systems

Many augmented reality systems are used to display information about the world directly onto it as a kind of floating wall text [Karahalios, 1998], [Rekimoto, 1995], . Like Installation, these systems calibrate virtual coordinates to real spaces, but they are quite different in their focus and intent. Augmented reality systems call upon the virtual to annotate the real. Iconic tags or symbols appear overlaid onto scenes to indicate for instance, if there is mail in your mailbox. There is little attention to the forms or space in the virtual, or their interactions with the real, and as a consequence the virtual layer is entirely dominated by the real, appearing as little more than an intelligent heads-up display.

By contrast, Installation places more attention on the virtual than the real. If there is a subordinate world in Installation, it is the real world, which appears as a reactive underlay for a richer virtual environment. Perhaps Installation is less augmented reality than augmented virtuality.

"Eye in hand" systems

George Fitzmaurice seems to have been among the first to describe and develop systems with handheld screens tracked in space. He called these "eye in hand" systems [Fitzmaurice, 1993]. (Interestingly, he used the very same tracking device I did ten years earlier. It is shocking how little the field of 3D tracking has progressed.) It is surprising, considering that they do in fact map the eye to the hand, how intuitive the "eye in hand" model is. This is seen to be a primary advantage of the technique [Tsang, 2002]. Since 1993, there have been several notable systems for augmented reality using handheld screens. One, the Virtual Car, by Art + Com, used an overhead armature to track the viewpoint of a screen used to display a highly detailed model of a virtual Mercedes [Art + Com, 1997]. The Boom Chameleon, a similarly car-oriented device also uses a hinged rig to track the screen [Tsang, 2002]. This device traces its lineage directly back to Fitzmaurice's original concept.

Figure #: A rendering of Art + Com's Virtual Car system. [Art + Com, 1997]

Figure #: The Boom Chameleon. [Tsang, 2002]

There even appears to be a related product on the market, *WindowsVR* from Absolut Technologies in Brazil. Surprisingly, none of the other 3D augmented reality systems uses a live camera feed. As I will point out in my analysis, this was one of the most important features of *Installation*, and the easiest to implement. It is possible that they eschewed it out concern that reference to the real world would make small errors in calibration noticable. My research indicates that people are tolerant, even ignorant, of a great deal of misregistration as long as it is of the right kind.

Figure #: The WindowsVR rig has joysticks to register translation. [Absolut, 2002]

This list of precedents, most of which I was shamefully unaware of as I produced *Installation*, indicates that this work has a rich history and also an active present.

Evaluation and Critique

Installation removed the layer of spatial metaphor inherent in most graphical computing by dealing directly in the space of a room. An object created two feet in front of the user was two feet in front of the user. He was free to step around it to operate on it from the side. This kind of readjustment of viewing and working angle is exactly the kind of maneuver that we do continuously without ever thinking about it in the real world, but which we must master some interface to achieve in computational design. As Tsang points out, manipulation of viewpoint in “eye-in-hand” systems requires essentially no new learning. Furthermore, in traditional three-dimensional modeling, operations that change the position of objects viewed through the screen, implicitly change our physical position relative to the scene. But since we know that we have not moved, we must imagine that the entire virtual world displayed in front of us has reoriented without the slightest hint of inertia or other true physical effect. It makes the interaction feel cheap and unreal, and separates us from our work.

This problem with the traditional computational representation of space became obvious on watching people interact with *Installation*. They experienced delight that the objects they created behaved the way their intuition demanded they should. There was an immediacy to the interaction, which people had ceased to expect from machines. It is ironic, perhaps sad, that the operations that seemed magical to users of *Installation* are the most mundane features of our real physical lives. That lifting a viewing window and looking at a scene from a different angle was cause for wonderment, bespeaks the distressing inadequacy of typical human-machine interaction.

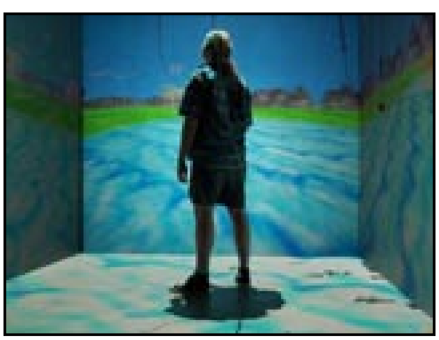


Figure #: An immersive CAVE simulation. Is this more convincing? [http://resumbrae.com/info/mcn01/session3/]

In the corner opposite augmented reality, privileging the virtual to the complete exclusion of the real are immersive virtual environments. What *Installation* called into question about these systems is whether it is necessary to jettison all of the richness and intricacy of the real world to create a convincing virtual experience. The ease with which *Installation* aroused a response from its users indicated that there is a sumptuous experiential quality to be gained by embedding a virtual world within a real one.

Forgiveness and relativity

Some human qualities that proved quite consistent over the course of my projects first became apparent with Installation. First, it was reassuring to discover how forgiving of certain discrepancies the human sensory system is. This might be expected given the tremendous extent to which our notions of a consistent reality are constructed from fragmentary sensory evidence and expectation. But it was a surprise to me. The linear algebra I was doing to reconstruct the scene as users moved the viewing window was only so good. It corresponded very roughly with what an actual window would see. Yet the illusion was fairly convincing. That had a lot to do with relativity of sensing. We have almost no absolute references for sensing anything. We gauge things entirely relatively to what else we are experiencing at the moment. This can be demonstrated in countless ways. There are color experiments that show that we perceive color values almost exclusively by value relative to the visual field surrounding a point. This is well-known to any photographer or videographer who has to take white-balance into account. We cannot perceive small global shifts in color temperature unless they happen quickly enough that we can compare them to a fresh memory.

I was fortunate also not to be overlaying virtual objects onto real objects, in which Azuma states discrepancies of 1/60th of a degree may be noticable. Instead there was a strong separation between the physical and the real objects, and I did not endeavor to tie them tightly to each other. Azuma in his survey of existing augmented reality applications notes that these discrepancies are

Figure #: Three paintings of Salisbury Cathedral by John Constable. They all use a different color palate to render the scene, but they are all convincing.

severely limiting for certain applications like medical imaging [Azuma, 1997].

Feedback

The believability of spatial connectedness was quite strong. Although the screen did not behave exactly as it would physically, it was impossible to say exactly how it was off, and it hardly seemed to matter since the misalignments were predictable, consistent, and could be counteracted by physical feedback. Azuma refers to a phenomenon called *visual capture*, in which any contradictory sensory information tends to be overridden by the visual. This effect was certainly noticable in Installation. Although the physical movement of the screen may not have exactly matched the screen's representation, the visual took precedence, and the discrepancy went mostly unnoticed.

The importance of feedback can hardly be overstated. As Norbert Weiner wrote, many control problems disappear in the presence of a human operator with sufficient feedback [Weiner, ????]. For instance, how hard should one push a door to open it? The answer is "hard enough." We don't know how hard we are going to have to push a door, so we adjust our own exertion based on instantaneous feedback we feel about whether the door is yielding. Everything is relative to momentary circumstance and experience. The feedback loops inherent in Installation, were very tight. The control of the 3D cursor onscreen by means of the stylus was one instance. The cursor was easy to control because it followed the hand directly and it provided onscreen visual feedback immediately. In fact, in object creation mode, there was an inherent spatial translation in effect that took the gesture being made from in front of the screen to behind it. Almost no user of the system

Figure #: A translation takes the gesture from in front to directly behind the screen.

even noticed it. An absolute translation became unnoticable in the face of tight feedback and good relative correlation.

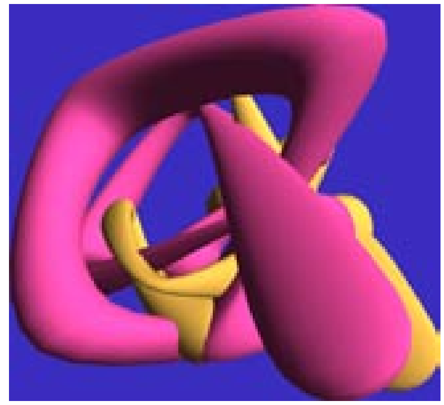


Figure #: Without the background. blobs are just blobs.

How little it takes

Another observation that became apparent accidentally during the operation of the system (when the camera stopped working) was how much I was getting from how little. All the camera provided was a live video feed of the room to be plastered behind the virtual objects. It was not calibrated or manipulated in any fashion. But the moment it was removed, the system became entirely flat. Even though users could still use the screen to view the virtual forms from different angles, the primary experience of their existing in the room was utterly gone. It was a shock, and worth remembering how powerful a simple live image can be to create context.

Figure #: Georges Braque's Fruit-dish uses many perceptual cues to give a rich illusion of depth without resorting to linear perspective.

Difficulty of depth

The challenge of conveying depth on a two-dimensional medium is ancient. *Installation* added to that discussion the capability to move the display surface through the scene. But many of the traditional problems of conveying depth remained. J. J. Gibson identified 13 different cues we use to perceive depth [Gibson, ????]. Not very many of them made it intact into *Installation*. Stereo vision, a favorite of many augmented-reality implementations, was gone. In the absence of any physical referent for the shapes, it was impossible to use their relative size in the scene as a depth cue. Almost the only things remaining to use for depth-cueing were occlusion (of the objects with themselves only), surface shading (but no shadows), and relative speed of movement in the visual field. It was this last that proved the most useful, and the illusion of depth was best when there were multiple objects in the scene at different depths and the user was actively moving the viewing window.

It was interesting also to note how difficult it was for users to draw in an unconstrained 3D environment. They were used to having the structure of a flat surface to press against when making an image. It was difficult for them to control the depth of their drawing. Often if they were drawing letters, for instance, they would be using as feedback only the single 3D view that the stationary viewscreen gave them. So they would close their shapes only to the point of visible closure in a single 2D projection. When they then moved the screen, they would see that their letters went way out of plane and did not topologically close at all. Most letters people drew were not legible from angles different from the viewing angle at which they were drawn. To the extent that this was a failure of the system to translate the spatial intention of the user, I think it should be addressed. What it represents is a failure of feedback. With enough spatial information, users could certainly close their forms. What it would require is a system that allowed for users to change their viewpoint easily as they drew so they could actively perceive their forms. This would probably best be attached to the eye so that head movement could be used in its natural way to disambiguate 3D projection.

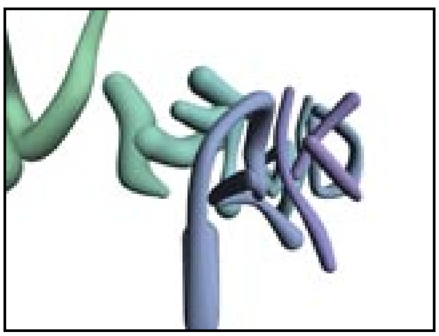


Figure #: These letters were legible from the front. I wonder what they said .

Simplicity

One of Installation's best innovations was a lack of any visible onscreen interface elements except for a cursor. This helped the system to disappear. In particular there were no graphical elements that called attention to the plane of the viewscreen as anything other than a window onto a 3D space. Any buttons, sliders, or text would have set up a virtual plane that would have been impossible to ignore. It would have distracted from the sense of pure transparency that *Installation* aspired to. Mappings were so clear and reactive that the systems driving them could be forgotten. This was achieved quite successfully in the throwing of the objects to client screens. There was a whole network architecture set up to facilitate this data transfer, but it was totally invisible to the spatial interaction, which was crystal clear.

The importance of this transparency was made obvious by its unfortunate lack in one case. One client, the printer, sat in exactly the wrong place to be aimed at by the system (way in front of the screen, behind the user). Therefore rather than have people throw their objects to the physical printer, I printed out a piece of paper with a picture of a printer on it and taped it to the wall in front of the system. It was essentially a physical icon for the printer. When people threw their objects to this icon, they printed out on the printer behind them. This separation of the icon from the actual device shattered the illusion of the object's spatial travel, and it exposed the network plumbing underneath it all. Anywhere that metaphor becomes visible, it exposes its separation from the reality for which it stands. It became an important design criterion to avoid metaphor and apparent interface wherever possible.

Future Work

A Platform for collaboration

The ideas explored in *Installation* become particularly powerful when we imagine several windows at once looking onto the same evolving environment. Then it becomes a model for

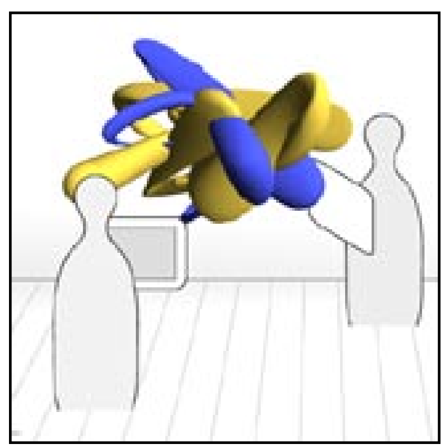


Figure #: Giving forms a shared spatial context allows them to be the objects of collaborative effort.

luxurious collaborative computation. This model is applicable to any kind of communal form-making, whether that's physical form or abstract information, meaning the ideas could equally find use in architectural design or large-systems engineering. The fundamental idea is that once a work object is placed into space it has a shared context for simultaneous manipulation. This facility is demonstrated by Tsang's system, which he explicitly proposed as a prototype for the collaborative 3D design markup and critique [Tsang, 2002].

Figure #: Microsoft Bob suggested the home as a metaphor for information organization. But it took place in a fictional iconic space.

Storage and retrieval

It is easy to imagine the ideas in *Installation* being used for storage and retrieval of information. What could be more natural than to look for something you placed in a physical location? A hierarchy of folders offers very little to the eye to act as retrieval cues. Under most conditions, we cannot even be sure that the position of an item will be constant on our screen. We spend time and energy orienting ourselves to the ad-hoc spaces that the machine tosses at us as fast as we can handle them. Instead why not let the machine orient itself to our own naturally inhabited space?

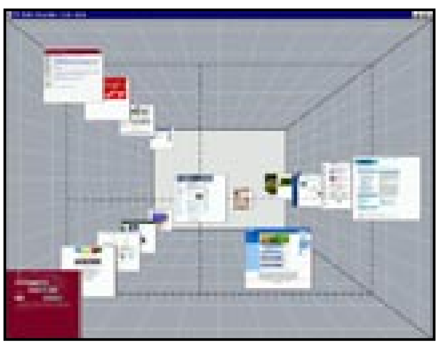


Figure #: [Dourish, 2000] studied storage and retrieval from a spatial model like this. It doesn't have much to say about our experience of real space.

There have been attempts to apply a physical metaphor to information storage, but few of them have used a real space as the containing envelope. Most of the spaces have tended to be iconic or pure raw regions of linear perspective. I believe neither one has the potential for association that a well-corellated real space has.

Figure #: Web Forager from Xerox Parc organized web data in a virtual library [Card, 1996].

Installation explores the mixing of real and virtual spaces, and in so doing, begins to fulfill the promise of models for computation that respond to our basic human facilities and intuitions.

Internaut

Figure #: A web space made into a virtual space by Internaut. A map of the area is shown in the upper right.

Introduction

After *Installation*, I turned to a slightly more abstract spatial problem. I wrote *Internaut,* a system for mapping internet structures into three-dimensional virtual environments and exploring them in a first-person game engine. As such, it did not meet the requirements for spatial computing as outlined above, but was, in fact, a project whose deficiencies were instrumental to my construction of that definition. The analysis of its flaws led directly to my understanding of the importance of spatial computing as opposed to purely virtual environments.

Technical Description

The Internet constitutes an enormous electronic architecture that defines places without regard to physical structure. We navigate these spaces with web browsers, moving from place to place with a click on a link. Internaut proposed that a physical architecture could be derived from the shape of the network and navigated with a first-person 3D game engine. This was a several-step process, which involved first making spatialized maps from web sites and then processing them into a form in which they could be virtually explored.

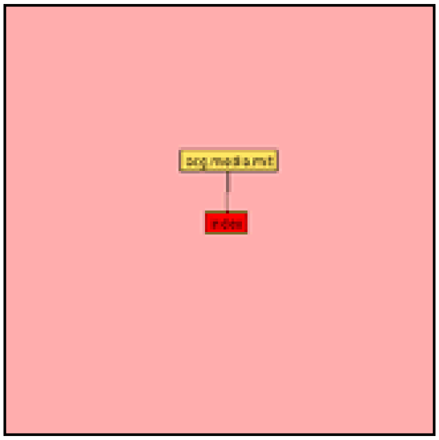


Figure #: A map begins from a web pages and trolls the links on that page.

The maps were generated starting from a given seed web page by a fairly simple procedure that guaranteed several criteria in the three-dimensional map that I deemed important for them to be meaningful. First, every page from the same site as the seed that was accessible by any path of links should be represented. Second, any two pages that linked together should be immediately accessible to each other. There are numerous ways to design a process to do this, but the one I implemented relied on a simple physics simulation running in Java.

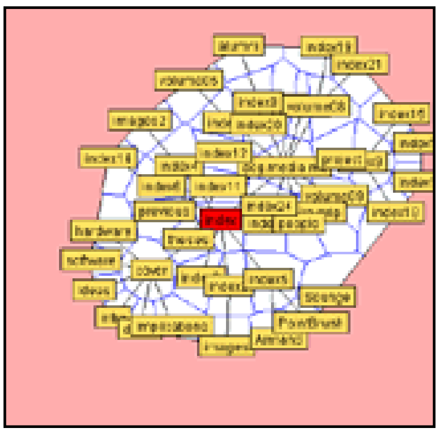


Figure #: A map grows. The root node is shown in red.

The first page was represented by a node in a 2D graph with a point location. All links on this page to pages at the same site were traversed in order, and these sites were added to the graph as nodes with springs connected to the root node. These simple simulated springs pull nodes together with a force proportional to their length plus a constant factor for their rest length. It should be no surprise, that these new nodes, which are added to the graph at random locations settle into a ring around the root site. A user was allowed to click and pull on any node in the graph at any time. All springs stretched to accommodate such manipulation, and snapped back into a relaxed configuration when released.

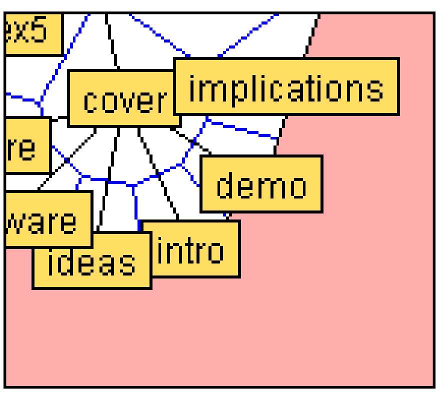


Figure #: In this detail we see that the page "cover" links at least to pages "demo," "intro," and "ideas." These are connectect by springs (black lines), which will punch doorways in the walls of the rooms (blue lines).

Each new page was then processed in the same way as the root node in the order in which it was added. The resulting network of nodes connected with springs was a stretchy gyrating mass that was constantly attempting to relax into the lowest energy state consistent with its topology of connections.

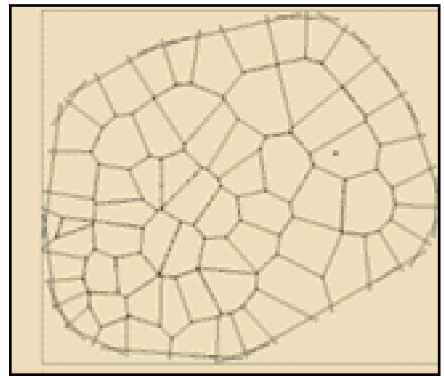


Figure #: The map is then processed in a Quake map editor.

The nodes were then separated from each other with walls that were the divisions of a Voronoi diagram. A Voronoi diagram associates each node with the area surrounding it that is closer to it than to any other node. This is always a lattice of convex polygons surrounding each node, guaranteeing that each node gets some share of physical space. The springs connecting the nodes intersected these Voronoi-generated walls at many points. Anywhere they intersected, a doorway was drilled in the wall, insuring that any link became a navigable path from one cell to another. This structure successfully located pages in a 2D map close to pages to which they were linked. Obviously there are linking conditions possible in web sites that are not possible to represent in a 2D map with strict adjacency, but the method guarantees that these will be pulled together more strongly the further they are separated, so it does a good job of creating spatial representations of web structures.

Figure #: I then had to demilitarize the game.

The next task was to go from a map in this Java application to a map usable in a 3D game engine. I chose a modified form of the Quake II engine from ID Software because it is now a mature open source project. I generated a map file for this engine with the largest image on any page tiled onto its walls like repeating wallpaper. This surface image was the only distinguishing feature of any room. I undertook extensive changes to the engine to demilitarize it, removing the guns and gangs of monsters bent on killing the explorer, and

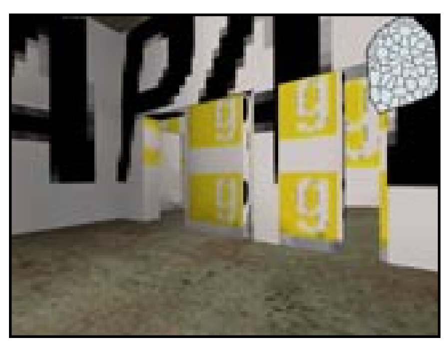


Figure #: After removing the gun and adding a mapping feature.

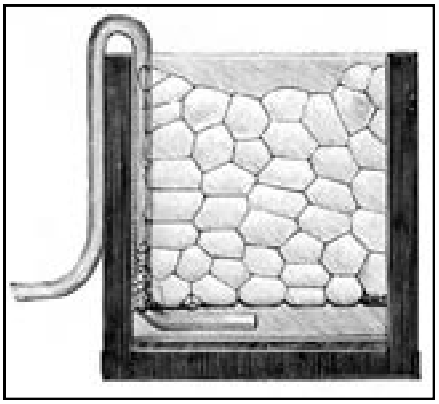


Figure #: Soap bubbles make voronoi patterns. [Boys, ????] Referenced from [www.snibbe.com/scott/ bf/ bubbles.htm]

Figure #: Scott Snibbe's Boundary Functions [http://www.snibbe.com/scott/ bf/]

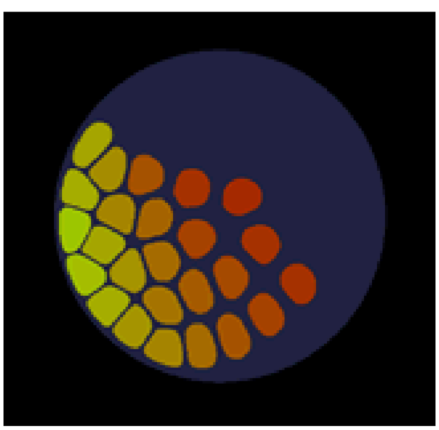


Figure #: Jared Schiffman's honey. [Shiffman, 2000]

adding a mapping feature which displayed a map of the entire site onscreen at all times. I retained the engine's capability to run in a networked mode in which multiple players could explore environment together, seeing each other, and even communicating via typed messages.

I showed the project repeatedly, letting users select the starting web site and then allowing them to navigate the resulting three-dimensional map. As I watched them try to orient themselves to this remapping of internet space, I became aware of many things that would inform my future work.

Precedents

For the self-organizing map component of the project, I had many good precedents. This sort of problem has interested scientific and artistic communities for a long time. Voronoi diagrams have broad application to many problems in analytic geometry and self-organizing systems. For instance they can be used to position nodes in self-organizing neural networks [Suanders, 2001]. And they arise naturally in many situations in which surface energies are being minimized as in soap bubbles. They appeal to computational artists and designers for their organic appearance and ease of production. Jared Shiffman used them for their organic visual quality in *Honey*, an exercise in cellular form [Shiffman, 2000]. Scott Snibbe used them for their partitioning ability in *Boundary Functions*, in which participants stepping on a platform are automatically separated from each other by boundaries projected from above [Snibbe, ????].

Simulated springs are even more commonly used in computational design. They lend movements a squishy, organic feel. Benjamin Fry has also used springs to organize web spaces in a way very similar to mine in *Anemone*, which tracks web traffic as a continually evolving network of nodes representing web pages, connected with springs [Fry1, 2000].

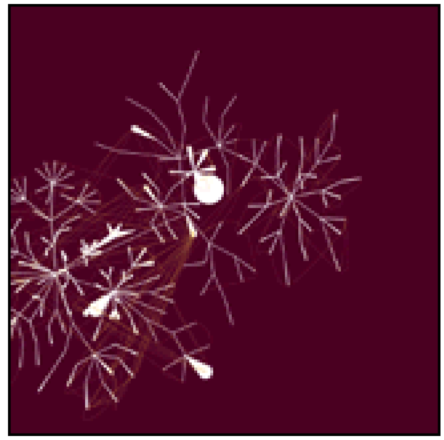


Figure #:Ben Fry's Anemone [Fry1, 2000].

Ideas of organic form and self-organization have become popular in theoretical architecture in recent years. Greg Lynn uses such forms as "blobjects" in his designs. Very little of such architecture has been built, and it may be for good reason—such spaces are very difficult for us to understand as we are used to understanding traditional architectures with choreographed hierarchy and sequence.

Figure #: A study model of Greg Lynn's.

Mappings of non-spatial networks into virtual spaces are not new either. Apple briefly promoted a 3D flythrough technology called Hotsauce for web page meta-information. AT&T Research produced a system called CoSpace, which used an additional layer of VRML on top of existing web pages to represent web spaces.

Figure #: Apple's Hotsauce meta-content 3D web flythrough plug-in. [http://www.inxight.com/news/apple_initiative.html]

Other networked virtual environments were designed spatially from the beginning. Certainly networked first-person shooter games like Quake III Arena have been successful. It is easy to convene teenage boys in a virtual space with the lure of their being able to shoot each other with impunity. We are currently experiencing a small explosion of nonviolent networked virtual environments that are not meant to represent existing web spaces, but to establish parallel virtual Internet spaces that are constructed and inhabited by a broad public such as the *Sims Online* [Electronic Arts, 2003], *Second Life* [Linden Labs, 2003], and *There* [There, 2003]. Several systems like these already exist, but do not find wide use. A strange feature of the places that users construct in these virtual environments is that they mimic structures in the real world. They put "roofs" on their "houses," for instance.

Figure #: CoSpace, a 3D web browsing system from AT&T research. [Selfridge, 1999]

Figure #: The Sims Online. [Electronic Arts, 2003]

Why? There isn't any rain, or in fact weather of any kind to defend against. And enclosure represents no protection from outsiders. It must be a desire for familiarity that drives people to make designs that take no advantage of the liberation that they might experience in these worlds without physical limitation.

Evaluation and Critique

Users enjoyed wandering around the spaces generated by *Internaut*, but found them confusing and difficult to navigate. Even with the help of a map, they had difficulty finding their way around or remembering where they had been. I think there were several probable contributing factors.

First there was the elimination of all text. I expect people would have difficulty navigating any web sites that had all the text removed and left only a single image to demark each page. Links would no longer describe their destinations in words, but be tiny thumbnail images of the image on the page they linked to. Navigating spaces like this would, I expect be somewhat bewildering too.

But even in the absence of text, there was a difficulty in navigating the structure due to its unfamiliar and inhospitable physical structure. There is a reason that we do not construct our building plans as Voronoi diagrams. The spaces that these generate tend toward spatially undifferentiated rotundas of doorways that make it impossible to identify a dominant spatial axis. Even when there is one, it is not shared by any adjacent cells. Under such conditions, it is often impossible even to identify the portal through which one entered a space.

We are careful in architectural plans to define circulation space. We do not expect rooms to function both as destinations and corridors for movement at once. The Voronoi plans make no such circulation. There are no clear means of passage between spaces that do not directly abut. To get from one end of the space to the other it is necessary to turn at every room, potentially even away from the final destination. There is no organizing logic that makes the space serve an intention other than aimless wandering.

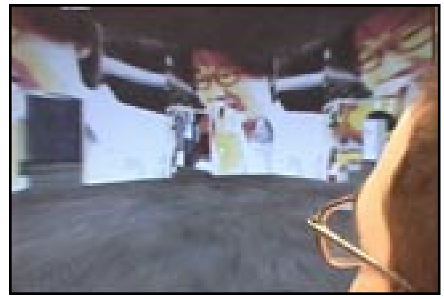


Figure #: Internaut tended to offer the user a bewildering array of self-similar doorways.

Use of an organizing geometry other than Voronoi could potentially help this. There are experiments in grammatical architectures that could help point the way to saner structures [Brown, 1997]. That is one possibility for future research. These geometries might allow for the use of more information from the web sites than simple topology. It should be possible, for instance, to identify the primary entrances to the web site. These should represent entrances to the virtual space as well. (In the existing geometry they are most likely to be buried at the center and surrounded by a ring of ancillary pages.) It is likely that some links from a page are more dominant than others—larger text or higher on the page. These should be represented by larger openings or grander access.

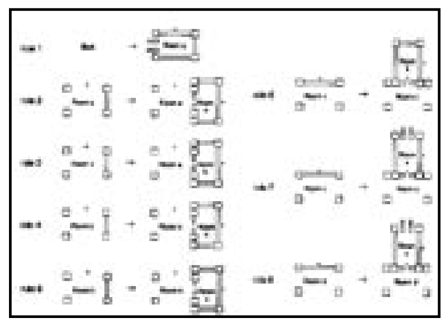


Figure #: Rule-based design from Gero [4.290 Production Systems, Fall 2002].

Another possibility is that part of what makes the Internet successful is that it is fundamentally non-spatial. Certain conditions of spatiality do not apply to it. For instance there is no such thing as a one-way connection in space. There are doors that lock from one side, but adjacency is commutative. Not so in a non-spatial network. One page may link to another that has no idea of the existence of the referrer. This network of one-way streets has the tendency to channel users toward sites that are commonly linked to. These have a higher chance of being useful than the sites that are seldom referenced. There is also a trail of breadcrumbs that web-surfing leaves that a user can always use to backtrack via the "Back" button. No such facility exists in real space, although it could be simulated by having movement leave a trace in a virtual environment.

The most damning concern may be that the fundamental property of Internet space is the collapse of distance. Distances are measured in the number of clicks the path takes, and a long one may be three. This implosion of space is necessary to what makes the Internet a useful complement to the real world. An advantage of shopping online is that every store is equidistant at a distance of one click, or the typing of its address. In order to spatialize this condition, it would require a bewildering portal—a spherical mall with thousands of openings that would be a thrilling sight, but hardly useful. It must not be necessary to cross any space to have access to another. Once the intended destination is identified, the need to "walk" there only represents wasted time. Access must be as fast as the delivery of information will allow. So perhaps the idea of a spatial internet is fundamentally flawed. Cyberspace as Jean Baudriallard puts it is

> Where all trips have already taken place; where the vaguest desire for dispersion, evasion and movement are concentrated in a fixed point, in an immobility that has ceased to be one of non-movement and has become that of a potential ubiquity, of an absolute mobility, which voids its own space by crossing it ceaselessly and without effort. [Baudillard, 1988, p. 32]

In a study of the necessity of legibility of virtual spaces, Ruth Dalton concludes that global intelligiblity is not important in systems such as the web where that structure is not used for navigation. Web space developed without any need for an intelligible global structure, and to try to impose one is likely a fool's errand.

Future Work

Lots of the issues raised in the first part of my critique could be addressed with sufficient further work. We could try to generate rule-based architectures that are more legible and easier to navigate. While I think the program of virtual representation of Internet spaces has something to teach us, I do not think it is generally useful outside of its value as an idea with cultural resonance. People suggest that it would be a good shopping interface, in which a user could walk around and see merchandise disposed around a space while talking to others virtually browsing with them. That is a possibility, and I think it would initially be exciting to some, but I don't think its long-term effectiveness would be any greater than nicely displaying merchandise on a web page. The Sims Online may well succeed, but I believe that that will have more to do with its nature as a game than as a networked space. Remeber that the non-online version of the Sims was wildly popular too. I have come to believe that there is more interesting territory to explore in the realm of spatial computing, in which the spaces involved are real spaces that the user already has attachment to and experience with.

Stomping Ground

Figure #: A kid engrossed in Stomping Ground.

Introduction

Stomping Ground is a permanent installation at the MIT Museum consisting of a musical carpet and a projection of live video with superimposed blobs. It is a collaboration between Joe Paradiso director of the Responsive Environments group at the Media Lab, who made the carpet and the radars, Kai-yuh Hsiao of the Cognitive Machines group, who wrote the music, and myself, who designed the space and programmed the visual interaction.

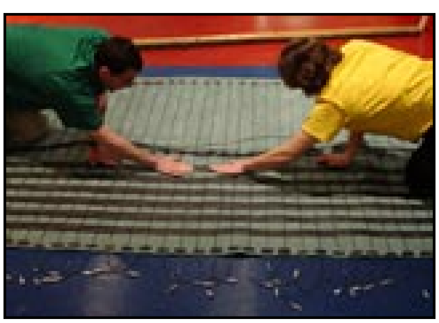


Figure #: Rewiring the carpet with piezoelectric wires. [Photo by Stephanie Hunt].

System Description

The carpet tracks the location and intensity of footfalls with a grid of sensors. Doppler radars mounted on the sides of the projection wall track the overall direction and intensity of upper-body motion. This information is used to create a musical composition that has two modes: one has a richly layered ambient sound, and the other is agressively percussive. The same data is fed to the graphics system, which produces blobs that grow upwards from the locations of footsteps. The blobs are superimposed on a live video image showing the legs and feet of people on the carpet (whole bodies of very small people). The video and the forms in it are warped by a virtual fluid simulation, which is stirred by stomping and upper-body activity.

Figure #: Kids enjoying the carpet.

Background and Related Work

As should be the case in the extension any good work, the prior work served as my foremost precedent. Prior to my involvement, the carpet had been exhibited as part of exhibits on musical instruments and hosted dance performances. I studied footage of these events, the sound and code of the music-making, and the technology behind the operation of the carpet. [Paradiso, 1997].

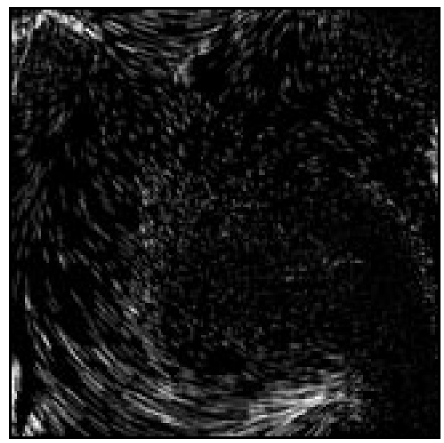


Figure #: I implemented a fluid flow model from [Ventrella, 1997] to warp the video image.

One of the music's modes has a watery background sounds, which led me to give the graphcis an undersea feel. I used an intuitive 2D fluid-flow model by Jeffrey Ventrella to warp the projection based on flow induced by "forces" from the radars [Ventrella, 1997].

The blobby forms I adapted from *Installation*, connecting their nodes with springs, and subjecting them to reverse gravity, which pulls them up from the base of the display and out of the picture.

Evaluation and Critique

It was an interesting challenge to come into a project that already had such a thorough life independent of visualization. I wanted both to fit into the framework as it existed—the expressive qualities of the music, the two modes—but I wanted also to make my portion of the project my own. I wanted the visual component in the end not to be not to be separable from the whole experience.

Invisibility

Stomping Ground represents an intermediate step in the integration of physical and virtual environments. The real space of the carpet is represented on the screen while virtual artifacts swirl around on top. It is an augmented and distorted mirroring. Unlike the direct and obvious form-making control users have with *Installation*, in *Stomping Ground*, the link between behavior and form produced is less obvious. More was being decided by the system, making the system itself more present as an agent. As much as it was a goal of *Installation's* to make the system invisible, it was a goal of the *Stomping Ground's* to become a focus of attention. It was the exhibit as much as the forms and sounds made by it. In that way it blurred the line between instrument and artwork.

Hotpants/LittleVision

Figure #: A bunch of LittleVisions running tiny movies.

Introduction

Hotpants was a handheld display device originally designed for use with the NYLON microcontroller system [nylon.media.mit.edu], which we produced to teach basic microcontroller design to undergraduates. Then as I became interested in the display's potential for autonomous operation, I untethered it from NYLON, renamed it LittleVision, and began to use it as a standalone device for the recoding and showing of short video segments.

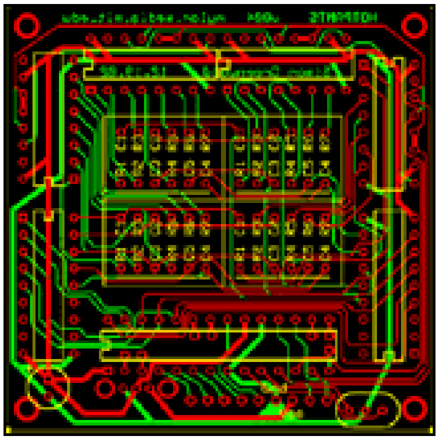


Figure #: A bunch of LittleVisions running tiny movies.

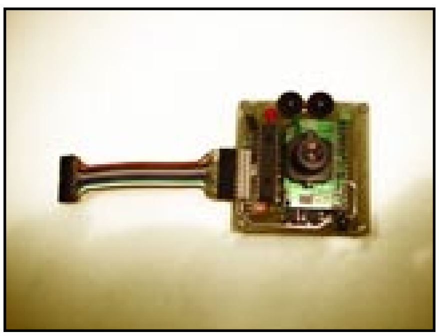


Figure #: A standalone camera board turns LittleVision into a self-contained tiny videocamera.

System Description

Hotpants/LittleVision consists of a very simple circuit which uses a PIC microcontroller to drive four shift registers and two current source chips, which in turn drive a matrix of 10 X 14 red LEDs. These LEDs can be set to display at full brightness, half, or off. The board exposes a set of programming pins, which are used to connect the board to a PC for downloading of new movies. The board stores about 300 frames, depending on how well they compress, and plays them back at 12 per second, for a total of 25 seconds of video. After this period (or shorter if the movie contains fewer frames), the movie loops. I have recently developed a second board, a camera board, which can be used to record movies directly to the LittleVision without the use of a PC. (It is functional, but not yet fully debugged.)

The circuit and its components are quite inexpensive, and were designed with that criterion in mind. There are much nicer display elements available than these red LED arrays, but they are all more costly. We have run several workshops in which participants film movies of themselves or other props and then burn them to the devices and take them home. In one two-day workshop, we had participants build their boards the first day and make movies the second day.

Technical Details

Hardware

The whole circuit is controlled by a PIC 16F876 microcontroller running at 20 MHz. It has 22 usable I/O pins. We are using it to drive four 5 X 7 LED arrays. The LED elements in the arrays are referenced by row and column, so we do not have

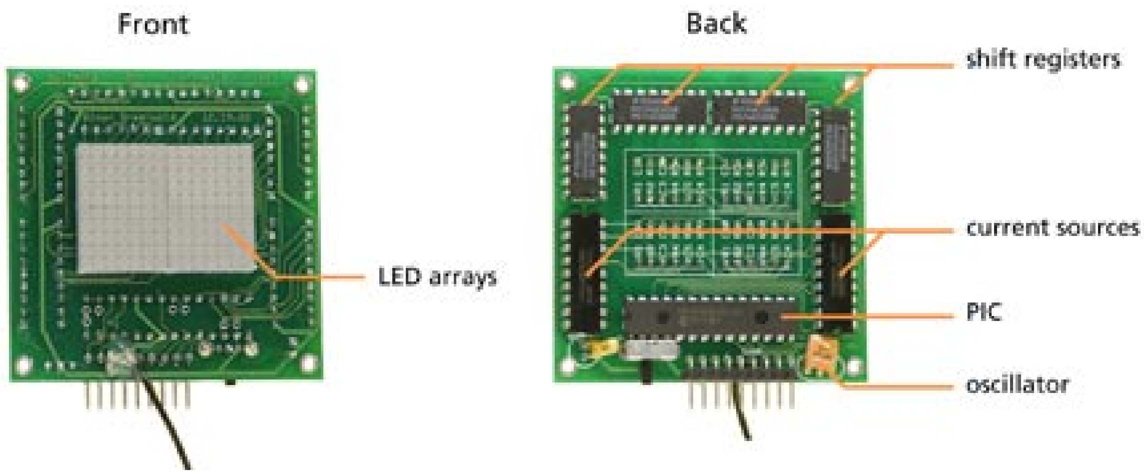


Figure #: Annotated images of the circuit innards.

simultaneous unique access to each one. Basically what we have to do is turn on one column at a time and light each row that is on in that column. Then quickly switch to the next column, and so on. That means that each column is only lit for a fraction of its possible time. This is sad, as it cuts down on brightness, but unavoidable. We do, however, play one nice trick, which is to treat the four arrays as two tall columns rather than one large array. That way we can control each LED while keeping the columns lit 1/5 of the time rather than 1/10, effectively doubling the brightness. (This may make more sense on inspection of the PIC code that drives it. [Appendix C])

Unfortunately, that means that we have to control two columns of 14 LEDs independently. So with 10 columns and 28 effective rows, we are saddled with a burden of 38 outputs, which we know the PIC can't provide by itself. So we use shift registers. Shift registers turn serial outputs parallel by piping clocked values to their output pins on a specific signal. So we hook up 4 shift registers in series, and end up with 32 extra outputs controlled by 3 pins on the PIC (data, clock, and output enable).

Finally we have a potential problem with constant brightness. We want all of the LEDs to be equally bright, but the PIC has a limited ability to sink or source current, which means that when it's lighting 14 LEDs at once, they'll be dim, and when it's lighting one, it'll be bright. So we run the PIC column outputs through a Darlington current source chip to give it muscle.

Software

There are several different incarnations of software for Hotpants because it has been used in a bunch of different contexts. All of the software for Hotpants to date has two components, one on a PC and one on the board. A system by Megan Galbraith allows you to write programs in the Nylon language and send them to Hotpants. A setup by Simon Greenwold lets you take movies with a webcam and send them to the board. The software on the PC side is different, and so is the firmware on the PIC. It is helpful to burn a bootloader onto the PIC ahead of time so that you can download different programs to it to change its functionality.

Figure #: Justin filming a tiny movie.

The basic operation of the firmware on the PIC is to change the values in the display buffer over time. That becomes an animation. The actual refresh of the screen column by column is done by timed interrupt, so it remains consistent no matter what else is going on on the PIC.

We get three pixel levels (ON, HALF-ON, OFF) by using two alternated screen buffers. A pixel that is half brightness is on in one buffer and off in the other. That way it gets half duty cycle. (Actually it only gets 1/3 duty cycle because we display the second buffer two times out of three. That was just because it made the contrast between all-on and half-on better.)

Precedents

Interestingly, precedents for Hotpants are somewhat hard to find. It seems that existing technologies are always either more or less than Hotpants. Handheld displays that do more than Hotpants/LittleVision are everywhere. These are on PDAs and the backs of digital cameras. There are begining to be backlit LCD picture frames sold, which are somewhat similar in spirit to Hotpants, but deliver more image fidelity than object-relationship. Products less than Hotpants are the LED array components themselves, which come in a huge variety of sizes and configurations but have no built-in control circuitry to drive them.

Figure #: A digital picture frame from Ceiva. [http://www.ceiva.com/]

Pixelated LED displays are everywhere as banners, and even architectural surfaces. People are starting to have video displays as small as watches. But all of these try for an imagistic resolution. Jim Campbell is an artist whose work with LED arrays explores pixelation, motion, blur, and form. His pieces led me to realize that putting a blurring filter over a highly pixelated display makes the image easier to decipher. His pieces also demonstrate how much recognition we get from motion.

Figure #: The Nasdaq exchange in New York has a full color LED wall.

Evaluation and Critique

Hotpants/LittleVision was a radical departure from my previous work. It brought my attention to the realm of the handheld object, a scale which allows users to form totally different kinds of attachments than room-sized environments. And interestingly, what LittleVision did was essentially compress room-scale activity and place it in the hand as a small electronic brick with a pleasant heft. Participants had a connection with the scenes they were filming, and then immediately thereafter to hold them in their palms was a very different experience than it would have been to see them on a television screen, or even on the LCD panel of a handheld video camera. This difference had a lot

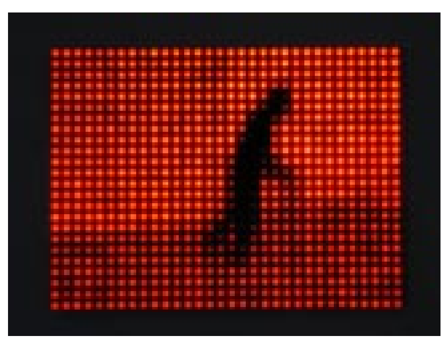


Figure #: From Motion and Rest #5, Jim Campbell, 2002. [http://www.jimcampbell.tv/]

to do with a level of abstraction that the limited resolution enforced.

Figure #: Can you tell what this movie is about? (Hint: It swims in the ocean and has big sharp teeth.)

10 X 14 is not very many picture elements. Complex scenes are not recognizable. Typically no more than two large shapes are intelligible at once. This forces an act of imagination onto the experience of viewing a LittleVision, that, like the cartoon rendering discussed above, removes the distracting quality of near-perfection. The viewer can slip in and out of seeing figure or ground or even individual pixels. This slippage is also tied tightly to the distance at which the object is viewed, which makes people experiment with it, bringing it close to their faces or holding it as far away as possible.

As with Campbell's work, scenes that were impossible to understand would sometimes snap into focus when they started to move. Interestingly, it was also motion that brought out the sharpest qualities of depth in *Installation*. It seems that human perception owes a lot to motion.

.does not happen with a handheld computer such as a Palm or PocketPC. The screens on these are designed not to confuse the issue of pixel versus image. They display images as faithfully as they are able at high enough resolution so that they are instantly recognizable. Their displays are primarily surfaces of interface, which take up as much of one side as possible. The interfaces draw the user's attention to a flat space of text and buttons, which totally overpowers the substance of the object itself. Like an anorexic, they are always fighting their physical existence, trying to become thinner and lighter. They are rectangular to overlap the palm in one dimension. This makes it impossible to fold one's thumb down across the top of them—the natural desire for holding palm-sized objects. They are held like a stick, not a rock. There is something

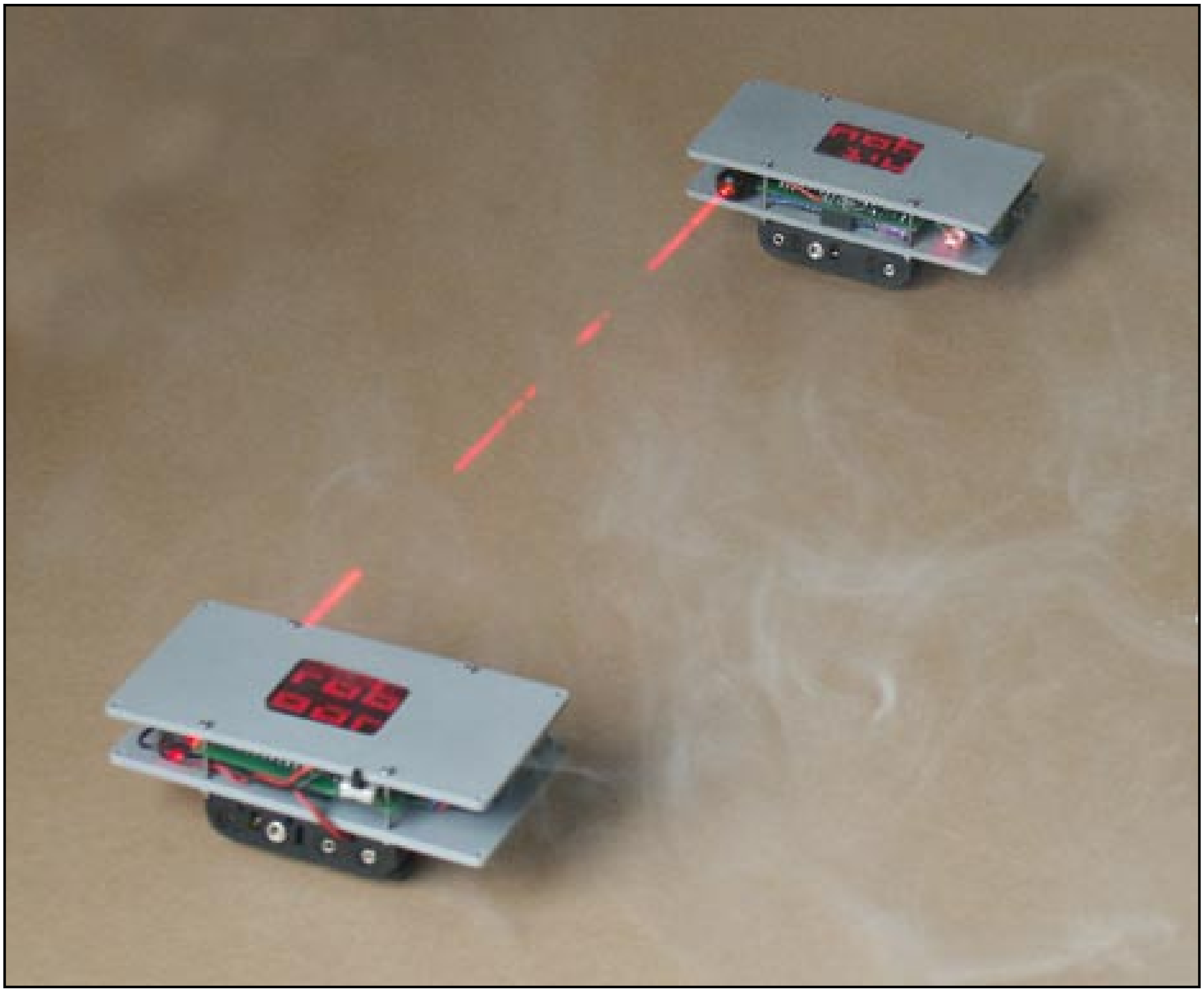


Figure #: Word Toss handhelds sending information over a visible laser.

Pointable Computing

Introduction

One way to understand remote communication is as a battle with the separating qualities of space. AT&T's old slogan "Reach out and touch someone," made that explicit. The phone was to be an electronic prosthesis for contact. But it has not only been long distances that we have put effort into nullifying. The "remote" in remote control typically connotes no more than 15 feet. This kind of spatial collapse attempts to bring things just out of the sphere of reach into contact with the fingers. It functions as an extension of touch, and most remote controls resemble the kinds of interface we would expect to encounter on an appliance itself. This is not an interaction about communication, however. It is strictly about control, and it operates unidirectionally.

Remote control has moved a step further in recent years to encompass remote data access. This has pushed the technology beyond the capacity of infra-red communication and into radio-frequency territory with 802.11 and BlueTooth. The spatial idea behind these technologies is different from the spatial singularity model of telecommunication and remote control. Instead, these technologies are proposed to replace wires. Wires are simply physical connectors designed to carry signals. They do exactly what their shape implies. It has been possible until recently to tell what a machine is connected to by tracing its wires. Suddenly the wires are going away, and it is totally unclear what connections are being made from machine to machine. A useful assumption may be that everything is connected to everything. There is no disconnect to make any one particular connection significant.

And that is a problem. Now that we have essentially conquered spatiality with communication technology, we are left floating in an undifferentiated spacelessness. True we may have eliminated the need to crawl around to the back of our machines to plug in devices, but we have replaced that inconvenience with a new burden of reference. We must assign everything we want to communicate with a unique identifier so that we can select it from a list of things in range of communication. We have essentially become like our machines, who have no notion of directionality or focus, and therefore must refer to things by ID. This is not a humanizing direction of development.

What I proposed in Pointable Computing was a solution to this crisis of nonspace in wireless communication.

Figure #: Word Toss handhelds sending information over a visible laser.

Description of the system

Pointable Computing was simply a handheld system for remote communication over visible lasers. It was the absolute epitome of directed communication. Until I learned spread the beam slightly, it was so sharply directed that it was hard to use at all. The purpose of the project was to explore the possibilities and experiential qualities of highly-directed communication and contrast it with undirected technologies.

Technical description

The system consisted of two handheld devices equipped with laser-diodes and phototransistors for sending and receiving of signals. I spread the beam slightly with a lens system to make it easier to control for distant targets and eye-safe. Each handheld had a display board (a repurposed Hotpants display) a single button and a control wheel. I also made a standalone wall-mounted receiver with three Hotpants displays. Each of these systems was driven by a PIC microcontroller.

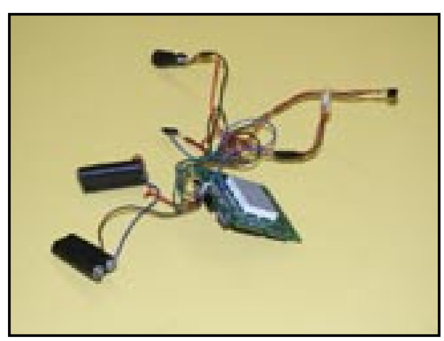


Figure #: Word Toss handhelds sending information over a visible laser.

The proof-of-concept application I designed for the devices I called Word Toss. Each handheld showed two words stacked vertically, a transitive verb on top and a noun on the bottom. In one of the devices, rolling its wheel changed the verb, and in the other device, it changed the noun. Each device’s laser was on by default. When the devices were aligned, their lasers hit the other’s receiver,

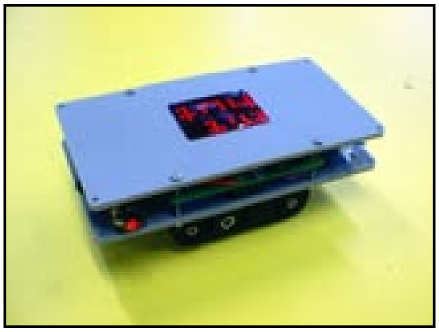


Figure #: Word Toss handhelds sending information over a visible laser.

and a pixel in the top right of the receiving device would light to indicate that it had acquired a signal. When either device's button was pressed, its laser was pulse-modulated to send a message to the other device. The message in word toss was simply the verb or noun selected with the wheel. The other device received the message and changed its word to match the word sent. It was also possible to use the handhelds to send words to the wall-mounted device, which displayed them. I was successful in sending messages from at least 30 feet away.

Background

Pointable Computing draws on a rich history of research and application in several fields including virtual reality, HCI, tangible interfaces, electronic communication, and networks.

[Do this properly.]

[It will be necessary to gain an understanding of the role of the human being in a computational environment. This will entail reading about theories of technology, interface, information, and virtuality. On the technical front, I will need to ground myself in distributed computing, optical networking, and the history of machine pointing and locating, from the earliest mice to six degree-of-freedom trackers and GPS. Gesture recognition systems, such as "Put-that-there,"[3] [FIGURE] will be important point of reference. Virtual and augmented reality systems will be necessary to study as a competing approach to the integration of space and computation.]

Use Scenarios

I developed several use scenarios to illustrate possible applications of pointable computing. They are somewhat more utilitarian than imaginative.

Universal remote

The most obvious use of Pointable Computing would be to make a universal remote. Pointing the device at any enabled object would turn the handheld into a control for that object. On the face of things, this seems to be a rather mundane application, and one that seems to run counter to the program of endowing objects with individuality and escape from metaphor. But this kind of control can bring autonomy to a previously overlooked device.

Figure #: Word Toss handhelds sending information over a visible laser.

Speakers are a good example of disenfranchised objects. Since they are the source of sound, it would make sense that to control volume you would manipulate them directly. This isn't, however, the case. Instead we have to reach to a separate box covered with controls and turn a knob. We know this drill because we have learned it, but it makes sense only if understood as a case for efficiency—all the controls are centrally located to save you the footwork of walking to your speakers and to save money in manufacture. If the speakers were outfitted with pointable sensors, they would be controllable from anywhere they were visible as fast as you could point at them. They would enjoy finally being addressed as the agents of soundmaking instead of the slaves of a central console. This kind of distributed object autonomy is exactly the condition that Pointable Computing facilitates.

Active Tagging

Imagine yourself walking down an aisle of products. You see one you would like more information

Figure #: Word Toss handhelds sending information over a visible laser.

about or two you would like to compare. You point your handheld device at them and they transmit information about themselves back to you. Why is this different from giving each product a passive tag and letting an active reader look up information in a database? Again the answer is about autonomy and decentralization. If the information is being actively sent by the object scanned, it does not need to be registered with any central authority. It means that no powerful agent can control the repository of product information, and anyone can create an active tag for anything without registering some unique identifier. Note also that in this scenario we see the likely condition that a non-directed wireless communication like BlueTooth would be useful in conjunction with a Pointable. The two technologies complement each other beautifully.

Figure #: Word Toss handhelds sending information over a visible laser.

Getting and Putting

In a vein similar to the Tangible Media Group's mediaBlocks project[2], it would make sense to use Pointable Computing to suck media content from one source and deliver it to another. Here again it is not necessary to display much on the handheld device, and one button may be sufficient. An advantage in media editing that the Pointable has over a block is that there is no need to touch the source. That means that it would be possible to sit in front of a large bank of monitors and control and edit to and from each one without moving. It may even make sense to use a Pointable interface to interact with several ongoing processes displayed on the same screen.

Figure #: Word Toss handhelds sending information over a visible laser.

Instant Wiring

In this simple application, the Pointable is used simply to connect together or separate wireless devices. If, for instance, you have a set of wireless headphones which can be playing sound from any one of a number of sources, there is no reason you couldn't simply point at your headphones and then point at the source to which you want to connect them.

Sun Microsystems likes to say, "The network is the computer." This is a fairly easy formulation to agree with considering how many of our daily computational interactions are distributed among multiple machines. Any form of electronic communication necessarily involves a network. The shrinking and embedding of computation into everyday objects implies that informal networks are being created in the physical fabric of our homes and offices. If we assume that the network of wireless devices around ourselves is essentially a computer, we must admit that we spend our days physically located inside our computers. Being located inside the machine is a new condition for the human user, and it allows the possibility of directing computation from within. A pointing agent, a kind of internal traffic router, is one potential role for the embedded human being.

Figure #: Word Toss handhelds sending information over a visible laser.

Reactive surfaces

Reactive surfaces are building surfaces, exterior or interior, covered with these changeable materials coupled to arrays of pointable sensors. They make use of new materials that have changeable physical properties such as LCD panels, electrochromic glass, OLEDs, or electroluminescents. It would be possible, for instance, to write a temporary message on a desk or wall or define a transparent aperture in an otherwise shaded window wall. Such an aperture might follow the path of the sun during the day.

Analysis and Critique

Pointable Computing takes as its starting point an emerging reality in which everyday electronic devices communicate wirelessly. These devices already have identities tied to their functions, be they headphones, storage devices, or building controls. They are not crying out for an additional layer of interface. How can we address the new capacity of things to talk to each other without further mediating our relationships with them? We need the remote equivalent of touch, an interaction focused on its object and containing its own confirmation. Pointable Computing offers that by way of a visible marker, a bright spot of light. You do not need to consult a screen to determine if you are properly aligned. It is apparent. The receiver may also indicate that is has acquired the beam, but that indication will always be secondary to the visual confirmation that the object is illuminated.

The system did feel substantively different from existing modes of wireless communication. And its primary difference was its spatial specificity. It felt much like using a laser pointer, which has a remarkable quality of simultaneous immediacy and distance. This I believe is due to its antiphysical quality of tremendous length with infinite straightness and lightness. It is like an ideal rod. Also like a physical pointer, it is usable because it offers feedback. As can be demonstrated by a game of “pin-the-tail-on-the-donkey” we cannot point very well without continuing to reference what we are pointing at. A laser spot is the perfect feedback for pointing—ask the military.

As Norbert Weiner pointed out, any system containing a human being is a feedback system. As a user, a person automatically adjusts his behavior based on the overall performance of the system[1].

What makes the Pointable Computing a robust communication system is that the feedback loop containing the human being is direct and familiar. The human eye has an area of acuity of 1–2°, implying that narrow, beamlike focus is the norm, not the exception for human perception. The rest of the visual field is sampled by eye movements and then constructed in the brain. Tight visual focus is the way we solve the problem of reference without naming in a spatial environment. The feedback loop that enables the act of looking entails our observing the world and correcting our body attitude to minimize error of focus. It happens so quickly and effectively that we do not even notice it. The same feedback loop can be applied to a point of focus controlled by the hands. It is not quite as immediate as the eyes, but it is close. And, as it turns out, it doesn't suffer from the kinds of involuntary movements that plague eye-tracking systems.

[I don't know where to put this if anywhere.] [Pointing is a natural extension of the human capacity to focus attention. It establishes a spatial axis relative to an agent, unambiguously identifying anything in line-of-sight without a need to name it. This brings our interactions with electronic devices closer to our interactions with physical objects, which we name only when we have to.]

Pointable Computing successfully takes computation away from the screen and into the space between things. It use of simple, inexpensive components, and its surreptitious hijacking of the human machine as a very fine controller make it more appealing than many other options like motion-tracking, inductive position sensing, or computer vision for establishing simple spatial relations to a user. It requires no calibration, it operates robustly under almost any conditions, and it weighs next to nothing. I expect to see more systems employing laser-directed spatial interaction.

[All about feedback and control. Look at what Carlos Rocha did. You can only do that by giving real feedback. Important. Reference Ryan again.]

[Add discussion of thingness. Non-screen based interface. Contrast to Ishii work in which empty tags are given meaning. Limit to how far that can go.]

EyeBox

Figure #: Word Toss handhelds sending information over a visible laser.

Introduction

[Why is 3D scanning a problem of interest not just to design and engineering?]

Why put the object in the machine? If we want machines to inhabit the same worlds we do, they are going to have to recognize and operate on the same physical reality we do. This includes not just real spaces, but the objects that fill and define those spaces.

We are going to need good ways to get machines to recognize objects.

How do you bring the object to the machine? What is the way to do this? Open it up and put it in. That is how you get anything into anything. A computer needs to have an inside. Right now to the extent that they do, they are intestinal, not homey.

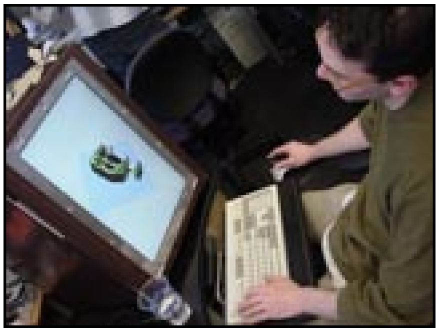


Figure #: Word Toss handhelds sending information over a visible laser.

Introduction

My final project at ACG turned my attention very much toward physical objects. It centered on finding a good way to get them into the computer. As many people such as Bill Buxton [REF] have noted, even as our machines get tremendously more powerful internally, our abilities to get things other than printed material in and out of them has not progressed very far. The engines of computation have digested very little of our world. In order for our machines to become fuller partners in our work and play, they are going to have to join us in our physical world. That means we are going to have to introduce them to the objects that form the substance of our lives. In EyeBox, I have made a computer that a user can sit in front of and work at. Then he can open it up and place an object inside. The object will be scanned in 3D and its form will become available for digital manipulation. However important it is as an inexpensive 3D scanner, it is, I think, more important as an example of a simple spatial interaction with a computer that seems exotic because nobody does it. Opening a computer to put an object inside it feels good, it turns out. It breaks the barrier of the screen by making use of the space behind it. It makes sense to everyone who experiences it.

Figure #: About to scan a small robot.

Description

EyeBox is made out of mini-fridge, three webcams, two fluorescent lights, a microwave turntable, and a flat panel display. [DIAGRAM + FIGURE] Any dark-colored object nine inches on a side or less can be placed into the box, and in approximately twenty seconds, the machine rotates the object once around and produces a full volumetric reconstruction of it from the visual hull of 24 silhouette images (eight from each camera taken during the rotation). A user begins by opening up the fridge. He places an object on the turntable inside, which has hash marks around its edge. He closes the fridge, and the turntable begins to spin. The user sees the camera images from the three cameras displayed onscreen as the object rotates. After a full rotation, the screen changes to a 3D projection showing the 24 silhouette images in their positions around the platform, and an iteratively refining 3D reconstruction of the object on the platform. Over the course of the next few minutes, the representation of the volume of the object gets progressively finer until it reaches a resolution of 512 by 512 by 512 voxels. Then it is filtered to smooth the voxels, giving it a smoother shape.

Motivations

A goal in the project was to keep costs low. Very nice 3D laser digitizers are available for $8,000. EyeBox is not as accurate as these, but it cost $100 to build (minus the flat panel, which is entirely optional). There is an obvious place for such inexpensive devices in industries such as rapid fabrication, design, and entertainment.

Less obvious, but perhaps more important in the long term is the need for computers to be able to recover geometries from the world simply to be more useful in problems that are meaningful to human beings. Computers are wonderful devices for cataloging objects. It would be great to be able to catalog objects as full three-

Figure #: About to scan a small robot.

dimensional reconstructions of themselves. These representations could be sent to others and printed out either locally or remotely, yielding respectively a 3D copier, and a form teleporter. Museums might be interested in this to catalog artifacts or to exhibit pieces in a way that users could place them in a cabinet to find out more about them. It could be used to let people leave impressions of objects in places where they would not leave the actual object.

Method

EyeBox uses a technique called visual hull reconstruction to recover volumes from the silhouettes of objects. Methods of visual hull processing fall loosely into three categories: image-based [REF], polyhedral [REF], and volume carving [REF]. All of these techniques rely on the same basic principle—that a silhouette relative to a calibrated camera produces a generalized cone of volume in which the object must be located. [FIGURE] These cones from several cameras can be intersected to produce a representation of the volume that they are all looking at. It takes surprisingly few cameras to get a fairly good approximation of most common shapes.

Techniques for reconstructing form from silhouette data are all capable of producing its “visual hull” relative to the views taken. Abstractly, the visual hull of an object is the best reconstruction that can be made of it assuming views from every angle. The visual hull, as discussed in Petitjean [3], is a subset of an object’s convex hull and a superset of its actual volume envelope. Specifically, a visual hull technique cannot ever recover a full topographical concavity, such as the inside of a bowl. Such an indentation will be filled in by the visual hull. This is because the technique reconstructs volumes from their silhouettes, and no matter what angle one views an object from a complete concavity will be obscured by its rim in silhouette. [FIGURE]

Image-based

Image-based techniques are the fastest because they do not reconstruct three-dimensional form at all. Instead they synthesize new views from any angle by selectively sampling from the source images directly. Since there is no volumetric representation produced, they are not suitable to true volumetric reconstruction problems. It is possible to imagine, however, reformulating many volumetric problems as image-based problems. For instance, volumetric object-matching may be construed as an image search for the best reconstruction to match a given image of an unknown object. The challenge would be making it fast enough to search all possible orientations of all possible matching objects.

Figure #: Image-based visual hulls from [http://graphics.lcs.mit.edu/~wojciech/vh/IBVH2000.pdf]

Polyhedral

Polyhedral techniques produce a surface representation of the object (easily converted into a volumetric representation if required) by geometrically intersecting polygonalized versions of the cones. This is relatively quick, and provides an unaliased representation without the need for iterative refinement. [Deal with this more completely. Extensions to this technique are able to fit splines to the hulls to let them curve as in Sullivan and Ponce [4].] This technique allows for easy texture-mapping of the original images back onto the reconstructed surfaces, giving another level of detail. I implemented this technique in several different ways, but each time I ran into the same problem: it is highly sensitive to calibration and numerical error. It is imperative

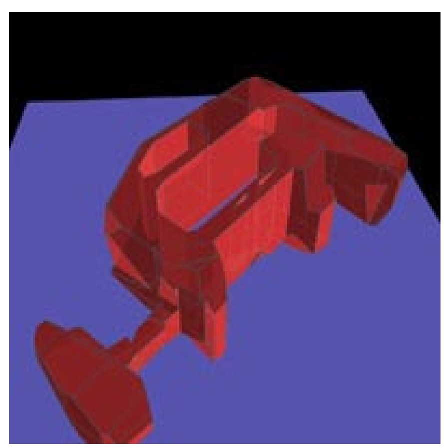


Figure #: Image-based visual hulls from [http://graphics.lcs.mit.edu/~wojciech/vh/IBVH2000.pdf]

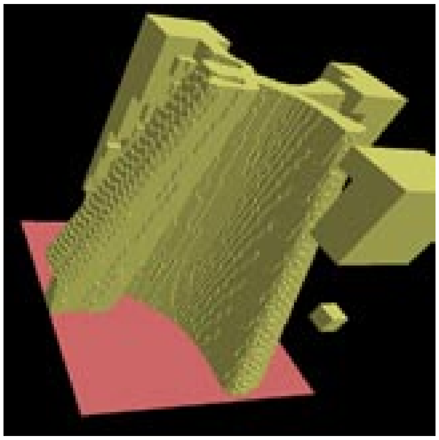


Figure #: Image-based visual hulls from [http://graphics.lcs.mit.edu/~wojciech/vh/IBVH2000.pdf]

that the geometric operations used to construct the volumes be numerically robust and have adjustable geometric tolerances. Methods for general volumetric intersection (constructive solid geometry) that have these necessary characteristics are challenging to implement and difficult to find as free software libraries. So although in theory this may be the best class of methods, it is very difficult to get it to work reliably on real-world data.

Volume carving

This is the simplest technique to implement and also the slowest. It projects voxels from world space onto each of the camera views. If a voxel projection falls fully outside any of the silhouettes, it can be discarded. This produces an explicit volumetric representation at the cost of voxel aliasing and lots of computation. I implemented it because I wanted a volumetric representation for matching purposes and it was the easiest to produce. It is also by means of its aliasing somewhat more tolerant of error in camera calibration than the polyhedral method. This proved to be a significant advantage in the turntable driven scanner.

Speeding it up

Octree subdivision

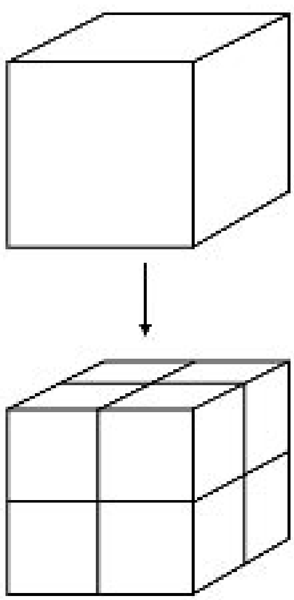


Figure #: Image-based visual hulls from [http://graphics.lcs.mit.edu/~wojciech/vh/IBVH2000.pdf]

Having chosen the volume carving method, I sped it up by representing the volume as an octree. That is an iteratively refined volumetric tree starting with a root node representing the entire volume to be scanned. When a projected node is found to be cut by the silhouette from any camera, it is divided into eight subnodes [FIGURE]. This way whenever a large node is found to be outside of any of the projections, it need never be subdivided or otherwise considered again. This speeds processing up dramatically. Another speed advance was to iteratively refine the octree representation by one level at a time, running it on each camera at each level. That way more large octree nodes were rejected earlier, and did not slow it down. Octree nodes that were wholly inside each silhouette were marked too, so that on each iteration, the only nodes that had to be processed were nodes that in the previous level intersected silhouette boundaries in some camera. This is tantamount to finding the substantial structures early and then iteratively refining the surface. It also means that you see the form improving over time and you are free to stop the process whenever it gets to a level you are happy with. I smooth the surface by applying a Gaussian filter to the voxel data and then finding an isocontour.

Background & Precedents

The phone booth guy.
The woman who scans and makes the small people.

The technique of reconstructing volume from silhouette data is not new. It is well worked out and documented in a variety of sources. Typical setups for the process involve a single well-calibrated camera viewing an object on a turntable as in Kuzu and Rodehorst [1]. The turntable is turned by hand or motorized to provide an arbitrarily large number of silhouette images to be acquired from a single camera.

Fixed multiple camera setups exist, notably Matusik, Buehler, and McMillan’s [2], which is capable of scanning people in a room in real time. This setup requires a computer per camera and one more as a central processor, so it doesn’t qualify as a low-cost solution, but their results are stunning.

Figure #: Image-based visual hulls from [http://graphics.lcs.mit.edu/~wojciech/vh/IBVH2000.pdf]

Figure #: Image-based visual hulls from [http://graphics.lcs.mit.edu/~wojciech/vh/IBVH2000.pdf]

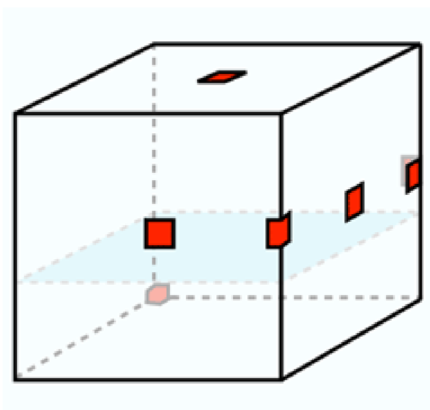


Figure #: Image-based visual hulls from [http://graphics.lcs.mit.edu/~wojciech/vh/IBVH2000.pdf]

It is also not designed for scanning handheld-sized objects.

Design and Operation

EyeBox as a mini-fridge is a second generation of the system.

Revision 1

The first version, a foamcore cube 18 inches on a side with six cameras at fixed locations and no turntable, was quite successful—in some ways more successful than its turntable successor. [FIGURE] The camera positions in the original version had to be carefully chosen to deliver the most amount of non-redundant information. Therefore they were not one-to-a-side, as might be supposed. Views separated by close to 180 degrees are primarily redundant. The camera placement was as shown in the [DIAGRAM].

The first step in the construction was the dismemberment of the webcams. Then I built an 18” X 18” X 18” cube out of foamcore and put a plexiglass shelf in it 7” from the bottom. I cut holes in the sides and top for the cameras and attached two small fluorescent lights to the inside. [FIGURE] shows the box with the top off and my calibration object, a laser-cut cube with color-coded edges, inside. Calibration of the cameras was a two-step process. The first step was camera calibration, which I accomplished by Tsai’s method [FIG] embedded in a calibration application I wrote for the system. Then I was ready to write the reconstruction software.

The first step was to acquire a silhouette image from each camera, which was very easy because of the well-controlled imaging environment. For each camera, I simply subtracted an image of the empty box and then thresholded the results.

The reconstruction proceeded as detailed in the octree method outlined above.

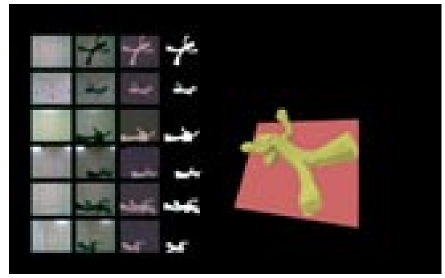


Figure #: Image-based visual hulls from [http://graphics.lcs.mit.edu/~wojciech/vh/IBVH2000.pdf]

Problems

There were some problems with the reconstructed objects. Many of them had to do with the white background. Light colored objects did not scan well at all. Specularities on objects are always white and tended to be seen as background, drilling holes in objects. In a future version of the system, I would use a blue background to make segmentation simpler. Reflections off the plexiglass were troublesome. Finally, the box was rather large for an effective scanning volume of 6” X 6” X 6”. That could have been improved with wider angle lenses, but the wider the field of view, the lower the quality of the reconstruction. There were also errors of volume just due to spaces not visible to any camera. This could have been helped with more cameras.

The second version of the system set out to solve some of these problems. It used a rotating platter to effectively multiply the viewpoints from three cameras into 24. The rotating platform also helped shrink the necessary size of the system. Since cameras were only looking at the object from one side, it was the only side that needed visual clearance. It imaged against a rounded background to get rid of dark corners in the empty volume.

Figure #: Image-based visual hulls from [http://graphics.lcs.mit.edu/~wojciech/vh/IBVH2000.pdf]

Revision 2

Revision 2 was housed in a mini-fridge. I chose a mini-fridge because it fairly closely matched the dimensions I determined were optimal, and I could not resist the feeling of the seal made by a fridge door. I gutted the fridge and drilled a hole in the back to run cables out. I decided to orient it standing up rather than lying down so as not to evoke a coffin. Instead it is very clearly a mini-fridge, and its hybridity is part of its strong appeal. I used a water-jet cutter to cut out a large opening in the door and mounted an Apple Cinema Display in it. I salvaged an AC gearhead motor from a old microwave turntable and mounted it inside the fridge with a shaft and a plexiglass turntable on it. I glued three webcams to the interior of the fridge looking slightly off-center at the turntable. I turned them off-center to maximize the probability that they would perceive the edges of objects—the source of all of my information. I was not concerned that they might not be able to see both edges at once because I rotated every object a full 360 degrees. I disassembled two small fluorescent lights and mounted them inside the cabinet pointing directly back onto the curved white back surface. My hope was that this would completely backlight the subject and get rid of all the problems with specularity. In fact it ended up still giving a strong side light. I mounted a reed switch on the door hinge to control the platter motor. When the door closes, the platter spins.

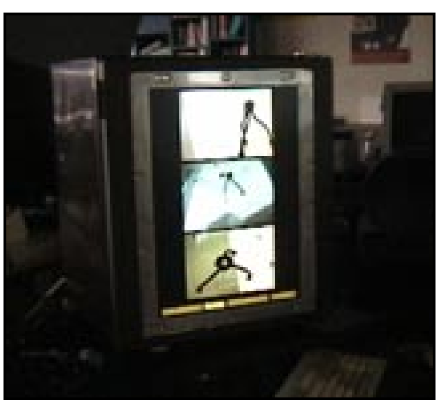


Figure #: Image-based visual hulls from [http://graphics.lcs.mit.edu/~wojciech/vh/IBVH2000.pdf]

My setup avoided having to carefully control the speed or position of the turntable by placing black marks at its edges in 45 degree increments. [FIGURE] The total light value from a small patch of the camera looking from the top is used to determine when the turntable is in position to use a single video frame from each camera as a still image from one angle. Two of the marks are not black—one is red, and one is cyan. These are present to indicate the starting position (which will be considered zero degrees), and the direction the platform is spinning. It is necessary to determine the direction in real time because the turntable motor is a cheap AC motor lifted from a microwave, and it is therefore impossible to know which direction it will turn when power is applied.

I calibrated the cameras by the same procedure as the first version. Because I had not constructed the whole system to engineering tolerances, I calibrated each of the 24 views by hand rather than calibrating three and performing rotations on them.

Results

All of the changes proved to be advantageous, and my results were somewhat better with the new system. The biggest disappointment was how little it improved. The fantastic advantage of the technique is that it takes so little information to give very good results. After the first several cameras, adding more gives diminishing returns. It may be that 24 views is more than is necessary, and rotating the object may therefore be as well. With the current cost of webcams at about $15, maybe I should just settle for 12 in a stationary setup. Not rotating has several advantages—easier, more consistent calibration, no moving parts, faster operation. The primary advantage, though, to not rotating the object is the improved magical quality of producing a transformable 3D reconstruction from an object that is totally stationary.

Figure #: Image-based visual hulls from [http://graphics.lcs.mit.edu/~wojciech/vh/IBVH2000.pdf]

Analysis and Critique

References

[1] Y. Kuzu and V. Rodehorst, Volumetric Modeling Using Shape From Silhouette, www.fpk.tu-berlin.de/forschung/sonder/pub/DT4_kuzu.pdf.

[2] W. Matusik, C. Bueler, and L. McMillan. Polyhedral visual hulls for real-time rendering. In Proceedings of Twelfth Eurographics Workshop on Rendering, pages 115-125, June 2001.
[3] S. Petitjean, A Computational Geometric Approach to Visual Hulls, Int. J. of Comput. Geometry and Appl., vol. 8, no.4, pp. 407-436, 1998
[4] S. Sullivan and J. Ponce, Automatic Model Construction, Pose Estimation, and Object Recognition from Photographs Using Triangular Splines, IEEE Transactions on Pattern Analysis and Machine Intelligence, 20(10):1091-1096, 1998.

Spatial Computing for Information storage and retrieval, a discussion.

The Associate is a system that provides users with a means to associate digital information with physical objects. Traditional file systems offer little other than file names and types by which to remind users of the contents or context of a document. The shortcomings of name-dependent filing are well documented [2]. Naming represents an overhead to the user; it demands the categorization of ideas before the work is complete; it is exclusively dependent on language memory cues, which are slow to digest and process; it requires that users formalize into hierarchies information that may or may not be naturally hierarchical; and it is difficult to recall or communicate full data “paths” with other users. It is telling that people do not name things in their environment in order to reference them. In fact, they name almost nothing that does not come when called. Instead they use spatial organizations—piles, shelves, drawers, rooms, etc.—and their focuses of attention to differentiate between objects. By attaching files to real physical objects, The Associate allows users to employ the same principles of spatial organization and associative recollection to store and retrieve their digital information that they use in their daily interactions with a physical world.

The state of the art and my contribution

[3D scanning techniques]
[Do a taxonomy. Talk about the ways we perceive depth.]

Summary Conclusions

I have not done enough to put forward a comprehensive theory of spatial computing. My hope is that that will never be possible, allowing me to work productively toward it for the rest of my career. But I have shed some light into its corners, and discovered what I believe are its fundamental principles. The variousness of my experiments, rather than being an impediment to this inductive process, has been essential. The qualities evident from experiments so widely disparate in scale and approach are likely to have some validity over the entire field.

It Doesn't Take Much

What this means is that suggestion of a link to space is often enough. Approaching perfection may do more harm to the feeling of connection than good. This was evident in several projects. First in Installation, the power of the live video feed demonstrated this. It was not much to add, and it certainly did not fool the eye. But it established context of the interaction. The mind did the rest of the work.

It was apparent again in LittleVision, which presented a highly abstracted, low-bandwidth, representation of a visual scene. It was just enough to indicate what was going on. The work of the mind to bring the scene into focus created an attachment to the interaction. The understanding was a collaboration between the object and the observer.

It is important to remember this. Do not try to fool the senses! It will not work. The senses are canny and aware of much more than we will ever be able to simulate. The closer we try to approximate reality, the more noticeable our failure will be. Instead, we must suggest what we want the user to experience and rely on him to do the rest.

Object resonance

There are many factors at work in whether an object will have resonance with an audience. My work has revealed several to me.

First, if it is to be held, it must have a form that is pleasing in size, weight, and texture. LittleVision demonstrated this admirably. It is also very important that it have no wires trailing off of it. Tethering destroys an object's autonomy and restricts its manipulation.

A second technique, not exclusive of the first, is to use objects with existing resonance and repurpose them. The mini-fridge cabinet of EyeBox gives it an appeal to many that no custom cabinet could.

Feedback, Relativity, Consistency, and Expectation

Immediacy of feedback is the single most important quality of interaction. We are set up to control our operations in the world only relative to feedback we receive about how they are proceeding. If a system does not provide such feedback it becomes impossible to control. We do not sense absolutes, but relative values. We can

Relativity

The relativity of sensory experience is something

How hard do you push a door? (As hard as you have to.) It's about testing and response and feedback. (Rocha)

Consistency + Expectation

Literalness

No icons.

The difference between throwing the thing to an image of the printer vs. throwing it to the real printer. When you throw it to the icon, the piping becomes apparent. Must hide it. It exists to disappear.

Transparency

Depends on intention. The system must disappear. To the extent that it's visible, it is broken.

I see no reason to deny either the real world or the limitations of digital devices. A computation that includes uncertainty and morbidity is a better friend to me. Instead of trying to replace reality with an incomplete and sanitized representation, why not engage it, do our best to fold into it? Sense it to the extent we can, pull in what measurements are available. Fail without embarrassment where inevitably we must. Let the edges of algorithms show. I am for a rough, degraded spatial computing that feels true.]

References

Dourish, P., W.K. Edwards, and e. al., Extending Document Management Systems with User-Specific Active Properties. ACM Transactions on Information Systems, 2000. 18(2): p. 140--170.

Julie Dorsey and Pat Hanrahan, “Modeling and Rendering of Metallic Patinas.” Proc. of SIGGRAPH ‘96. In Computer Graphics Proceedings, Annual Conference Series, 1996, ACM SIGGRAPH, pp. 387-396.

From Metropolis: Robert Venturi’s Disorderly Ode, Sept 2001

Lev Manovich, Generation Flash, 2002.

Lev Manovich, The Aesthetics of Virtual Worlds: Report from Los Angeles, 1996

“Eye Movements and Vision” by A. L. Yarbus; Plenum Press, New York; 1967.

Pausch, R., Proffitt, D., and Williams, G. (1997), Quantifying immersion in virtual reality, SIGGRAPH’97.

Robertson, G., Czerwinski, M., and van Dantzich, M. (1997), Immersion in Desktop Virtual Reality, UIST’97.

Deering, M. (1992), High resolution virtual reality, in Computer Graphics, 26, 2, 195-202.

Arthur, K. W., Booth, K. S., and Ware, C., “Evaluating 3D task performance for fish tank virtual worlds”, ACM Transactions on Information Systems, vol. 11, no. 3, July 1993, pp. 239-265.

Slater, M. (2002) Presence and the Sixth Sense, PRESENCE: Teleoperators and Virtual Environments, MIT Press, 11(4) 435-439.

Witmer, B.G., Singer, M.J., “Measuring Presence in Virtual Environments: A Presence Questionnaire”, Presence, 7 (3), 1998, pp. 225-240.

C. Kline and B. Blumberg. The Art and Science of Synthetic Character Design. Convention of the Society for the Study of Artificial Intelligence and the Simulation of Behavior (AISB), Symposium on AI and Creativity in Entertainment and Visual Art, Proceedings, Edinburgh, Scotland, April, 1999.

E. H. Gombrich
Art and Illusion, a study in the Psychology of Pictorial Representation, 1969, Princeton University Press, Princeton, NJ.

Colby, Grace, and Laura Scholl. “Transparency and Blur as Selective Cue for Complex Information.” Proceedings of SPIE’92. 1992.

Ishii, H. and Ullmer, B., Tangible Bits: Towards Seamless Interfaces between People, Bits and Atoms, in Proc. of Conference on Human Factors in Computing Systems (CHI ‘97), (Atlanta, March 1997), ACM Press, pp. 234-241.

In “Human-Computer Interaction in the New Millenium,” John M. Carroll, ed.; © Addison-Wesley, August 2001, pp. 579-601.
Emerging Frameworks for Tangible User Interfaces
Brygg Ullmer and Hiroshi Ishii

Ullmer, B., et al., “mediaBlocks: Physical Containers, Transports, and Controls for Online Media,” in Proceedings of SIGGRAPH ‘98, ACM Press, pp. 379-386, 1998.

Ullmer B., Ishii H. The metaDESK: Models and Prototypes for Tangible User Interfaces. Proc. of UIST'97, pp.223-232.

Matt Laibowitz, Joe Paradiso, "Phenomenological Model for Distributed Systems," http://www.media.mit.edu/resenv/phenom/

Interfacing to the Foot: Apparatus and Applications, J. Paradiso, K. Hsiao, A. Benbasat, Proceedings of the ACM CHI 2000 Conference; Extended Abstracts, pp. 175-176.

Tsang, M, Fitzmaurice, G., Kurtenbach, G., Khan, A. & Buxton, W. (2002). Boom Chameleon: SImultaneous capture of 3D viewpoint, voice and gesture annotations on a spatially-aware display. Alias|Wavefront. Submitted for publication. [http://www.billbuxton.com/boomChameleon.pdf]

Zhai, S., Buxton, W. & Milgram, P. (1996) The partial-occlusion effect: Utilizing semitransparency in 3D human-computer interaction ACM Transactions on Computer-Human Interaction, 3(3), 254-284.

Buxton, W. (1997). Living in Augmented Reality: Ubiquitous Media and Reactive Environments. In K. Finn, A. Sellen & S. Wilber (Eds.). *Video Mediated Communication*. Hillsdale, N.J.: Erlbaum, 363-384. An earlier version of this chapter also appears in *Proceedings of Imagina '95*, 215-229.

Dunne, A. and Raby F (2001), "Design Noir, the Secret Life of Electronic Objects" August / Birkhäuser, Berlin

Billinghurst, M. and Kato, H. (2002). Collaborative Augmented Reality. *Communications of the ACM*, *45*(7), 64-70.

[Kar98] Karrie Karahalios. Diorama.In Grisson et al. [GMA
+98],
page 297. ISBN 1-58113-046-5, 1-58113-048-1 (CDROM). LCCN T385 .S541998b. URL http://www. acm.org:80/pubs/citations/ proceedings/ graph/280953/ p297-karahalios/

WindowsVR system www.abs-tech.com/Produtos/ 3D_VR/VR-Hardware/hmds1/Virtual_Research/ win_vr.html

Art+Com *Virtual Car for Daimler-Benz AG*, 1997 www.artcom.de

Bolt, R.A. Put-That-There: voice and gesture at the graphics interface. SIGGRAPH Computer Graphics, 14(3), 1980, 262-270.

Jun Rekimoto, “The Magnifying Glass Approach to Augmented Reality Systems”, International Conference on Artificial Reality and Tele-Existence ‘95 / Conference on Virtual Reality Software and Technology (ICAT/VRST ‘95).

Unifying Augmented Reality and
Virtual Reality User Interfaces
Wayne Piekarski and Bruce H. Thomas
Technical report January 2002, University of South Australia

George W. Fitzmaurice, Situated information spaces and spatially aware palmtop computers, Communications of the ACM, Special issue on Augmented Reality and UbiComp, July 1993, 36(7), p.38-49.

Azuma, R., T., A survey of Augmented Reality, in Presence: Teleoperators and Virtual Environments, 6, 4, (1997), p. 355-385.

Fig. 52 from Soap Bubbles, Their Colors and Forces which Mold Them. C.V. Boys.

Robert Saunders, Simplified ART, August 2001
http://www.arch.usyd.edu.au/~rob/java/applets/neuro/SimplifiedARTDemo.html

Jared Shiffman, Honey 2000
http://acg.media.mit.edu/people/jarfish/honey/

Ben Fry, Anemone, 2000 - 2002
http://acg.media.mit.edu/people/fry/anemone

Ben Fry, Tendril, 2000 - 2003
http://acg.media.mit.edu/people/fry/tendril/

Card SK, Robertson GG, York W. The WebBook and the Web Forager: An information workspace for the World Wide Web. Proc CHI '96 ACM Conference on Human Factors in Computing Systems. New York: ACM Press, 1996; 111-116.

Selfridge, P. Kirk, T. "Cospace: Combining Web-Browsing and Dynamically Generated 3D Multiuser Environments", SIGART 10, 1, 24-32, 1999.

Electronic Arts, 2003
Sims Online
http://www.eagames.com/official/thesimsonline/home/index.jsp

Linden Labs, 2003
Second Life
http://lindenlab.com/

There, Inc., 2003
There
http://www.there.com/

Ruth Conroy Dalton, Is Spatial Intelligibility Critical to the Design of Large-scale Virtual Environments?

nternational Journal of Design Computing, vol. 4, 2002
http://www.arch.usyd.edu.au/kcdc/journal/vol4/dalton/index.html

Brown K N, 1997 “Grammatical design” *IEEE Expert: Intelligent Systems and their Applications* **12** 27-33

Jean Baudrillard: The Ecstacy of Communication, New York: Semiotext(e), 1988

The Magic Carpet: Physical Sensing for Immersive Environments J. Paradiso, C. Abler, KY. Hsiao, M. Reynolds, in Proc. of the CHI ‘97 Conference on Human Factors in Computing Systems, Extended Abstracts, ACM Press, NY, pp. 277-278(1997).

Arrows in the Fluid Mind
A 2D Fluid Dynamics Model for Animation
Based on Intuitive Physics
Jeffrey Ventrella
1997
http://www.ventrella.com/Ideas/Fluid/fluid.html

I developed an extended use case for Eyebox as a direction for future research, in which I propose its use in a system for associative interface. The idea is that digital information could be permanently associated with physical objects and then organized and retrieved using them as physical proxies. The proposal is included here as Appendix [BLORF].

Associative Interface

Associative machine memory as outlined in Poggio and Girosi [16] has been an active topic of research for some time. It has achieved some polish and effectiveness in automatic clustering by content of Internet sites by such engines as Google. CiteSeer is a similarly effective automatic associative engine for technical publications [17]. I will be using automatic clustering of information by content in order to group documents in specific places as in the semantically clustered filing system of Gifford et. al. [18] and the Remembrance Agent of Rhodes [12], but I do not expect to be breaking new ground in the field. I will implement existing algorithms as this capability is not the crux of the system. The Associate differs from these precendents in its emphasis on user-generated associations with objects.

Spatial mappings of data are not new either. There are countless systems and frameworks for the visualization in virtual space of abstract data such as Robertson's Data Mountain [5], scatter graphs [13], navigable virtual environments [14], and mapping onto familiar forms such as cities [15]. Recent studies indicate that the addition of the third dimension to such systems is not helpful to users in storage and retrieval, and in fact, adds clutter and frustration [19], [20]. I argue that these may not apply generally, but may be tied to two problems of representation. First is a general insensitivity to most modes of human spatial awareness. Typically "virtual environment" conjures images such as figure 2, taken from a study of the utility of the third dimension as a retrieval cue. What we see is an image that is three-dimensional in exactly two senses: perspective of size and linear perspective. Perceptual psychologist James Gibson identifies thirteen different means of human perception of depth [21]. To implement two of them to the exclusion of all others has bearing on the utility of "virtual environments" for storage and retrieval only in so much as nearly all virtual environments ever created implement exactly the same two means of representation of depth. Consider an image such as figure 3, Georges Braque's Bowl of Fruit. The depth that is conveyed in this image is an intimate, human perception-oriented depth. It is accomplished without perspective of size or linear perspective at all. In fact Braque considered them thin tricks that did little more than confuse the eye [22]. The depth in Braque's piece is an operational depth. It is one that we can relate to as though the fruit bowl were right in front of us—graspable. The strict analytic perspective of figure 2 has next to no relationship to real human place as it is perceived.

The Associate makes a strong distinction between space and place. Space may be represented as in figure 2, the span of three orthogonal bases projected into two, but place must be represented as something much fuller. Toward this end, I will limit my use of linear perspective and focus more on perspectives of blur, movement, texture, color, and shade. This will still constitute a virtual environment, although it may not be three-dimensions mapped to two as they canonically are. In addition I hope to employ eye-tracking to change the viewpoint of the scene as in [23] and [24] to make the perception of place active rather than passive. Hall and Thorndyke both point out that active perception, the natural mode of human environmental acquisition, is far better for learning spaces than passive reception of visual information [22], [25]. This reconception of virtual space in light of seemingly forgotten principles of art, design, and perception presents a significant contribution to the field.

Problems

I do not expect with this thesis to strike a mortal blow to the hegemony of naming. Naming is often indispensable, and in fact, so that I may refer to it in this paper and in speech, the system I am making to demonstrate the plausibility of anonymous storage and retrieval has a name—The Associate. I am restricting myself to a zero-name diet in its implementation only to demonstrate that it is possible. Words may still be visible in The Associate, hanging off of spatialized information, but they will be words automatically extracted from the information being stored. They will operate as retrieval cues, not names per se.

I have worked out the majority of the primary technical hurdles in the experiments leading up to this thesis, so I do not expect to fail technically. However, The Associate may not help me retrieve documents faster or more accurately. There are other criteria by which to evaluate storage and retrieval systems, which are largely ignored by the literature perhaps because they are difficult to measure (discussed in detail below in the Evaluation section).

Evaluation

The Associate makes no claims to make storage and retrieval faster or more accurate. Instead it offers some of the benefits of “reminding” that Freeman [2] and Rhodes [12] outline. Results of speed-of-retrieval tests in virtual storage environments are equivocal, which I argue points to a strong dependence on specific implementation details ([5], [19]) and specific user characteristics ([26], [27]). I will not be assessing The Associate by its utility as a completed system. I do not have the time or resources to iterate and refine its design to the point that a quantitative analysis of its benefits would be fair to it. Instead I hope to complete a qualitative assessment of the value of the overall direction of the research—is object-associated digital storage a fruitful avenue for continued exploration? In order to gauge this, I will evaluate The Associate much as Mander, Salomon, and YinWong evaluated their “pile” metaphor for casual organization of information—through user observation and brief survey [1]. I will ask a set of users to perform a variety of storage and retrieval tasks in both a traditional file system and then with an associative physical interface. Then I will ask them a series of questions about their experiences to determine what works and what needs work.

For instance, I will ask people to indicate on a scale their level of frustration in searching for information; to indicate whether the experience overall was pleasurable or frustrating and to what degree; and whether they think they would use the system in practice and under what circumstances. In another set of tests, I intend to ask people to perform a set of filing tasks and then ask them back in two weeks’ time to try to retrieve what they filed. Through this analysis, The Associate will help us discern productive directions for the use of physical memory for information organization.