The I/O Bulb and the Luminous Room

John Stephen Underkoffler
B.S., MIT, 1988
S.M., MIT, 1991

SUBMITTED TO THE PROGRAM IN MEDIA ARTS AND SCIENCES,
SCHOOL OF ARCHITECTURE AND PLANNING,
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY, MASSACHUSETTS INSTITUTE OF TECHNOLOGY

February 1999

(author)	John Underkoffler	Program in Media Arts & Sciences
(accepted by)	Nicholas Negroponte	Director, MIT Media Laboratory
(accepted by)	Hiroshi Ishii	Associate Professor, MIT Media Laboratory
(certified by)	Stephen A. Benton	Chair, Program in Media Arts & Sciences

The I/O Bulb and the Luminous Room

John Stephen Underkoffler

Submitted to the Program in Media Arts and Sciences,
School of Architecture and Planning, on 8 January 1999:
in partial fulfillment of the requirements for the degree of
Doctor of Philosophy, Massachusetts Institute of Technology

ABSTRACT

We introduce a new device called the *I/O Bulb*, an evolution of the common lightbulb capable of emitting structured light and performing simultaneous optical capture. This basic notion is extended to suggest an architecture called the *Luminous Room*, in which every portion of an enclosed space is treated by one or another of many coordinated I/O Bulbs. We present a series of implementations – hardware & software systems that mock up I/O Bulb environments; two full I/O-Bulb-based applications; assortments of smaller interaction experiments; and finally a prototype Luminous Room system – in order to demonstrate and substantiate these core ideas.

Doctoral Dissertation Committee

Nicholas Negroponte	Director, MIT Media Laboratory
Hiroshi Ishii	Associate Professor, MIT Media Laboratory
William Mitchell	Dean, MIT School of Architecture & Planning

Contents

Acknowledgements

Las Vegas knows no gambler like Hiroshi Ishii, who despite the poor odds attributed it by lower rollers placed an unlikely side bet on behalf of this research, then a wandering mendicant. That was 1997; the dice are now nearly at rest; and so we shall see – but a hopeful early glance seems to show the pile of chips a bit bigger. I wish that I could somehow repay his wise guidance, his humane dedication, and his AEC-suspicion-inducing personal energy.

Nicholas Negroponte played Schrödinger to the cat of my PhD work. His persistent doubt kept lifting the lid to check, and thus did many lesser kitties die. What you'll read about following is the hardy survivor of that brutal but rightful triage, for which process I am grateful.

Bill Mitchell lent his keen analytical eye and redoubtable enthusiasm to each stage of the project. A man with alarmingly many pies, he amazes with as many fingers. The present pie has benefitted enormously from the presence of those phalanges. I thank him.

My family is the funniest I've ever come across. The humor is manufactured to exacting tolerances in the marrow of my parents (who I'll wager used up several industrial drums of their most concentrated just putting

up with their brain-gimped son's unending graduate studies) and was passed by standard genetic means full-strength to my brothers (who've buoyed me with it oftener than I deserve). I respect no-one more than these four people: James, Karen, Thomas, Peter. Great idea-founts, perpetual humanists, and indefatigable workers, they are collectively my role model and the best I can imagine. The pride I feel to be among them is I think no deadly sin.

Words are fickle creatures that falsely promise clear expression. Mostly we can make do with their shabby approximations. Occasionally we know in advance not even to bother trying. This is so as I come finally to Wendy Plesniak. A quandary: but still. With a single sentence, on a Sunday in waning 1996, she catalyzed this entire project. The privilege of her friendship includes uncounted such detonations of brilliance, explosions the more surprising for their innate comic essence. Conjoined to her fine mind is a fierce will. Certain quietly selfless deeds born of this will have taught me that more is possible than I knew. By her unique alloy – wisdom & humor, intelligence & humanity, sanity & profanity – she has made this work at all possible; as she has me.

To Charles, an engineer and thinker,
and Mildred, a poet and musician;

and to Karen and James, who are everything.

The I/O Bulb and the Luminous Room

Prolegomenon

These pages tell the story of some work toward a single simple goal: the pervasive transformation of architectural space, so that every surface is rendered capable of displaying and collecting visual information. Part of the story's plot is the search for an appropriate mechanism: how can you make architecture do such a thing? That's a good challenge, and the answer you'll find offered here is half technology (how it can be done now) and half philosophy (how it should be done eventually). An equally important piece of the narrative concerns a harder question: once your room has evolved these new information talents, how does it use them? What does it make sense to do there? The full answer, should the question continue to be interesting, will be decades or more in forming, but a beginning – a sketched-in corner of what looks like a very large canvas – occupies many of the chapters here.

To get started, though, we'll return to the first question. What device or technique could be devised to enable a room to do the visual display and capture we've envisioned? And where in the room will it do that from?

1 Idea

A certain Canadian media theorist considered the electric light the only perfect medium: by syncopating human pursuit away from the insistence of diurnal rhythms, by turning night into day as necessary, the lightbulb qualified for him as *pure information* whose message was *total change* and which suffered no restrictions on its unequaled *transforming and informing power* [11].

Electric lighting changed everything, forever.

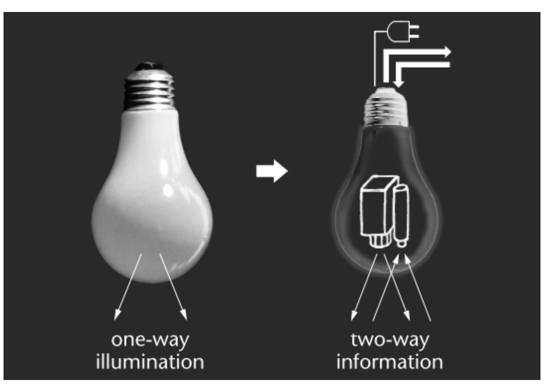
This overly dramatic claim – that the electric light is not only a powerful transforming force but in fact immutably perfect – leaves apparently little room for improvement; still, if we ignore its absolutism the basic sentiment is both clear-sighted and intriguingly suggestive. Our purpose here is to follow the implicit suggestion and show the evolution of the lightbulb.

Does that leave any room for further change?

1.1 I/O Bulb

To do so, we will start with the bulb that Edison invented for us about 110 years ago and that is still available, basically unchanged, for purchase today. For the moment we'll suppose this familiar object to be a primitive sort of digital projector – a digital projector of spectacularly low resolution whose wall switch controls the binary state of its lone pixel. But now the first half of our bulb upgrade program is clear: we will build a higher resolution model. What are the implications of a 1000x1000 pixel lightbulb?

The lightbulb as 1x1 projector. What about a higher-res version?



Meanwhile, the shell that forms the bulb's familiar shape and separates its interior workings from the exterior world is glass and thus – like any good window – permits optical flow in either direction; yet the typical incandescent lamp takes advantage only of the outward

Our modern lightbulb – the *I/O Bulb* – will also see.

direction. What of light that flows from the outside in? Might the inside of a lightbulb be a site not only for radiating information but for collecting it as well? Thus the second half of our bulb reform curriculum: we need a model that looks out at the surroundings it serves, performing continuous visual capture of whatever may be going on there. So there will be a tiny video camera inside our new bulb.

Not unlike a human head: a basically spherical site where lots of sensing and output is concentrated.

We call this new two-way information transducer an *I/O Bulb*: a high-resolution lightbulb that can see. The notion of using projection onto architectural or other real-world surfaces to render them 'information-bearing' is not wholly new (see, for example, the *Put-That-There* system described some chapters hence), and the use of machine vision as a primary input mechanism has had both proponents and occasional successes. But binding these two techniques together in a single device is a critical distinction, one which suggests also the emergence of a new kind of pervasive infrastructure.

luminous + tangible is a crucial mix

Too, the applications that have succeed as clients of this new optical information spigot have so far evinced a unique flavor, characterized in part by a careful division and balance of their tasks between physical and digital/-projective components. So that, conceptually powerful as a 'pure' digital vision of I/O Bulb use might seem (i.e. it communicates with you by projecting and you communicate with it by gesturing at and moving through its projections), it will be one of the chief conclusions of this work that a hybrid version incorporating physical articles – *stuff* of one sort or another – will best succeed. This approach to interaction we will broadly describe as *luminous-tangible*.

1.2 Luminous Room

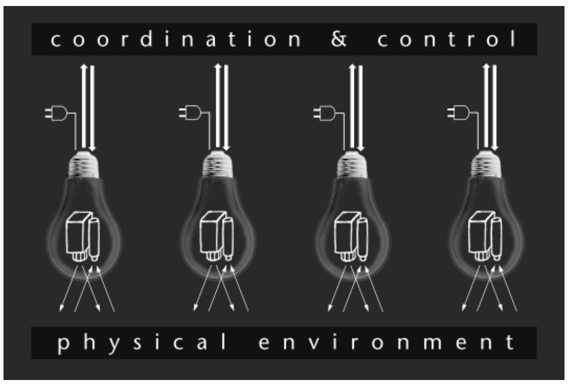
When there's an I/O Bulb for each part of the space, we get a *Luminous Room*.

If the I/O Bulb is the atomic unit of transformation that the work represented here would foist on the world, then the *Luminous Room* is the infrastructure that results from seeding an extended space with a multiplicity of these units – enough, specifically, so that every part of the room is within the purview of at least one I/O Bulb. This association of many cooperating camera/projector nodes serves to transform the surrounding architectural space, making of each surface a site of available interaction [19].

It's far too early for this (or any other) document to pretend to understand all the implications of a working

Luminous Room. But the notion of a pervasively reactive architecture is the single prodding impetus behind our work, and so the dissertation's job is, in part, to illustrate some of the ways in which such an environmental facility

What would it mean to have a real Luminous Room?



might be well used; to imagine others; and to provide the rudiments of a framework for analyzing (and catalyzing thought about) such augmented spaces. In particular, we'll show the successful extension of one of the originally-single-I/O-Bulb applications to operate in a multiple-I/O-Bulb environment and discuss some of the issues involved in developing a comprehensive system for coordinating such distributed activities.

Our approach to figuring this out is to build as much of one as we can.

1.3 All Together

This document concerns the I/O Bulb and the Luminous Room. Even though these two concepts are the main protagonists of the story, they're somewhat shadowy protagonists; instead, the tale is told mostly through more immediately accessible characters: the applications built to illustrate the capacities and potentials of the I/O Bulb and of a Luminous Room.

Mostly, we'll talk about the applications that prove our more general ideas.

The chapter following this one introduces the dissertation's thesis proper, two simple questions to which the body of work undertaken intends to be an adequate aggregate answer. The chapter also introduces some basic software techniques – including a simple two-part approach to machine vision – that underlie most of the demonstrative applications built thus far.

Ch. 2: Central arguments; basic hardware & software common to everything else.

The succeeding chapters (3 & 4) introduce and discuss in some detail the pair of finished, large-scale applications – an optical layout prototyping system and an urban planning workbench environment – on which

Ch. 3 & 4: *Illuminating Light* and *Urp*, our two big applications.

most of the work's eventual understandings & conceptual analyses & claims to success are based. These are joined by an additional chapter (5) that describes a number of smaller applications, comprising one early experiment constructed before we'd quite got the hang of what I/O Bulbs are good for and a handful of more recent 'design sketches' illustrating divergent interaction styles.

Ch. 5: Earlier & smaller experiments.

The subsequent chapter (6) more formally develops the Luminous Room idea and presents a first implementation of one, including a set of extensions to the existing low-level I/O Bulb infrastructure and a sample application that operates throughout disjoint portions of a larger space.

Ch. 6: Luminous Room proper; a real implementation; a real application.

The antepenultimate chapter (7) begins with a brief review of distant-relative research projects from the past three decades – the history behind and context around the present work – and continues with an explication of some central issues of luminous-tangible interaction: the proper design of systems that incorporate both physical artifacts and digital/projective elements.

Ch. 7: Related work; *luminous-tangible* design.

The final two chapters conclude our business, suggesting a number of further directions for expansion of the I/O Bulb and Luminous Room work and offering a summary of the current work's findings.

Ch. 8: Further directions.

Ch 9: Conclusion.

2 Fundamentals

It is the purpose of this dissertation work to substantiate the core I/O Bulb and Luminous Room ideas, mainly through practice: realization of these ideas in a handful of convincing forms. The implementation work that does this – not just hardware, but demonstrative real-world applications as well – has been directed at providing two critical bits of evidence: that the technology can be built; and that the technology should be built.

Can: is it realistically possible to construct an I/O Bulb that functions as we have imagined? Will the modest sensing techniques available today – machine vision, certainly, but eventually others too – reliably extract enough information to make the Luminous Room a satisfyingly reactive environment? Might citizens living a generation hence naturally screw Sylvania 100 Watt I/O Bulbs into their lamps and ceiling fixtures? In short, does some combination of extant, emerging, and immediately developable technologies allow us to build an operational I/O Bulb and the applications that use it?

Is it really possible to make this stuff work?

Should: is a Luminous Room the (or at least an) appropriate way to think about emerging forms of architectural-scale interactions? Is an I/O Bulb a useful quantum of mechanism? Are these concepts fertile, so that they serve not just to describe this year's one-shot demonstrations but also to prescribe next year's research directions? Most importantly: even if it were possible to install I/O Bulbs in our living and work spaces, would there be anything useful that we could do with them, or is the I/O Bulb no more than an academic conceit?

And if so, is it a good idea: how far can it be taken?

Naturally, we take the preceding questions to have foregone answers; but for the benefit of those whose belief (quite reasonably) cannot be bought with this promise alone: all the words beyond this point work at stitching together the implementations, experiences, and analyses comprising the project's effort into a tapestry of proof. What's to be found from this point onward is a collection of hardware and software systems that realize the I/O Bulb apparatus in several ways and that provide applications illustrating at least part of the style and range of interactions possible in a Luminous Room environment.

We've built some experimental bulbs & applications that should let us decide.

2.1 Basic Hardware Infrastructure

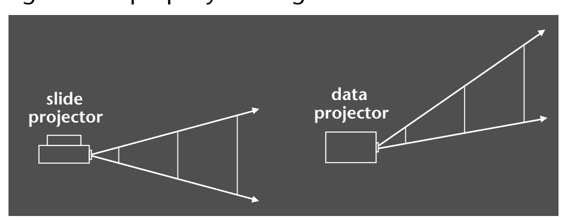
A temporary hardware provision.

The application fragments, full applications, studies, and miniatures that form the core of this dissertation make use of a fixed, overhead I/O Bulb prototype comprising a moderately bulky (compared to a real lightbulb, that is) Epson digital projector and a compact medical-industrial video camera.

2.1.1 A Word On Collocation & Coaxiality

Data projectors are supposed to sit on tables but project up onto walls.

The anti-keystoning mechanism inseparably integrated into the data projectors of today (it's 1998, and all projector manufacturers assume that their products are used pretty much exclusively in business presentations, and will thus only ever be placed on conference-room tables or mounted on ceilings) means that the projection expands along a frustum not centered on the normal to the lens, although each parallel focus plane of this frustum is of course still properly orthogonal to the lens normal. This means that a projector that points toward the horizon, straight ahead, will deposit its image well above the 'horizon-aim' of its lens and the image will be properly rectangular. This is in contrast to,

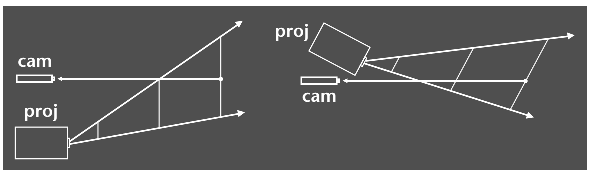


Normal 'straight-ahead' v. 'anti-keystoned' projection.

say, a standard slide projector, whose projected image center will always coincide with the aim-point of its lens; moving the image higher on the wall necessitates propping up the front end of the apparatus, but the image then becomes trapezoidal ('keystoned').

An I/O Bulb needs its camera to look at the same part of the world that its projector is lighting up.

The I/O Bulb idea, meanwhile, clearly only works if the region seen by its input-camera is the same as the region painted by its output-projector. Given that available projectors cannot truly project 'forward', we are left with two prospective geometries for achieving proper coincidence of input and output: we can either separate the camera and the projector, so that the regions treated by each are precisely the same; or we can keep the camera and projector together (at least as close as is geometrically possible) and tip the projector downward to bring the center of the projection into alignment with the center of the camera's view. The significant shortcoming of this latter arrangement is that not only is the resulting projection trapezoidally distorted – requiring every software client to compensate by using expensive counterdistortion processing on any imagery that's to be displayed – but the plane of focus is also tipped with



The annoyance of anti-keystoning: two camera-aligning tactics.

respect to an orthogonal projection surface: correct focus is now impossible.

With this in mind we have elected to embrace (for now, at least) the spatially-separated-camera-and-projector option. One objection to this is philosophical in origin: the I/O Bulb is supposed to be a compact device in which the optical input and output mechanisms are more or less collocated. This ideological disparity we are willing to tolerate in the short term, as long as it presents no impediment to the development of the end applications that are our goal.

We'll choose the one on the left.

But a more serious concern surrounds the issue of whether the two system components are optically coaxial or not: is there really no parallax between the camera's view and the projection axis? Some maintain that only a true 'zero-parallax' (precisely coaxial) system can ever work; and indeed there are reasons that minimizing parallax is advantageous. The fact is, however, that for an arrangement in which all objects and projections are restricted to a single plane – which is our arrangement – the parallax issue is moot. (As always, of course, the proof is in the pudding. Our pudding, described below, is a temporary-measure I/O Bulb prototype with a very large amount of camera-projector parallax. It works quite well.)

Does the central axis of the camera's view have to coincide optically with the projection's?

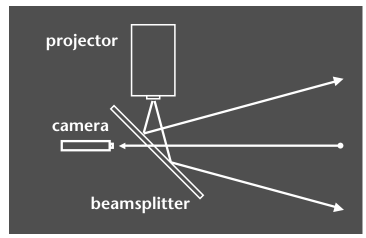
No, as it happens.

As a side note that nevertheless appears in the main flow: until some enterprising lass or lad succeeds in fabricating an integrated silicon device in which 'projection pixels' are literally interleaved with 'camera pixels', the only way to achieve a zero-parallax optical I/O system is through the use of a beamsplitter (half-silvered mirror). In fact, we built a version of the I/O Bulb early in the research that used this technique and immediately discovered the fundamental drawback that renders such an

… though we did build just such a zero-parallax I/O Bulb.

approach completely unworkable: scatter. Even an absolutely clean beamsplitter scatters a small fraction of the light that's incident upon it; since output visibility in a

A beamsplitter gives us zero parallax, but scatters too much projector light into the camera.

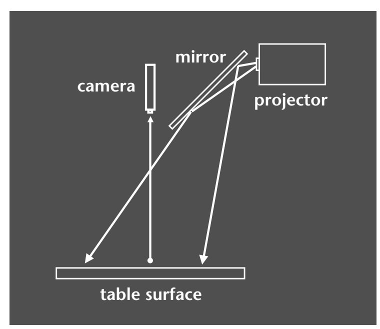


normal work environment requires the projection component of the I/O Bulb to be very high-intensity (the more so because the beamsplitter 'throws away' part of the light that would normally reach the projection surface), a fair amount of light is unavoidably scattered from the beamsplitter surface. The camera must look through this surface as well, and whatever it might have seen of the environment outside the I/O Bulb is now drowned out by this scattered projection light.

2.1.2 But So Anyway:

The upshot of all this is that our current development environment uses an I/O Bulb mock-up whose camera and projector are separated by a fair distance and exhibit a large amount of parallax. And while this two-piece

The I/O Bulb we'll use isn't in its ideal or final form, but for now it's close enough. For one thing, it works.



construction is certainly antithetical to the prescribed form of the I/O Bulb – i.e. collocated projection and optical capture – the resulting workspace is still a perfectly adequate development environment for the applications that are, after all, much of the point.

People can't figure out where the light comes from.

Curiously, we find that in practice and with almost no exceptions users of the applications served by this apparatus cannot easily locate the source of the projections, though we have made no effort to hide or disguise the projector. These visitors typically give up the search, or request an explanation, or conclude that the projected light originates within the lamp-like camera housing; and thereby – inadvertently – the illusion of a dogmatically correct I/O Bulb is maintained.

An unforeseen advantage of this physically distributed design emerged subsequently. For typical 'workbench' applications in which operators stand or sit at the front of the table (i.e. on the left side in the diagram above) and at its sides, occlusion by an operator is less of a problem than would be the case with a zero-parallax system: shadows from an operator's hands and arms tend to be thrown forward – that is, toward the operator herself.

Our 'broken' I/O Bulb actually mitigates operator self-occlusion problems.

We will thus, at least for the moment, assume the existence of the essential I/O Bulb hardware. For most of the experiments and sample applications that make up the dissertation we rely on this 'temporary' I/O Bulb implementation. A stab at building a 'true' version of the I/O Bulb apparatus is described later.

For the purposes of application development, we've got our I/O Bulb.

2.2 Basic Software Infrastructure

What's ultimately more interesting, since it's not so hard to believe that binding a miniaturized projector and a tiny camera physically and optically together will soon enough be a straightforward thing, are the fundamental software components required for the successful implementation of Luminous Room and I/O Bulb scenarios as we imagine them. What follows will discuss the basic vision algorithms on which our existing applications are based.

The hardware we can decide to approximate for now; but how to make the I/O Bulb see?

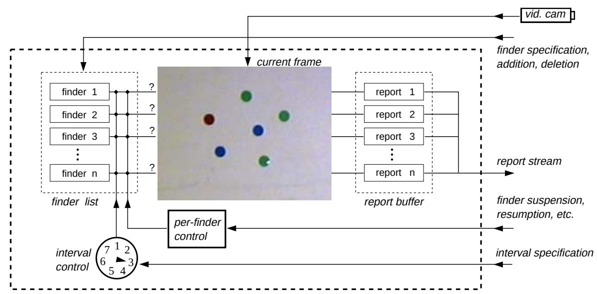
2.2.1 Simple Vision: glimpser

For reasons both of reliability and of computational efficiency, we have decided to build upon a very modest 'raw vision' model: the *glimpser* program (now also in use in several other unrelated projects around the lab) simply identifies colored dots in its visual input. *glimpser* accepts commands from its master application to define, create, destroy, and condition 'finders'. Each finder is an independent directive to locate within the input frame a

Simplest possible vision method buys us speed and reliability.

specific-sized region of some particular color. Finders,

We'll look just for colored dots.



once created, can be restricted to a certain subregion of the input field, can be temporarily deactivated or fully reactivated, and can be 'de-emphasized' to be evaluated less frequently in order to streamline the search when input movements are known to be slow or very sporadic. Finally, each finder may be instructed to report only one color-spot location per frame, to report up to some fixed number of spot locations per frame, or to report fully as many spot locations as may be found per frame.

This dot-finding program – *glimpser* – is by nature networked, so it can do its job anywhere.

glimpser is implemented as an isolable server in which requests and directives are received and resulting reports are transmitted over a TCP/IP connection. In this way *glimpser*'s predictably heavy computational demands may be fobbed off onto another CPU altogether, leaving the 'main' CPU freely available for the full simulation and rendering needs of the actual application in question; or, for lighter tasks, *glimpser*'s low-level vision efforts as well as the application-specific calculations can be assigned to the same machine, in which case the TCP/IP connection that links *glimpser* with the end application devolves (at least under respectable operating systems) into a high-speed and low-latency internal channel. *glimpser* has been used with satisfactory results in both guises.

2.2.2 The Utility of Dots

Patterns of colored dots make unique object ID tags.

The point of this color-dot-finding is that, in nearly all of the applications we'll describe, individual physical objects are tagged with unique colored-dot patterns: for a variety of reasons, not least of which is the desire to maximize reliability and stability while minimizing per-frame computational cost, we decided at the outset of all our implementation to eschew higher-level machine vision techniques (like template-matching) that attempt to identify objects through shape and other per-object attributes.

Instead, our intent was a kind of object-independent tagging scheme that – while enjoying the benefits of machine vision, like inherent multiplexing – would exhibit a special flexibility. For example, if we decide that an application needs to be able to recognize a new object, we need only declare the unique dot pattern that will be affixed to this object; depending on the structure of the application and the intended function of the new object, this addition may not require recompilation, or indeed even restarting the application. An object-centric vision scheme would, on the other hand, require some form of 'retraining'. At the same time, our dot-pattern vocabulary is arbitrarily extensible, limits being imposed only by available physical space (obviously, we need the patterns to be small enough to fit on the object they identify) and the syntactic richness of the pattern space we establish.

A color-pattern language that lets us add new objects without retraining the software.

An important implementation issue is the reliable isolation of genuine color dots from an unpredictable background. To wit: even with highly saturated colors chosen as pattern-dot 'primaries', the dots are at best still Lambertian reflectors. Thus there is no way to guarantee (1) that the same hue will not be present in garments, skin pigments, or unrelated objects in the environment, or (2) that brightly-illuminated surfaces in the environment may become color isomers of the dots' own hues through aliasing of the CCD's chromatic response curves. So irrespective of the sophistication of *glimpser*-level algorithms, false positives will be reported and genuine dots ignored with crippling frequency. Making the dots self-luminous (say, by embedding small LEDs) would solve the problem by boosting the luminance of each to an unambiguous level in the video input field, but would also violate our policy – here during Act 1 of the research, anyway – that objects used by Luminous Room applications be passive.

But these dots: how can we make sure *glimpser*'ll pick them out of the background reliably?

CCD characteristics and illumination issues conspire constantly to make other stuff look like our dots.

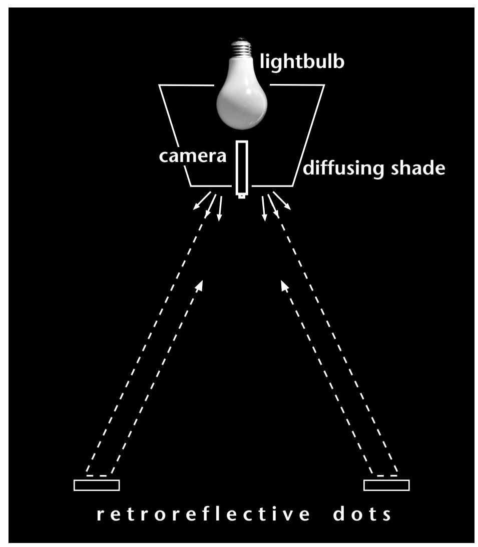
Instead, we've elected to use retroreflective dots complemented by a low-intensity, diffuse light source around the I/O Bulb's camera. A first round of dot-design employs a disk of panchromatic 3M ScotchLite material covered with a colored gel (red, green, or blue). At the same time, a moderate 60W lightbulb (of the tra-

Retroreflective material beneath the dots' color gels and a soft light source near the camera guarantee much brighter dots.

ditional variety) is incorporated into the I/O Bulb structure, placed directly above the slim video camera. A diffusive shade is positioned around the whole, with the lens of the camera protruding from the bottom.

Most of the dots' light goes back into the camera.

Each dot is then illuminated by the annular diffuser and, irrespective of the angle of the light's incidence on it, reflects a gel-filtered version of most of this light directly back into the camera's lens. Because of this



Thus an old-fashioned lightbulb turns out to be a critical component of the I/O Bulb.

angularly selective reflection, human viewers do not perceive the dots as other than normal surfaces; they seem no brighter than anything else. But from the privileged position of the camera, the dots glow fiercely: typically 2-4 stops brighter than any other part of the visual field. The critical result of all this is that it is now necessary to stop down the camera (either optically or electronically) in order to bring the high-luminance dots within its dynamic range – but doing so renders most of the rest of the input field black. Only dots are left to be seen; reliable dot isolation is thereby assured.

The dots are consequently so bright that the camera must be stopped down – thus removing the pesky background.

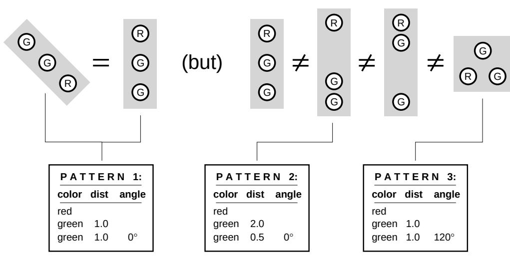
New, even more chromatically selective dots are now being constructed as a single layer, cut directly from new tinted ScotchLite sheets. The color selectivity of these materials is good enough that we have also added yellow and brown to our corral of recognized dot colors, extending *glimpser* accordingly.

glimpser has thus far provided a way to locate, video frame by video frame, the individual dots that comprise our identifying patterns. Now we must somehow sort through each frame's cloud of unassociated dots to extract the patterns within.

2.2.3 voodoo

An application-independent geometric parsing toolkit called *voodoo* interprets the simple colored-dot-location output of the *glimpser* program. *voodoo* analyzes each unorganized per-frame collection of found color dots into a list of unique patterns that have been regis-

voodoo: a way to get from uncorrelated dot-clouds to patterns.



tered with it by the application it serves. These patterns specify a sequence of colors; associated with each pair of adjacent color dots in a pattern is a distance, and with each contiguous triplet of dots a angle. These two parameters – the distance between each pair of dots and the angle through each triplet, along with the dots' color sequence – are enough to uniquely define any arbitrary pattern; and so one such pattern is assigned to each of the client system's known objects, both physically (colored dots are pasted to the top of the object) and computationally (the pattern is registered with *voodoo*).

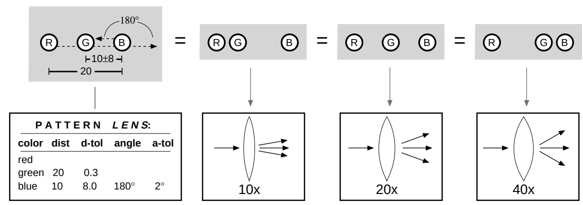
one dot: a color
two dots: a distance
three dots: an angle

Meanwhile, each distance and angle specification has associated with it an individual tolerance within which range a 'match' will still be recognized. The intent of this provision is twofold. First, such a measure permits *voodoo* to absorb the inevitable inaccuracies and occasional single-pixel indecisions of machine vision algorithms – without this kind of allowance, our vision-based pattern matches would simply fail most of the time. Second, the tolerance specification makes possible the definition of 'parametric' patterns: for example, a lens in an optical simulation system might be identified as the sequence 'red, blue, green' with a some distance and a small tolerance specified for the red-blue pair, but with a 180° turn required between the red-blue and blue-green

Because of video noise and finite resolution, we need to build some slack into each pattern.

segments and a large tolerance for the blue-green distance. This means that a lens would be identified wherever a red and a blue dot are appropriately spaced and have a green dot *somewhere* along the line between

We can use the same pattern-slack to make adjustable objects.



them; but the application can then use the relative position of this intermediate green dot to define the power of the lens (10x, 20x, 40x, etc.). Thus, definable distance and angular precisions can provide a kind of end-user-adjustable degree of freedom – a way to build simple sliders and dials.

Take into account time-to-time occlusion by the operator's hands.

voodoo also provides an 'object persistence' mechanism: it's necessary to assume that low-level vision will occasionally fail (for a frame or two) to report the presence of extant color dots, and – more critically – that users' hands will intermittently occlude dots. In these cases, we would like the object-representing patterns identified in previous frames to exhibit a bit of 'temporal inertia'. The persistence mechanism, then, allows objects to continue to exist for a short while even in the absence of positive visual data, and is implemented as a coherence algorithm that attempts to produce a one-to-one match between the patterns detected in the current frame and the patterns from the previous frame. The algorithm allows for a certain amount of translation and rotation frame to frame; the parameters specifying these amounts may be adaptively adjusted to reflect changing frame rates and expected user-induced object velocities.

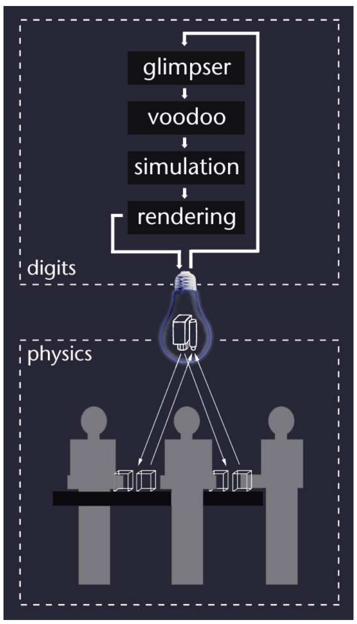
Match *before*'s objects with *now*'s objects to see which are new and which are only moved.

What's important is that, unlike techniques that rely on temporal averaging or statistically based prediction, this approach adds no delay: rather, it provides a kind of 'latching' or one-way hysteresis so that every positional/attitudinal change for each object is instantaneously reported to the client simulation and reflected directly in the simulator's output, while occlusion or low-level vision dropouts result in a freeze of the object's understood position but not in its disappearance.

This method provides extra reliability, yet incurs no delay.

2.3 The Way of Things

The general model of operation followed by most of the applications constructed so far involves two simultaneous interaction loops bound together through the



People work with tangible stuff; the machines work with numerical stuff; and the I/O Bulb binds together both kinds of participants with light.

functions of an I/O Bulb. The first is fairly self-evident and is the domain of humans, who manipulate physical objects ('output from human') and apprehend both the instantaneous state of these objects and the luminous augmentations projected into real-world alignment by the I/O Bulb (all of which, physical and luminous, constitutes 'input to the human'); this cycle continues.

Simultaneously, the second loop is a cyclical pipeline in the computational realm – whether it's localized in a beige box somewhere or dissolved into the architecture itself is irrelevant here – and begins with a stream of consecutive video frames of the environment provided by the I/O Bulb; these undergo low-level vision analysis (*glimpser*), which in turn reports to the object identification module (*voodoo*); the identities, positions, and orientations of the objects thus found are made available to the particular application in use for the purposes of updating its underlying simulation; the application is then responsible for rendering graphical output (using metrics provided by the I/O Bulb that describe its geo-

From video of the world, to low-level vision, to high-level vision, to simulation, to rendering, back out to the world.

metric relation to the real world) and projecting this luminous output back into physical space.

It should go without saying that this second interaction loop, while the object of our engineering attention, should remain as invisible to participating humans as possible. So it is that a primary goal of the work recorded here is to design the technology so that the interactions it makes possible simply seem part of what the surrounding architecture *does*.

Various other vision schemes are possible.

Of course, the duumvirate of *glimpser* and *voodoo* is only one of many possible answers to the machine vision needs of every I/O Bulb-based system; in general, these two steps can be replaced as necessary with other vision schemes, as indeed is the case with Chapter Five's standalone fluid-flow application.

2.4 The Right Applications

I/O Bulb idea's true proof only through true usefulness.

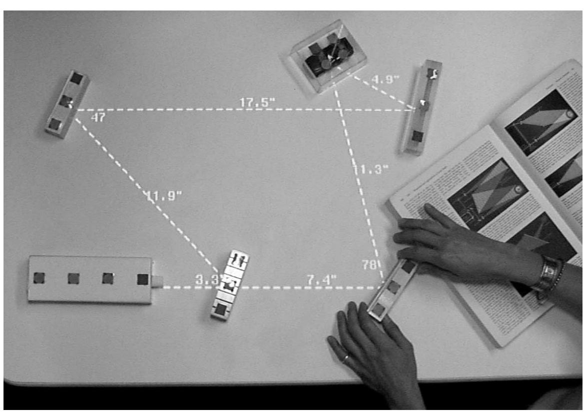
It is crucial to our cause of vindicating the I/O Bulb and Luminous Room ideas that we be able to show how they might support significant, useful applications. The best way to do this is to build several significant, useful applications. In doing so, we not only approach an answer to our second thesis question ('should we?') but can also – if the vindicating applications are well chosen – begin to form an understanding of what domains are well-suited to treatment in I/O Bulb form and how a given domain's translation from real-world practice (or traditional on-screen practice) to luminous-tangible practice can best be carried out. Naturally, these higher-level understandings can also benefit from analyzing those applications and avenues that don't necessarily have strict utility.

(Though we'll learn from uselessness too.)

The next four chapters describe the working applications (some a full 'professional' success, others mere sketches or design studies) that demonstrate use of I/O Bulb and Luminous Room structures. Following that we offer a preliminary discussion of luminous-tangible design issues.

3 First Application: Illuminating Light

Two optical engineering students stand at an ordinary table. One pulls from a box a stylized plastic object - it looks roughly like a laser - and places it on the table. Immediately a luminous beam appears to shoot forward from the laser model's aperture. The student moves the laser from the center to the corner of the table, and the beam tracks along with it, always originating at the same point on the laser's front surface. The second student places a small aluminum representation of an optical-grade mirror on the table, and then moves an additional model representing a beamsplitter into the path of the existing laser beam. At this point of intersection a second, weaker beam is generated, bouncing off the splitter's surface. The student rotates the beamsplitter model in place (the partially-reflected beam sweeps across the table in response to the changing orientation of the splitter) until the reflected beam strikes the mirror set out earlier. The first student, meanwhile, is grasping this faux mirror and swivels it until the beam now also reflected from it runs the length of the table, parallel to the part of the original laser beam that continues through the beamsplitter.



Optics and holography: physical models, digital laser beam.

During these and subsequent manipulations, the various optical components – though simple inert representations (unwired and sensor-free) – behave very much as their real counterparts would, directing and modifying the light that passes through them; and these physically accurate 'beams' of light are wholly simulated and projected down in careful registration with the optics. The students continue adding and adjusting components until a complete hologram-recording setup has been constructed. As they work, *a continuously updated display at the far end of the table shows the layout's relative optical pathlengths as well as a rendered simulation of how the object would appear in a real, analogously recorded hologram.*

3.1 The Domain: Optical Engineering

For a variety of reasons, holographic-optical engineering emerged as an ideal first field for our attentions. High-quality optical elements are simultaneously expensive and notoriously susceptible to damage: a single fingerprint can destroy a two-hundred-dollar front-surface mirror instantly and permanently. The breadboarding tables on which experiments are constructed and prototypes built – often floated on sensitive vibration-isolation air pistons – are a scarce resource. At the same time, the precision required of laser-based optical systems frequently results in long setup and iterative refinement times (times that generally also exceed the time spent running the actual experiment).

Work with real optics – which're expensive yet fragile – is slow & painstaking.

All of this suggests that a well-designed 'simulated optics workbench' could be a valuable tool. Such a workbench should permit the optical engineer to tinker with a setup, interactively manipulating an accurate simulation of the evolving layout and its operation. Having eventually arrived at an optimal configuration 'offline', the engineer could then rapidly reproduce the setup on the real table to perform the end experiment.

But if it were possible to design experiments and setups away from the lab?

Several powerful mouse-and-CRT-based optical layout and analysis packages exist (LightTools, ACCOS, ZEMAX, OptiCAD, &c.). However, intuition for the behavior of optical systems and for their proper design comes principally through physical interaction with real-world components; for many of the field's students, theory does not gel until the effects promised in textbooks can be observed and manipulated firsthand in the laboratory. Thus, a simulator whose input and output were arranged to emulate the real thing – not just visually, but haptically and spatially as well – could foster and exploit such geometric understanding skills. In short, we set out to provide a virtual optical workbench with which experimenters could physically manipulate three-dimensional stand-ins of different optical components and directly observe the results.

That's possible via CAD-style programs, but without the fluidity and spatial intuition of using real optics.

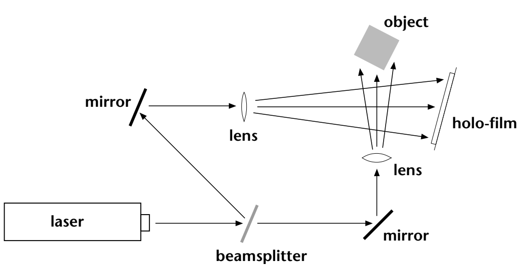
Additionally, in applied holography 'correct' design solutions are generally discernible from 'incorrect' solutions, allowing us to evaluate the usefulness of our system: can practitioners build optical design X more easily, more quickly, more imaginatively with the system than without? Finally, the presence of an established and ongoing program in holographic imaging within the university promised a ready supply of holographers, both student and professional, who could be invited to use the system and be observed doing so.

We were lucky to have a nearby supply of quality holographers.

3.2 Basics of Holography

The mechanics of holographic recording are conceptually simple. A fine-grained photographic plate is exposed simultaneously to two beams of light: one, called the ‘object beam’, comprises light scattered from laser illumination of the object or scene being recorded; the other, called the ‘reference beam’, is uniform, unmodulated laser light [6]. In order for a stable (and thus photographically recordable) interference pattern to result from the overlap of these two beams, they must originate from the same laser. This is accomplished with a beamsplitter – usually a partially silvered mirror – which allows some of the laser's light to pass undiverted through it while reflectively redirecting the remainder into a second beam. Moreover, because of the limited coherence provided by prevalent Helium-Neon lasers, it is a typical constraint of holographic setups that the length of the object path and that of the reference path (as measured from the beamsplitter, where these two paths become distinct) must be equal. Additional geo-

Normal photography uses only one light-path (scene to film). Holography requires a second simultaneous path (laser to film).



One standard geometry for recording a transmission hologram.

metric requirements are often imposed on the setup, such as the desirability of some particular angle of incidence of the reference beam on the recording plate. Finally, the distance from and angle at which the object is illuminated are inevitably of great import, as these factors directly control the eventual appearance and aesthetics of the three-dimensional image produced by the finished hologram. Thus the principal challenge of designing a working holographic layout is the simulta-

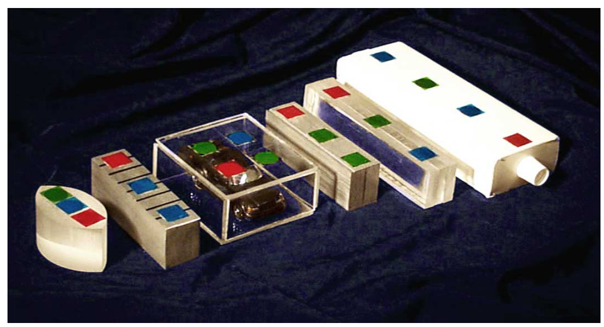
So holography, like many other optics problem, is about arranging elements correctly in space.

neous satisfaction of various geometric requisites with a single configuration of optical components.

3.3 Optics As Optics

No interface: grab a mirror when you want to use a mirror.

The intent was that using the simulator should be as close to working with the real thing as possible. This necessarily suggested an interface built to provide a 'direct manipulation' style of access to the elements in question. Thus, instead of providing indirect tools – a single general tool, say, for instantiating many different optics, another for sketching beam paths, etc. – we would provide the optics themselves. If an experimenter needed a lens, rather than using some physical analog of a menu to 'create' a virtual one she would just grab the object that looks like a lens, placing and adjusting it within the



L to R: lens, beam-splitter, object, mirror, film, laser.

setup as desired. The lens object would exactly recapitulate the functions and effects of a real-world lens (at least to the limits of the optical simulator); the laser-object would work like a real laser; the mirror-object like a real mirror; and so on: optics as optics.

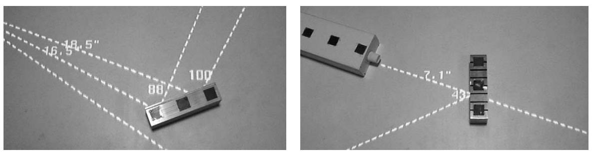
voodoo dots on each.

We designed a set of simple objects, each intended to suggest visually its real-world counterpart (with the exception of the holographic 'recording subject', a detailed automotive model). Affixed to the top of each optical model was the unique pattern of colored dots that would identify it to the system by way of *glimpser* and *voodoo*.

3.4 At Last, Then

Our first full application: *Illuminating Light*, for laser and optics experiments.

The result of all this was the first full system demonstrating the I/O Bulb infrastructure: *Illuminating Light* – as illustrated by the scenario at the outset of the chapter – is a general ray-based optical simulator with additional holography domain knowledge. Users of Illuminating Light freely manipulate simple models of optical elements like lasers, mirrors, and lenses; the position and orientation of each element is used by the simulation to determine the path(s) of the laser beam. These continuously updated and evolving paths are then rendered and projected visibly back into alignment with the real-world models on the tabletop.



Mirror reflects, beam-splitter reflects & transmits.

Additional information of particular interest to optical and holographic engineers, like the distance between successive elements traversed by the beam and the bounce angle of the beam as it reflects off mirror surfaces, is unobtrusively projected directly into appropriate locations in the physical setup. Finally, once an optically viable hologram-recording setup has been constructed the Illuminating Light system displays a rendered simulation of what the corresponding real-world hologram would look like, were the layout to be replicated in a real lab with a real laser, real film, etc.

It's easy to put useful ancillary information right in the real-world geometry.

3.5 Discussion

Illuminating Light has proven a substantial success. Those who've experimented with it report that the illusion it offers – inert optics models brought to accurate life – is convincing enough to be ignored. The domain that it addresses is a real one; practitioners would normally design their experiments using either actual optical components (expensive, fragile, confined to specialized lab spaces) or one of various CAD-style simulators (whose mouse-keyboard-CRT interfaces, even were they well designed, greatly hamper efficiency and flexibility in what is after all a physical undertaking). Illuminating Light demonstrates something somewhere in between, in which the advantages of dextrous manipulation acquire the safe freedom and greater flexibility of the digital. Also, small children who know nothing of optics have been observed to fight each other for access to Illuminating Light.

A handy lab away from Lab.

Existing spatial skills respected, new ones fostered.

Kids love it too.

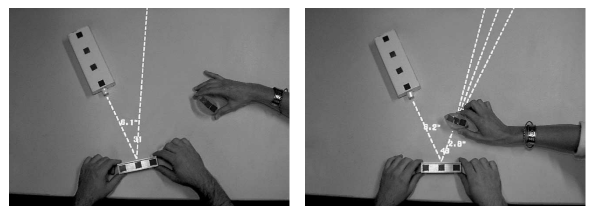
A facility of the system that became quickly apparent, though it had never been explicitly designed, was for two-handed manipulation. By their nature, vision algorithms perform a kind of spatially-distributed input in which as many changes as may be present at any particular time are always simultaneously apprehended.

No such thing as a two-person mouse.

This is in clear contrast to the model of input that depends on mice and tablets; there, the fantastic mechanical dexterity resident in human hands is forced through a tiny keyhole – a pointing device – that registers a single degree of freedom (position) only. Two positions are impossible to indicate simultaneously with a mouse, as are quantities like twist and orientation.

Two-handed and collaborative manipulations for free.

The system similarly supports collaborative activity within its workspace (in which case the two, or five, or



twenty hands just happen to belong to different people). Obviously, these convenient properties – multi-handed and collaborative input – belong not just to Illuminating Light but will be shared by every application that we build with the I/O Bulb.

CHI (like children) appreciated Illuminating Light.

The Illuminating Light application was presented at (and published in the proceedings of) CHI '98 [19].

3.6 Further Issues

Once we had built, experimented with, and observed others experimenting with Illuminating Light a host of questions regarding the larger implications of I/O-Bulb-based applications began to form. From among these the following two are appropriate for approach now.

3.6.1 All These Objects

Although a much more detailed discussion of the nature of the physical objects used in luminous-tangible interactions is undertaken in Chapter Seven, we may begin here to consider the somewhat simplified case that obtains in the Illuminating Light application. Here, models of optical elements stand in, one-to-one, for real optics. But what, fundamentally, are these models?

What really are the optics models?

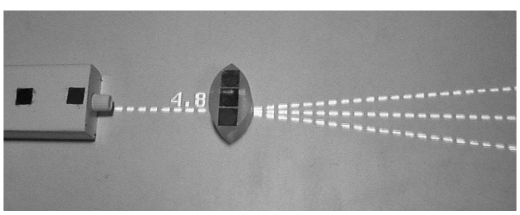
Earlier work in MIT's Tangible Media Group had introduced the term 'phicon' (that is, 'physical icon') to refer to corporeal artifacts whose use in digital systems was a close analogue to the use of icons in more familiar screen-based windowing environments. The most potent incarnation of the phicon concept was a collec-

Are they 'phicons'?

tion of small acrylic models of particular buildings from MIT's campus; one of these building-models placed on a special rear-projection display system running the 'Tangible Geospace' [16] application would cause a free-floating map of the entire campus to attach itself, aligning its position so that the map's representation of the building lay just under the physical model. An amateur cartographer who moved the phicon-building would cause the projectively attached map to follow along. Placing a second building on the surface would provide a second alignment point, so that now relative manipulations of the two phicons could be used to both scale and rotate the map below.

Ullmer & Ishii's *Tangible Geospace* used model building phicons to control display of a map.

The suggestion exists that the optics models in Illuminating Light fulfil a role similar to the building-phicons of Tangible Geospace; that notion stems simultaneously from two observations: one to do with the visual appearance of the models and the other to do with their function within the simulation. The first point pertains mainly to the lens, whose highly stylized shape can be seen as



A lens doing what lenses do.

an extrusion of the two-dimensional symbol for a lens used in many optics texts (rather than the radially symmetric 'surface of revolution' that is the more common form of a real lens). The argument goes that Illuminating Light's lens must be a phicon because it is a physical incarnation of a graphical icon (from the publishing domain). The second claim is that the optics models perform functions that are abstractions of the functions performed by their real-world counterparts.

Our lens is a literal extrusion of textbooks' lens symbol.

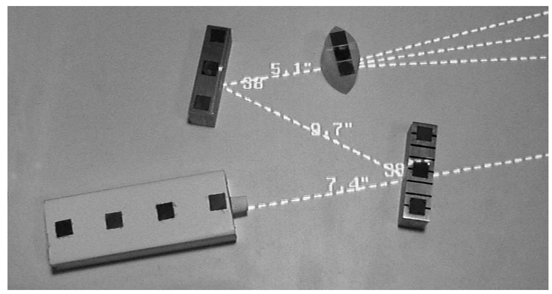
We would argue against this reasoning. While the boundaries of the term 'phicon' are nowhere exhaustively declared, these limits clearly have to do with the meaning of an object – which in turn is more critically tied to function than to form. The optics models are indeed iconic in form. But they could just as easily be 'actual' optics (with the voodoo tag applied, any object will perform the lens function in the simulation, irrespective of how much it looks or does not look like a lens) or again just as easily be completely unrelated objects. So

But what's the function of our lens-model? Is it abstract (icon-like)?

we come to function.

What's important here is that – because of the nature of the simulation – the optics models used in Illuminating Light are not abstractions; from the point of view of the operator, they *are* optics. The lens model functions

Each of Illuminating Light's optics is neither more nor less than its real-world counterpart.



just as a real lens would. Again, the accuracy or assumptions of the current underlying simulation should not be the issue. At the moment it's ray-based; we could evolve more and more complexly accurate simulations to include chromatic effects, diffraction effects, quantum effects, &c., until the range of phenomena exhibited by the lens model's interaction with the rest of the system was as detailed as the real thing. Critically, the model lens *implies* neither more nor less than a real lens: the meanings are isomorphic. Contrast this now with the function of the building models in Tangible Geospace: these models in no way recapitulate the function of their real-world analog; rather, they serve as iconic representations of the position of the building in question. (None of this is criticism of what is an honorable piece of research; the point's just that the *function* of the physical artifacts is in that case very different.) A Tangible Geospace pilot would never claim that the building models *are* buildings.

Tangible Geospace's little buildings aren't meant to be used as real buildings.

Computer-desktop icons are severe abstractions of other digital entities (files, for example), and neither pretend to nor are able to fulfil the same roles as what they stand in for. Typically, these icons simply denote *existence* (i.e. that a thing exists) and static or evolving *location* (i.e. the file's here, and now you're moving it there). Icons are not isomorphic to what they represent: you can't edit text in the icon that represents a text file, and you don't edit a text file icon in a text editor. The same general relationships must hold between phicons and the entities they represent.

A computer desktop's icons distill what they represent down to mere *existence* and *location*.

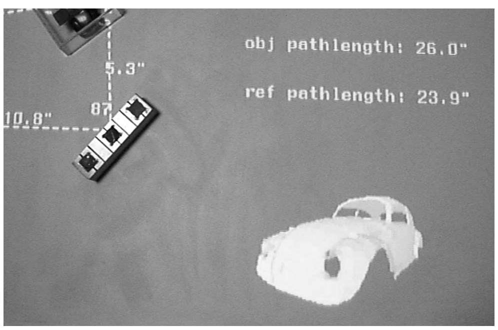
So the optics models of Illuminating Light are not phicons. What precisely they are will be taken up again in Chapter Seven.

Our optics models are not phicons.

3.6.2 Continuity of Space

A similar and similarly intriguing issue exists regarding the meaning of space. By and large, the Illuminating Light application transforms the table surface on which its models rest into a space where simulated physics is resident and in which experiments may be carried out. This reading of simulation space as real space is complicated, however, by the presence of the small 'pure display' region in which textual readouts (pathlength matching information) and the three-dimensionally ren-

Two different spaces: one where physics happens; another where meta-information is shown.



The small 'playback' region of Illuminating Light's table.

dered playback of the resulting hologram are placed. So in fact the application's space on the table surface is broken not only by a geometric discontinuity but by a cognitive discontinuity as well: the meanings of the two adjoined but separate spaces on the table are quite different.

Is the juxtaposition of two disjoint meaning-spaces in an application like Illuminating Light proper? The question, seemingly reasonable, is in fact wholly meaningless. Are we allowed to build spaces like this? Are we disallowed? There are no rules; rather, the issue is one of design, a domain in which there are always examples that work and examples that don't. A better-formed question might be this: is the juxtaposition cognitively tractable? Can experimenters understand it and work with it? Well, yes. There's no reason to think that anyone'd be any more confused by the partitioning of space in the Illuminating Light application than they are by the partitioned use of space on, say, their office desk. Do you confuse the blotter on your desk with the phone on your desk, perhaps trying to place a letter on the

Is it kosher to break up the space like this?

Sure, as long as whoever has to use the space understands it.

phone in order to sign it? Typically no. The uses and meanings of these two desk spaces are wholly disjunct, but we handle the disparity reliably and with ease.

In the real world spaces are broken up constantly, but we manage there quite well.

Of course, the visual manifestations of the phone and the blotter are good clues and cues to their identities and differences. The phone looks like a phone, the blotter like a blotter. What if they looked identical? Even then we could still use the space properly so long as we knew which was located where and were assured that the positions would not switch or migrate. So this is one preliminary piece of the crux: operators of I/O Bulb applications can probably use various abutted spaces without confusion so long as they either have a way to correctly anticipate the locations of the disparate spaces or are provided with visual (or other sensory) cues as to the identity of each one.

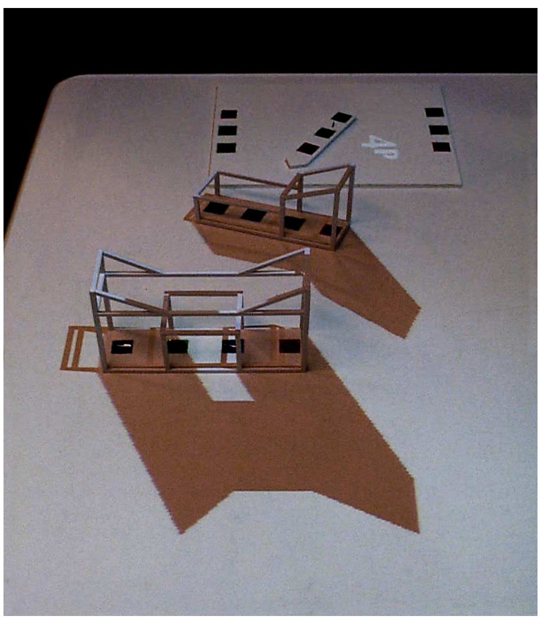
Illuminating Light has little ways of letting humans know which space is which.

The spaces in Illuminating Light implicitly subscribe to both these approaches. After a moment's experimentation, an operator will have learned that the optics space is on the left, the 'readout' space on the right. More fundamentally, though, the constant presence of text and graphics in the readout space is itself a good marker of that space's type – without a more explicit demarcation of one space from the other.

We will see these issues of space recur.

4 Second Application: Urp

Two urban planners, charged with the design of a new plaza, unroll onto a large table a map showing the portion of the city that will contain their project. They place an architectural model of one of the site's buildings onto the map. Immediately a long shadow appears, registered precisely to the base of the model, and tracks along with it as it is moved. They bring a second building model to the table and position it on the other side of a large fountain from the first; it too casts an accurate shadow. "Try early morning", requests one of the planners. Her colleague places a simple clock on the map; a glowing '3pm' appears on the



Giving life to inert architecture models.

clock's face. The colleague rotates the hour hand around to seven o'clock, and as '3pm' changes to a luminous '7am' the shadows cast by the two models swing around from east to west.

It is now apparent that in the morning the second building is entirely shadowed by the first and will receive no direct sunlight. The urban planners decide to try moving the first building south by eighty yards, and upon doing so can immediately see that this solution restores the second building's view of the sun. The just-moved building is now only twenty yards to the north of an east-west highway that borders the plaza on the south; one of the planners places a *long road-like strip of plastic on top of the map's representation of the highway, and tiny projected cars begin progressing at various speeds along its four lanes. The other planner brings a wand into contact with the nearby building, and the model's facade, now transformed to glass, throws a bright reflection onto the ground in addition to (but in the opposite direction from) its existing shadow. "We're blinding oncoming rush-hour traffic for about ninety yards here at 7 AM", he observes. "Can we get away with a little rotation?" They rotate the building by less than five degrees and find that the effect on the sun's reflection is dramatic: it has gone from covering a long stretch of highway to running just parallel to it.*

The urban planners position a third building, near and at an angle to the first. They deposit a new tool on the table, orienting it toward the northeast: the prevalent wind direction for the part of the city in question. Immediately a graphical representation of the wind, flowing from southwest to northeast, is overlaid on the site; the simulation that creates the visual flow takes into account the building structures present, around which airflow is now clearly being diverted. In fact, it seems that the wind velocity between the two adjacent buildings is quite high. The planners verify this with a probe-like tool, at whose tip the instantaneous speed is shown. Indeed, between the buildings the wind speed hovers at roughly twenty miles per hour. They slightly rotate the third building, and can immediately see more of the wind being diverted to its other side; the flow between the two structures subsides.

4.1 Background

To rather harshly simplify a very complex field, the domain of urban planning involves the relationship between architectural structures and existing settings.

Urban planning: putting architecture in the right place.

4.1.1 Urban Planning Issues

This chapter's work focuses on the arrangement of architectural forms to both fulfill certain aesthetic goals and at the same time satisfy a variety of practical constraints. Among the primary constraints we will consider are the following:

- **shadows:** Does the proposed placement of a tall structure mean that from dawn until 10 AM no direct sunlight will reach an existing building that was formerly able to see the sunrise? Could such a situation be the source of a lawsuit? (Yes, in fact.)

• **proximities:** Is a building too close to a roadway? Is the distance between two adjacent buildings too small to allow adequate pedestrian flow? Is a building too far from an intersection?

• **reflections:** When a building with all-glass sides is erected as proposed, will low-angle sunlight (in early morning or late afternoon) be reflected directly into the eyes of oncoming motorists on a nearby highway? For what distance along the highway will this glare be present?

• **wind:** Does the placement of a building into an existing urban configuration result in a constant 80 km/h airflow over its north face? Does it result in a low-pressure zone on its east side that will make opening doors difficult?

• **visual space:** How will what pedestrians see change with the addition of the new structure? Will the space become visually claustrophobic? Will the new structure introduce a pleasing regularity into the skyline?

4.1.2 Standard Approaches

A host of traditional techniques exists for the treatment of these different constraints. Shadow studies are often undertaken by placing a strong light source above a model of the site in question; the exact position of the source is specified by a special table indexed through time of day, season, and latitude. This task is somewhat arduous, very difficult to adjust, and ultimately not quite correct (the source throws shadows from a finite distance, while the true sun's rays are essentially parallel as they reach our planet). Distances are of course easy to measure by hand. Reflections present further difficulties, however: adapting the shadow-technique (positioning light sources above the models) for reflections requires placing small patches of reflective material on various of the models' surfaces, but the difficulty of obtaining extreme flatness and full registration of these patches makes accurate results less than likely. Each of these concerns can also of course be addressed solely on paper using drafting techniques that involve tedious by-hand calculations [22].

By-hand methods are available to answer most of the questions (with lots of work).

Airflow analysis is another proposition altogether. Here, the only viable non-computational approach is the immersion of the model or models in a wind tunnel; smoke sources released upstream from the subjects can be used to show overall flow patterns. No matter the

Wind is more complex.

level of detail imposed on this kind of setup, however, the eventual scale of the phenomenon being tested differs from that of the simulated setting – fluid dynamics is sensitive to scale – so that observations are valid only to a certain extent.

As always, we could use on-screen tools, but at the expense of real-world context.

More recently, computational approaches to each of these analyses have become available. There are several CAD-style architectural applications (AllPlan FT, ArchiCAD, 3D Studio Max, AccuRender, etc.) that incorporate on-screen facilities for shadow and reflection studies. Airflow simulation is still a difficult matter; solutions to the prevailing *Navier-Stokes* equations are always approximate and expensive, and no known system allows real-time rearrangement of architectural structures within the ongoing simulated flow field.

4.2 Implementation

The models already exist; they may as well tell us a bit more about themselves.

It was our intent to construct an interactive workbench for urban design and planning that would collect together functions addressing the various concerns listed above; the novel aspect of our system would be that its information would all be centered on or attached to actual physical models of the architecture in question. The result of this effort is *Urp*.

4.2.1 Functions & Requirements

Shadows

We can see what shadow patterns will be like, just by moving the models and observing.

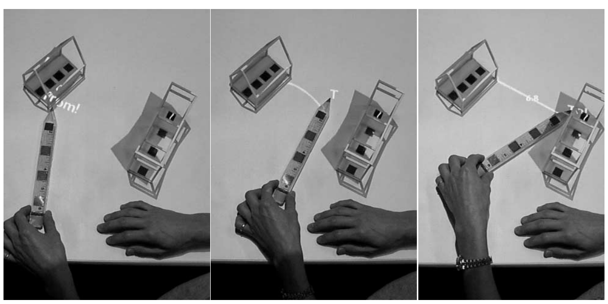
The shadow-casting facility was the first portion of *Urp* to be constructed, and was in fact the original catalyst for thinking about the field of urban planning: we'd asked ourselves "what if little models of buildings could cast adjustable solar shadows?". This function is very simple; any building placed in the working environment continuously casts a shadow, and the only influence available to the urban planner is the clock-object, whose instantaneous setting determines the time of day and thus the position of the computational sun (see the figure on this chapter's first page). If the clock object is removed from the workspace, time is 'locked' at its most recent value.

An inverse clock changes the time – and the sun's position in the sky.

An early incarnation of the shadow function allowed time to jump instantaneously between different values as the clock – which is quantized at every-hour-on-the-hour values – was adjusted. The resulting visual discontinuity was somewhat confusing, particularly during rapid changes from mid-morning to mid-afternoon: the shadow appeared to flop around in a way that (wrongly) suggested an inaccuracy of the system. Particularly when compounded with the inevitable small positional uncertainties that result from (genuine) video-noise-based imprecisions in the machine vision pipeline, this proved fairly confusing. Instead, the current system interpolates from one time value to the next using a cubic spline (the transition duration is about one second). This gives rise to appealing shadow transitions, whose smooth 'swinging' motions strongly recall time-lapse cinematography.

Distance Measurements

An initial test in which every building and road constantly displayed its distance from every other such structure left the workspace far too cluttered and visually



To show the changing proximity between structures, just connect them.

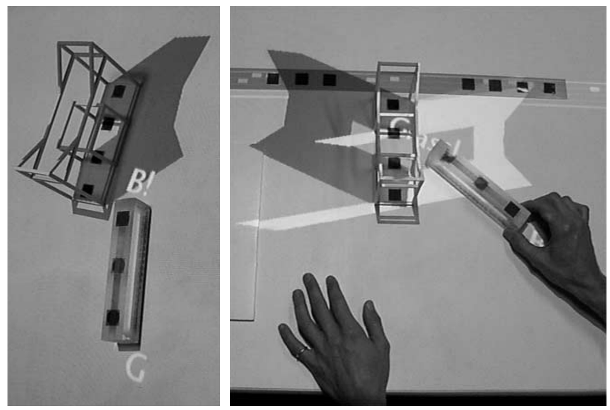
distracting. Rather, *Urp* now provides a distance-tool (shaped like a pencil but with the image of a ruler stretching between the pencil tip and eraser) that can be used to connect together *selected* structures. To do this, an urban planner touches the tool's tip to one building, on which one end of a sinuous line is then anchored; stretching the tip-end of the line away and eventually touching a second building or a road then connects the two structures, the line's curves flattening to leave it straight. A display of the line's length floats along and around it, and this number naturally changes as the connected structures are moved. When the distance display is no longer desired, touching the eraser end of the tool to either connection point disconnects the line.

Reflections

Long, thin *voodoo*-tagged strips represent roads; placing these in the environment engages a traffic simulation, whose automotive components are projected onto the plastic strips. Crossing two strips at any angle automatically generates an intersection with implicit traffic-control signals, so that cars come to a standstill in one direction while the cross-direction flows.

We have roads. But we shouldn't reflect low-angle sunlight into drivers' eyes.

A transparent wand placed onto the table shows a **B** at one end and a **G** at the other. Touching the **G** end of the wand to any building causes its facades to become glass, so that solar reflections are generated and pro-



A material wand changes buildings from brick to glass, and back again.

jected onto the ground. It is apparent that reflections are far less intuitive for most people than are shadows – in part because of the angle-doubling that occurs at the bounce surface, and in part because not all of the components of the reflection are necessarily in contact with the object itself; some small 'polygons of light' can be thrown huge distances away from the building that generates them, depending on the angle and orientation of the parent surface.

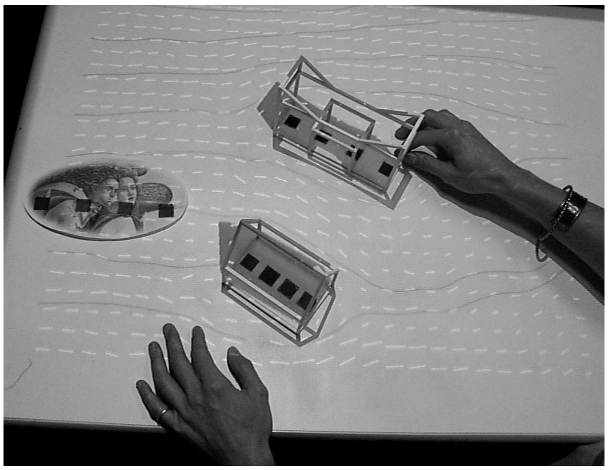
Intuition for reflections is much harder than for shadows.

Incidence of reflected sunlight onto the various roadways is always immediately evident, and it is easy to experiment with the small angular adjustments that give rise to large changes in these reflected patterns. Finally, touching the **B** end of the wand to a glass building transforms its facades back into brick, and the reflections disappear.

Wind Effects

Urp's airflow simulation is engaged simply by placing the wind-tool – a kind of inverse weather vane – anywhere on the table; orienting the tool selects one of eight quantized directions (the eight major compass points). The simulation is displayed as a regular grid of white segments, whose direction and length correspond to the instantaneous direction and magnitude of the wind at that position. In addition, ten red contour lines are shown, generated simply by 'connecting the dots' from location to location according to the local field vectors. These displays take a qualitative form; for more pre-

Start the wind blowing and see its flow everywhere in the workspace.



Obstacles like buildings affect the airflow; the change is readily visible and comprehensible.

cise measurements, the anemometer-object is available. Placing this arrow-shaped tool within the field samples and numerically displays the flow magnitude at the precise position of the tool's tip. Periodically, these numbers break off from the tool and go floating through the field as a further means of conveying the larger-scale flow patterns.

Although the airflow simulation is the most computationally expensive part of *Urp*, the entire system remains usably interactive and responsive at a modest eight Hertz – so it's possible to move buildings around the workspace and immediately view the effects on wind flow.

Even with the costly wind simulation, the whole thing remains real-time.

Site Views

The most recently added functionality provides a mechanism for 'previewing' a configuration of buildings from various points of view. Since the model buildings' three-dimensional forms are already resident in the system (necessary for the calculation of shadows), it is a simple matter to render them in perspective and with simple shading parameters. A camera object is provided for this purpose; driving this camera about the workspace results in the updating of a real-time rendering of the current arrangement of buildings in the site, as viewed from pedestrian height and the position and orientation of the camera.

Live snapshots of what we've designed.

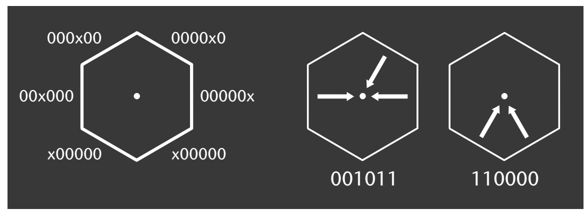
4.2.2 Wind Simulation Method

We employ a particular species of cellular automaton called a 'lattice gas' to efficiently simulate pedestrian-

An economical airflow simulation scheme.

level airflow through *Urp*'s workspace. The lattice gas computation [4] involves a grid of hexagonal cells, each of which can support up to six gas 'particles' – one for each face. The state of each hex-cell is represented at every instant as a modest six bits: if a bit is on it implies

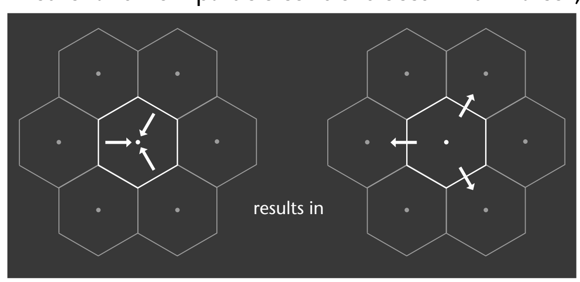
A lattice gas is made of hexagonal cells, each of which can contain up to six particles. Here's how they're coded, with two examples.



the presence of an incoming particle, understood as travelling toward the center of the cell through that bit's corresponding side. At each timestep, every cell is 'reacted' according to a small set of rules that determine whether and how particle collisions occur within a cell;

Simple rules give rise to complex and accurate fluid behavior.

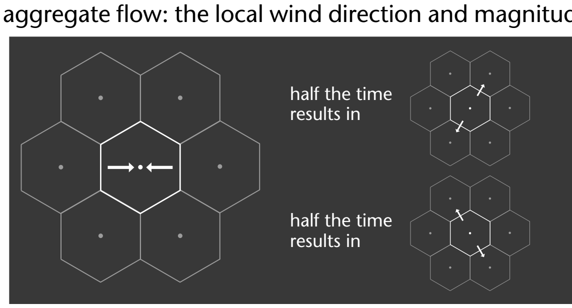
Only two interactions are recognized. One is a three-way collision, which just reverses the incoming particles.



the rules are arranged to preserve momentum. After reaction, the redirected particles from each cell undergo transport to the boundaries of the six surrounding cells, and the cycle then repeats.

We use a 100x100 grid of lattice gas cells to simulate windflow in the workspace. The motions from contiguous 4x4 sub-blocks of cells are averaged to find an aggregate flow: the local wind direction and magnitude.

The other is a head-on collision, which 'scatters' its particles into one of two equally likely outgoing configurations.



Any other distribution of particles passes through the cell: no collision.

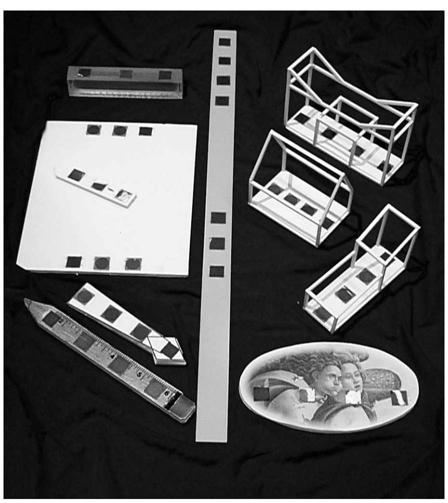
Obstacles – i.e. the bases of buildings – are represented by 'filling in' the appropriate cells, disallowing them from containing particles and causing incident particles to bounce directly back from their boundaries. Meanwhile, because such a small grid displays preferential anisotropy along its three major axes, it's not possible to represent arbitrary flow directions accurately. Instead, the grid is held fixed, with particles injected predominantly from the right side to flow leftward, while the world (i.e. building footprints) is rotated opposite the intended wind direction and analyzed into the grid. The resulting simulation is then rotated back once more, so that the airflow is moving in the originally specified direction, and projected down into alignment with *Urp*'s objects.

Our air-grid calculator prefers horizontal flow, so we always present the world to it accordingly; afterwards we counterrotate the results back into reality.

4.2.3 Objects

Irrespective of the range of functions attached to them (understanding of which is a topic in a later chapter), the *forms* of the various physical elements employed in *Urp* represent a small design exploration. The architectural models, of course, cannot be other than they are, inasmuch as the system is predicated on the core notion of attaching variegated graphical information to pre-existing models. The road-object, too, must correspond at least in its dimensions to the simulation that will be overlaid on it.

For the remainder of the objects, however, no partic-



Objects that are literal representations have self-imposed forms.

But is there a prescribed shape for other tools?

ular form is *a priori* demanded. Some, like the wind-tool and the distance-measuring-object, attempt to denote their function through suggestive pictorial elements. Others, including the clock-, anemometer-, and material-transformation-objects, are abstract in form and hint only vaguely at their intended use: operators are assumed to have 'learned' the association between these forms and the functions they represent. So no one specific design methodology has yet been chosen.

Function can be suggested through shape, through pictorial associations, or through amalgams of these.

As we build more and more *I/O Bulb* applications, and as the accessible complexity of each increases, objects will unavoidably multiply. Without yet addressing the problems of a resultant overpopulation, we acknowledge that the general issue of object form is an important one. Should there be a standardization of form (at least where the semantic demands of a particular application leaves some freedom), so that a recognized vocabulary of object appearances can be exploited? Or should an application designer be free to assemble arbitrary forms, with the understanding that end users of a system are necessarily semi-expert and may thus be expected to learn its individual 'language'?

An eventual standard for object-design would allow rapid understanding of new tools.

4.3 Discussion & Experience

Reaction to *Urp* has been overwhelmingly positive. From one point of view, an easy claim might be that all of *Urp*'s interactions are adequately presaged by *Illuminating Light*; that, having already worked with *Illuminating Light,* no-one should be particularly surprised by *Urp*. But the way *Urp* addresses its domain is fundamentally more complex – particularly as regards the use and meaning of its component object-tools – than *Illuminating Light*'s approach. We'll see why in Chapter Seven.

4.3.1 Professionals

Close to two dozen architects and urban planners (both practicing and academic) have either watched demonstrations of or directly experimented with *Urp*. Their overall impressions have been uniformly favorable; critically, most of these professional visitors said that they would voluntarily begin using a system like *Urp* immediately if it were available. Academicians cited its likely usefulness in teaching and 'quick turnaround' student prototyping. The practicing architects mentioned that not only would the system aid in their own personal design efforts, but that it could be invaluable for client presentations (in which three-dimensional models are, at the moment, necessarily separate from animations and simulations of light & shadow, windflow, etc.). Further, several younger subjects stated that such an application would help them in communicating ideas to seasoned,

Architects want *Urp* as soon as possible.

older practitioners within their firm (especially founders!) who have otherwise resisted attempts to 'computerize' aspects of their art.

Reassuringly, several made the unprompted assertion that *Urp* "hits all the major points of urban planning". Several commented that it was significant to find so many major functions collected into a single application, and all responded excitedly to the use of the architectural models themselves as the system's principal 'interface'. One insider was particularly delighted at seeing wireframe architectural models cast solid shadows, while insisting "and yet it doesn't bother me at all – the shadows are entirely believable".

Professionals believe *Urp* to be comprehensive.

4.3.2 Others

Perhaps as many as two hundred visitors with no special expertise in the urban planning field have also observed or directly operated *Urp*. The application strongly engaged nearly all of them – more reliably than did *Illuminating Light* – possibly because of the easy and universal familiarity of its domain (and its simulated physical phenomena). Several expressed interest in seeing an expanded functionality that would encompass not just the effects of interest to urban planners but also other distinctly nonphysical processes that could also be simulated and attached to the geometric distribution of structures in *Urp*. In particular, questions arose about economic simulations (what's the effect if the bank or the post office is twice as far away, or is turned wrong-way-round so that the door is on the other side?) and production flow simulations (can we increase efficiency by building a second warehouse and interposing it between the manufacturing plant and the shipping building?).

Nonprofessionals immediately envision their own fields' concerns treated by an extended *Urp*.

4.3.3 Known Problems

A small shortcoming of our object-mediated interaction style becomes apparent through the use of *Urp*'s site-view camera. Because an object with physical extent (i.e. the camera object) must be employed to designate the desired position and orientation of the view to be rendered, it's simply not possible to get immediately next to an existing structure. That is, if we want to see a rendering of an architectural structure in some proposed location as viewed from, say, the doorway of another building, we'd need to place the camera object closer to the building object than the physical extents of both

The camera's too big to go everywhere.

together will allow. In the real world, of course, this is no problem at all because of the vastly different scales of a building and a camera. Inside our simulation world, however, all objects and tools must be represented at essentially the same scale.

Sometimes it's nice to have all objects at the same scale; but it can also be a liability.

So the same properties of physical objects that are advantageous in some circumstances (e.g. three-dimensional collision detection is computationally expensive, but the impossibility of interpenetrating *Urp*'s architectural models is a convenient constraint that automatically mirrors the desired impossibility in the real situation being simulated) can simultaneously be detrimental in other circumstances (our inability to position the *Urp*-camera 'in the doorway' of a building, when that would present no difficulty at all for a real camera).

We eventually need a more appropriate fluid-flow simulation.

The lattice gas used to simulate airflow in *Urp* is admittedly inappropriate in several ways. Most important is that we use a two-dimensional structure to approximate what is ultimately a three-dimensional phenomenon – our patterns are somewhat inaccurate because *Urp* air cannot flow *up*. The scale of the simulation is incorrect as well: with the grid dimensions we are constrained to (in the interests of real-time operation), what is being simulated is closer to the micron domain than the meter domain. This scale mismatch then has implications for resulting fluid properties, including viscosity and Reynolds number.

But we've illustrated how we can always integrate a new simulation into the space.

Nevertheless, we feel that the point that simulations can be incorporated directly into the physical environment is well served by our approximating lattice gas; as we come up with better techniques it is a simple matter to substitute them.

4.4 Ongoing Work

More detailed functionality to make *Urp* 'real': zoning, seasons, terrain, etc.

Based largely on the comments of professional architects and urban planners, it seems worth considering a significant expansion on each of the individual functions in *Urp*, by way of bringing the application nearer to 'actual usability'. A multitude of such enhancements are immediately evident: examples include built-in zoning knowledge, so that automatic warnings are generated when proximity or other positional violations occur; additional clock-like controls for specifying latitude and season; a light-and-shadow integration tool that would cause the cumulative light incident over a year's time to be calculated and displayed within the workspace, as an aid to landscape architects; and the incorporation of topographic information, so that non-planar sites can be accurately treated and displayed.

To this latter end it will be important to introduce a facility for projecting 'absent' components into the workspace: buildings that are part of the site but for which no model is available, or whose positions cannot be changed by the planner. These elements would of course still cast shadows and exhibit the various forms of interaction enjoyed by the physically present models.

All buildings need not be physically present as models.

4.4.1 Remote Collaboration

These projection-only components will also represent real models manipulated at a remote location by colleagues with whom the urban planner is collaborating. Construction is now under way of a new, experimental design space in MIT's Architecture Department, and a second Urp workbench will be installed there. Extensions to the *voodoo* toolkit (already implemented and discussed in Chapter Six) will allow planners at the two installations to collaborate directly: objects manipulated at each location will be projectively represented at the other.

Some buildings may be 'ghosts' of models present at a remote collaboration *Urp* site.

What's not yet designed or implemented – and this is a centrally important issue for such remote collaboration tasks – is the protocol for manipulating objects that are locally present and objects that aren't. When F. Gehry places an upright parallelepiped on his *Urp2* table in Los Angeles, L. Woods immediately sees a representation of it (and its shadow, &c.) on his table in Manhattan. But Woods believes that the new World Dental Headquarters should be placed fifty yards to the south. Is he allowed to move the building? Clearly (for now, at least) he cannot move the physical model located a continent-width away. On the other hand, a great deal of the 'collaboration' would surely be lost if he cannot or may not influence the position at all.

But how do you manipulate a building that isn't there?

Perhaps *Urp2* supplies Woods with a tool for temporarily 'moving' virtual objects, so that he can show Gehry exactly where he wants it placed: Gehry sees a representation of the building break away from the actual model on his table and drift to Woods's intended position, but the change is only momentary; as soon as Woods releases the virtual building on his end it is rubber-banded back to its 'real' position (the one defined by the real model in Los Angeles). Of course, Gehry retains ulti-

What if you had a 'bungee' tool that let you pull virtual buildings around (but only temporarily)?

mate control over the location – access to the physical model makes him the building's 'owner' – and may or may not concede the point, simply by shifting the model to coincide with Woods's momentarily moved version.

Many more complex issues arise when the workspace is half local & half remote.

This is of course a single example of the complex issues that arise when the multiple people have simultaneous access to the same information multiply presented; it's also a fairly simple such example. Beginning to sketch the full design space for interactions of this kind will be a critical next step.

4.4.2 Constructive Activities

Could we design buildings in addition to merely moving them around?

Urp as it is construed currently is a tool for broad analysis and for creative experimentation at a particularly coarse scale (we move and arrange entire buildings). An appealing extension to Urp would permit architectural work as well as urban design work: we'd like to be able to select from a group of canonical building blocks and arrange these into new buildings.

We at least need to be able to stack blocks.

The key additional technological trick required here involves the third dimension. At the moment, we cannot stack objects to build upward. There are at least two approaches to remedying this deficit that suggest themselves, however.

One way to do that without too much work involves transparent blocks with interleaving *voodoo* patterns.

One requires little or no modification to the existing I/O Bulb underpinnings (i.e. *glimpser* and *voodoo*) and is in fact already under development. This scheme uses transparent geometric building blocks that have been designed to interlock bottom to top, so that the way they stack is constrained. Each building block has a unique *voodoo* pattern, even though the building blocks are not unique, carefully engineered to 'interleave' with its siblings (since all surfaces are transparent, lower patterns show through higher blocks). So if a block labeled 'R-B-B' is placed atop another labeled 'R-G-G-G', the application can either use the apparent spatial coincidence of these two *voodoo*-patterns to conclude that one is above the other or can 'find' a separate pattern defined to represent the composite stack: 'R-R-G-B-G-B-G'.

But you can't go up very high that way.

The drawbacks to this scheme are two, and both limit the extent of stacking that can be supported. The first is the clear difficulty of designing a large lexicon of mutually interlocking *voodoo* patterns. Given that each colored dot has finite extent, given that these must be spaced sufficiently far apart to allow the dovetailing of other such patterns, and given that the whole (dots spaced far apart) must fit on a block of moderate size, it is easy to see that we could never hope to be able to stack more than two or three blocks. The second drawback regards a more fundamental limitation of *glimpser* itself. As a stack of blocks grows Babel-like heavenward, the *voodoo* dots toward the top of the pile grow quite large in the I/O Bulb's camera view (an inescapable effect of perspective geometries). *glimpser*, however, implicitly assumes roughly planar activities, so that all dots appear to be of the same size. The effect of larger dots is to increase the uncertainty of location (since glimpser is free to find the fixed-size region it's looking for anywhere within the now larger patch of constant color), until eventually – when a dot's image becomes more than twice as large as its expected size – several dots will be 'found' for each that is present. At this point the reliability of the technique breaks down altogether.

An alternate approach would require relinquishing the *glimpser*/*voodoo* foundation in favor of something a bit more specialized: if the construction set consists of a finite number of preordained forms, each painted black with a regular grid of white laid over all surfaces, then it is a tractable problem to build a vision system that can both identify each object as well as its orientation and proximity to the camera. Note that such a system would have to perform SHRDLU-like deductions about stacking, based on the cumulative historical state of the work surface. That is, *SHRDLUrp* must keep track of what's already on the table, since any vertically-stacked block will necessarily occlude the lower object that supports it. This is in contrast to most of the glimpser/voodoo-based applications built so far, which endeavor to work as statelessly as possible – so that the complete disposition of a workbench is extractable at any given time without having to know anything of its past state.

We need a new vision system that identifies gridded blocks directly.

That requires keeping track of history, so that we're not confused by occlusion.

4.4.3 Topography

One of the most frequently encountered questions about Urp is the possibility of its treating urban planning scenarios that do not occur on flat, planar sites. As long as the topography in question is not severely hilly (so that objects at higher altitudes begin to encounter the dot-size-distortion problems discussed above), this is not a particularly difficult extension beyond what already exists. A solution might choose to employ a scale model

Non-flat urban design sites: either work directly on a model of the terrain or let the system project topographic contours.

of the terrain in question or instead to project iso-height contours onto the flat workspace surface (in the style of topographic maps). In either case, however, the system would have to be apprised of this terrain geometry in order to properly calculate shadow and reflection effects.

5 Gallimaufry: First & Small Works

5.1 Early Experiments

Before the *Illuminating Light* application had solidified the direction and viability of I/O-Bulb-mediated maneuvers, an early configuration demonstrated the augmentation of an entire single wall in a small office. The application that filled this space was not yet conceptually strong or cohesive enough to fully vindicate the assertions we wanted to be able to make, but it did illustrate several kinds of luminous-tangible manipulations.

An early attempt, not yet mature.

The first of these involved the notion of digital storage in a physical container. Documents – images, text, and live video – could be created and then dragged around the wall-workspace literally manually, using a colored paddle held by the human operator. These documents could be brought into loose association with a large vase simply by being placed in its vicinity; a graphical spring would emerge from the vase and attach itself to the document to indicate the association. Once some number of documents had thereby been attached to it, the vase could be spun around its vertical axis to bring them fully inside. The vase was then free to be moved about the space, its cargo safe within it (as we know from the real world, containment must continue irrespective of container-position). At any later time a second twist of the vase would cause it to explosively disgorge its contents, which would then be visually manifest and available once more for paddle-manipulation.

Digital storage in physical containers.

A simple gesture for insertion and extraction.

Move the container and the stuff is still inside.

Vertical chess was a second capacity of this same early I/O-Bulb-augmented space. At any time (i.e., even during the prenominate vase maneuverings) an outsize chessboard could be brought into the space; this signalled the system that it was time to play the world's arguably most boring game. Animated chesspieces

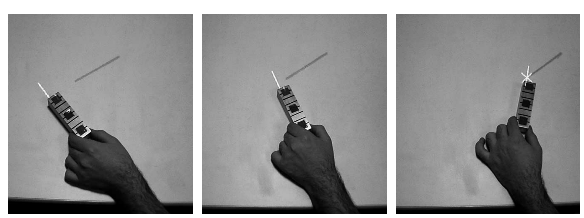
A chess partner anywhere, just by pulling out the board.

would immediately emerge from various points along the wainscotting and jump into position on the board. The board could be moved arbitrarily around the space, the sprightly pieces always hopping about to regain their appropriate places on it. Though never ultimately taken so far, the intended completion of the scenario would have provided the human competitor with physical chesspieces to be placed on the board. Without explicitly distinguishing one species from another, the system would have been able to proceed from the known initial configuration of pieces to understand the distribution of tokens on the board at every moment, so that a back-end chess algorithm could provide Shallow Blue's responses to the human's moves.

5.2 Scraper

Generic tools: tweezers, scrapers, wipers.

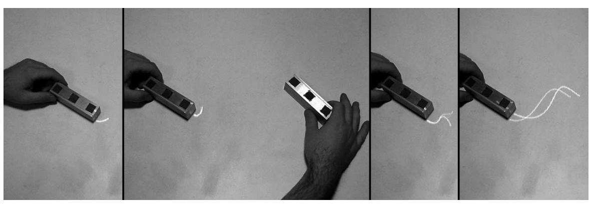
We have begun to experiment with a collection of reusable luminous-tangible tools that are the digital analogs of scrapers, spatulas, tweezers, wipers, and so on. The need for such implements arises because manipulations will eventually require finer control (than do our current applications) over what and how information is associated with various physical artifacts. To wit, we might imagine a need to remove – to pry loose, essentially – one of several distinct pieces of digital information that are attached to a single real-world object.



Using a spatula to sweep an obstacle to the side.

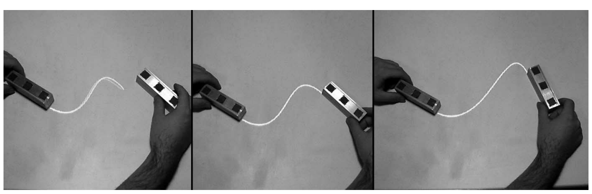
These generic tools consist typically of a physical handle out of which sprouts one form or another of projectively-bonded filament – some rigid, some elastic, some straight, some curved; in the current hypothetical case, we'd likely insert the luminous business end of a simple linear scraper between the physical object in question and the digital element that needed to be dislodged, and then sweep this latter away.

Sometimes, details of execution turn one kind of tool into another. A corralling instrument with two physically separate handles between which was unwound an arced



A spline path snakes outward from one handle...

digital filament – intended for broad collecting tasks in the same way that two curvingly joined hands can



... to connect to another; the spline can then be interactively reshaped.

sweep crumbs from a countertop – was eventually observed to be a good way to interactively specify splines. Indeed, the tool's luminous 'jump-rope' had been implemented as a Bezier curve. This has since led to an investigation of luminous-tangible tools for designing curves and surfaces.

Digital curve design using real handles and tools?

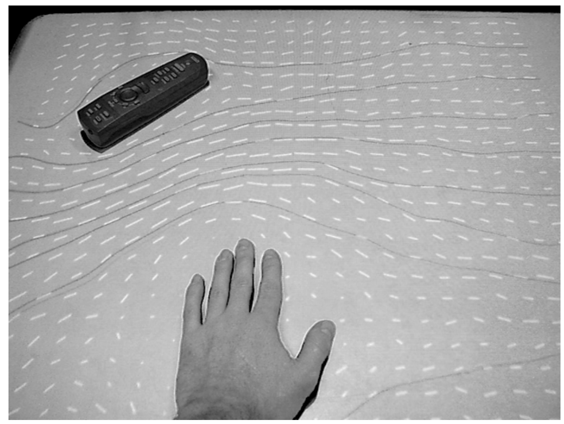
5.3 Standalone Fluid-Flow

The first I/O Bulb application to be built without the use of *glimpser* and *voodoo* is a simple fluid dynamics workbench called *seep*. The same lattice-gas simulation deployed in *Urp* runs here, but instead of taking as input the position and orientation of structures known in advance (i.e. *Urp*'s various architectural forms), *seep* allows arbitrary objects to be placed in the flow path. The shapes of these objects are extracted from the visual field captured by the I/O Bulb using rudimentary frame-differencing techniques; these silhouette shapes then serve as obstacles appropriately positioned within the

A generalized physics simulation environment: suddenly it's not necessary to know objects in advance.

flow simulation's boundary

The result is a real-time simulation in which fluid appears to flow from right to left across a table surface; any object (non-inanimates like hands are also valid) placed on the table rapidly diverts the flow, which for



Real-time simulated fluid flow around arbitrary obstacles.

example exhibits increased field velocities in certain places – as one would expect – in order to maintain the overall right-to-left flux. Moving the obstacle-objects produces beautiful and fluid-dynamically satisfying transient effects, including slipstreams, eddies, sloshing, and all manner of swirls. Although *seep* is in no sense a complete application – there's no facility for extracting quantitative measurements, or for affecting the simulation parameters, for example – it is a promising new kind of tool for providing intuition for complex physical phenomena and their interaction with real-world objects and forms.

6 Luminous Room

With a single prototype I/O Bulb and a brace of well-received proof-applications in hand, we felt it possible to move toward our original goal: a system architecture for a real Luminous Room. As suggested early in these pages, the way to do that would be to distribute enough I/O Bulbs throughout a room so that every surface could be addressed. Between a *single* I/O Bulb and *many* I/O Bulbs the first and most important step is *two*; below is the account of those two.

If we can get many I/O Bulbs in different parts of the same space cooperating, then we'll have built a Luminous Room.

6.1 Fundamental Issues

The central technical challenge of building a Luminous Room – as distinct from a single I/O Bulb – is synchronization.

We need to consider spatial synchrony: where I/O Bulbs' regions of influence abut and overlap a bit, work is required to align the projective and metric spaces along the seam. How can an atomic chunk of projected information that needs to cross two bulbs' zones be treated? What problems of geometry do we face when a new I/O Bulb is added to a space in which others are already resident or (as we describe later) when some I/O Bulbs are themselves mobile? Can we devise automatic methods for calibrating each bulb to the others and to the surrounding architecture?

If two I/O Bulb's zones overlap, how do we make sure they line up?

How do we handle objects and projections that straddle zones?

There's temporal synchrony as well. What would be the networking and delay issues in using the Luminous Room to implement a pervasive *ClearBoard*-like system [7] for interpersonal communication? Is it reasonable or inadvisable for an object of which there are two physical copies to exist (as far as the Luminous Room is concerned) in two locations simultaneously?

One object, two places at the same time?

Finally, computational synchrony must be addressed. Should a single, centralized computational stream simultaneously serve all the I/O Bulbs in a Luminous Room space, so that an object that straddles the reach of two separate bulbs is first cooperatively 'sensed' by both, then singly 'conceived of' by the central process, and finally augmented through pieced-together projections? Or does each I/O Bulb rate its own independent brain, so that the handling of straddle-objects requires more explicit process-to-process communication?

Luminous Room: one central master brain, or many cooperating ant-like brains?

As we might expect, the full extent of such basic, pervading issues is outside the sweep of this dissertation work, but a modicum of preliminary effort has begin to address them.

6.2 Software Extensions

Binding together more than one I/O Bulb.

We'd built a second prototype I/O Bulb; we needed a layer of software at some level or other that would make them mutually aware. The place at which it seemed most advantageous to do this – for the particular explorations we had in mind, at least – was at the *voodoo* stage of processing.

Extensions to *voodoo* are the best way to achieve this, for now.

voodoo in its 'solitary' guise was already responsible for reporting to its master application the identities and locations of all objects seen by its I/O Bulb. Now it would make the same report to every other I/O Bulb system as well. Thus *dee-voodoo* ('distributed *voodoo*') is a set of extensions to the existing software – entirely transparent to the higher-level implementer – whose first task is to connect over a dedicated TCP/IP port to all other *dee-voodoo* processes on the nearby network. Each of the various resulting links is then used to effect a bidirectional transfer of geometry information: the initiating *dee-voodoo* describes its I/O Bulb's position, orientation, and associated surface dimensions, all with respect to some globally acknowledged reference, and receives in return the distant I/O Bulb's complementary particulars.

One *voodoo* per I/O Bulb; each *voodoo* talks to every other, so that each one believes that it sees every object.

Following that preliminary exchange, every object recognized by the *dee-voodoo* process serving I/O Bulb **A** is reported not only to application **A**, but is also relayed to I/O Bulb **B**'s (and **C**'s and **D**'s and ...) *dee-voodoo* process, which in turn reports the object to application **B** *as if the object had been seen by I/O Bulb **B***. (This entails a small preparatory step in which **A** first transforms the reported object's geometric description into its **B**'s local coordinate system.) The object-exchange is also of course reciprocated from **B** to **A**, and so on, in an ongoing relay mesh with n^2–n links. Although we have at present only two I/O Bulbs, we have tested use of *dee-voodoo* to synchronize up to five independent application processes, using 'manual' mouse-and-keyboard manipulation of objects for the three bulb-less systems. Even with a consequent twenty point-to-point links in simultaneous operation, the participating systems evinced negligible lag between the movements of local objects and those of distant objects.

6.3 Distributed Illuminating Light

The first inhabitant of our preliminary Luminous

Room was (fittingly) an elaboration of the first complete I/O Bulb application: Illuminating Light, our trusty optics simulation environment, now works at more than one location at the same time. With almost no change to the applications itself – simply by linking with the *dee-voo-doo* library instead of the *voodoo* library – starting the Illuminating Light system automatically seeks out other running versions of the same system, announces itself to them, listens for a description of the real-world geometric transformation by which it differs from each, and begins to receive object reports from the other versions' I/O Bulbs.

With this basic structure prepared it is possible to run the system in two different modes. If each I/O Bulb properly identifies its position and orientation within the room, then the resulting aggregate of separate Illuminating Light applications acts to provide 'windows' of interaction on a continuous space. Thus, a laser aimed off the edge of one of the tables will reappear on any

Physical 'windows' on a continuous optics space.

other table that lies in its path, so that conceptually the entire room is a vast, continuous physical optics space: the beam of faux-laser light appears able to pass even through regions of the room that lack an I/O Bulb. Of course, the most natural configuration of a limited number of I/O Bulbs would likely be to build a single larger contiguous workbench surface (rather than the isolated 'islands' of optical simulation that we've described). Still, there's no essential distinction between the two cases.

If, on the other hand, every I/O Bulb is configured to claim that it occupies the same position and orientation in the room's global coordinate system (they all believe that they're at the origin, for example) then the result is a collaborative setup: every table displays a synchronized version of the same space. A mirror placed on one table will appear (virtually) on every other, and will have an identical effect within each simulated optical environment; whether it's physically present or not is irrelevant. Of course, those objects introduced into one I/O Bulb's environment from another site must now be graphically represented, and for this reason the application must be able to distinguish between local and remote objects. But this simple variation on the dee-voodoo theme (the modification is a no more than a few words in a configuration file describing the I/O Bulb's location and orientation) has laid the foundation for a very different use of the Luminous Room architecture than originally imagined: CSCW (Computer Supported Collaborative Work).

A collaborative workspace that presents the same scenario many places at once.

Fabulously, several of the system's earliest operators found themselves trying to avoid standing immediately between the tables, clearly in an attempt not to block the beam's propagation. It would then take them a moment (brief, but *still*) of conceptual effort to conclude that of course this concern was unnecessary. Even then, this same false apprehension recurred for some. The anecdote highlights not only the strong perceptual effect of collocated visual input and output – display *in* the real-world leads to expectations *of* the real-world – but also certain puzzling design issues: maybe it *should* be possible for an interposed human to block the beam's transmission from table to table. That'd be physically accurate, in some sense, and wouldn't oppose natural intuitions. But in other ways it's convenient for human physicality to be ignored; after all, this faux laboratory is already intended to ameliorate some of the oni associated with 'real' optical design, and so – unless the point is to remind engineers not to carbonize their limbs in forty-Watt beams – there may be little point in simulating this particular phenomenon.

The experts, fooled.

We must design to only selectively reproduce real-world phenomena and conditions – sometimes it's more useful to be non-natural.

6.4 Discussion

An implication of using *dee-voodoo* to realize a simple Luminous Room architecture is that every individual I/O Bulb is rendered effectively omniscient: each one seems to observe all objects everywhere. This in turn implies a particular set of answers to the fundamental questions posed at the start of this chapter.

Spatially: we have an aggregate system in which geometric registration is smoothly handled. So long as the participating I/O Bulbs have accurately reported their real-world disposition at the outset of intercommunication – and in fact it is entirely permissible for an I/O Bulb to subsequently broadcast revised geometry-information if for some reason it's moved – then real-world alignment of graphical elements is assured. (This point is naturally moot when the various I/O Bulbs each use an identical offset from the global coordinate origin to provide conceptually collocated workspaces: e.g., the remote-collaboration version of *Illuminating Light*.)

Distributed *voodoo* permits smoothly continuous spaces from more than one I/O Bulb.

Temporally: we have shown, as already described, a system that exhibits very small lags. For a 'same-space' interpretation of the Luminous Room idea (inevitably relying on a local network) we may expect that any application with moderate frame-to-frame object descriptions will enjoy similarly brisk performance. However, a wider reading of the Luminous Room notion – one in which, say, distant spaces are connected for the purposes of collaboration – is more complex. Given today's global network infrastructure, disparate sites would experience lags anywhere between ten milliseconds and several seconds; and there is not necessarily evidence to suggest that this will change for many years to come.

Lags are small for communication between I/O Bulbs in the same space. What about intercontinental collaborations?

One acceptable way of addressing this, we believe, is to establish conventions that allow an application to directly acknowledge the lag. If representations of remote object can be subtly labeled with some indication (either quanti- or qualitative) of the delay that they suffer, then human participants can adjust their expectations accordingly. Such a measure is not a lazy dodge, though the accusation is easy. Consider, by analogy, an international phone call whose substantial delay (50 – 200 ms) disrupts the conventions of normal verbal communication: conversants who are not aware of the lag are liable to interpret the resulting ungainly overstep-

Sometimes alerting people to the existence of a lag is enough to make it bearable.

pings and pauses as rude, angry, or hesitant; but callers who acknowledge the lag can quickly adapt their conversational strategies to accommodate it.

One processor per I/O Bulb – deliberate isolation.

Computationally: on the one hand, the implementation of *dee-voodoo* does not distinguish between the case of two processes running on separate CPUs and the case of both running on a single multitasking CPU. (Our choice of the separate-CPUs scenario has largely to do with practicalities: the machines we used were adequate for a single complete system's execution but would have been overtaxed by two; further, each machine provided only one video input and one video output.) On the other hand, a fairly severe I/O Bulb solipsism is implicit in our current Luminous Room formulation. To wit, communication among neighboring I/O Bulbs is sparing, and the overall architecture of applications is somewhat antithetical to a model in which (for example) a single application process receives contributions from a roomful of I/O Bulbs, applies machine vision analysis to a vast composite optical field, performs its simulation task, and finally generates a 'roomful' of graphical output to be automatically segmented and sent back to member I/O Bulbs for display.

7 Discussion & Big Picture

7.1 Context

The ideas defining this work are hardly without precedent; some of the pieces have been seen before in other forms.

7.1.1 Broad

As early as the late 1970s, MIT was demonstrating the idea of wall-sized-computer-projection-as-architec-

ArcMac: info wall that listens and watches.

ture [1]. Within the space called the 'Media Room', members of the Architecture Machine Group had built applications like *World of Windows* and *Put That There*, allowing human inhabitants to interact with the information displayed on the far wall through a synthesis of methods as various as physical gesture, voice recognition, and eye tracking.

Anyone who employs video projection as part of some interactive system would be ill-advised not to mention the work of Myron Krueger, beginning in the 1960s

Krueger: a video shadow is your digital marionette.

and continuing through the present. Especially via the ever-revised face of his *VideoPlace* installation [10], Krueger was an early pioneer in the use of video as simultaneous input and output.

7.1.2 Inspirational

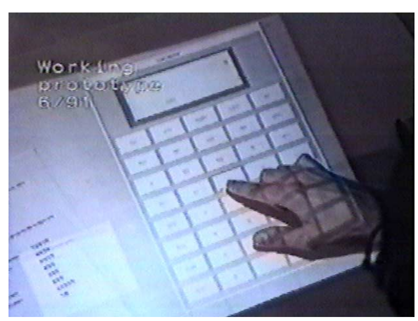
One of the most gorgeous applications of projective schemes is Michael Naimark's *Displacements* (1984). In this installation at San Francisco's Exploratorium Naimark had constructed a faux livingroom environment, replete with chairs, books, potted plants, lamps, occasional fixtures, and so on. A number of actors moving about the space and manipulating its contents were filmed by a special camera centrally located and rotating slowly to look progressively around the room. The camera was

Naimark: a dead room with life projected back into it.

then removed and the film developed; meanwhile, Naimark and his assistants painted the entire room – including its formerly mobile contents – a uniform matte white. The final exhibit consisted of the 'erased' room with a rotating projector in the exact position of the original camera, now showing the developed film. As visitors watched, a finite window of color and movement swept around and around the sterile environment, briefly reanimating each part of it.

Closest in spirit to our present work is the *Digital Desk* project of Pierre Wellner, then working at Xerox's EuroParc [21]. Wellner's system acted in effect to migrate the

Wellner: the computer desktop, pried loose from the screen.



computer desktop off the familiar monitor and back once more onto the physical desktop. This was accomplished with an overhead video projector pointed down onto a desk surface; a video camera also gazed at the desk and provided the system with much of its input. The intent was a seamless mingling of digital and physical documents so that (for example) a user could employ two fingers to indicate some region of a text-filled paper atop the desk to be visually 'read' by the system, OCRed, and then inserted back into a purely digital text composition displayed projectively on the same surface.

7.1.3 Immediately Environmental

The work of Professor Hiroshi Ishii's Tangible Media Group [8] has provided the present work with a critical component missing from its earlier conceptions, and that is of course the tangible member of the luminous-tangible symbiosis. The early application (the chess and vase/container system) remained an only modest success in part because its too-sparse physical implements were unwieldy, literally as well as conceptually. The majority of TMG's projects (see, e.g., [16]) have worked to show how well-designed physical implements, gracefully deployed in an ordinarily digital realm, can provide a very real advantage in the handling of certain kinds of tasks.

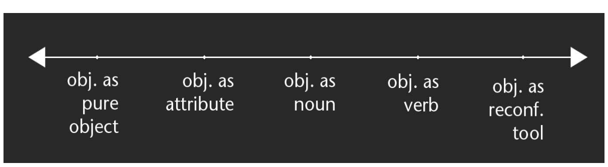
Ishii: many things work better if there's something to hold onto [8].

7.2 Luminous-Tangible Issues

7.2.1 Object Meanings Axis

When you work with physical objects and aligned light, the interaction is *luminous-tangible*.

What are the different ways in which a luminous-tangible system can understand or make use of an object?



Every object is here somewhere.

We offer a design space that arrays all possible interpretations along an axis that moves away, in both directions, from a center representing a maximally 'real-world' object reading.

Note that these classifications are intended to apply only to objects considered in the context of a luminous-tangible system – we are not attempting a generic scheme appropriate for arbitrary TUIs (tangible user interfaces) [8]. Moreover, we are not proposing a grammar (as does Ullmer in [17]) for the prescription or analysis of TUI-based object-to-object interactions; the

A classification scheme for luminous-tangible systems only.

Object Meanings axis classifies *individual* objects. It should be understood too that the words 'noun' and 'verb' are used merely as a convenient way to suggest certain properties, and not in any attempt to imply a full mapping between luminous-tangible objects and linguistic parts of speech.

Object As Noun

These objects occupy the center of the axis and are likely the most obvious in their behavior. They are fully literal, in the sense that they work in their luminous-tangible context very much the way objects 'operate' in the real world – an Object As Noun exists in our applications simply as a representation of itself: an immutable thing, a stand-in for some extant or imaginable part of the real-world. All the objects in the *Illuminating Light* application are of this type – each of the optics models is meant to be understood (in function) as its real-world counterpart. The buildings and roads in *Urp* are also of this variety.

The most literal kind of object: a little stand-in for a real-world counterpart.

Object As Verb

As we move to the right along the continuum, away from Object As Noun, inherent object meaning is progressively abstracted in favor of further – and more general – functionality. The material-changing wand in *Urp*, for example, is an Object As Verb. It is not understood as 'present' in the world of *Urp*'s simulation, but exists to act on other components that are, or on the environment as a whole. The clock and wind objects do just this, in affecting ambient conditions like time, solar angle, and wind direction. However, both these tools exist somewhere in the continuum between Object As Noun and Object As Verb, inasmuch as they are each in some sense a metonymic proxy for objects that do conceptually occupy the simulation's world – here, the sun and the aggregate phenomenon of 'wind'.

A tool that does one kind of thing to other objects.

Object As Reconfigurable Tool

This variety of object-function is fully abstracted away from 'objecthood' in a way perhaps loosely analogous to a GUI's mouse-plus-pointer. The paddle in the chess-and-bottle is of this sort, but where a WIMP-style interface typically uses a series of menus to change the function of the mouse, the paddle depends for these alterations of meaning on context and state. Since that single early use of this kind of object, however, we have temporarily avoided its further deployment: to simply transplant some variation on the mouse-and-menu idea into our applications is too easy, and flies in the face of

A generic tool, the way that the mouse pointer on a CRT can mean more than one thing.

the basic tenets of building luminous-tangible systems in the first place. We do believe that there exists a proper (non-menu) method for introducing such reconfigurable objects into the world of the *I/O Bulb* – and this solution will soon be required to combat the inevitable proliferation of objects that results from constructing ever more complex applications.

Object As Attribute

As we move to the left away from the center position on the axis, the object is stripped of all but one of its properties, and it is this single remaining attribute that is alone considered by the system. The arbitrary objects that act as flow obstacles in the *seep* application are one example: here, nothing matters but the *shape* of what's placed in the workspace; all other attributes of the individual objects used are ignored. Other systems might consider (for some purpose or other) only the color of an object, or the object's size, or its velocity.

A simplified view of an object, where one specific property is all that we care about.

Object As Pure Object

This last category is the most extreme, and represents the final step in the process of stripping an object of more and more of its intrinsic meanings. In this case, all that matters to a luminous-tangible system is that the object is knowable as *an object* (as distinct from *nothing*). It may or may not be important that the object be uniquely identifiable; to take an example in which it is, we can imagine extending the digital-storage-in-physical-bottle scenario to a full Luminous Room setting in which information can be stored in arbitrary objects, wherever we may happen to be. Thus, just as we might scribble a phone number on anything nearby – an envelope, a magazine, even a hand – the following scenario would make sense: "Where did you put the directions to the restaurant?" "Oh – they're in the scissors."

An object that could be anything, as long as it's something – universal by virtue of total abstraction.

The scissors don't matter as scissors; all that's relevant is that they exist and are distinct from other objects that might have been used instead – and that they're where the restaurant directions are.

It is at this far end of the meaning spectrum that we suddenly find that the axis is not linear, but in fact connects to itself, end-to-end: if an object has been shorn of all inherent meaning, then paradoxically it is free to be assigned an arbitrary functionality. So if we move beyond Object As Pure Object we can find ourselves suddenly back at Object As Reconfigurable Tool.

A wraparound classification space.

7.2.2 Straddle-Balance

By definition, every luminous-tangible system locates meaning and functionality simultaneously in two contrasting places: in physical objects, which are directly manipulable by human clients of the application, and in projected digital elements, which are not. It has become apparent that the way in which an application distributes its tasks between corporeal objects and noncorporeal projection – straddling the graspable/corporeal and the digital/projective – has a great deal of bearing on its ultimate behavior and form.

How much is physical, how much projected?

Illuminating Light, for example, posed little question as to which parts of the application would be projected and which would be physical; in setting out to directly parallel the way in which optics experiments are constructed and carried out in the real world, we automatically obtained an elegant balance: physical models would represent physical optics, and projected *I/O Bulb* light would represent actual laser light. So as the real-world engineering pursuit became a luminous-tangible simulation, noncorporeal remained noncorporeal and manipulable remained manipulable. In a sense, the system very conveniently dictated its own design.

For optics, the answer was a clarifying gift: optics as optics, light as light.

Urp represented a somewhat more complex design circumstance. However, the same pattern of solid-to-solid and nonmaterial-to-projective mappings emerged: light and shadow effects became aligned projective complements to the architectural models, as did the airflow simulation.

It's more complicated with more functions, many of which don't concern light directly.

It is important to note that the buildings in *Urp*, through their geometric arrangement, carry no less meaning than the more 'exciting' shadows and reflections attached to them – the placement and orientation of structures is, after all, the end goal of urban planning. That is to say: in *Urp* the disposition of physical building models itself contains information; they are not just 'input' but 'output' as well.

Objects themselves may convey information simply through their arrangement.

A very different kind of meaning distribution is demonstrated by the early Chess-&-Bottle system. Here, the scenario's objects carried little specialized meaning: the chessboard was simply an inert stage for the antics of the animated chesspieces, and the bottle – being a container – was essentially unrelated to the digital constructs that it contained. Instead, nearly all the functionality in the system had been concentrated into one physical

Fewer physical objects can imply that each is more abstract & powerful.

tool: the color paddle. This single significant instrument was used to create documents, to move them about the space, to associate them with the bottle, to trigger the bottle to absorb them, and so on. To a certain extent, the paddle acted much like the featureless but infinitely assignable mouse of a GUI.

Clearly, applications that have very few projective components and rely mostly on physical objects lean toward 'just being the real world'; while applications that tend to ignore physical objects in favor of complex or standalone graphical components (e.g. the paddle system) encroach on familiar GUI territory. But each extreme can also be appropriate, depending on the needs it addresses and the context in which it's deployed.

At the two extremes, we get either the real world (no digital) or the familiar old GUI (no physical).

Ultimately, we do not yet have a large enough body of telling luminous-tangible applications to formulate general prescriptive rules, but we can state that such straddle-balance issues will remain central to proper luminous tangible design.

7.2.3 Graphical Dynamism

Each of the five major applications discussed in this document (*Urp, Illuminating Light, Chess-&-Bottle, seep, Distributed Illuminating Light*) is marked by a constant graphical dynamism. Indeed, many opportunities have been taken to incorporate subtle motions into most graphical constructs that do not, by the nature of their content and meaning, demand stasis (shadows, for example, are obviously not free to dance around, and so in *Urp* they don't; but *Illuminating Light*'s laser beams, whose context clearly precludes lateral translation, are nonetheless represented by a dashed line that marches ever forward). We find that, as a general design principle for luminous-tangible interactions, these small visual gestures are desirable for the following reasons:

We jiggle our pixels whenever possible.

• *Apparent life*. Slight ongoing motions reassure the Luminous Room inhabitant that the application is still running. They also lend a modicum of personality to the application: a quantity not strictly necessary, but generally welcome.

This lets you know that the program hasn't crashed.

• *Disambiguation of the real and the virtual*. Early tests with largely static graphical systems showed that with fairly dense, interpenetrating collections of physical objects and digital projections, some confusions could arise regarding the status of the projections. Slight

It shows what's real and what's virtual.

motions of a sort unlikely to attend physical objects help to signal graphics' identity.

It allows the HVS to perceive a higher resolution for text and graphical constructs.

• *Increased resolution*. Because the resolutions at which our current luminous-tangible applications operate (32 dpi down to 4 dpi) are significantly lower than those commonly provided by other displays, human parsing of text is often hampered. But, as these glyphs are anti-aliased, even sub-pixel motions can dramatically increase their comprehensibility. Text aside, the apparent resolution of all projected I/O Bulb graphics is increased when these constructs are in motion.

It is aesthetically pleasing.

• *Aesthetics*. If we understand the aesthetics of an interaction to be a function of clarity and detail, then the combination of the three effects just described certainly leads to a 'pleasanter experience'. More ineffably, applications that apply subtle motions to different parts of their graphical apparatus simply look better than those whose elements are static.

8 Onward

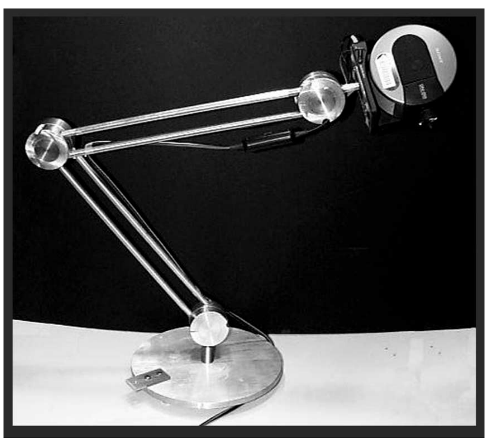
8.1 Hardware

Additional implementations of the basic I/O Bulb structure have been planned and, in some cases, implemented in prototype form.

8.1.1 Luxo-format I/O Bulb

As a complement to the fixed I/O Bulb architecture on which most of our example applications have been based we are currently also building an adjustable version. This structure, modeled loosely after the familiar Luxo lamp, is an articulated armature whose joints are instrumented with high-resolution optical encoders. The

The new I/O Bulb screwed into the old Luxo lamp.



end of the linkage supports a comparatively small Sony CPJ-200 NTSC projector and a lipstick video camera. Forward kinematic calculations based on the precisely known inter-joint distances and the continuously measured joint angles are used to determine the exact three-space position of the projector-camera pair. This knowledge is used, in turn, to apply the proper predistortions to the imagery fed to the projector.

Always know where the camera & projector are.

Conceptually we intend that this mobile I/O Bulb structure would be used in a manner analogous to the operation of the articulated desk lamp that it resembles: if we desire more *illumination* at some particular place on our desk, we simply pull the Luxo's head toward that point. Similarly, the 'I/O Luxo' can be dragged toward some location or object on which we desire to cast more *information*.

Need more info? Pull the I/O Luxo closer.

8.1.2 Real I/O Bulb

Building a real I/O Bulb.

Even as we accumulate evidence for the significant usefulness of a Luminous Room environment by building pertinent applications atop mocked-up (though entirely functional) hardware platforms, a parallel research track demands attention: the construction of an 'actual' I/O Bulb. By this we mean a monolithic device of modest size that performs collocated projection and video capture.

Toward this end we have been collaborating with a Media Lab sponsor company, which has curiously requested anonymity, on the basic technologies. A prototype currently exists, sporting an eminently reasonable profile of 6"x3"x4" (design of the next generation, half again as big, is under way). The embedded camera is an astonishingly mere 2" long with 0.4" diameter and ~5 lux sensitivity. The projector, while boasting a promising new underlying technology – not to be disclosed here – is unfortunately still so dim as to preclude actual use: it's visible at all only when room lights are extinguished and the projection surface is not much more than twelve inches distant.

Still, we expect technical innovations to keep their normal vigorous pace, so that soon enough a fully practical I/O Bulb implementation will exist. This inevitable object will send an important message: it will itself be the best possible symbol of the ideas it makes possible.

8.2 Vision

The use of a very simple vision algorithm, one which does nothing more than locate colored dots in its input field, is predicated in part on its author's comparative lack of sophistication in the vision-algorithm-design realm. But, too, the bare-bones nature of the technique – combined with the subsequent yet substantially visionless voodoo interpretation step – results in a highly streamlined object recognition system with good reliability (meaning also: predictable in its failure modes). Glimpser and voodoo, considered together as a reusable toolkit, have proved ideal for a large range of applications that take evolving object-layouts or spatial distributions as their principal input.

We could get rid of the colored dots...

This momentary satisfaction notwithstanding, there's plenty more to do in the vision domain. The voodoo object-tagging technique is convenient, cheap, and mostly reliable; but a rightful objection exists to the aesthetic compromise it demands: colored dots pasted everywhere.

One obvious replacement for such explicit optical tagging would be to use template-matching techniques to identify objects directly from their respective shapes – and of course template-matching is also hardly the only method for doing shape-based object recognition (e.g., [0]). As always, each of these approaches can be compared to the others and found to have ad- and disadvantages; we'll state simply that the investigation of all these different tacks is a bit peripheral to the main program at hand. For now we will content ourselves with the understanding that any one of the more sophisticated means is likely to be far more computationally demanding than what we've already built: we'll suffer our spectral blemishes and leave the search for variations on the vision theme to future generations.

More immediately interesting are vision systems capable of other than fixed-object recognition. For example, we intend to add to the existing vision capability a parallel analysis that identifies human hands or hand gestures (e.g. [15]). So far, the simulators we've built 'understand' the world only through the instantaneous disposition of certain simulation-specific objects ('phicons'). A system that can additionally see hands can make use of the information to increase the accuracy and reliability of its spatial understanding (e.g.: "no hand has come near the Studebaker model, so its small positional variations must be due to video noise and should be ignored"). Moreover, gestural communication becomes possible. Eventually, too, larger-scale I/O Bulb installations – those that address a substantial portion of a room – will require tracking of entire humans (cf, e.g., [0], [13]).

We can extract more meaning from visual input by interpreting it in more than one way at the same time.

8.3 Third App: Cinematic Previsualization

The third panel in our triptych of proof-applications would address the needs of the film production community, as follows:

Making movies is so expensive that you'd better plan really, really well.

Commercial cinema is an ever more expensive proposition. Budgets are increasingly bloated; the complexity of sets created and shots attempted grows yearly; and postproduction assumes a larger and larger responsibility for creating fundamental content and solving incidental problems. All this has led to a critical reliance on careful preproduction, of the various techniques of which so-called 'previsualization' methods have gained distinct popularity. Previsualization intends – whether via a tiny video camera scooted around a model of the set or via computer-graphics rendering – to show in sketch form what a completed shot will eventually look like. In addition to assisting the design of shot composition, previsualization can help with financially significant decisions: if the camera is pointed partly downward throughout the entire sequence, do we really need to pay to construct a ceiling on our set? If the lens is so distortingly wide, shouldn't we hire half the number of extras, since it's now clear that the shot won't even show the back of the crowd?

One way of planning is by pretending you've already made the movie.

Our intended previsualization system begins with a projected blueprint of a set – presumed already to have been designed, perhaps in a CAD-like digital system – that shows not only representations of architectural structures but also of the props, actors, and miscellaneous scenery that will remain stationary during the shot. To this are added small voodoo-tagged models of every actor, piece of furniture, or set dressing that will need to move during the scene. Finally, a voodoo-tagged model of the camera itself is provided: this is the system's principal tool.

We'll let the director pre-shoot the movie with a voodoo camera on a voodoo set.

Use of the application proceeds by maneuvering the camera model through the miniature set, additionally moving each actor-model, or prop-model, or etc. at the appropriate times throughout the scene's progression. Directly projected back into the set are graphical indications of two critical pieces of spatial information: first is the view cone, originating at the camera and expanding outward through the set at an angle determined by the focal length of the lens chosen for the shot. Only those objects and portions of the set within this wedge will be seen in the frame. Second are the nearest and farthest planes of focus: determined by the currently chosen aperture setting, these planes enclose the volume within which objects will appear in the final frame to be crisply focused. Additionally, a continuous three-dimensional rendering of the camera's view is available outside the boundary of the set, allowing the operator to see how the shot will actually look when executed.

The set itself will show what the camera can and can't see at each moment.

You can also look at a 3D rendering of what the real shot will eventually look like.

When the shot has been completed, the portions of the set that have been cumulatively seen by the camera are visually highlighted; the significance of this provision

If the camera never sees it, you don't have to pay to build it.

is that, for example, if we find that no part of the set's back wall has been met by the camera's gaze then we need not even construct that wall. On a typical big-budget feature, that knowledge alone could mean saving a week of construction and tens to hundreds of thousands of dollars in construction costs.

8.4 Polyluminism

If realizing the Luminous Room were our sole aim (and we intended to work backwards from that goal) then clearly the front-projective approach of the I/O Bulb would not be the only one available. Getting a room's surfaces to bear mutable information could also be accomplished through rear projection or through tiling of the surfaces themselves with, say, plasma display panels. Less immediately realistic but more ultimately appropriate would be the prospect of an addressable surface coating, like Jacobsen's E-Ink [9].

Projectors aren't the only way to build a Luminous Room.

There are, certainly, drawbacks to each technique, as there are advantages to each. For example, use of front-projection always risks physical shadows near the base of any object of appreciable thickness – i.e., one or another portion of the surface on which it rests will be geometrically inaccessible, no matter the position of the projector. On the other hand, a back-projected or self-luminous surface cannot hope to paint the upward-facing part of an object atop it, a task easy enough for a remote projector.

Other ways of making everything glow.

In the end, a combination of several (perhaps even all) of the available means for attaching visual information to objects and surfaces will likely be desirable: a Fully Luminous Room. For the moment, however, and inasmuch as it is the only option both sufficiently mature and architecturally noninvasive, we elect to rally our efforts round the standard of projection from afar. In particular, the I/O Bulb is our instantiation of the distinguished case in which the point of optical dissemination and the point of optical collection are collocated. And at any rate: it is the I/O Bulb that is our point of departure and (ultimately) the concept of our true affections.

Ideally, we use more than one method at the same time.

But for now: projection out, video in.

A simple experiment could show how projection from above (with video capture also from above, as before) can be combined with a self-luminous surface to provide object augmentations impossible with either alone. We'd temporarily erect the usual projector-and-camera pair over a recently donated Fuji flat-panel

A first step: projection from below *and* above?

plasma display. The projector aloft would be used solely to project onto the surfaces of physical objects moved around the workspace, while all other display would be handled by the supporting plasma display surface. The two projective spaces would be aligned so that graphical structures can cross the silhouette boundaries of the application's objects with apparent continuity.

9 Coda

We have taken the first steps toward a pervasive augmentation of interior architectural space: architectural surface as interaction site.

This dissertation and the work it represents have introduced the I/O Bulb and Luminous Room concepts; our efforts have been aimed at answering two simple questions with regard to these proposed structures: 'can we?' and 'should we?'.

9.1 Can Revisited

'Can we?' referred to the technological feasibility of what we'd imagined.

9.1.1 Hardware

To date we have constructed several working I/O Bulb prototypes using off-the-shelf components. While larger than desirable (projectors are still bulky in 1998) and not altogether of the prescribed form (the camera and projector are spatially separated), these prototypes have been entirely adequate as a testbed environment for the applications that are part of our conceptual proof. Further, positive reaction to the particular applications we've built so far suggests that at least in professional contexts these 'I/O Bulb stations' would be perfectly acceptable in their present, temporary form (i.e., a special table where you go when you want to use the urban planning application, the optical prototyping workbench, &c.).

Of course, the full goal is still an I/O Bulb compact enough to be discreet and cheap enough to become ubiquitous. A round-and-a-half of prototyping with our unnameable corporate bulb-partner has shown remarkable progress toward miniaturization. Alternate technologies are evolving rapidly as well, and there is good reason to think that the most promising avenue may involve literal integration of the output mechanism and the camera sensor at the fabrication level: a single chip surface at which projection and detection occurs. Such a scheme will permit true coaxial I/O Bulb operation without the inefficient light-loss that's an inevitable consequence of the beamsplitter approach.

9.1.2 Software

The other fundamental challenge to building a working system is the basic software that interprets the environment as seen by the I/O Bulb. As an initial solution we have developed *glimpser* and *voodoo*, a two-part vision scheme that identifies objects through unique geometric color codes. This approach has proven reliable and efficient: in all of our demonstrations to date, both vision steps share a single modest CPU not only with each other but with the end application they serve, and a respectable frame rate is nonetheless maintained.

glimpser and *voodoo* are only an initial measure, however. Eventually it will be desirable to bring to the problem more sophisticated vision techniques, likely in cooperation with other non-vision identification and positioning techniques (contact-free tags, for example).

An extension to the *voodoo* module that serves each I/O Bulb – called *dee-voodoo* – allows it to communicate with arbitrarily many other I/O Bulbs, sharing with them information about the objects in its purview. *dee-voodoo* is thus the key component in one simple architecture for realizing the Luminous Room idea.

9.2 Should Revisited

'Should we?' referred to the applicability of what we were developing: would there be something interesting to do once we'd succeeded with the basic I/O Bulb technology?

To date we have constructed two comprehensive applications (and a host of smaller design explorations) that have helped to show why the I/O Bulb was worth fussing over: *Illuminating Light* is an optics layout environment that closely mimics the way real optics and optical phenomena behave, allowing an experimenter to rapidly prototype experiments; and *Urp* is an urban planning workbench in which an urban designer's pre-existing architectural models can be 'brought to life', interacting with each other and a simulated physical environment.

Reaction to these systems, both from professionals within the fields the systems were designed to serve and from general nonprofessional observers, has been extremely positive. Architectural and optical engineering professionals have expressed a great interest in incorporating these new tools into their daily practices; and representatives of other fields have frequently made a generalizing leap, imagining I/O Bulb applications that could address their own particular concerns.

9.3 The Importance Of Objects

The facility of the I/O Bulb for binding together physical objects and projected information gives rise to a new style of human-computer interaction which we have introduced as *luminous-tangible*. And although the projective component is certainly important – this light is, after all, the sole *literal* output of the I/O Bulb – it's only with a careful balance between projected graphical information and solid physical objects that I/O Bulb-based interactions are offered to best advantage.

We have made a small foray into a theoretical understanding of the different ways objects behave, the ways they can be used, and the ways they are perceived when they are components in luminous-tangible interactions. As we build additional (and more topically varied) applications for the I/O Bulb and Luminous Room we will continue to develop our understanding of what characterizes proper luminous-tangible interactions and how best to design them.

9.4 At Home

Our demonstration of the I/O Bulb's and Luminous Room's worth ('should we') has been predicated on applications designed to serve various professional communities.To be ideologically frank, though: while we do indeed believe that we have succeeded in showing a beneficial evolution of the lightbulb, the professional realm in which we've done this is only half the game.

The other half – and it's a much harder half, ultimately – is a more ordinary venue: the home. If every house's rooms become Luminous Rooms, what really will someone want to do there? The needs of a scientist, of a designer, of a planner, of a person at work are much different from the needs of an individual at home (even if it's that same person).

So as the research begun here continues and broadens, a critically important branch will be the one that investigates and designs useful luminous-tangible interactions for the home.

But that's another story.

References

[0] Azarbayejani, A., Wren, C., and Pentland, A. Real-Time 3-D Tracking of the Human Body. *Proceedings of IMAGE'COM 96*, May 1996.

[1] Bolt, R. "Put-That-There": Voice and Gesture at the Graphics Interface. *Proceedings of SIGGRAPH, 1980*, 262-270.

[2] Feiner, S., MacIntyre, B., Seligmann, D. Knowledge-Based Augmented Reality. *Communications of the ACM*, Vol. 36, No. 7, July 1993

[3] Fitzmaurice, G. W., Ishii, H., and Buxton, W. Bricks: Laying the Foundations for Graspable User Interfaces, *Proceedings of CHI '95*, 442-449, 1995.

[4] Frisch, E., Hasslacher, B., and Pomeau, Y. Lattice-Gas Automata for the Navier-Stokes Equation. *Physical Review Letters*, 56, 1505-8

[5] Gaddis, W. The Recognitions. Penguin Books, 1993.

[6] Hariharan, P. Optical Holography: Principles, Techniques and Applications. Cambridge University Press, 1984.

[7] Ishii, H., Kobayashi, M., Arita, K. Iterative Design of Seamless Collaboration Media. *Communications of the ACM*, Vol. 37, No. 8, August 1994.

[8] Ishii, H. and Ullmer, B. Tangible Bits: Towards Seamless Interfaces between People, Bits and Atoms. *Proceedings of CHI '97*: 234-241, March 1997.

[9] Jacobsen, J., Comiskey, B., et al. The Last Book. *IBM Systems Journal*, Vol. 36, No. 3, International Business Machines Corporation, 1997.

[10] Krueger, M. Artificial Reality II. Addison-Wesley, 1991.

[11] McLuhan, M. Understanding Media. MIT Press, 1994.

[12] McNerney, T., and Keays, B. The metaField: A Human-Scale, Interactive Projection Floor. Submitted to *CHI '99 late-breaking results* session.

[13] Niyogi, S., and Adelson, A. Analyzing and Recognizing Walking Figures in XYT. *Computer Vision and Pattern Recognition*, IEEE Computer Society Press, June 1994.

[14] Niyogi, S. and Freeman, W.T. Example-based Head Tracking, *Proc. 2nd International Conference on Automatic Face and Gesture Recognition*, 374-378, 1996.

[15] Starner, T., Weaver, J., and Pentland, A. A Wearable Computing Based American Sign Language Recognizer. *Proceedings of International Symposium on Wearable Computers 1997*.

[16] Ullmer, B., and Ishii, H. The metaDESK: Models and Prototypes for Tangible User Interfaces. and Atoms, in *Proceedings of UIST'97*, 223-232, October 1997.

[17] Ullmer, B. and Ishii, H. Formal Representations for TUI Primitives: Towards a Theory of Tangible Interfaces. Submitted to *CHI '99*

[18] Underkoffler, J. A View From The Luminous Room. *Personal Technologies*, Vol. 1, No. 2, 49-59, June 1997.

[19] Underkoffler, J., Ishii, H. Illuminating Light: An Optical Design Tool with a Luminous-Tangible Interface, in *Proceedings of CHI '98*: 542-549, April 1998.

[20] Underkoffler, J., Ishii, H. Urp: An Luminous-Tangible Workbench for Urban Planning and Design, in *Proceedings of CHI '99.* May 1999.

[21] Wellner, P. Interacting with Paper on the DigitalDesk. *Communications of the ACM*. Vol. 36, No. 7: 87-96, July 1993.

[22] Yee, R. Architectural Drawing. John Wiley & Sons, 1997