Published in the Proceedings of CHI '98, April 18-23, 1998, ©1998 ACM

Illuminating Light: An Optical Design Tool with a Luminous-Tangible Interface

John Underkoffler and Hiroshi Ishii
MIT Media Laboratory
Cambridge, MA
{jh,ishii}@media.mit.edu

ABSTRACT

We describe a novel system for rapid prototyping of laser-based optical and holographic layouts. Users of this optical prototyping tool – called the *Illuminating Light* system – move physical representations of various optical elements about a workspace, while the system tracks these components and projects back onto the workspace surface the simulated propagation of laser light through the evolving layout. This application is built atop the *Luminous Room* infrastructure, an aggregate of interlinked, computer-controlled projector-camera units called *I/O Bulbs*. Philosophically, the work embodies the emerging ideas of the *Luminous Room* and builds on the notions of 'graspable media'.

We briefly introduce the *I/O Bulb* and *Luminous Room* concepts and discuss their current implementations. After an overview of the optical domain that the *Illuminating Light* system is designed to address, we present the overall system design and implementation, including that of an intermediary toolkit called *voodoo* which provides a general facility for object identification and tracking.

Keywords

engineering simulation, optics, holography, luminous interface, tangible interface, augmented reality, prototyping tool, interactive projection, tangible bits

SCENARIO

Two optical engineering students stand at an ordinary table. One pulls from a box a stylized plastic object – it looks a bit like a laser – and places it on the table. Immediately a luminous beam, projected from above onto the table's surface, 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 from 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, reflecting off the splitter's surface. The student rotates the beamsplitter model in place (the partially-reflected beam sweeping 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.

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.

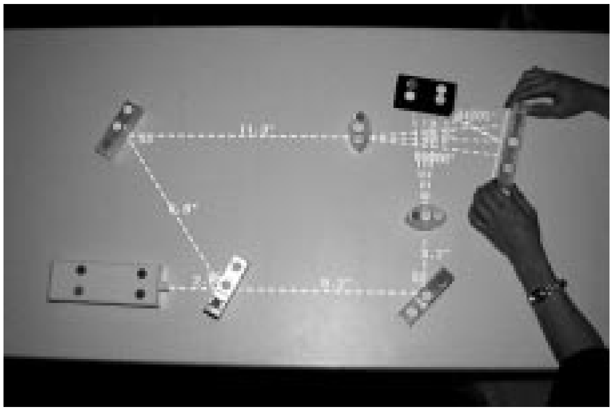

FIGURE 1: THE ILLUMINATING LIGHT SYSTEM IN USE

INTRODUCTION

The scenario described above is *Illuminating Light*, a working application of the *Luminous Room* infrastructure and the central topic of this paper. It is built atop *voodoo*, a toolkit for constructing layout-based interactive simulations, and employs a Medium-Scale *I/O Bulb* for display and scene capture. Its half-physical interaction style also extends *Tangible Interface* ideas explored elsewhere.

I/O Bulb

We offer a conceptual generalization of the familiar lightbulb, as follows: from a slightly unconventional point of view, we can see an ordinary incandescent bulb as a projector – albeit an especially low resolution one. Indeed, such a projector has a resolution of 1x1 pixels; this single, large pixel originates at the position of the bulb and is overlaid on the entire surrounding room. The information stream that drives this 'projector' happens also to be its power feed. From here, however, it is easy to imagine increasing the bulb's resolution, so that the intensity and color of the emitted light is directionally dependent. Now the lightbulb is capable of projecting images into the space around it. At the same time, we imagine incorporating a video camera into this device, so that there is a mechanism not only for

---

output but for input as well; this new kind of lightbulb is always ‘looking where it’s going’. To this novel bulb’s power feed we must now add a two-way information stream.

In the context of this paper, the *I/O Bulb* is the atomic unit of required physical mechanism, i.e. a single, compact device that performs coincident (or near-coincident) projection and video acquisition. For most of what follows we will simply presume that such a facility is available, although we briefly describe below three different experimental *I/O Bulb* apparati, all constructed using commercially available projectors and cameras optically bound together. (A separate, parallel line of research, not discussed here, is the development of a ‘true’ *I/O Bulb* – a single glass housing containing an optically and electronically integrated miniature projector and tiny camera.)

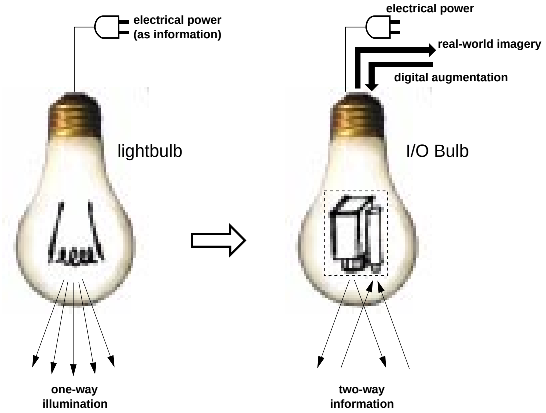

FIGURE 2: FROM LIGHTBULB TO I/O BULB

Luminous Room

The notion of a *Luminous Room* infrastructure involves extrapolating from just one to a collection of many *I/O Bulbs*, interlinked and distributed throughout some architectural space. The resulting aggregate of two-way optical nodes – exhibiting various levels of resolution, some nodes positioned to provide overlapping ‘regions of influence’ – acts to extend the meaning of architectural space, making every surface a potential site for digital interaction, display, and manipulation [9].

Implementation Scales

An idealistic vision would see the *Luminous Room* structure implemented by supplanting every extant lighting fixture in a room with an *I/O Bulb* of appropriate size; in the exploratory meantime we have built three contrasting but complementary prototypes of such an apparatus, each addressing a different ‘scale’ of detail, extent of addressable space, and motility.

Large Scale (ceiling-mounted gimbal system)

This prototype *I/O Bulb* consists of a high-resolution video projector mounted in a computer-controlled gimbal; it is designed to be suspended just below the ceiling of a room. Through a combination of mechanical and optical degrees of freedom its projection can reach every unoccluded part of the surrounding space, including the ceiling itself. Its associated video camera is subject to these same optomechanical rotations and thus is constantly aimed to view the portion of the room currently projected onto. This gimbal-based apparatus is designed to enable ‘room-scale’ interactions: this means both interactions with mobile access to the entire room (e.g. manipulable information that tracks a user around the space) and interactions of significant spatial extent (e.g. fixed, reactive architectural annotations along an entire wall).

FIGURE 3: LARGE-SCALE I/O BULB APPARATUS

Small Scale (‘Luxo’ desk lamp configuration)

The Luxo-style prototype comprises a very small video projector and coaxially-aligned miniature video camera mounted on an articulated arm. The camera-projector mount and the arm’s joints are outfitted with high-resolution encoders that report the instantaneous angular disposition of the device’s five degrees of freedom; forward kinematic calculations then provide the exact three-space position and orientation of the active elements.

FIGURE 4: SMALL-SCALE I/O BULB APPARATUS

In many respects, this apparatus acts in a manner complementary to that of its large-scale counterpart: where the gimbal system moves autonomously, the small-scale version is moved volitionally by the user; where the large-scale system is intended for broad, room-scale tasks, the small-scale apparatus is intended for spatially constrained, high-detail work; and where the gimbal rotates from a central position to operate ‘radially outward’, the Luxo-style device typically moves about the boundary of the interaction space it addresses, looking ‘radially inward’.

Indeed, we intend that its use should be suggested by that of an analogous desk lamp: just as such a lamp can be moved within a small volume to shed more light on some particular region or object, so the small-scale *I/O Bulb* can be moved around its working space to ‘shed more information’ at a desired point.

Medium Scale (fixed conference table system)

The Medium Scale *I/O Bulb* prototype is intended as a 'dedicated' fixture: an immobile ceiling-mounted projector-and-coincident-camera apparatus that serves some restricted area, like a conference table or a workbench. Being fixed allows the system to make stronger assumptions about the space below it; in particular, certain machine vision problems become simpler, since users' hands and arms will in general be the 'largest portion of human' that will occlude the table's surface. At the same time, such a system is able to provide higher resolution to its target area than would the gimbal-mounted version.

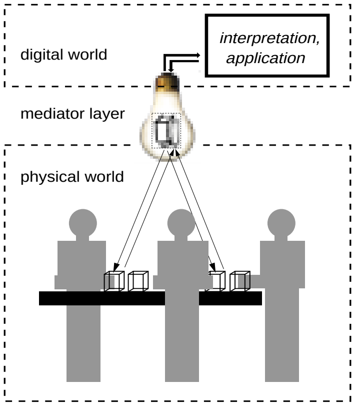

FIGURE 5: MEDIUM-SCALE I/O BULB SCHEMATIC

This paper describes an application that was implemented as a 'proof of concept' of the *Luminous Room* infrastructure in general and of the Medium Scale *I/O Bulb* concept in particular; a proper proof required identification of a domain in which a genuinely useful system could be built and evaluated.

APPLICATION DOMAIN: HOLOGRAPHY

For a variety of reasons, holographic 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 setup and iterative refinement times that greatly exceed the time spent running the actual experiment. 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.

Several powerful mouse-and-CRT-based optical layout and analysis packages exist (LightTools, ACCOS, ZEMAX, OptiCAD, etc.). However, intuition both 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 are 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 both foster and exploit such geometric understanding skills. In short, we set out to provide a virtual optical workbench with which experimenters can physically manipulate three-dimensional stand-ins of different optical components and directly observe the results.

Additionally, in applied holography 'correct' design solutions are unambiguously distinguishable 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 our university promised a ready supply of holographers, both student and professional, who could be invited to use the system and observed doing so.

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 [4]. 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 – often 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 geometric 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 simultaneous satisfaction of various geometric requisites with a single configuration of optical components. An example of such a layout is shown in Figure 6.

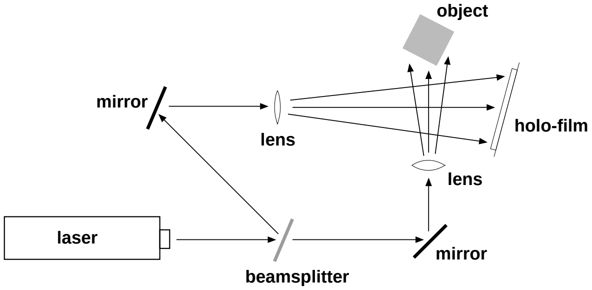

FIGURE 6: TYPICAL HOLOGRAM-RECORDING LAYOUT

SYSTEM DESIGN REQUIREMENTS

Our intent was to build, using the facilities of an *I/O Bulb*, a prototyping tool for holographic recording setups that would accurately simulate the interaction of laser light with various optical elements. These elements would be represented by physical models, while the 'beams of laser light' would be projected from above in careful alignment with the models. Although the optical elements and the laser beams – the former moved volitionally by human users, the latter computationally generated and projectively inserted into the real space – would be *implementationally* decoupled, the application would convincingly cause them to *appear* causally coupled.

Direct manipulation of the optics models would be close enough to working with the 'real thing' to be at once instructive to students and helpfully familiar to professionals. The system would furnish a selection of fairly standard optical components and allow their arrangement in arbitrary ways, but would also provide certain additional information (like relative optical pathlengths) of particular relevance and necessity to the construction of holographic recording layouts. The system would also be able to detect when a 'successful' holographic setup had been constructed.

We required that the system reject the standard 'one object at a time' restriction of mouse-and-keyboard input. Instead, users must be allowed to manipulate as many objects concurrently as necessary or convenient. The comparative efficiency of two-handed or simultaneous-translation-and-rotation handling has been clearly established [3]; we also demand of our application that it accommodate collaborative work, with two or more participants making simultaneous adjustments to many optical components.

Finally, the system should provide a memory feature, so that the instantaneous state of an optical layout could be recorded pictorially and then, later, projectively overlaid at proper scale on the empty workspace – either for purposes of subsequent 'off-line' review or so that users could rapidly and accurately replicate an earlier setup.

From this checklist we constructed the holographic workbench simulator called *Illuminating Light*.

BACKGROUND

The full concept of a *Luminous Room* infrastructure is related in a variety of ways to many recent and emerging strands of research that are often loosely collected under the tag 'Computer Augmented Environments' [8]. *Illuminating Light* considered as an individual application, however, is related a few particularly relevant works.

We find Pierre Wellner's *Digital Desk* research inspirational [10]; the extensive mixing of physical and digital artifacts through the use of video projection is a powerful notion. However, where Wellner proposed a migration of the virtual desktop off the CRT and back onto the real desktop, we are interested in enhancing or assisting the execution of real, physical activities – here, optical engineering – that are ordinarily outside the realm of 'computing'.

The *Illuminating Light* application represents an approach that is complementary to the usual methods of *Augmented Reality* systems, such as Steven Feiner's *KARMA* [2]; the unwieldy nature of head-mounted displays and the difficulties of ensuring real-world alignment and calibration make the prospect of 'untethered' interaction attractive. Further, the difficulty of providing a common reference when *physically personal* equipment is used to provide augmentation (a problem shared also by *Wearable Computing* rigs, as in [7]) can similarly be addressed with reality-aligned direct projection. Work at Stanford (the *Two-User Responsive Workbench* [1]) has shown one way to address the issue of 'shareable' virtual space – two independently head-tracked users equipped with stereo goggles are able to refer without spatial confusion to the same three-dimensional virtual objects, thanks to a four-processor, two-Infinite-Reality-pipeline SGI Onyx that generates two stereo image pairs for each timestep. We are following a contrasting path in which a single, external projection into the common, real-world workspace instantly offers shareability to an arbitrary number of users without requiring the proliferation of tracking, rendering, and viewing hardware.

The *Tangible Bits* work at the MIT Media Laboratory has provided a strong direction, prompting thought on seamless couplings of the digital and physical worlds [5]. The idea of 'phicons' – physical, functional icons – is of particular relevance to our optical simulation system. But where much of the Tangible Bits research has made use of phicons with various *symbolic correspondences* between digital meanings and physical manifestations, the objects in our application will have a *direct correspondence* to other physical objects (real optics) with definite, distinct uses and significance.

IMPLEMENTATION

The *Illuminating Light* system is built as a hierarchy of hardware and software systems. The physical apparatus on which it is dependent – a 'Medium Scale' *I/O Bulb* – is assumed for the duration of this paper. The software portions and application-specific physical components of the system and its supporting hierarchy are detailed here.

Overview

Users of *Illuminating Light* manipulate models of optical elements on which are affixed unique patterns of small colored dots. Visual input of the workspace is passed from an overhead *I/O Bulb* to a succession of vision analysis systems (*glimpser* and *voodoo*) that parse the dot patterns into recognized objects with attendant positions and orientations. The spatial configuration of objects thus identified is then used by a ray-based optical simulator to determine the resultant path of laser light; this path is visually rendered and, together with ancillary numerical and graphical information, accurately projected via the same *I/O Bulb* back into the workspace.

Vision

Machine vision is not yet reliable as a *general* input mechanism; however, under certain constrained circumstances it can be made to function reasonably well. In the case of the *Illuminating Light* system, vision analysis of a live video stream comprises the principal input.

For reasons both of reliability and of computational speed and efficiency, we decided to build upon a very modest 'raw vision' model: the *glimpser* program, generalized from an earlier version built by the authors and already in use in several projects around our laboratory, simply identifies colored dots in its visual input. Built as a client-server facility, *glimpser* accepts commands over a network connection to define, create, destroy, and condition 'finders'.

Each finder is an independent directive to locate within the input frame a specific-sized region of some particular color. Finders, once created, can be restricted to a certain subregion of the input field, can be temporarily deactivated and 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.

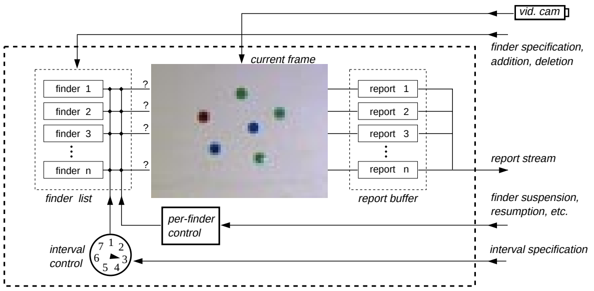

FIGURE 7: SIMPLE VISION ANALYSIS: GLIMPSER

Simulation Toolkit (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 registered 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 required distance, and with each contiguous triplet of dots a required angle. Every pattern is defined by a unique disposition of these parameters.

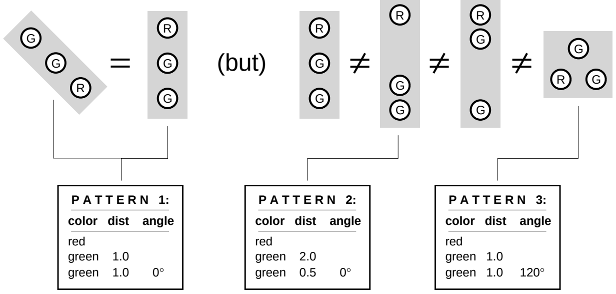

FIGURE 8: PATTERN PARSING WITH VOODOO

Further, each distance or angle specification has associated with it an individual tolerance within which range a 'match' may 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, vision-based pattern matches would simply fail most of the time. Second, the tolerance specification makes possible the definition of unique but 'parametric' patterns: for example, a lens in the *Illuminating Light* system is identified as the sequence 'red, blue, green' with a certain distance and a minimal tolerance specified for the red-blue pair, but with a 180° turn required between the red-blue and blue-green segments and a large tolerance for the blue-green distance. This means that a lens will be identified wherever a red and a blue dot are appropriately spaced and have a green dot *somewhere* along the line between them; but the application then uses 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.

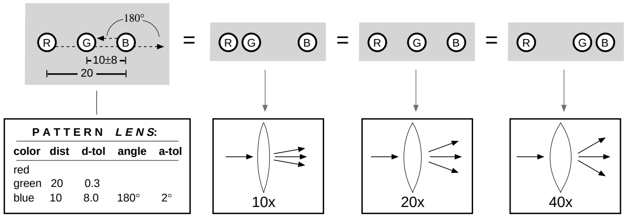

FIGURE 9: VARIABLE SINGLE OBJECT RECOGNITION WITH VOODOO

voodoo also provides an 'object persistence' mechanism: it is wise 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 periodically 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.

voodoo, as an independent toolkit, implies and makes easy a whole range of *Luminous-Room*-based simulation applications that use evolving layouts or distributions as a primary input. For example, we are currently constructing an urban planning system in which city engineers can continuously arrange *voodoo*-tagged architectural models and observe resultant simulations of traffic and pedestrian flow projected in perfect alignment back into the miniature urban space. Indeed, nearly any simulation that proceeds from the instantaneous and evolving arrangement of physical models, be they purely symbolic or more literally representational, will find *voodoo* a sturdy and convenient backbone.

Physical Representations of Optical Components

The holography setups to be executed with the *Illuminating Light* system require six basic optical elements: a laser, mirrors, beamsplitters, lenses, a 'holo-object' (the physical thing being visually recorded; in the present case, a small car), and the holographic film plate itself. From the point of view of the system's strictly technical implementation, these elements could be perfectly well represented with nothing more than their individual arrangements of colored dots, perhaps pasted onto cardboard strips. With regard for actual human users, however, we felt that carefully designed physical representations of these components would be an important element in the finished system.

Clearly, the objects have to be easily graspable; so a certain amount of corporeality is in order. The size of the workspace and the demand for a given spatial resolution from the *glimpser* dot-finder dictate an approximate scale. Each object needs to have its identifying dot pattern affixed to a top plane parallel to the workspace surface, and needs also to rest reliably on this surface – thus 'extruded' shapes with flat tops and bottoms are required. In the current implementation, all are about 1.25 inches tall and most about four inches long. The model mirrors, beamsplitters, lenses, and holographic plate are roughly an inch wide.

We require of the objects that they be simultaneously evocative and aesthetic (visually *and* haptically), balanced between direct representation and suggestive visual abstraction, and appropriate for the technical requirements of the vision system. We considered ourselves on the right design path when passers-by would more often than not stop to touch, pick up, or manipulate the object prototypes left on our work table even when the system was not turned on; the six current incarnations are shown in Figure 9.

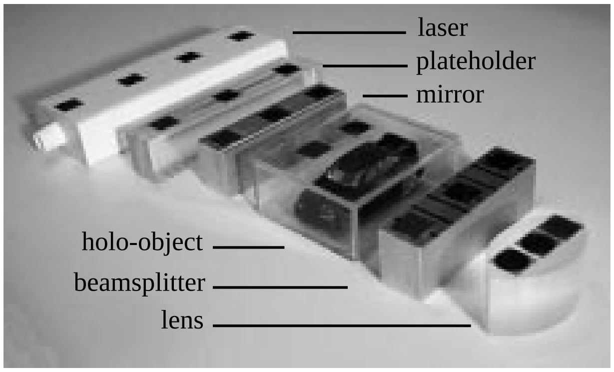

FIGURE 9: ILLUMINATING LIGHT'S SIX OPTICAL COMPONENTS

Optical Simulation

The outward, user-level function of the *Illuminating Light* application is holographic-optical simulation. The underlying simulator is heavily ray-based, both in its implicit treatment of optical behavior and in its implementation. Its three principal software objects are the *Ray*, the *OptElement*, and the *OptSystem*. Each *Ray* specifies an origin and a direction, and may also name a terminus – the downstream point at which the ray intersects an optical element and is transformed by it in one way or another. Each *OptElement*, meanwhile, represents some optical element and is defined by two essential functions: the first calculates, for any ray, whether the ray and the element intersect and (if so) the distance from the ray's origin to that point of intersection. The second generates a list (possibly empty) of new rays generated by the effect of the optical element on an intersecting ray. Each distinct kind of optical element is represented by an *OptElement* with different versions of these two functions, whose particular specification is alone enough to describe every possible sort of ray-based optical behavior.

Thus, for example, a beamsplitter is a kind of *OptElement* that generates two new rays for each one that intersects it: one continues forward undiverted and the other emerges from the intersection point at the calculated 'mirror-bounce' angle.

An *OptSystem*, finally, is a 'bookkeeping' object that contains a list of *Rays* and a list of all participating *OptElements*. Each simulation cycle involves first allowing every generator element – i.e. every element that, as does the laser, spontaneously generates output rays without first being intersected by an incoming ray – to produce its initial ray. Each of these initial rays is then made to terminate at the closest intersecting optical element (if any), and that element is allowed to 'process' the ray and transform it into zero or more output rays, according to the element's individual behavior. This same process is repeated for each of these secondary rays, and so on.

Miscellany

Display

The visual output of *Illuminating Light* is simple; its principal component is the rendered path of laser light. An initial implementation of this path as a set of static, unbroken lines projected in careful alignment between the model optical elements proved not entirely satisfactory: so long as components were being moved, the path of the beams would of course constantly evolve, and thus remain visually prominent; but when the setup was allowed to remain untouched, even for a short while, the beams tended to 'disappear' perceptually. The current implementation shows each beam as a dashed line whose segments move slowly forward (though the line's endpoints are fixed). This modest dynamism not only causes the beams to remain permanently in perceptual view, but also imparts a subtle and inviting 'life' to the system. Of course, it also serves conveniently to make the direction of beam propagation clear.

Numbers indicating the lengths of individual beam segments and various angles throughout the evolving optical layout are projected at appropriate locations. To these visual elements we also impart a small smooth sinusoidal motion that lends them unobtrusive visibility and assures that in regions of high graphical density each number is periodically moved 'out of the clutter'.

Finally, a real-time computer graphics rendering of the setup's holo-object, shown beside the active area of the workspace, simulates the visual output of a real hologram recorded according to the current optical configuration. This rendering shows only a silhouette if the setup is incomplete or improperly realized, but is shown fully shaded once a valid arrangement has been constructed.

All display elements are projected in grayscale; this is a precaution that acknowledges our colored-dot-based machine vision algorithm: introducing sufficiently saturated colors into the workspace can cause *glimpser* to report 'input' where there is none.

Memory Feature

The application automatically saves a frame of its host *I/O Bulb*'s video at regular intervals (every twenty seconds seems to work well). Additionally, a user may 'cheat' with a keystroke that causes the current visual state of the workspace to be immediately stored. These frames may be subsequently 'played back' into the workspace, either singly or as a sequence.

Hardware

glimpser and the *Illuminating Light* application work as a client server pair; however, they currently both run on one machine, a single R5000-processor Silicon Graphics O_2. An InFocus LitePro 620 provides projection onto a table surface from sixty-five inches overhead; a Panasonic KS-

152 miniature video camera is used for visual input.

RESULTS

Quantifiable

Even with both processes running on the same CPU, the application executes from thirty-two to forty complete simulation-cycle and output-frame-rendering iterations per second, while *glimpser* performs its work on input video frames at eight to twelve Hz. The comparative slowness of the input portion of the system (i.e. *glimpser* plus *voodoo* only managing an update of objects' positional and angular changes at 8-12 Hz) is made less noticeable by the high update rate of the animated laser beams and measurement displays – the 'world' always seems to continue running smoothly.

User Testing & Experience

A group of eight holographers and holography students has worked with the *Illuminating Light* application singly, in twos, and in threes; an additional unknown number of passers-by has also experimented freely with the system.

FIGURE 11: COOPERATIVE WORK WITH ILLUMINATING LIGHT

The immediately apparent advantage of using the *Illuminating Light* application is that it permits much faster prototyping than would be usual with real optics deployed on a breadboard or vibration-isolation table. The principal reasons for this are the evident durability and inexpensiveness of the model components and the comparative ease of their manipulation. Users were able to abandon the slow, deliberate care that is mandatory when handling real optical elements. Indeed, most of the holographers and holography students who helped test the system evinced a certain delight in being able to simply grab components and move them very quickly, not worrying about getting fingerprints on expensive optical surfaces or having to loosen and retighten magnetic bases. During collaborative use, a typical working suggestion like "let's move the laser and then swap the mirror and the beamsplitter" would usually be carried out in little more than twenty seconds. The same operation in a 'real' lab setting would require at least five minutes of a seasoned optical engineer and twenty or more minutes of an average team of holography students.

At the same time, the coupling between the physical optics models and the projected laser-path simulation was tight enough so that users tended to dismiss the distinction as irrelevant. Users indicated that they quickly came to "believe" the non-substantive laser beam (after the novelty of manipulating physical objects and seeing collocated projections respond had worn off).

The test subjects felt that the floating component-to-component distance measurements and the ray-to-ray angular measurements were helpful – in a real setup these quantities have to be measured carefully by hand and recorded in a lab notebook. They also responded favorably to the pathlength matching information presented outside the boundary of the 'active holography workspace', although several commented that it was a little strange to find these numbers in a separate 'display' area – that it would be more convenient if the information were incorporated into the evolving setup. Finally, users were unsure about the effectiveness of the rendered 'finished hologram' view presented in the side display area.

DISCUSSION & FUTURE WORK

The comments of test users suggest that we might do well to explore alternatives to the separate 'display' region on the table's surface, folding its functionality into the active optics region. The pathlength matching information could receive straightforward treatment as additional, appropriately positioned textual and numerical annotation throughout the setup. The 'finished hologram' view is a bit more difficult to incorporate into the workspace, particularly as the system can only project onto the '*x-y*' plane while most of a hologram's relevant visual information resides in the inaccessible '*y-z*' plane. This would be an opportunity, however, to demonstrate the profitable overlap of two separate *I/O Bulb* instances: the Small-Scale 'Luxo-style' prototype could be placed in the workspace to project such 'orthogonal' details into the setup.

We have demonstrated that under certain circumstances use of *Illuminating Light* permits much faster prototyping than is possible on a real optical table with real optics. However, as this application is poised somewhere between physical optical system layout and a software tool (like ZEMAX) intended for highly precise optical design and analysis, its comparison against this latter class of tool will be of clear importance. We intend therefore to perform a series of tests (again with real holographers and holography students) to establish the relative merits of these two approaches, both in the context of education and of professional engineering.

There is an ongoing issue, endemic to such physical interface systems, regarding control of parameters beyond the narrow bounds of the simulation. For example: how, other than via the keyboard or mouse, should a user of *Illuminating Light* be able to switch the display of automatic angle and distance measurements on and off? What is an appropriate way to trigger the capture and storage of memory frames, and then later select among various saved frames for retrieval and display? Such manipulations are ordinarily handled quite adequately (if somewhat inelegantly) by screen-based GUI elements. But to introduce additional 'control objects' that are tracked throughout the workspace in the same manner as the optical elements would not only dilute the purity and directness of the interface to the simulation but would also likely necessitate cordoning off a portion of the workspace as a distinct 'control zone'. Finding a solution to this dilemma is important.

While *voodoo* is an efficient and reliable means of performing object tracking, the colored dots that it employs do somewhat compromise the aesthetic design of the objects themselves. An alternative is the use of a template-matching vision algorithm – such as the one reported in [6] – that can be 'trained' on the appearances of the objects themselves. Although there are predictable tradeoffs (e.g. partial occlusion of an object by a grasping hand is more likely to confuse template recognition than *voodoo* recognition, and redesign of an existing object or addition of a new object requires a full retraining instead of the simple application of new colored dots) we intend to build a version of *Illuminating Light* that uses such a vision system.

Illuminating Light has helped to prove the effectiveness of its hardware platform (i.e. the Medium-Scale *I/O Bulb* apparatus) and of *voodoo* in rapid construction of simulation systems. We are already at work on several new, contrasting systems. In addition to the urban planning scenario detailed earlier, we are exploring a less physically literal application: a signal-processing tool that allows designers to manipulate physical representations of digital elements (i.e. delays, adders, gain elements, etc.) and view intermediate versions of the signal shown along the connections between elements.

A larger issue concerns the projective philosophy of the *Luminous Room* and *I/O Bulb* structures. While active-surface systems (e.g. the *MetaDesk* system in [5]) never suffer from occlusion problems, a projective system has certain other advantages: it is possible to place information not only *around* but also *on* objects in the workspace. Further, it is possible to projectively address objects that may be located away from a supporting surface. Indeed, the large-scale *I/O Bulb* can address an entire room; a similarly ambitious active-panel approach would require quite a few panels.

CONCLUSION

We have described a system that, in accordance with the aims of the *Luminous Room* architecture, uses real-world surfaces as an arena both for display and for direct-manipulation input. The application, called *Illuminating Light*, succeeds through its mimicking of an optical workbench in marrying light and physicality: although its user-handled physical optics models and the computer-simulated path of laser light are in reality very distinct (at least from the point of view of the system), their close cognitive cause-and-effect coupling tends to mitigate the perception of the one as an *input* channel and the other as an *output* channel.

The system heavily exploits the advantages of control via graspable implements (as explored in other tangible interface work), but the additional strength of this particular application domain is that the system's components act not just as physically instantiated abstractions but as direct representations of the 'real thing'. This allows *Illuminating Light* to provide constant visual feedback in a form that is already intrinsic to the simulation's real-world counterpart – and so the 'virtual' part of the application does not seem distracting or glaringly distinct from its 'real' part. We may note that we do not perceive our interactions with normal, non-digital reality as characterized by *input* and *output*, but rather as a participation in a kind of continuous *causality.* In natural consonance with this interaction approach, the system cleanly permits two-handed and collaborative use.

Even though machine vision is not yet a reasonable general-purpose input mechanism, in controlled circumstances like the ones detailed here it can be acceptably effective (and even efficient: the entire *Illuminating Light* system, comprising one process for low-level vision and another for visual input parsing and the actual optical simulation with attendant rendering and display, runs on a single low-end SGI machine). We have found the *voodoo* toolkit, together with the *glimpser* vision software, useful for rapidly assembling such layout-based simulation systems; construction of the portions of the *Illuminating Light* application hierarchically above *voodoo* took roughly one week.

As one test-subject holographer noted, the 'in-plane' restrictions resulting from our projective architecture reduce what is in general a three-dimensional task to two dimensions. Indeed, while the beam path in most real holography setups is actually purely planar, running parallel to the table's surface and elevated some eight inches above it, a few setups require vertical inclination of the beam. Thus, while *Illuminating Light* accommodates perhaps ninety percent of typical holographic layouts, there remains a number of layouts that fundamentally cannot be addressed using a projection-based system.

ACKNOWLEDGEMENTS

We thank David Kung and Kevin Parent for early formulating discussions of reagent grade; Matt Slayter for dogged & relentless fabrication of several *I/O Bulb* prototypes; Andy Dahley and Paul Yarin for indefatigable help with optics model design & construction and for clarifying camaraderie, early and late; and Wendy Plesniak and Rob Poor for countless lambent insights at every step.

REFERENCES

[1] Agrawala, M., Beers, A., Fröhlich, B., et. al. The Two-User Responsive Workbench: Support for Collaboration Through Individual Views of a Shared Space, in *Proceedings of SIGGRAPH '97*: 327-332, August 1997.

[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, in *Proceedings of CHI '95*,: 442-449, 1995.

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

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

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

[7] Starner, T., Mann, S., Rhodes, B., et. al. Augmented Reality Through Wearable Computing, to appear in *Presence* Special Issue on Augmented Reality, 1997.

[8] Underkoffler, J. Antisedentary Beigeless Computing. *Personal Technologies*, Vol. 1, No. 1, March 1997.

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

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