MITSUBISHI ELECTRIC RESEARCH LABORATORIES
http://www.merl.com

Interactive Projections

Paul Beardsley, Ramesh Raskar, Clifton Forlines, Jeroen van Baar

TR2004-107 January 15, 2005

Abstract

Projectors have traditionally been used as fixed devices making fixed displays. But projector dimensions are shrinking and projectors are just now on the threshold of being compact enough for handheld use. Should handheld projection prove viable, it offers revolutionary new possibilities for display, with opportunistic projection onto nearby surfaces like walls and tabletops to create a display wherever needed. But there is a missing element here. Personal projection involving only passive viewing is of limited use, and it needs a mechanism for interaction to be truly versatile. Consider how viewing an electronic diary would be almost useless without selection and scrolling. Existing UI solutions for handheld devices don't obviously translate to a handheld projector. Our work attempts to address this, and we describe a recently introduced technique to demonstrate how familiar mouse-interactions can be done in the context of handheld projection, and via a natural single-handed pointing motion of the projector. We show how interactive projection supports both traditional applications in a new medium, and also novel applications.

IEEE Computer Graphics and Applications

This work may not be copied or reproduced in whole or in part for any commercial purpose. Permission to copy in whole or in part without payment of fee is granted for nonprofit educational and research purposes provided that all such whole or partial copies include the following: a notice that such copying is by permission of Mitsubishi Electric Research Laboratories, Inc.; an acknowledgment of the authors and individual contributions to the work; and all applicable portions of the copyright notice. Copying, reproduction, or republishing for any other purpose shall require a license with payment of fee to Mitsubishi Electric Research Laboratories, Inc. All rights reserved.

Copyright © Mitsubishi Electric Research Laboratories, Inc., 2005
201 Broadway, Cambridge, Massachusetts 02139

Interactive Projections

Paul Beardsley Ramesh Raskar Clifton Forlines Jeroen van Baar

Mitsubishi Electric Research Labs, Cambridge MA[*]

Abstract

Projectors have traditionally been used as fixed devices making fixed displays. But projector dimensions are shrinking and projectors are just now on the threshold of being compact enough for handheld use. Should handheld projection prove viable, it offers revolutionary new possibilities for display, with opportunistic projection onto nearby surfaces like walls and tabletops to create a display wherever needed.

But there is a missing element here. Personal projection involving only passive viewing is of limited use, and it needs a mechanism for *interaction* to be truly versatile. Consider how viewing an electronic diary would be almost useless without selection and scrolling. Existing UI solutions for handheld devices don't obviously translate to a handheld projector. Our work attempts to address this, and we describe a recently introduced technique to demonstrate how familiar mouse-interactions can be done in the context of handheld projection, and via a natural single-handed pointing motion of the projector. We show how interactive projection supports both traditional applications in a new medium, and also novel applications.

1 Introduction

Compact data projectors are already small enough to be transported easily, with a typical commercial model being maybe 6" square and weighing 2 pounds. This compactness is generating new modes of use, away from fixed projector installations, and towards more mobility and opportunistic deployment of projectors. In tandem with this, advances in self-calibration mean that an arbitrarily placed projector, maybe at an angle to the display surface, can automatically produce a desirable projection - keystone-corrected and with upright-orientation [Sukthankar et al. 2000; Raskar and Beardsley 2001]. Portability and self-calibration will likely lead to increasing deployment of projectors in homes, retail stores and other settings.

But the scenario above is still a familiar one – a fixed projector making a fixed display for mainly passive viewing. A much more significant innovation will see projectors used as handheld devices. There is a driving force for this progression. Cellphones and PDAs are becoming smaller for portability, but this is in direct conflict with the need for a fixed display that is easily visible to the user. A projector can address this problem because new advances are continually shrinking the physical dimensions of the hardware, while still producing good-sized projections. However, passive projection on its own is of limited use, and the focus of this paper is *interactive projection* - a technique for moving a cursor across a projection, allowing all the familiar mouse-interactions from the desktop to be transposed to a handheld projector, and without the use of a secondary device like a mouse, stylus or pointer.

1.1 Interactive Projection

How can we transpose familiar interaction methods from the desktop or handheld device to do interaction with a projection? There is well-developed work on laser pointer interaction with displays in fixed installations [Olsen, Jr. and Nielsen 2001], but we are concerned with the handheld case in arbitrary environments. It might seem straightforward to provide mouse-interaction by putting a touch-pad mouse on the handheld projector. But this adds bulk to the device, which we are trying to avoid. It also implies two-handed use. And there is a resolution issue, in doing fine control of a cursor on a potentially large projection area using a small touch-pad.

Our technique avoids these problems, and allows a cursor to be tracked across the projection by a one-handed pointing motion of the projector. Once the cursor is at the desired location, items are selected in the usual way using buttons on the handheld device.

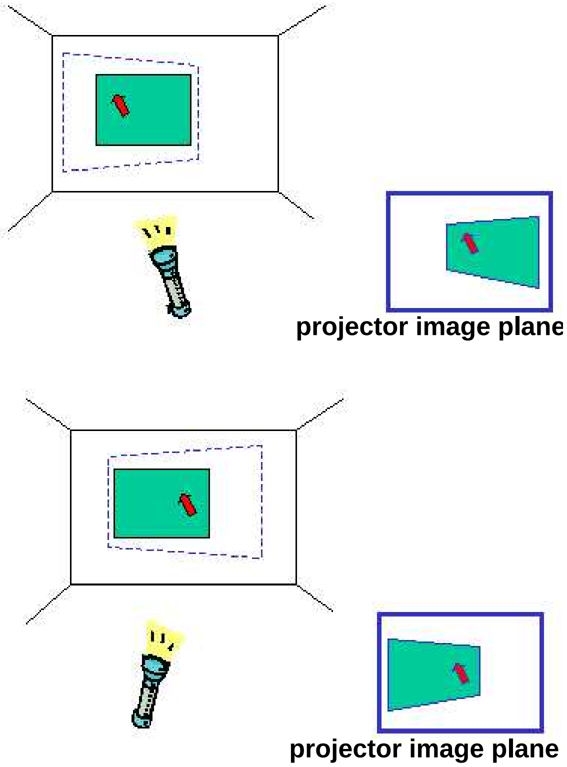


Figure 1: The handheld projector points toward the left- and then right-side of the display area. The main projection (green) stays static on the display surface, due to the adjustments made on the projector image plane. The cursor (red arrow) moves in direct correspondence with the motion of the projector, because the cursor graphic occupies a fixed position on the projector image plane. The user sees a static projection with the cursor tracking across it. The blue dashed line indicates the boundary of the full projection area.

The approach is as follows. First assume a way to compute the 3D motion of the projector relative to the display surface. Knowing the projector's 3D location at each instant, we can create any desired projection on the surface – in particular we can factor out the projector motion to create a projection that is static on the dis-

[*] email:[pab,raskar,forlines,jeroen]@merl.com

play surface. Figure 1 illustrates how the desired projection (green area) is modified on the projector image plane to attain the desired static projection on the surface. Now consider what happens if part of the projection is allowed to follow the motion of the projector i.e. some pixels on the projector image plane always have the same fixed graphic. Figure 1 illustrates this with a cursor graphic at the center of the projector image plane. The effect for the user is to see a static projection with a cursor moving across it in response to pointing motion of the projector.

One-handed pointing motion feels natural as a way to direct the cursor. Furthermore this leaves the fingers free to hover over the mouse buttons, ready for a button click. This is in contrast to the touch-pad where a single finger must do a context-switch between touch-pad and buttons, or different fingers are brought into play, but which is anyway an inherently more complex interaction for the user.

A constraint of our technique is that the projection data can only occupy part of the projector image plane as shown in Figure 1. About text-entry, this has not been a topic of our work so far, but a thumboard is a common solution for a handheld. An interaction that used pointing-motion to direct the cursor, the index finger for mouse-buttons, and the thumb for text-entry would be feasible, particularly since text-entry rarely occurs simultaneously with mouse-interaction.

1.2 Applications

What types of applications are supported by handheld projection and interactive projection? We propose three broad classes. The first class works using any clean display surface for projection. An example might be a PDA-projector, in which the projector is used to view a diary and scroll the contents. The projector's advantage here is purely that it allows opportunistic creation of a display on any convenient surface, but the particular surface is irrelevant.

The second class of applications creates a projection that is tailored to the physical surface. Typically this is projected augmented reality. The first stage is object recognition and recovery of projector pose relative to the object, and the next stage is projection of an appropriately positioned overlay that provides some information about the object. Interactive projection allows the user to, say, switch overlay modes, or to indicate a point of interest to obtain extra information.

The third class of applications is based on selection of 'physical regions-of-interest'. Consider how a typical way to define a rectangular region-of-interest on the desktop is by mouse hold-and-drag. We can transpose this to the physical world, using hold-and-drag of the projected cursor, with the projector visually indicating the selected rectangle on the physical surface. The selection might be used as input to computer vision processing e.g. the user might draw a physical region-of-interest around a word on signage in a foreign city, and request an automatic translation.

There are other interaction possibilities that we mention in passing. **Gestural motions** of the projector - such as horizontal or vertical shakes - to control the projection. **Pose-awareness** in the environment to determine projection content e.g. a projected augmented reality overlay might alter according to the distance between handheld projector and object. **Projected sketching** on a physical surface, using projection to show a persistent view of the cursor track across the surface. **Multiple handheld projectors** with intelligent combination of the projections in a pooled display.

1.3 Hardware

Our prototype handheld projector is shown in Figure 2. There is a hand-grip on the base with click buttons under the index finger for IO. The components are

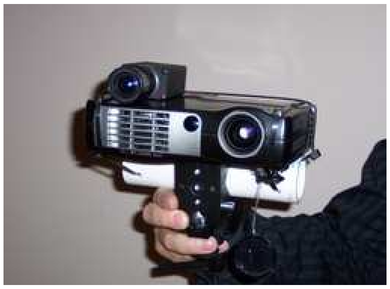


Figure 2: Prototype handheld projector.

- a Plus V-1080 projector, 1024x768 pixels, 60Hz framerate, dimensions 7×5×1.5 inches and weight 1kg,
- a Basler A602F camera, 640x480 pixels, 100Hz framerate,
- four rigidly attached laser pens (not shown in the figure),
- umbilical to a computer.

The prototype weighs about 2.5lb. It is suitable for experiments, but is too heavy for true portability. While there are challenges in creating a true handheld projector, some small projection devices do already exist – for example, the Canesta projected keyboard [www.canesta.com 2004], and the Siemens Mini-Beamer [www.siemens.com 2002]. These trends indicate that hardware solutions for a handheld projector are feasible, and we put discussion about hardware aside to concentrate on functionality and supporting algorithms for handheld projection.

2 Basic Techniques

Interactive projection was introduced in Section 1.1. Here we provide more detail about the algorithms – Section 2.1 describes how to make a static projection on the display surface even under motion of the handheld projector. Section 2.2 describes interactive projection.

2.1 Making a Static Projection

There are two issues to address when using a handheld device to create a projection. The first is keystone correction to produce projected data that appears undistorted and with the correct aspect ratio, even when the projector is at an angle to the display surface. The second is removal of the effects of minor hand motion and hand-jitter so that the projection appears static on the surface.

To address these requirements, we use fiducials (distinctive visual markers) on the display surface to define a coordinate frame. We emphasize that the use of fiducials - which implies a fixed workspace, contrary to one of our goals - is not a hard constraint. Any motion-tracking system for the projector would give the required information, and a good candidate is to have one or more cameras attached to the projector that are dedicated to motion tracking based on imaging the surrounding environment. But using fiducials is a straightforward approach and has allowed us to concentrate the work so far on applications, rather than on supporting software.

Figure 3 illustrates the basic task. The goal is to place the projection in the target area (red dashed line), which is defined relative to the four fiducials. This is achieved by using the camera to sense the fiducials, and hence to infer the target area in camera image coordinates. This information is transformed to give the target area in projector image coordinates. Finally the projection data is mapped to these coordinates on the projector image plane, giving correct placement of the projection. The whole process is repeated for each new time-step. The detailed steps are

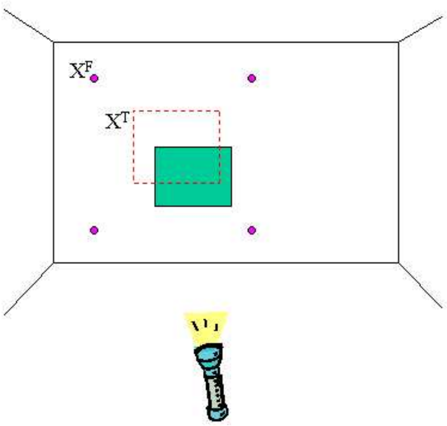


Figure 3: The purple points are fiducials on the display surface. The red dotted-line indicates the target area for a projection - this is defined relative to the fiducials but is not physically marked. The green area indicates the current projection. Projection motion (e.g. due to hand-jitter) causes the discrepancy between the target area and the projection, and this discrepancy undergoes a correction at each time-step.

- Detect the camera image coordinates c_i^F of the fiducials. Compute the homography H_{SC} between the display surface and the image, using the known surface coordinates X_i^F of the fiducials, and the points c_i^F.
- Apply H_{SC} to the known surface coordinates X_i^T of the target area, to obtain the camera image coordinates c_i^T of the target area.
- Detect the camera image coordinates c_i^P of the vertices of the current projection. Compute the homography H_{CP} between the camera image plane and the projector image plane, using the known projector image coordinates of the vertices of the projection, and the points c_i^P.
- Apply H_{CP} to c_I^T to generate p_i^T, the coordinates of the target area on the projector image plane.

The algorithm is projective. An alternative approach would be to work in euclidean space, explicitly computing the location and orientation of the projector relative to the display surface. This is sometimes useful (for example if we wish to make explicit the projector's distance from the display surface, for some other use). But the projective algorithm is lighter computationally.

In practice, there is a modification to step (3) above and the computation of c_i^P. Detecting c_i^P in the camera image is difficult when the projection has arbitrary appearance. Instead we detect laser spots from the four laser pens (Section 1.3), which is straightforward because the spots are bright. These spots provide a way to compute H_{CP} as described in [Beardsley et al. 2004] and the remaining processing is the same.

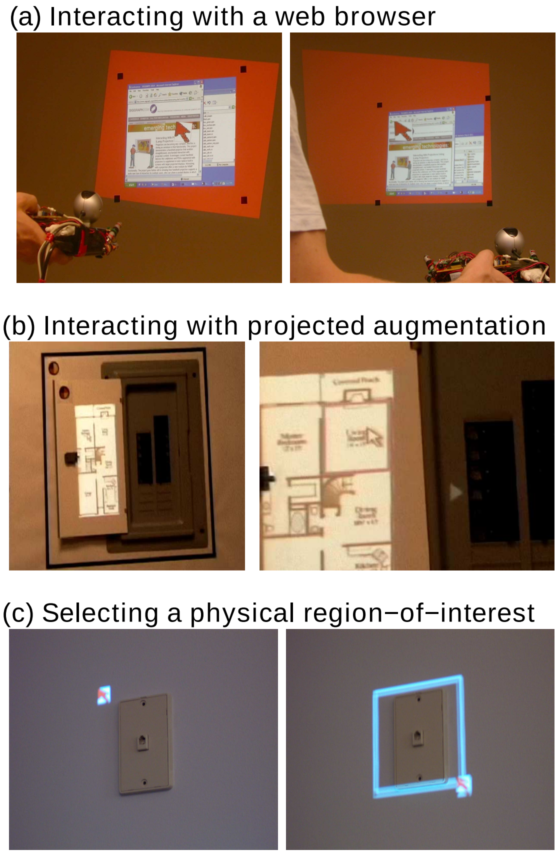
2.2 Interactive Projection

At each time-step the projector image plane is updated to achieve a static projection on the display surface, as described in Section 2.1. In conjunction with this update, the center pixels of the projector image plane are overwritten with a graphic for the cursor.

The effect on the projector image plane is that the main projected content is continually being updated, to factor out projector motion, while the cursor graphic is fixed. The effect on the display surface is that the main projected content appears static, while the cursor moves across the surface in direct correspondence with projector motion.

3 Applications

This section describes three example applications, one for each of the main classes of application proposed in Section 1.2. See Figure 4.



(a) Interacting with a web browser

(b) Interacting with projected augmentation

(c) Selecting a physical region−of−interest

Figure 4: **TOP:** The user interacts with a projected web browser. The browser remains static on the surface during the interaction, while the cursor moves in correspondence with the user's pointing motion. The outer red area indicates the full projector image, and has been made distinctive for the purpose of illustrating program operation. **MIDDLE**: Augmentation of a fuse box. **Middle-left:** The projection at left presents the user with a map of rooms served by the fuse-box. The dark vertical strips at right are the fuses. **Middle-right:** A close-up of the projection. The user has positioned the cursor on one of the rooms in the projected map, and performed a button-click to invoke information. In response, a projected arrow indicates the fuse for that room. **BOTTOM:** Defining a physical region-of-interest, preparatory to doing computer vision on the selected area. **Bottom-left:** the projected cursor is positioned upper-left of the area and the user starts a mouse hold-and-drag operation. **Bottom-right:** the defined rectangle. Note that all the selection operations that are familiar from editing and drawing packages can be transposed to interactive projection, to make a physical selection.

3.1 Projected Web Browsing

The first application is a familiar desktop application running under handheld projection. We take an ordinary desktop windows environment and modify it so that the display goes to the projector, and so that mouse input is taken from the click-buttons on the handheld device. Figure 4a shows a user interacting with a web-browser including clicking links, scrolling, and selecting menu items.

3.2 Projected Augmented Reality

Projected augmented reality can support various applications, and we consider task guidance on a control panel as an example. We envisage a user maybe interleaving handheld projection with opportunistic placement of the projector on a small tripod around the workspace. This work is likely to involve different interactions - for example, pointing to request the identity of a specific component, or navigating through a series of instructions plus projected overlays for a particular task. Figure 4b shows an example of projected augmentation on a fuse-box - a projected map of the environment served by the fuse box is shown to one side, the user employs interactive projection to select a location on the map, and the appropriate fuse is highlighted.

The processing is similar to that in Section 3.1, but the four fiducials are replaced in this case by a black rectangular border. The colored circles visible at top-left in Figure 4b-left encode identity, and are used to automatically index the correct augmentation for the object.

Projected augmented reality has some advantages over augmentation via a handheld-screen. It avoids a context-switch between the physical object and screen. It avoids resolution problems if a complex scene is being rendered on a small screen. Temporary, opportunistic placement of the projector allows the user to operate around the workspace without holding a handheld device, while still having the augmentation immediately available. And when required, projection is the most natural approach for shared multi-person viewing of augmentation.

3.3 Selecting a Physical Region-of-Interest

Interactive projection allows the user to define an arbitrarily-shaped region-of-interest (ROI) on a physical surface in the world, while being presented with visual feedback about the selection. For example, the user can do a mouse-button hold-and-drag to define a rectangular ROI as in Figure 4c, utilizing familar interactions from the desktop.

Unlike the examples in the previous two sections, there may be no fiducials when interacting with arbitrary parts of a scene in this way. But if the area is sufficiently textured, then the texture will stand in for the fiducials - it serves to define a homography between successive camera images and hence place all successive cursor positions in a single camera coordinate frame. Knowing the cursor track relative to the texture in this camera coordinate frame, we can transform to projector image coordinates as in Section 2.1, and then present the user with appropriate visual feedback about the selected ROI.

One use of physical ROIs is to allow the user to outline a specific area to be processed by computer vision. An example given in the introduction was selecting foreign words from signage, and then invoking OCR and translation. As a second example, this section describes a straightforward way to attach and retrieve electronic notes at arbitrary (textured) locations in an environment. Assume a user wishes to use the handheld projector to attach an e-note to, say, a light switch. The intention is that on a subsequent occasion, the user directs the handheld projector at the light switch, selects it, and is automatically presented with a projection of the e-note.

• The user directs the cursor at the light switch, and defines a ROI around it.

• The camera image data for the ROI – a ‘texture key’ - is stored along with the e-note which is to be stored.

• On a subsequent visit, the user again selects a ROI around the light switch.

• The image data for the new ROI is matched against all stored texture keys by template-matching, and the e-note associated with the best match is projected onto the surface.

The functionality in this application can be achieved, as with other applications in this paper, without using projection - a handheld device with camera and fixed display could capture an image of the required part of a scene, the user could employ a stylus to indicate the required area of texture on the handheld display, and the e-note is attached and retrieved as before. But the interaction is significantly less direct, bringing in a context-switch from physical surface to the screen. In contrast, direct interaction with physical texture by pointing is much more immediate and minimal.

4 Why Projection?

Projection is not suited for all applications. For example, visibility may degrade under some ambient lighting conditions. But handheld projection also has some unique strengths, and the question is whether there is a class of applications for which it could prove the most appropriate technology. Our work is an investigation of mobile, opportunistic projection and we have described some of its advantages. Firstly, handheld projection has the potential to make many everyday surfaces into displays, a vision of the ‘world as the desktop’. Device size is not linked to display size, making projectors a good technology for a portable handheld display. Furthermore, projection has some specific advantages when making augmented reality overlays – it avoids a context-switch between fixed display and physical scene, and avoids resolution problems when viewing or interacting with a scene via a small handheld display. We also showed in Section 3.3 how interactive projection can provide a direct and elegant interaction with physical features.

5 Conclusion

Handheld projectors have the potential to extend the way we think about display, moving beyond fixed screens to allow opportunistic display on everyday surfaces around us. In this context, this paper has described a new technique for interacting with projected data which is essential if handheld projection is to be truly useful. To illustrate the versatility of the idea, we described three applications – projected web browsing, projected augmented reality, and selecting physical regions-of-interest. The latter is of note because it indicates how handheld projection can support innovative new techniques. While there are other ways to achieve the same functionality, interactive projection provides a very natural and minimal interaction, suggesting its potential in other applications that mix physical texture and digital data.

References

BEARDSLEY, P., VANBAAR, J., AND RASKAR, R. 2004. Augmenting a Projector-Camera Device with Laser Pointers. TR 2004/035, MERL.

OLSEN, JR., D. R., AND NIELSEN, T. 2001. Laser pointer interaction. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, ACM Press, 17–22.

RASKAR, R., AND BEARDSLEY, P. 2001. .A Self-Correcting Projector. In *IEEE Comp. Soc. Conference on Computer Vision and Pattern Recognition (CVPR’01)*.

SUKTHANKAR, R., STOCKTON, R., AND MULLIN, M. 2000. Automatic keystone correction for camera-assisted presentation interfaces. In *Proc. Intl. Conf. Multimedia Interfaces ’00*.

WWW.CANESTA.COM, 2004. Canesta projection keyboard.

WWW.SIEMENS.COM, 2002. Siemens Mini-Beamer.