

# Annotating Remote Reality with iLight

J. Foote, D. Kimber,  
FX Palo Alto Laboratory  
3400 Hillview Ave. Bldg. 4  
Palo Alto, CA USA 94304  
{foote, kimber}@fxpal.com

T. Iyoda, T. Abe, K. Hirata, & J. Miyazaki  
Fuji Xerox Co. Ltd.  
430 Sakai, Nakai-machi, Ashigarakami-gun  
Kanagawa, Japan 259-0157  
jun.miyazaki@fujixerox.co.jp

## Keywords

camera/projector systems, active video, augmented reality, annotation, remote collaboration

## Categories and Subject Descriptors

H.5.3 [Information Interfaces and Presentation]: Group and Organization Interfaces – Synchronous interaction; H.5.1 [Information Interfaces and Presentation]: Information Systems – Artificial, augmented, and virtual realities

## ABSTRACT

The iLight system allows remote users to draw using light projected on a real-life object or scene. Using an aligned camera/projector system, live video from the scene is presented as a drawable video image to one or more remote users. Graphics drawn directly on the video canvas are projected onto the scene visible in the camera. Thus remote users may annotate particular scene objects or features for local viewers, and can also display text and project arbitrary digital images. Drawn objects are pixel-aligned with the video, and unlike previous systems, there are no video feedback problems, nor are computer vision techniques required to detect objects or actions. We present two versions of the system: a lightweight, inexpensive 2D version suitable for shared whiteboards, and version with specialized optics designed for annotating three-dimensional objects.

## 1. INTRODUCTION

iLight is a camera-projector system that allows a user to draw on a video image of a real-world scene using digital ink. The drawn marks are projected onto the scene, and are visible to remote users through the video image, as well as to anyone in the vicinity of the projector. (For discussion, we say the camera-projector system and scene are “local,” while users annotating the scene are “remote.”) The iLight system thus allows remote users to draw on real-world objects, using familiar drawing tools on an active video interface. Given an audio link, remote users can collaborate with

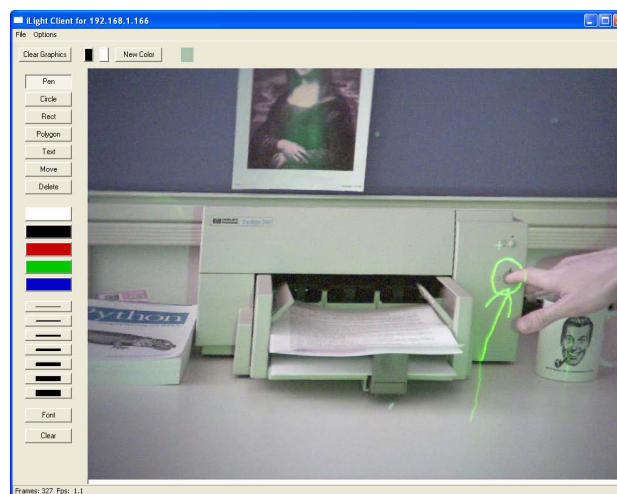
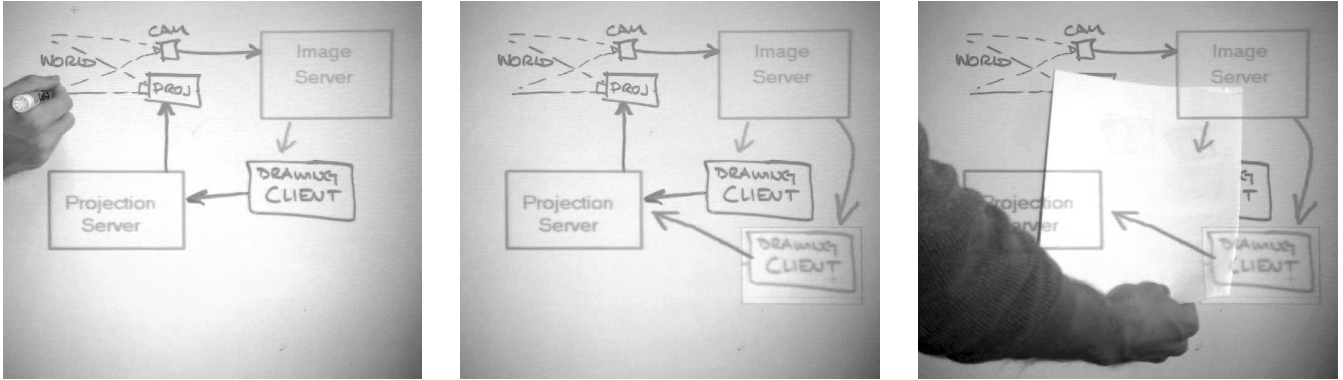


Figure 1: iLight video drawing client. A remote user has drawn an arrow and a circle directly on the video image. The drawn annotation is then projected on the real-world object.

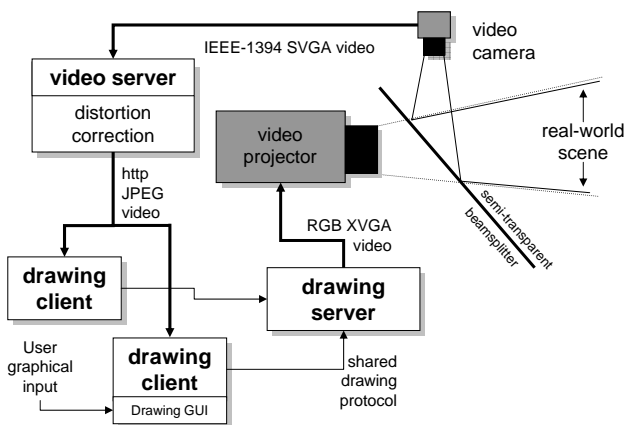
local personnel over a real object. Though not explicitly part of the system, we assume there is a separate channel to allow voice communication between the local and remote users. Because it is not critical to have audio synchronized with the video, any audio or side channel can be used. In practice, we use the telephone system (though a VoIP audio channel could be integrated with the iLight system, or even a text messaging facility).

Figure 2 shows a shared whiteboard as a motivating example. The iLight system’s video projector “draws” remote images on a local whiteboard, while sending live whiteboard video to remote users. In operation, remote users draw on the camera image with familiar graphical tools such as rectangles and (digital) pens, while local users draw directly on the whiteboard using (physical) dry-erase markers. The local projector, fixed with respect to the video camera, projects the remotely drawn images onto the whiteboard. Because the iLight camera and projector are calibrated and aligned, the projected image is drawn exactly where the remote user intends. Local users see the ink and projected light intermixed on the whiteboard, while remote users see a camera image of the same, as in Figure 2. Thus local and remote users can freely draw and annotate each other’s



**Figure 2: Shared whiteboard using iLight. Remote users draw with projected light while local users draw with ink. Remote users may copy and paste sections of the image, as shown center and left.**

work as if they shared a local whiteboard. Though neither can erase the other's marks, the remote user has functionality not available with physical ink. For example, the remote user may project any arbitrary image (as in Figure 12), as well as copy-and-paste ink marks (as in the central image of Figure 2).



**Figure 3: Projector/camera system**

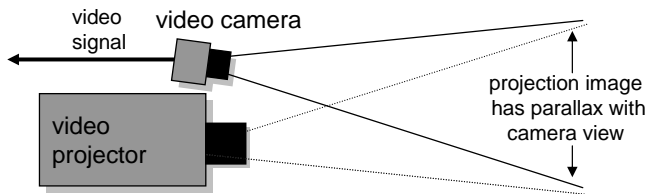
## 2. RELATED WORK

The iLight system furthers a long tradition of video-enabled collaboration tools. Space does not permit a full review, but we will review a few relevant precursors. A primary inspiration is the 1993 Digital Desk from Xerox's EuroPARC [11]. A principal difference between the Digital Desk and the iLight is that at least one iLight user is interacting with a computer screen, with all the affordances available in that domain. In the Digital Desk system, users interact only with the physical desk. Because the Digital Desk relies on computer vision to detect actions and documents, an implementation robust enough for everyday use has yet to be developed. A similar but more recent project named "Tele-Graffiti" was developed at CMU [9]. In this system, a pair of cross-coupled camera-projector systems allow two participants to mark on what is effectively a single surface. Like the Digital Desk, all interaction is via physical marks on pa-

per, and computer vision is required to detect and register drawing surfaces.

Another related system is Tang and Minneman's VideoWhiteboard developed at Xerox PARC [4]. The VideoWhiteboard system used only video projections of drawn images, and thus iLight functionality such as projecting digital text or images is not available. The "ClearBoard" is another system using cross-coupled projectors and cameras to simulate a transparent markable board [16]. Cross-coupled camera-display systems like VideoWhiteboard and Clearboard suffer from video feedback artifacts such as trails, oscillations, and blooming, which requires elaborate countermeasures like polarization filters or synchronized LCD shutters. In comparison, the iLight system has no direct video feedback, and thus does not suffer from this problem. Yet another shared whiteboard system, developed at IBM Research, allows users of a virtual-reality collaboration system to interact with physical whiteboard using a projector and a stroke-capturing pen input device for use on the whiteboard [6]. The Drawing Over Video Environment (DOVE) developed at CMU [17, 18] allows the user to annotate a video image with pen gestures for collaboration. Software detects and corrects drawn gestures like circles and quadrangles. However, a user interacting with a real-world object can only view the drawn annotations in a video monitor. In contrast, the iLight system projects the annotations directly upon the object, which is hopefully a more direct and intuitive approach. This is the method used by the "Everywhere Display" developed at IBM research [13]. In this system, a steerable mirror projects interactive controls onto the real world. A vision system enables gesture recognition for interaction with projected controls. Work at the MIT Media Lab has generalized some of these approaches into the concept of the "I/O Bulb," a combined projector/illumination source as in Figure 4. However this concept is intended primarily for interaction with tangible objects or computer simulations, and thus computer vision techniques are necessary for registration [5].

Several systems developed at the University of Tsukuba enable remote control of a steerable camera and a steer-



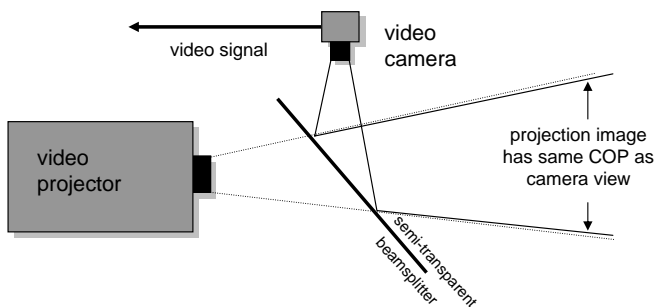
**Figure 4: The iLight 2D system does not require additional optics.**

able laser pointer [19]. Mouse gestures or other controls are translated into movements of the laser pointer “spot.” These systems were developed and demonstrated for remote instruction purposes, and include a bidirectional audio channel. A related system using a wearable, steerable laser pointer is discussed in [12]. Neither of these systems support video-aligned drawing or multicolor projection like the iLight system presented here.

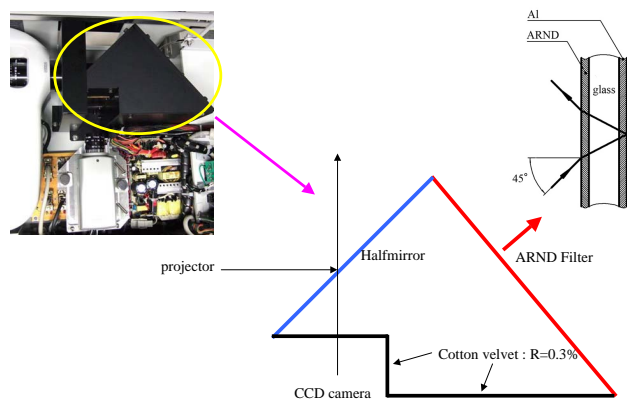
Many researchers have worked on so-called “augmented reality” (AR) systems [2]. This is a popular field and space permits only a brief overview. A typical AR system uses a transparent video display, often head-mounted, to overlay computer-generated information on real-world scenes. Displays must be precisely registered with the user’s environment so that displayed information is correctly aligned with real-world objects — a challenging task, especially for head-mounted devices. Not considering where the annotation data comes from in the first place, it is clear that the iLight is substantially simpler and requires no spatial registration other than that between camera and projector.

### 3. SYSTEM SOFTWARE

We first discuss the system software, as it works with any hardware configuration. We then present two projector/camera configurations. Figure 3 shows the general system architecture. The video server delivers video images to one or more drawing clients. Each drawing client implements a drawing GUI with drawing tools over the live video. Drawing clients send drawing commands to the drawing server, which displays the drawings on the projector. We discuss each module in more detail:



**Figure 5: The iLight 3D system uses a half-mirror beamsplitter to remove parallax between the projector and camera.**



**Figure 6: Optical system for combining projector and camera while reducing stray reflections from the projector.**

### 3.1 Image Server

The image server is responsible for delivering video. The camera delivers a video signal to the image server, which corrects for camera distortion and makes the video image available to the drawing clients. However good the imaging sensor, optical lenses will typically generate nonlinear barrel or pincushion distortions in the video image. We use an image warping algorithm that corrects for these radial distortions, so that straight lines in the real world map to straight lines in the video delivered by the server. Video frames are calibrated and corrected using routines from the Intel OpenCV library [14]. The function `cvUndistortOnce` corrects camera lens distortion for each served video frame, using a matrix of the camera intrinsic parameters and distortion coefficients calculated by the function `cvCalibrateCamera`. The server also corrects for any rotation in the camera with respect to the projector, so that lines projected at a particular angle result in a line with the same angle in the camera image. Finally, the server crops the corrected image to the projector’s field of view. The server delivers video as a series of JPEG frames via the widely-used HTTP protocol. Though this uses more bandwidth than interframe-coded video formats, it has the advantages of simplicity, arbitrarily good image resolution and quality, low latency, robustness across



**Figure 7: iLight 3D system showing projector, camera, and optical box including half-mirror and optical trap.**

firewalls, codec independence, and graceful behavior over a slow connection. Over a typical 100BaseT intranet we have no problem achieving a full frame rate of 15 frames per second.

### 3.2 Drawing Client

We have two drawing clients’ a prototype implemented in Python with image processing routines linked from the OpenCV library, and a production client written as a Java applet for portability. The most elegant approach is to run the client on a wireless pen tablet, however it can be used on any modern PC with a pointing device. The client supports a graphical user interface for drawing on the video image. It displays the video as a canvas upon which a user may draw using familiar tools such as pen, line, rectangle, text, and selection tools, with selectable ink colors, pen widths, and text fonts. The drawing tool lets users access the background video image; for example, an “ink dropper” tool lets users select a pen color from the video image. In addition, users may select a region of the video image, and copy and paste it elsewhere in the drawing. Another feature supports image drag-and-drop, where users may drag an image from their desktop onto the drawing canvas, where it can be moved and resized as it is projected on the real-world scene. The drawing client aligns the drawing canvas with the video image using a simplified affine transform. Because rotation and distortion have already been corrected in the image server, only translation and scaling is necessary to align the drawing coordinates to the video coordinates. We notate this as  $x' = c_x x + x_0$ ;  $y' = c_y y + y_0$ . Because this changes only when the camera is moved, this need only be set once; our approach is to project a rectangle that spans the image, and manually identify the corners. These four points are sufficient to solve for the four unknown affine parameters  $c_x$ ,  $c_y$ ,  $x_0$ , and  $y_0$ .

### 3.3 Automatic Camera-Projector Registration

Besides a manual calibration described above, we have also implemented an automatic calibration system. Our approach solves only for the four affine parameters described above, so is practical only for a planar projection surface. To register the images, we need to find the location of at least four points in both. To do this, we first capture a reference video frame after projecting a uniformly white image. Next we project a series of blue circles at various points in the projection image, and capture additional calibration frames. After subtracting the reference frame from each calibration frame (to compensate for differences in illumination and reflection), the centroids of the circles are detected by thresholding and averaging the difference image. These centroid points, with the centers of the projected circles, allow us to solve for the four best-fit affine parameters  $c_x$ ,  $c_y$ ,  $x_0$ , and  $y_0$ . This works in practice, though more accurate centroid estimates could be found by better estimates of fiducial point location, for example by convolution. Even more sophisticated calibration can compensate for non-planar projection surfaces, as has been reported elsewhere, such as UNC [8].

### 3.4 Display Server

The drawing client sends drawing commands to the display server, using a homebrew protocol. The display server renders these commands into an overlay image, which is then

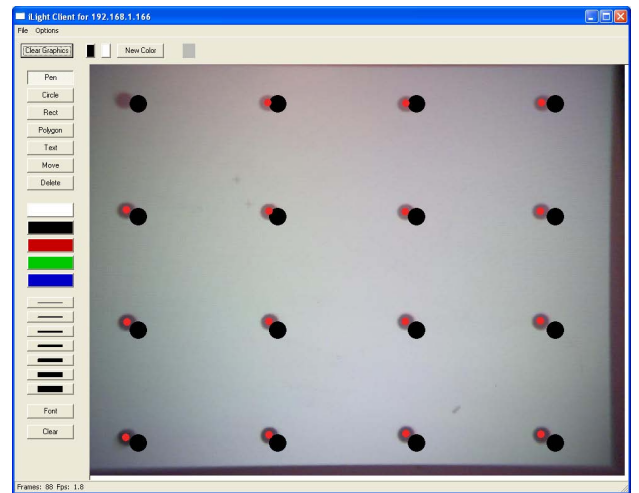


Figure 8: Automatic calibration: black fiducial circles are projected and detected (small dots) in the camera image. After calibration dot centers coincide (not shown).

projected onto the scene. If the projector system distorts the image we could correct it here as we do with the camera input; happily we find this is unnecessary because most projectors are quite linear. The display server can take input from an arbitrarily large number of drawing clients, which means the system may be shared by a number of remote users. Note that the projector projects the overlay image, which contains only digital ink and possibly bitmapped images. Because live video is never projected back into the camera view, video feedback with its consequent problems is not possible.

## 4. SYSTEM HARDWARE

Earlier camera-projection systems used analog video cameras by necessity, and suffered from the inherent low resolution and notoriously poor color fidelity [16]. Today, however, megapixel video cameras are not only available but surprisingly inexpensive, and the iLight system takes advantage of this by using high-resolution cameras with substantially sharper images and excellent color. We have implemented two “flavors” of the iLight system. The first we call **iLight2D**, which is a lightweight system useful for shared whiteboards or other planar objects. In contrast, **iLight3D** uses optics to remove parallax for use with three-dimensional objects. We discuss each configuration in turn.

### 4.1 iLight2D Hardware

**iLight 2D**, is a lightweight and inexpensive system primarily useful for shared whiteboards or tables. This system uses no additional hardware beyond off-the-shelf cameras and projectors; both are physically aligned but do not share a common center of projection (COP). Because of this, the system has inherent parallax, and is most practical for planar objects, hence the “2D” in the name. However the system’s simplicity and low cost makes it practical for many applications. We use an off-the-shelf Sony projectors combined with an high-resolution (1280 × 1024) FireWire (IEEE 1394) camera such as the Point Grey Research Dragonfly or the Veo Velocity Connect. To better match the camera’s

field of view with the projector, we replace the standard camera optics with a longer focal length lens, such as the 6mm lens sold by Supercircuits (part number ML6SG). This has a roughly 44° field of view when used with a 1/3" CCD.

## 4.2 iLight3D Hardware

Though iLight 2D is simple to construct and low in cost, it only works well for projecting on reasonably planar surfaces such as a whiteboard. For objects with substantial depth, we have implemented **iLight 3D**, which uses beam-splitter optics to align the centers of projection of both camera and projector, as shown in Figure 5. This reduces the disparity between objects and the projected image at any distance from the camera. Figure 7 shows the interior of the 3D system including projector (at left), camera (at bottom) and optical box (black triangle). We have implemented the optical trap depicted in Figure 6 to reduce stray projector reflections into the camera. Light from the projector not deflected by the half-mirror is reflected by an anti-reflective filter onto a highly absorptive black field. This arrangement reduces stray reflections by several orders of magnitude. The camera used is SONY DFW-SX900 (SXGA 1,280 × 960) color C-mount camera featuring a 1/2" CCD.

The 3D iLight system is intended for use on non-planar objects, yet the depth-of-field of commonly available video projectors is surprisingly short, sometimes only a few centimeters. This is insufficient for most real-world objects. Consequently, we have obtained a custom FUJINON projector lens (focal length 52.2mm, F number 1:2) with substantially higher depth of field (80-120 cm).

## 5. USER EXPERIENCE

In this section, we present the iLight system from the perspective of a remote user. Figure 1 shows a screenshot of the iLight user interface. Drawing tools are at the left, while the video canvas is at the right. Though there exist richer drawing environments, we have implemented basic tools including a pen, circle, rectangle, polygon, as well as moving and deletion tools. The line widths and colors of the drawing objects may be selected from a palette.

Operating iLight will be familiar for anyone who has ever used a computer drawing tool such as MacPaint or Adobe Illustrator. The chief difference is that drawing canvas is not a blank screen but a live video image. When the cursor is in the canvas, a user sees the camera image of the projected cursor. As the image has been precisely aligned with the drawing canvas, the cursor appears exactly where one would expect. When a user draws something, the drawn image is *not* typically rendered in the remote drawer's display; rather it is rendered locally as an overlay image, projected on the real-world scene, and is visible to the drawer through the video image. Again, because the overlay image has been precisely aligned with the drawing canvas, it appears exactly as it should on the remote user's computer screen. (Naturally, varying reflectances of a real-world scene will cause differences in the video of the overlay image; we discuss these issues in Section 5.3.)

### 5.1 Local mirroring

Though it is not necessary to render the overlay image on the remote user's screen, we have found that doing so is an



**Figure 9: Interface without (left) and with (right) local overlay image displayed. Locally showing the overlay image preserves the immediacy of drawing even with high video latency.**

excellent way to alleviate latency problems. The time delay between when the image is drawn and when it is visible can be significant for several reasons. The chief culprit is usually the video encoding delay. This is the time it takes to acquire and encode the video frames, followed by transmission, decoding, and rendering times. However, significant delays can arise simply from the speed of light delay between widely remote sites. In our system (which we have not yet optimized to minimize latency) we find delays to be the order of several hundred milliseconds between the time a figure is drawn and when it is visible in the camera image. This can be disconcerting as the pen seems to lag behind a user's mouse motions. A simple solution is to very briefly display the overlay image immediately as it is drawn, then have it rapidly fade away, revealing the camera image of the drawn figure "underneath." This nicely preserves both the immediacy of the drawing actions and the "transparency" of the camera image. This makes the system behave surprisingly well even in low or unpredictable bandwidth situations. We find iLight to still be perfectly usable at slow frame rates of only a few frames per second. Figure 9 shows the video image with and without the local overlay, which normally obscures the projected image.

### 5.2 Throwing Images and Cloning

The iLight system is able to project arbitrary images. By dragging an image file from the desktop to the iLight window, the image becomes an object that can be resized and moved around the video canvas. Naturally, the image is projected onto the real-world scene so it is visible by local users – we call this "throwing" an image. Because of the high resolution of typical projectors, it is entirely reasonable to throw a page of text for local users to read (though it might not be legible in the camera view). In a shared whiteboard scenario, local users can then mark up the projected image, as in Figure 13. As an additional feature, it is possible to "clone" a region of the video image and re-project it elsewhere. This is particularly useful in a shared whiteboard setting, as it gives copy-and-paste functionality to real-world drawings. For example, in Figure 2 the box entitled "DRAWING CLIENT" has been cloned in the middle and left images. Though not currently implemented beyond the "print screen" function in Windows, it would be

straightforward to save captured video images as a record of collaboration [3].



Figure 10: Annotating reality

### 5.3 Imaging issues

The real world is not an ideal projector screen. Problems may arise with excessively dark or specular (mirror-like) objects. Fortunately, these do not dominate typical scenes. If an annotation is not visible because of absorption, reflection, or shadow, this will be apparent to the remote drawer who can then modify the annotation to be more visible. Two major problems occur during iLight use, though both may be overcome in practice. The first is parallax; that is, the local users will not see exactly what the remote user sees unless they are in the exact same location as the camera. This is because projected images are distorted by the topology of the surface they are projected upon. This is not a huge obstacle for several reasons. Firstly, most annotations will indicate features or aspects of the indicated object, and thus are reasonably invariant to orientation. For example, a planar feature circled in the video image results in an ellipsoidal projection when the surface is not perpendicular to the projector. However it is still obvious to a local viewer which feature has been indicated. In the worst case, the remote user can direct a spot of light using a filled rectangle or circle, as in the center image of Figure 11, which should serve for any highly irregular topology. Figure 10 shows the iLight3D system in use to annotate connections for a local user. Two Ethernet jacks that must be connected have been highlighted, and a picture of a necessary adapter has been projected on the wall. Note that the projected annotations are still visible even on the shiny satin steel of the top left component, as well as on the glossy black finish of the top right camera.

Surfaces having specular (mirror-like) reflectivity will cause problems because the projected image will be only marginally visible on the desired surface, and may even be reflected in undesirable directions. However, very few objects are entirely mirror-like in appearance and we have not found this to be a substantial drawback.

### 5.4 Usage Scenario

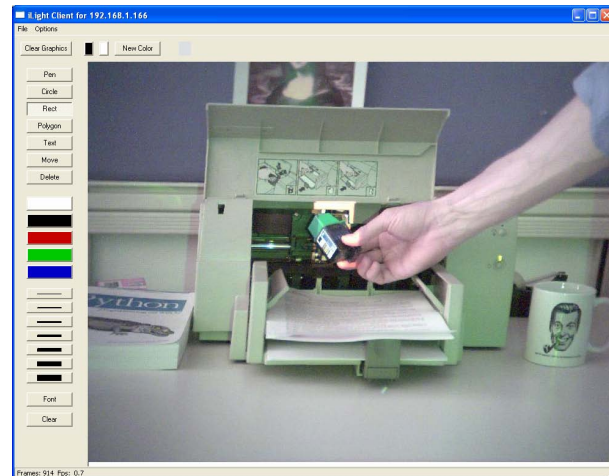
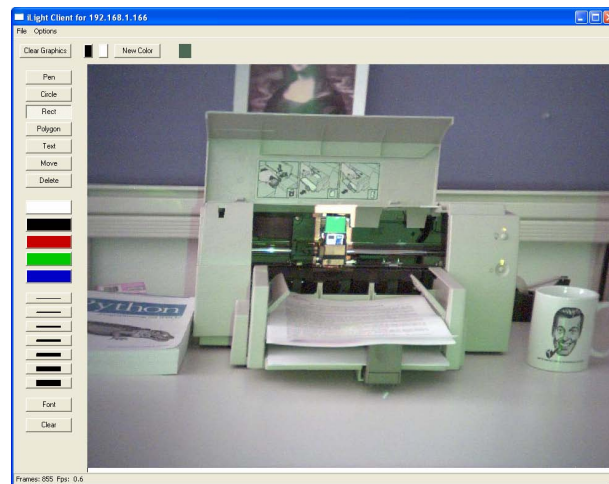
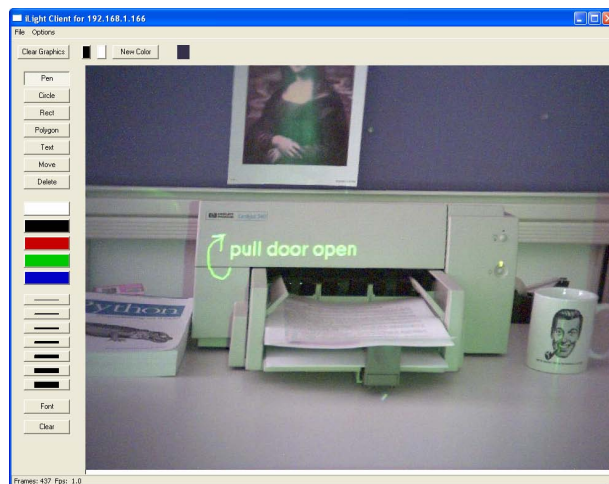


Figure 11: Remote service scenario: remote expert shows how to open printer (top), indicates the ink cartridge (center) and assists user in removing it (bottom).

To motivate the use of the iLight system, we illustrate a scenario where an office worker telecommuting from home is

having difficulty maintaining a printer. Though the worker can call on their employer's support staff, the office staff is obviously poorly located to help the telecommuter at home. However, using the iLight system, the company's expert support personnel can assist the telecommuter without having to transport the printer or the expense of a "house call." By directing the iLight system at the troublesome device, the telecommuter allows an expert to inspect the printer remotely, who can then indicate where and how to manipulate the device. Figure 11 illustrates this scenario. Over the voice channel the expert learns that the home worker needs to change the print cartridge, but does not know the procedure. From the iLight image, the expert can determine the make and model of the printer, and indicate how the printer may be opened (top). The telecommuter can then take the suggested action to reveal the ink cartridge inside. If it is not obvious, the expert can highlight its location by drawing a white rectangle over it. (center) This indicates the desired location even though the object is both irregular and poorly reflecting. With this assistance from the remote expert, the telecommuter is able to successfully locate and replace the cartridge (bottom of Figure 11).

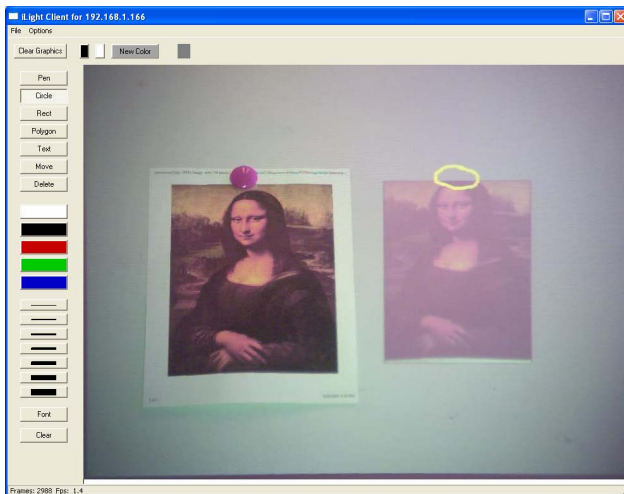


Figure 12: Virtual annotation of cloned image

## 5.5 Additional Applications

We anticipate several potential applications of the system:

- Remote service: In this scenario, a remote expert would guide a local user in the servicing of equipment or in some other task requiring human manipulation. The remote expert can indicate parts and features to the local user or users, and all can share information through the audio link. In addition, the remote user can project other information, for example diagrams or schematics, and annotate them for the instruction of local users. Local users can help in the interaction, for example a pad of paper could be put in the camera view as a ad-hoc screen for projected information.
- Telesurgery: Here, a remote doctor could indicate areas, organs, or operations via the projected light. A projected white area could add additional light for operations in cavities. As in the remote service application, information can be shared via the audio link.

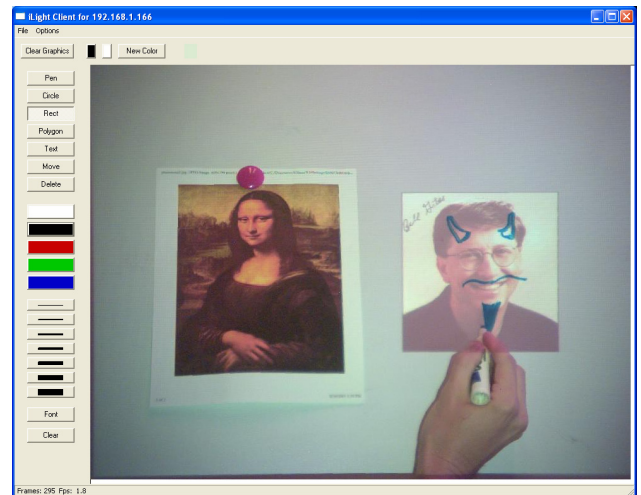


Figure 13: Real annotation of projected image

- Entertainment, sports or training applications: It is easy to imagine a performing arts scenario where a play director could indicate actor movements ("blocking") via an overhead camera-projector. Given a wireless pen tablet, the director could sit in the audience seating and indicate precise directions to a particular actor via arrows (for example) projected on the stage floor from above. Given the availability of high-luminosity laser projectors, such a scenario could be extended to a sports field or even a nighttime battlefield, using an infrared laser and IR-sensitive goggles for troops.
- Though this is not an "augmented reality" system in the classical sense [2], certain aspects support the literal sense of "augmenting" reality. For example, if the iLight system includes an optical character recognition (OCR) subsystem and a list of keywords, detected keywords in the camera view could be automatically highlighted by projecting a contrasting color on their bounding box(es). Such a system might help a human scan a large printed list for particular words or strings [15].

## 6. FURTHER WORK

We have begun the iLight project as a platform to investigate how automatic methods might be usefully combined with a camera/projector system beyond approaches already explored in the Digital Desk or Tele-Graffiti projects. This work has explored only a small set of a comparatively rich design space. We are currently investigating using camera feedback to enhance the projection onto surfaces with non-uniform reflectivity. By monitoring the reflection visible in the camera image, it is possible to modify the projected image to compensate for color and brightness variations in the scene. For example, when projecting on a checkerboard surface, the projection can be brightened in the dark squares and dimmed in the light squares to result in a more uniform perceived illumination. In a similar vein, to preserve the color balance when projecting on a colored surface, the color or background may be shifted to compensate so the perceived color is closer to that desired. In another approach, straightforward motion detection can estimate cam-

era motion, and allow an iLight system to be pointed in different directions (perhaps by a tilting mirror system as in [10]) while keeping annotations “attached” to the local objects. These approaches are relatively lightweight and robust, in contrast to the fragile computer vision or object detection requirements of other systems.

## 7. REFERENCES

- [1] J. Rekimoto and M. Saitoh, “Augmented Surfaces: A Spatially Continuous Workspace for Hybrid Computing Environments.” In *Proceedings of CHI’99*, 1999. <http://www.csl.sony.co.jp/person/rekimoto/as/>
- [2] R. T. Azuma, “A Survey of Augmented Reality.” In *Computer Graphics (SIGGRAPH ’95 Proceedings, Course Notes #9: Developing Advanced Virtual Reality Applications)* pp. 1–38, August 1995 <http://citeseer.nj.nec.com/azuma95survey.html>
- [3] Eric Saund, “Image Mosaicing and a Diagrammatic User Interface for an Office whiteboard Scanner.” In *Proc. of CoBuild’99. Second International Workshop on Cooperative Buildings*, Pittsburgh. <http://www2.parc.com/spl/members/saund/papers/zombieboard98.pdf>
- [4] J. Tang and S. Minneman, “Videowhiteboard: Video Shadows to Support Remote Collaboration.” In *Proc. of CHI*, pp. 315–322. 1991, <http://www1.ics.uci.edu/~jpd/NonTradUI/p315-tang.pdf>
- [5] J. Underkoffler, B. Ullmer, and H. Ishii, “Emancipated Pixels: Real-World Graphics in the Luminous Room.” In *Proc. SIGGRAPH 99* pp. 8–13 1999
- [6] Z. Shae, B. Tseng and W. Leung, “Immersive Whiteboard Collaborative System.” In *Annals of Software Engineering*, Vol. 12, pp.193-212, December 2001. [http://www.it.cityu.edu.hk/~howard/annssofteng2001\\_howard.pdf](http://www.it.cityu.edu.hk/~howard/annssofteng2001_howard.pdf)
- [7] T. Abe, K. Tokai, Y. Yamaguchi, O. Nishikawa, T. Iyoda, “New Range Finder Based on the Spatial Re-Encoding Method and its Application to 3D Modeling,” . In *Proceedings of SPIE Vol. 4661*, January 2002
- [8] R. Yang, M. S. Brown, W. B. Seales, and H. Fuchs, “Geometrically Correct Imagery for Teleconferencing.” In *Proc. ACM Multimedia 99* <http://www.cs.unc.edu/~ryang/publications/acm99.pdf>
- [9] N. Takao, S. Baker, and J. Shi, “Steady-State Feedback Analysis of Tele-Graffiti.” In *Proceedings of the IEEE International Workshop on Projector-Camera Systems*, October, 2003. [http://www.ri.cmu.edu/pub\\_files/pub4/takao\\_naoya\\_2003\\_1/takao\\_naoya\\_2003\\_1.pdf](http://www.ri.cmu.edu/pub_files/pub4/takao_naoya_2003_1/takao_naoya_2003_1.pdf)
- [10] F. Kjeldsen, et al. “Interacting with Steerable Projected Displays.” In *Proc. of the 5th International Conference on Automatic Face and Gesture Recognition (FG’02)*, Washington, DC. 2002. <http://citeseer.nj.nec.com/kjeldsen02interacting.html>
- [11] P. Wellner. “Interacting with paper on the DigitalDesk.” In *Communications of the ACM*, **36**(7):86–96, July 1993. <http://citeseer.nj.nec.com/wellner93interacting.html>
- [12] S. Mann “Telepointer: Hands-Free Completely Self Contained Wearable Visual Augmented Reality without Headwear and without any Infrastructural Reliance.” In *Proc. Fourth International Symposium on Wearable Computers (ISWC’00)*, Atlanta, Georgia p. 177 October 2000 <http://wearcam.org/telepointer.htm>
- [13] F. Kjeldsen, C. Pinhanez, G. Pingali, J. Hartman, T. Levas, and M. Podlaseck, “Interacting with Steerable Projected Displays.” In *Proc. of the 5th International Conference on Automatic Face and Gesture Recognition (FG’02)*. 2002. Washington, DC. [www.research.ibm.com/ed/publications/fg02.pdf](http://www.research.ibm.com/ed/publications/fg02.pdf)
- [14] Intel OpenCV Library <http://sourceforge.net/projects/opencvlibrary/>
- [15] Definition: Optical Grep <http://www.faqs.org/docs/jargon/V/vgrep.html>
- [16] H. Ishii, M. Kobayashi, and J. Grudin, “Integration of Interpersonal Space and Shared Workspace: Clearboard Design and Experiments.” In *ACM Transactions on Information Systems* **11**(4), October 1993 <http://web.media.mit.edu/~ishii/CB.html>
- [17] J. Ou, X. Chen, S. Fussell, and J. Yang, “Drawing Over Video Environment.” Demonstration, in *Proc ACM Multimedia*, 2003
- [18] J. Ou, S. Fussell, X. Chen, L. Setlock and J. Yang, “Gestural Communication over Video Stream: Supporting Multimodal Interaction for Remote Collaborative Physical Tasks.” In *Proc. of the 5th International Conference on Multimodal Interfaces*, Vancouver, British Columbia pp. 242–249, 2003.
- [19] K. Yamazaki, A. Yamazaki, H. Kuzuoka, S. Oyama, H. Kato, H. Suzuki, and H. Miki, GestureLaser and GestureLaser Car: Development of an Embodied Space to Support Remote Instruction.” In *Proc. ECSCW’99*, pp.239-258, 1999, <http://www.kuzuoka-lab.esys.tsukuba.ac.jp/~kuzuoka/publications/ecscw99/ecscw99.html>