PICOntrol: Using a Handheld Projector for Direct Control of Physical Devices through Visible Light

Dominik Schmidt[1,2], David Molyneaux[3,2], Xiang Cao[1]

[1]Microsoft Research Asia
Beijing, China
xiangc@microsoft.com

[2]Lancaster University
Bailrigg, Lancaster, UK
schmidtd@comp.lancs.ac.uk

[3]Microsoft Corporation
Redmond, WA, USA
davmo@microsoft.com

ABSTRACT

Today's environments are populated with a growing number of electric devices which come in diverse form factors and provide a plethora of functions. However, rich interaction with these devices can become challenging if they need be controlled from a distance, or are too small to accommodate user interfaces on their own. In this work, we explore *PICOntrol*, a new approach using an off-the-shelf handheld pico projector for direct control of physical devices through visible light. The projected image serves a dual purpose by simultaneously presenting a visible interface to the user, and transmitting embedded control information to inexpensive sensor units integrated with the devices. To use PICOntrol, the user points the handheld projector at a target device, overlays a projected user interface on its sensor unit, and performs various GUI-style or gestural interactions. PICOntrol enables direct, visible, and rich interactions with various physical devices without requiring central infrastructure. We present our prototype implementation as well as explorations of its interaction space through various application examples.

Author Keywords: Handheld projector; physical devices; visible light communication

ACM Classification Keywords: H.5.2 [Information interfaces and presentation]: User Interfaces. – Input devices and strategies

General terms: Design; Human Factors

INTRODUCTION

In everyday environments, we are increasingly surrounded by a growing number of electric devices. At the same time, many of these devices are shrinking in physical size. Both developments come along with their own challenges when it comes to controlling these devices. On the one hand, a plethora of devices imply that not all of them will be in physical reach when needed. Therefore, many people may prefer control at a distance for convenience. On the other hand, small devices are less able to accommodate control interfaces on their surfaces. In fact, user interface hardware is a major limitation to device miniaturization [19].

A common solution to these challenges is to decouple and externalize the user interface onto a separate control device. This not only allows for control at a distance, but may also provide access to a larger set of functions than is accommodated or desired through an on-device interface. For example, current remote controls for home entertainment systems often provide far more functions than are accessible directly on the device itself. Despite this appeal for a decoupled control interface, it also has several drawbacks. Above all, the interaction is no longer *direct*. Users have to divide and switch their attention between the controlling and the controlled. Further, if the user wishes to use a single control device to operate multiple devices in the environment, an explicit selection needs to be made first to indicate the intended target. This becomes cumbersome if the candidate list is long, and requires mental load to map abstract selection to spatially distributed physical targets. Finally, the user is often limited by the size and (lack of) display capability of the control device in hand, which in turn constrains the interaction richness.

Driven by these challenges, we explore *PICOntrol*, a new interaction approach that employs an off-the-shelf pico projector as the handheld control device for users to directly operate various physical devices in the environment. Figure 1 presents a simple example. To use PICOntrol, the user points the handheld projector directly at the device to be controlled (implicit and intuitive selection), and casts a projected graphical user interface (enlarged display capability) directly over a photo sensor unit integrated with the device (no divided attention) to perform a variety of GUI-style or gestural operations (enriched interactions) in order to control the device.

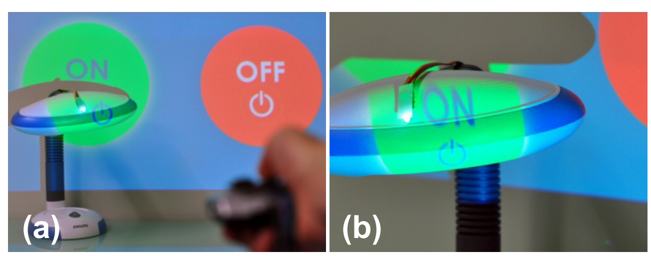

Figure 1. A simple example of PICOntrol interaction. (a) Pointing the handheld projector at a lamp. (b) Casting the projected button "ON" over the sensor unit and pressing an activation pushbutton on the projector turns on the lamp.

PICOntrol is made possible by having the projection simultaneously serve a dual purpose, both for presenting the visual interface to the user, and for transmitting control information to the sensor unit by embedding binary codes in the visible light. In doing so, PICOntrol removes the need for a dedicated channel or central infrastructure for communication. This also results in PICOntrol's technical simplicity. Both the physical size and the hardware complexity of the sensor units are minimal, making them easy to integrate into various physical devices. Based on the prediction that miniaturized projectors will become widely integrated in handheld devices such as mobile phones [17], and since PICOntrol does not require modification or augmentation on the projector, we anticipate future users may use PICOntrol spontaneously without carrying additional devices.

RELATED WORK

Direct Control of Physical Devices

Infrared (IR), radio frequency (RF), or WiFi based remote controls are becoming more widely used to operate various physical devices in the environment. However, given the weak or lack of directionality of the control signal (in the case of IR, the IR light is intentionally diffused, otherwise it is nearly impossible to aim the invisible single light beam to the receiver), it becomes necessary to explicitly select the intended device to control. Currently this is either done by physically choosing from multiple dedicated controllers, or in the case of "universal remote controls", the target device is selected through an abstract hardware or software user interface (e.g. button or touch screen). In addition, the user's attention is divided between the control interface and the device to control. Such an indirect control mechanism may cause both inconvenience and inefficiency.

Motivated by similar goals as us, researchers have explored direct pointing interaction with physical objects using a laser pointer. The laser dot may be detected directly by a photo sensor (e.g. [3, 23]). However, hand tremor makes precise pointing at distant targets difficult [18]. This has caused several systems to increase their sensor size, requiring additional space on the target device [16, 23]. Further, as only a single dot is visible, the richness of direct-pointing interaction using a laser pointer is rather limited. To enrich the interaction space, researchers have explored using a situated camera to track the laser dot position in space, supporting gestural input for devices (e.g. to guide home robots to perform various tasks [12, 13]). Alternatively, as a middle ground between direct-pointing and indirect control, Ringwald [23] uses a laser pointer to select a device, and a handheld device (PDA) as the user interface to perform richer but indirect control. In comparison, PICOntrol not only provides a much richer projected user interface, but also alleviates the hand tremor problem since the user only needs to overlay a relatively large user interface component on the sensor unit rather than aligning a dot to it.

More comprehensive approaches, which aim at supporting both direct and rich interaction, generally rely on a substantial amount of infrastructure. For example, XWand [1] enables natural interaction in smart environments through pointing, gesture, and speech input. The wand's location is tracked by two situated cameras, and each device to be controlled is first represented as a 3D model. WorldCursor [28] avoids cameras, but adds a ceiling-mounted steerable laser pointer to provide a real-world cursor controlled by the wand's motion. Light Widgets [9] uses a set of cameras tracking the users' hands to turn arbitrary surfaces into interactive widgets for controlling physical devices. Further work investigates using photos (e.g. [25]) or augmented live video (e.g. [5]) of the devices as a proxy to "directly" interact with them from a distance. In addition to external tracking of the handheld control device or prior knowledge of the environment or both, all of the above also require a central communication infrastructure to which each device is networked. Neither is needed for PICOntrol, as it directly communicates with the device through visible light in a peer-to-peer fashion, and any motion sensing involved is achieved through the sensor units themselves and relative to the device to be controlled (to be detailed later). In addition, with the user interface projected on the device to be controlled, the directness of the interaction is further increased compared to other solutions.

Handheld Projector Interaction

Using handheld projectors as a novel interaction device has attracted interest of many researchers. However, the majority of work concerns presenting and interacting with digital content, such as for exploring virtual information spaces using a flashlight metaphor [6, 21], supporting GUI-style software applications [2], or augmenting physical objects with digital content [2, 6, 22]. Unlike PICOntrol, all these systems are based on the principle that the projected digital information should remain geometrically static to the physical world, thus relying on constant active stabilization and rectification of the projected image. This requires either an external sensing infrastructure or an onboard camera system to track the projector and compensate for its movement, as well as calibration of the projector and the environment. In contrast, MotionBeam [27] and SideBySide [26] employ an alternative interaction style, not attempting to mitigate projector movement, but instead using it to directly move a digital game character that follows the projection.

In general, many of these works [6, 22] attempt changing the projected information in response to the physical context, but few have explored the opposite direction: controlling the physical world with handheld projection. MotionBeam [27] provides one simple example. Another exception is CoGAME [11] which adopts handheld projectors to manipulate mobile robots from a distance, using a central communication and external tracking infrastructure for both the projection and robots. PICOntrol, in comparison, does not rely on tracking the projector or stabilizing the image, therefore does not require any central infrastructure or calibration.

Encoded Projection

Several researchers have explored transmitting code via visible projection to be received by photo sensors. For example, such code may enable localizing of photo sensors for automatic projection calibration [14], tracking movable surfaces [15, 24], or transmitting payload information such as audio stream [20]. Many of these works rely on Digital Light Processing (DLP) projectors (some with custom hardware modification [15]), given that their fast-flipping mirrors lend themselves to encoding the projected light. Nii et al., in contrast, demonstrate using general luminance variations to embed information using an arbitrary projector [20]. Perhaps the most related to our work is RFIG Lamps [22], which uses dedicated Gray code images that are projected by a handheld projector equipped with RF communication and an onboard camera. This is for the handheld device to recover the 3D locations of active RF tags augmented with photo sensors, and in turn the geometry of surfaces or objects they are embedded in. Like many other handheld projector works, this knowledge was mainly used for subsequently projecting augmenting information to the physical environment, but without the intention to physically influence it. In contrast, PICOntrol addresses a distinctive set of design and technical challenges to support rich interactions to directly control physical devices. By embedding both command and location codes into projected user interfaces, the projection is both to be interpreted by the device being controlled, and at the same time to be viewed and operated by the user.

PROTOTYPE IMPLEMENTATION

PICOntrol consists of two components, the handheld projector as the control device, and the sensor units to be integrated with physical devices to be controlled (Figure 2). The image projected by the handheld projector serves a dual purpose: it presents a visible interface to the user and at the same time embeds codes to be detected by the sensor unit. An off-the-shelf pico projector and inexpensive photo sensors are used to implement PICOntrol. As communication takes place in a peer-to-peer fashion, PICOntrol does not require central infrastructure (e.g. wireless networks); and as selecting the device to control happens implicitly by projecting on it (thus communicating to it), PICOntrol does not need prior knowledge of the layout of physical devices in the environment. Unlike other handheld projector systems, we do not apply image stabilization or rectification, thus do not require tracking or calibrating the projector. In fact, we explicitly use the changing location of the projected user interface for interaction.

Consistent with most of today's remote controls, communication in PICOntrol currently takes place in a unidirectional fashion from the controller to the controlled device. This simplicity allows us to maintain a minimalist system design, not requiring any augmentation on the handheld projector (thus preserving a wider applicability), while still supporting a rich interaction space. Nonetheless, the addition of a similarly simple and peer-to-peer backward communication channel from the sensor unit to the projector (e.g. using similar encoding schemes with the LED on the sensor unit and adding a single photo sensor on the projector) could certainly provide further functionalities, such as automatically selecting a specialized user interface for the device or displaying its internal states to the user using the projector, which we leave for future explorations.

Handheld Projector

We use an unmodified off-the-shelf Microvision SHOWWX pico laser projector to project the user interface. Our encoding and transmission scheme, however, is not reliant on a certain projection technology, and can be generalized to all pico projectors. In our prototype, the projector is connected to a laptop which generates the required image data at 60 Hz. Further, we added a set of simple pushbuttons to the top of the projector (Figure 2a). The buttons are connected via an Arduino nano microcontroller to the same laptop and are used for basic input (i.e. one button to activate control, and two buttons for cycling through different projected user interfaces). In real world usage, however, we envision the projector to be integrated with an existing handheld device (e.g. a mobile phone, as early models already becoming available in the market), and the user can use the input mechanism available on the handheld device in place of the pushbuttons.

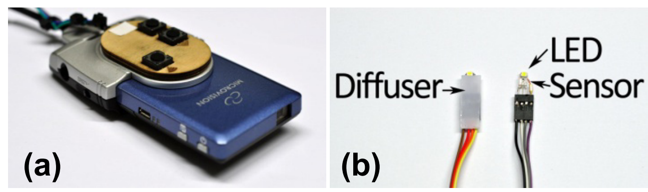

Figure 2. PICOntrol prototype. (a) Pico projector with pushbuttons for basic input. (b) Two sensor units, with and without diffuser attached.

Sensor Unit

A sensor unit (Figure 2b) includes an LED to indicate that a projection signal is being detected, and one or several photo sensors to decode signals (the actual number depends on the target application, as detailed later). Each TAOS TSL13S-LF light-to-voltage photo sensor is connected via a resistor-capacitor (RC) circuit to the same Arduino nano 328 microcontroller, using its analog input pins. The RC circuit acts as low-pass filter to smooth the projected signal and facilitates detection; it comprises an 8 kOhm resistor and a 0.01 μF capacitor. An acrylic diffuser is attached in front of the sensor for it to robustly detect projections from all angles and without requiring accurate pixel alignment. The photo sensor(s) and the LED are the only parts that are visible to the user (costing under $2), while we envision the RC circuit and the microcontroller to be embedded inside the controlled device in the future. Therefore, the required space on the outside of a device is minimal (5 mm^2 for the sensor plus an additional 3 mm^2 for the LED in a single-sensor setup). Moreover, as suggested by Dietz et al. [8], existing LED on the device (such as power indicators) could potentially double as photo sensors by alternating them between emitting and detecting light, further decreasing the space requirement.

Code Transmission

The handheld projector transmits codes to a sensor unit by systematically changing the brightness of pixels in a projected user interface over time. As pixels can be changed independently, multiple arbitrary codes can be transmitted concurrently using different projection regions. However, a sensor unit only receives the codes contained in the projection user interface region directly over it.

Encoding Scheme

We support two types of temporal binary codes: command codes and location codes. A command code encodes an individual command, represented by a discrete region within the projected user interface, such as on or off buttons. In contrast, location codes identify the location of the sensor unit within the projected image, for example to input continuous values when adjusting a virtual slider, or to perform a gesture. Similar to [14, 15, 21], we use Gray codes to encode locations. Each frame consists of bright and dark stripes, subdividing the projection area successively. We first encode horizontal positions, followed by vertical positions, which are projected in a sequence to uniquely identify locations in a grid (Figure 3).

Figure 3. Gray code patterns, encoding an 8 x 8 grid in six consecutive frames.

Both types of codes are transmitted on demand by pressing the activation pushbutton mounted on top of the projector. The same code is transmitted repeatedly while the activation pushbutton is being pressed. To transmit any binary code via the visible light channel, the brightness of pixels is varied accordingly over time. We represent 0-bits by low and 1-bits by high intensities respectively. In doing so, the absolute sensed intensity is not critical, as long as the contrast between low and high is sufficiently large to be detectable. In addition, our mapping ensures that different interface colors yield consistent intensity readings to the hue-insensitive photo sensor. Therefore, we can vary the brightness value of each pixel to represent the binary code while preserving its hue value, and transmit codes at any point in the projected image despite the various user interfaces displayed.

Embedded Synchronization and Calibration Signal

PICOntrol does not require an additional channel for synchronization, but instead embeds this information into the visible light channel used for data transmission. In particular, we project a single frame of black pixels within the code transmission regions, at the beginning and end of each code word to surround and hence delimit it, allowing for variable code lengths. Black image frames are particularly easy to detect with a laser scanning line projectors such as the one we use, as no light is transmitted during them. Minimal backlight during black frames with other projection technologies can also be accounted for by adjusting the detection thresholds accordingly. Such a single black frame (i.e. 16.7 ms) is further highly unlikely to be confused with other conditions (e.g. if the projection was moved away or covered temporarily, this would result in a longer interval of missing signal), and thus serves as reliable indicator to distinguish between normal projection and the start/end of code transmission.

While using PICOntrol, distance and angle between the projector and the sensor unit naturally vary as users are controlling devices from different locations. The sensors thus may receive different light intensities for the same projected brightness, depending on how the projector and the sensor unit are located towards each other. To ensure correct mapping of the sensed light intensity to the intended bit value, we additionally include a short calibration sequence after the synchronization frame, but before transmitting the actual code word. This proved to be more robust than solely relying on brightness contrast. The calibration sequence consists of one 0-bit and one 1-bit to set reference intensity values for the code word that follows. In addition, these two bits also indicate the type of the code being transmitted (i.e. 0-1 for location code, and 1-0 for command code). Figure 4 illustrates the complete transmission sequence for a single code, consisting of header (synchronization and calibration), actual code word, and tail (synchronization). The time that the sensor unit requires to receive and decode a code varies depending on the code type and length, resulting in the system response time ranging between 200 and 400 ms.

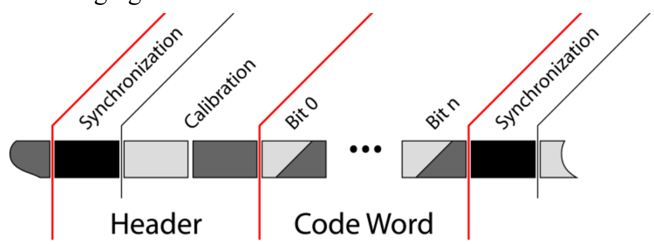

Figure 4. Code transmission sequence.

Rapidly changing the pixel brightness for code transmission inherently results in image flicker. However, this perceivable flickering only happens when the user presses the button, and is further restricted to regions of interest in the user interface. It incidentally serves as a direct visual feedback to the user, indicating an operation is being performed. Further, to ensure that the projected user interfaces are still recognizable by the user during code transmission, they are rendered based on hue contrast rather than brightness contrast. This results in color differences perceivable by human eyes, but the same intensity detected by the photo sensor. During our informal user trials with four participants (one hour each), none complained about the flicker.

Signal Detection

Although the encoding, transmission, and decoding scheme is generic, the way that the sensor unit physically detects the visible light signal may slightly vary between projection technologies, but is generally straightforward. For example, detecting signals from DLP projectors using photo sensors has been demonstrated (e.g. [14]). In our prototype we use a Microvision SHOWWX laser projector, which employs a scan-line-based approach. Three modulated laser sources (red, green, and blue) are combined into a single beam which is then reflected off a scanning mirror to construct each image pixel sequentially in a zigzag fashion. Thus the sensor unit overlaid by the projected image only receives projected light at the time when the laser beam passes over it, and an image frame is detected as a peak in the sensed light intensity. By maintaining the average intensity over a recent time window of 3 μs, we remove the influence of varying ambient lighting on the peak detection. In our prototype implementation, the sensor unit can detect the projector's signal from a distance of up to 4 m.

SENSOR UNIT DESIGNS AND INTERACTIONS

Here, we discuss three distinct sensor unit designs based on the above hardware and code transmission method. These include a single-sensor unit, a dual-sensor unit, and a color-sensor unit (Figure 5). The varying number of sensors enables different sets of interactions in a range of application scenarios. The single- and dual-sensor units, which we focus on in this paper, capture light intensity only. The single-sensor unit detects discrete interface components as well as location and lateral motion within the projection. The dual-sensor unit further enables detection of in-plane rotation and distance between the projector and the unit. Finally, the color-sensor unit enables further interaction possibilities based on color. All sensor units feature an LED that indicates detection of projection, and also facilitates the user to further position the projected interface.

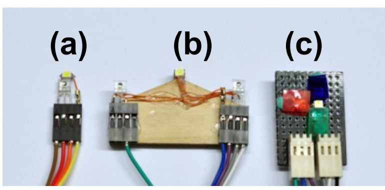

Figure 5. Sensor unit designs (shown without diffuser): (a) Single-sensor unit. (b) Dual-sensor unit. (c) Color-sensor unit.

Interaction Style

PICOntrol interactions share a common style**.** The projected image comprises one or several active regions which represent interface components, such as buttons or sliders, or an area for gestural input. An interaction is prepared by aiming the projection at the target device and overlaying the interface component in question on the sensor unit. Choosing the device and the function are hence achieved in a single continuous step. To trigger the interaction, the user presses the projector-mounted activation pushbutton. While the button is pressed, all active regions in the image keep transmitting their respective codes simultaneously, but only the code that is directly over the sensor unit will be received. Depending on the application, this may conclude the interaction (*static interaction*, e.g. to execute a command), or may be followed by moving the projector while keeping the activation pushbutton pressed (*motion-based interaction*, e.g. for adjusting a value using a slider or a knob, or for gestural input). In some cases, the active region may cover the entire image to enable larger scale gesture detection by tracking the sensor unit's location across the whole projection area.

Common user interface components (e.g. on-off switches, or sliders and knobs for adjusting a single parameter such as volume) could be made universally applicable to various devices and be interpreted accordingly by the respective device. Therefore, many typical projected interfaces can be reused across various devices with the same layout and codes. Some other functionality may be device-specific, requiring a particular control interface. In this case, device identification headers can be added to the transmitted code to avoid duplicate codes across devices. In our current implementation, the user can cycle through the available set of control interfaces using the two other pushbuttons on the projector. In future applications where the projector is integrated in a smart handheld device, we envision the user may use its touch screen to select or customize a specialized user interface to project. Finally, as we discussed, in future PICOntrol explorations which may feature a backward communication channel, the projector may identify the device it is pointing at and automatically project a specialized user interface, thus eliminating the need for user selection altogether.

Single-Sensor Unit

This unit uses a single sensor to capture projected light intensities (Figure 5a). It decodes the information embedded within the active projection region directly over the sensor to enable the interactions detailed in the following. Both static (using command codes) and motion-based interactions (using location codes) are supported. In both cases, the actions can be thought of as an inverse cursor interaction metaphor: instead of moving a cursor to indicate the location in a static user interface as in conventional GUI, here the sensor unit representing the "cursor" is fixed and the projected user interface as a whole is being moved. This bears some similarity to the Toolglass interface [4] and allows for general two-dimensional input equivalent to conventional GUI.

For the sake of simplicity, we first chose a basic lamp scenario to illustrate both kinds of interaction. In doing so, we attached a single-sensor unit to an off-the-shelf desk lamp. For easy prototyping purposes only, the sensor unit controls the lamp via a digitally switchable power outlet to simulate a direct integration with the lamp. With multiple such lamps, the user can use the same projected interface to individually control each of them simply by pointing at it.

Static Interaction

Figure 1 provides an example of static interaction using a basic interface designed to switch a lamp on or off with two dedicated buttons. Here the user overlays one of the projected buttons on the sensor unit mounted on the lamp's shade, and then presses the activation pushbutton on the projector. While different command codes are transmitted within the two buttons at the same time, the sensor unit only receives the command from the button directly over it. Feedback is provided directly in situ by the lamp's lighting state, without the user's attention being divided between the control interface and the target device.

Motion-Based Interaction

Motion-based control enables a range of different interactions. As shown in Figure 6a and 6b, the sensor unit's location can be used to implement crossing-based interfaces [1]. This further mitigates the difficulty of precise freehand pointing and involuntary movement introduced by pressing the activation pushbutton. Here, the user first overlays the sensor unit with either half of the interface, and then horizontally swipes the projection while holding down the activation pushbutton. Upon crossing the center line, the lamp is switched on or off. This interface uses the entire projection area as an active region by filling it with location codes when the activation pushbutton is being pressed.

Projector motion can be further mapped to set continuous values, such as a lamp's brightness. Figure 6c shows a visual slider, where the region of the slider is filled with location codes when the activation pushbutton is being pressed. Slider positions can be mapped to the brightness values of the lamp. In this case, the user can either drag the slider to adjust values continuously, or directly "click" on one position to set the value. Alternatively, the relative motion of the slider can map to the relative change of brightness to achieve a finer control granularity. In this case, the user can perform a clutching action by releasing the activation pushbutton and drag again.

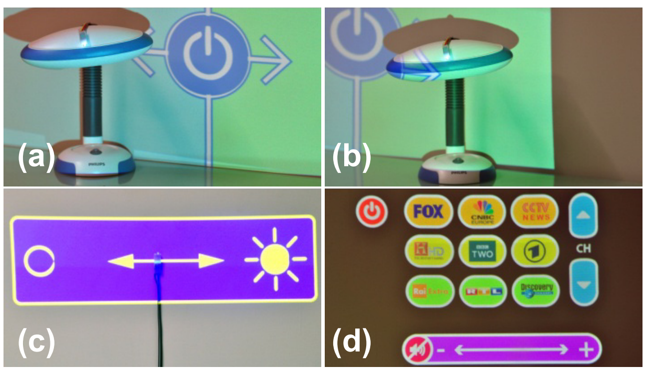

Figure 6. (a, b) A crossing-based switch. (c) Slider for adjusting continuous values. (d) Mixing interface components for controlling a TV.

PICOntrol also supports arbitrary two-dimensional gestures by projecting a location code pattern that spans over the entire image. Again, this pattern is only projected on demand, as long as the user holds down the activation pushbutton. Hence each gesture is demarcated by a pair of button press-release-actions, avoiding random movement to be interpreted as input. To demonstrate this, we simulate a digital photo frame on a standard tablet PC which receives instructions from the sensor unit control circuit via USB (Figure 7). Here, the user can navigate through galleries and photos by aiming the projection at the frame, holding down the activation pushbutton, and flicking to the left or right for going backwards or forwards respectively. A downward flick or an upward flick switches between the photo view and gallery view respectively.

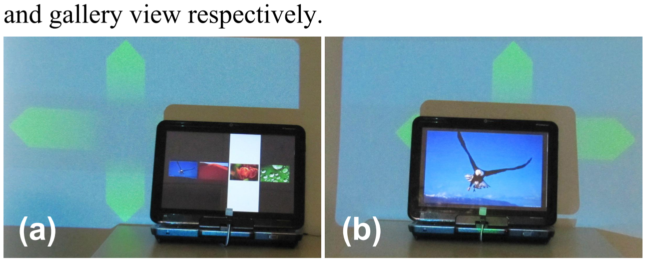

Figure 7. Navigating a digital photo frame with gestures (indicated by arrows) in (a) gallery view (b) photo view.

Obviously, static and motion-based interface components may be mixed to create more complex projected interfaces. Figure 6d illustrates a TV control interface that mixes buttons (static, for selecting channel etc.) and a slider (motion-based, for volume adjustment).

Multi-Device Interaction

As target devices are implicitly selected through pointing, no explicit or abstract device selection step is required. In addition to using the same projected interface to control multiple devices individually, users may also opportunistically combine interactions with multiple devices into one. For example, using the crossing-based switch in Figure 6a, users may hold down the activation pushbutton and swipe the projection over multiple devices to control them all in a single motion. This somewhat resembles the real-world interaction of throwing multiple neighboring light switches at once with a whole hand. In another example, two single-sensor units may be used to control the volume of two neighboring audio speakers respectively, each playing a different audio channel. Instead of using a single slider to control each individually, the user may also use a projection with two parallel vertical sliders to control both sensor units simultaneously. This effectively turns the two independent interface controls into one "macro" control and enables higher-level intuitive operations. Moving the projection up and down changes the overall volume of both channels, while rotating the projection adjusts the balance between the two.

Interaction with Small Devices

PICOntrol interfaces can also provide visible and expressive input for devices too small to accommodate a user interface of their own. To illustrate this, we built a minimalistic music player based on Arduino, which does not have any on-device input mechanisms, but can be entirely operated by PICOntrol using a simple interface echoing the iPod shuffle controls (Figure 8a).

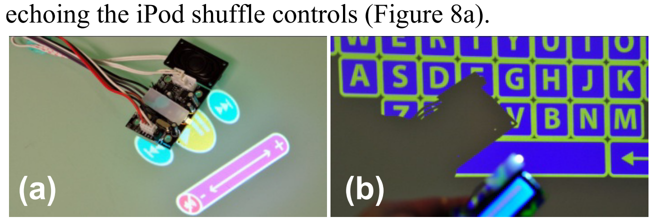

Figure 8. (a) Operating a minimalistic music player. (b) Using the shadow of a device itself as a pointer to enter text.

Similarly, PICOntrol can be used for text input on devices too small to have their own keyboard by projecting a virtual keyboard over it. Following the basic interaction described before, users may input text by moving the projection to overlay the desired character on the sensor unit and then press the activation pushbutton. However, note this motion is relative: the user may instead project the interface statically on a nearby surface and move the device itself to the desired letter, thereby effectively using it as a pointing device for direct interaction with the projection. Alternatively, the user may hold the device between the projector and the surface, hence casting a shadow [cf. 7] on the projected key (Figure 8b). In all cases, when the user presses the activation pushbutton, the respective key input will be transmitted through the sensor unit to the device.

Dual-Sensor Unit

The dual-sensor unit (Figure 5b) extends the previous design by adding a second sensor of the same kind. Both sensors are mounted in the same plane (at a distance of approximately 4 cm in our current prototype, which showed to be a reasonable trade-off between unit size and sensing resolution, to be discussed later) and are connected to the same control circuit. Using location codes as described before, the dual-sensor unit detects two separate locations within the projection simultaneously, one from each of its sensors. Bearing some conceptual similarity to how Minput [10] uses two mouse sensors on the back of a small device to sense its lateral and rotational motion, our dual-sensor unit enables sensing of additional degrees of freedom of the projection (rotation and distance), and extended gestural interaction.

Rotation

The two locations in the projection coordinate space (determined using location code, i.e. Gray codes), in conjunction with the known physical layout of the sensors on the dual-sensor unit, allow for recovering the in-plane (i.e. along the optical axis of the projector) rotation angle between the projector and the sensor unit by the following formula, where (x_1, y_1) stand for the detected location of sensor 1, and (x_2, y_2) for that of sensor 2 respectively:

$$\alpha = atan2(y_2 - y_1, x_2 - x_1)$$

Figure 9 illustrates this. In general, the maximal angular resolution is determined by the detected distance of the two sensors in the Gray code coordinate system, which is in turn dependent on the physical distance between the two sensors (larger distance results in higher angular resolution), the grid density of the projected Gray code (denser grid results in higher angular resolution), and the projection distance between the projector and the sensor unit (smaller distance results in higher angular resolution). In the worst case, if both sensors fall within the same grid cell and report the same location, the rotation angle cannot be determined. While the first two parameters (sensor distance and grid density) are more or less fixed in the system design, and subject to compromise with other design variables (e.g. sensor unit size and code length), the projection distance can be easily changed by the user on the fly, so that the user can choose the best tradeoff point between resolution and convenience that serves them best at the moment.

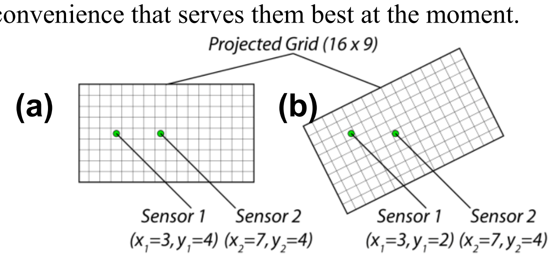

Figure 9. Detecting rotation on a dual-sensor unit. (a) Detected rotation: 0°. (b) Detected rotation: 27°.

Distance

Similarly, the distance between the projector and the sensor unit can be inferred. As shown in Figure 10, moving apart the projector and the sensor unit results in a larger projection, therefore the detected distance between the two sensors with respect to the Gray code coordinates decreases (and vice versa). This allows the dual-sensor unit to detect relative change in projection distance. In addition, if the projector's *ThrowRatio* (i.e. the ratio between the projection distance and the physical width W_{Proj} of the projected image) is known, the absolute physical distance between the projector and the sensor unit, D_{proj}, can also be calculated as:

$$D_{proj} = ThrowRatio \times W_{Proj}$$

where $W_{proj} = \frac{ResolutionX_{code} \times D_{sensors}}{D_{code}}$

$D_{sensors}$ refers to the physical distance between the two sensors, and $ResolutionX_{code}$ refers to the number of Gray code grid cells along the projection's X axis, both of which are fixed. $\boldsymbol{D_{code}} = \sqrt{(\boldsymbol{x_1} - \boldsymbol{x_2})^2 + (\boldsymbol{y_1} - \boldsymbol{y_2})^2}$ refers to the detected sensor distance in Gray code coordinates, from which we can calculate W_{proj} (physical projection width) and in turn D_{proj}. Here we assume the Gray code grid cells are square, however the formula can easily be adapted for non-square cells. The maximal detectable projection distance is reached once both sensors fall within the same Gray code grid cell (i.e. report the same location).

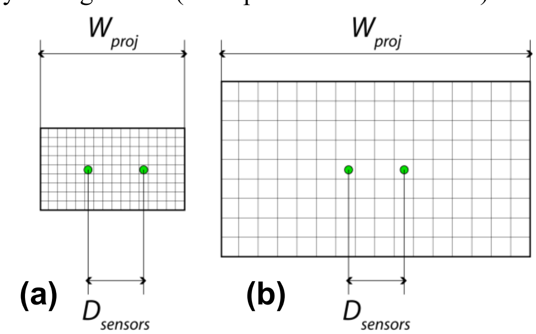

Figure 10. Detecting projection distance on a dual-sensor unit. (a) D_{code}=7. (b) Moving the projector further away, D_{code} decreases to 4.

Note rotation and distance is detected without internal sensors on the projector (e.g. accelerometers/gyros, or camera and RF transponder as used in RFIG lamps [22]) or external tracking infrastructure. Further, as we directly sense the relative position and orientation between the projector and the sensor unit, all projector interactions based on them are also relative to the sensor unit, agnostic of its absolute position and orientation (e.g. it could be mounted on the ceiling facing downwards). This inherently matches the point-and-interact paradigm of PICOntrol and allows the deployment of the sensor units to be entirely flexible. The sensor unit or the device it is connected to may be freely repositioned by the user, or itself may even move during the interaction, as demonstrated in later examples. To achieve the same with absolute sensing such as with internal sensors or external tracker, it would require continuously tracking the positions and orientations of both the projector and the controlled device as well as real-time coordinate transform between the two, a much more complex system to maintain.

The rotation and distance detection algorithm of the dual-sensor unit assumes that the projector's optical axis is approximately perpendicular to the sensor plane. Projecting at an angle may cause less accurate detection results. However, for interaction purposes such inaccuracies are unlikely large enough to be detrimental, as the obliquity is usually reasonably small thanks to the user trying to project a legible user interface. We could potentially create a three-sensor unit capable of sensing full 3D rotation and translation between the projector and the unit, however we suspect interactions based on them may become too complex and subtle to be useful for controlling everyday devices.

Example Applications

We now demonstrate two example applications using a dual-sensor unit. Similar to the linear slider input used with single-sensor units, rotation can also be used to input continuous values. For an intuitive example, to set the hour or minute of an analog clock equipped with a dual-sensor unit, the user points either the minute or the hour control component at the clock with the projected hand at the desired angle. Pressing the activation pushbutton transmits the location codes within both interface components. Thus the rotation between the projector and the clock is detected and used to directly set the respective clock hand to the indicated angle. Continuing rotating the projection while holding down the activation pushbutton allows for further tuning. As shown in Figure 11 we simulated a virtual analog clock displayed on a computer screen.

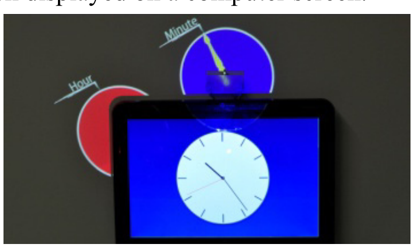

Figure 11. Adjusting a clock by directly setting angles.

In the interactions discussed so far, only the projector is moved or rotated. PICOntrol, however, also lends itself to interaction with moving devices (e.g. home robots such as Roomba™). When controlling such devices, both the projector movement and device movement need to be taken into account. Thanks to our relative detection mechanism, these two types of movements are not differentiated by PICOntrol, and can be seamlessly incorporated into the same interaction, agnostic of whether one or both entities are actually moving. Based on this principle, we demonstrate remotely controlling a toy car with a dual-sensor unit mounted on its roof, using two interaction paradigms: *direct-manipulation* and *follow-the-center*.

With the direct-manipulation paradigm, the user can overlay the projection with the car's dual-sensor unit, press the activation pushbutton, and rotate the projector to rotate the car on the spot. The sensor unit records the initial angle between itself and the projector when the location code is first projected, and subsequently keeps rotating the car if this angle is being changed in order to match the initially recorded angle. By doing so, the car preserves its initial angular alignment with the projection, as if the user is directly rotating the car itself (Figure 12a).

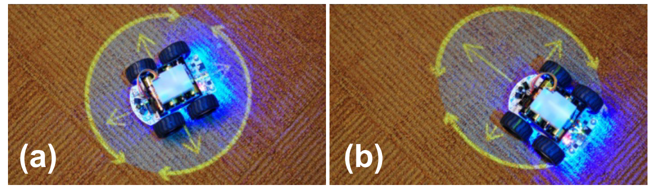

Figure 12. "Direct-manipulation": (a) Rotating on the spot. (b) Swiping to accelerate or decelerate the car.

Swiping the projection along the car's main axis while holding the activation pushbutton will cause it to start moving in the direction of motion (i.e. either forwards or backwards as shown in Figure 12b). The projection itself can be at an arbitrary angle to the car, as long as the direction of the swipe roughly aligns with the car's main axis (as the sensor unit can compensate for lateral motion in different directions by knowing the angle between itself and the projection). Further, if the car is currently already moving when the swiping happens, it will accelerate or decelerate depending on whether the detected swiping direction is the same as or opposite to the current car motion. As the swiping motion is sensed relatively between the projection and the car, this also inherently allows a static projection to serve as a barrier to slow down and eventually stop a moving car, as from the car's perspective a motion opposite the current driving direction is detected.

Note that although the car is interpreting all the control relative to itself, from the users' perspective they are always directly manipulating the car in absolute terms (i.e. world coordinates), as if using their hand to directly manipulate it. Hence the control mechanism remains the same regardless of which direction the user and the car face. This poses a clear advantage over most of today's remote control car interfaces, where the user must operate from the car's perspective and perform counterintuitive mental rotation when facing a different direction.

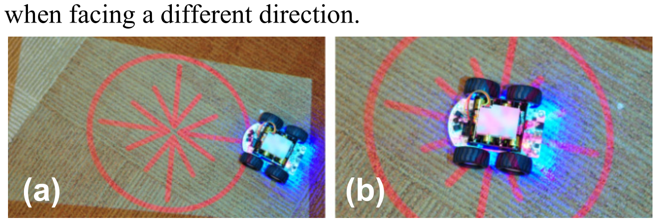

Figure 13. "Follow-the-center": (a) The car moves to and follows (b) the projection's center.

With the follow-the-center paradigm, the robot car always moves towards the center of the projected image, marked by inwards pointing arrows (Figure 13). In this mode the projector sends location codes continuously. By sensing the rotation and location of the sensor unit with respect to the projection, the car can determine the position of the projection center relative to itself and move accordingly. Users can "catch" the car by aiming the projection at it, and the car will move to and follow the projection's center, with the projection acting as a virtual leash. Hence, the user can guide the car through an arbitrary route.

Color-Sensor Unit

While this paper focuses on sensor units that sense light intensities, here we briefly explore a unit that detects RGB color. We prototyped our color-sensor unit (Figure 5c) using three photo sensors, each equipped with a different narrow band-pass color filter matching the wavelength of the respective color channel (R, G, B) of the projector. The three sensors are mounted with minimal spacing in a triangular arrangement to sense the same projected area. Other off-the-shelf RGB sensors may also be used for this purpose.

The color-sensor unit can support direct codification of locations within the projection based on variations in color in a static image, rather than a temporal code sequence. However, the color-sensor unit may potentially also be combined with temporal codes, for example to increase the transmission bandwidth by using n-ary codes with n colors, or by separating code transmission and user interface to different color channels to minimize interference and perceived flicker. We leave a full exploration of the capability of color-sensor units to future work, and provide one simple application example here: Mood lamps, such as Philips' LivingColors product, allow the user to set the emitted light to an arbitrary color in order to create a certain ambient atmosphere. PICOntrol's visual control interface can make this more intuitive, enabling the user to directly choose a color in a single interaction. The projector projects a color spectrum. To set a specific color, the user presses the activation pushbutton and moves the projection to align the desired color with the lamp's color-sensor unit (Figure 14). While the projection is moved the lamp's color is continuously updated, and to confirm a color selection the user releases the activation pushbutton.

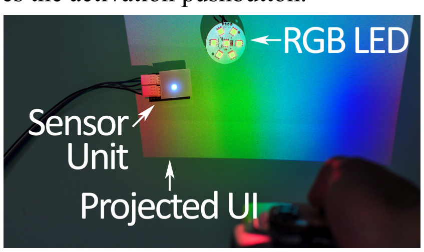

Figure 14. Setting the color of a mood lamp.

DISCUSSION

As mentioned earlier, while we consider a systematic evaluation premature for our proof-of-concept stage, we conducted an informal user trial with four participants (about one hour each) for preliminary feedback. All participants understood the PICOntrol interaction style easily, and were able to use it immediately. They frequently commented on the intuitiveness and "coolness" of the approach, and appreciated its usefulness in supporting everyday tasks. Such feedback supported that our exploration of PICOntrol interactions was successful.

That said, PICOntrol is not without its limitations. Although no participants complained about the visual flicker when control information is transmitted, and it incidentally serves as a feedback mechanism, we do aim to further lower its intrusiveness. We have conducted initial experiments with promising results, which limit code transmission to specific color channels that human eyes are less sensitive to so as to reduce perceivable flicker. This generic solution would work with any off-the-shelf projector. Whereas in scenarios which allow employing specific projection technologies (e.g. higher-frequency projectors, or those based on DLP [15] technology or include an IR projection channel), the visible flicker can be further reduced or removed entirely.

Another limitation is the need for the user to choose a specialized projected interface in some cases. However, with careful interface design, the number of such cases can be considerably reduced. Generic interfaces such as on/off buttons and sliders or knobs for parameter adjustment can be used across a variety of devices, not to mention multiple devices of the same category. In some other cases, the difference between two interfaces is purely visual, for easier understanding by the user, while the underlying codes being transmitted stay the same (e.g. the gestural interface used in Figure 7 and the "follow-the-center" interface used in Figure 13 are both transmitting location codes across the entire image), therefore the interfaces can indeed be used interchangeably. As a result, the burden for the user to select a user interface is lower than that of selecting a device to control with conventional universal controllers. Furthermore, as mentioned previously, the requirement for manually selecting interfaces could ultimately be eliminated by expanding the PICOntrol concept to include a lightweight (likely also via visible light) peer-to-peer backward communication channel in the future.

In everyday environments, the controlled device may not always have a large enough surface or be surrounded by a relatively clean background to accommodate a complete, clear, or undistorted projection. Although this may cause slight inconvenience for users (especially novices) to visualize the full user interface, it does not pose any difficulty to the system, as technically as few as one pixel needs to be projected on each sensor for the code to be detected. Whereas from the user's perspective, once they are sufficiently familiar with the interface layout, only the one interface component being triggered (or part of it thereof) needs to be visible for aiming the projector and controlling the device.

Interfaces become larger when projected at a farther distance. For the most part, this is a beneficial property. Not only their apparent visual size remains relatively constant to the user (since the user is also looking from a farther distance), but also it requires the same amount and precision of angular movement to overlay an interface component on the sensor unit regardless of the distance. Having said that, when several devices are positioned in close proximity and the user is controlling from afar, there is a risk of involuntarily covering and controlling multiple devices at the same time. Introducing mechanisms for the user to optically or digitally zoom the projection on the fly may solve this.

Finally, although we have demonstrated PICOntrol with a variety of concrete and familiar application examples, these are merely starting points for exploring an even richer set of applications. For example, the ability to intuitively control moving devices, as exemplified by the toy car example, may open a new space of human-robot interaction using handheld projectors. We should also note that PICOntrol is not necessarily the best answer for all scenarios, as each control mechanism has their pros and cons, but it should be seen as an alternative interaction approach and technology that addresses some of today's challenges and offers a new interaction space.

CONCLUSION

This work explored the concept of using handheld projectors for direct control of physical devices, using the visible projection to simultaneously present the user interface and transmit the control information. PICOntrol provides direct, visible, and rich interactions with various devices using off-the-shelf handheld pico projectors and inexpensive photo sensors without requiring central infrastructure. We have successfully demonstrated the feasibility of this new approach and presented a first exploration of its rich interaction space. As pico projectors become increasingly integrated with mobile devices, and because our sensor units can be easily integrated with devices at little additional cost, we hope our approach can be readily adopted in the future.

ACKNOWLEDGMENTS

We thank Masahiko Inami, Paul Dietz, Desney Tan, Martina Pagura, and colleagues at Microsoft Research Asia for valuable suggestions and discussions.

REFERENCES

1. Accot, J., and Zhai, S. More than dotting the i's — foundations for crossing-based interfaces. *CHI* (2002), 73–80.
2. Beardsley, P., van Baar, J., Raskar, R., and Forlines, C. Interaction using a handheld projector. *IEEE Comp. Graph. and App.* 25 (2005), 39–43.
3. Beigl, M. Point & Click - interaction in smart environments. *HUC* (1999), 311–313.
4. Bier, E. A., Stone, M. C., Pier, M. C., Buxton, W., and DeRose, T. D. Toolglass and Magic Lenses: the see-through interface. *Computer Graphics and Interactive Techniques* (1993), 73–80.
5. Boring, S., Baur, D., Butz, A., Gustafson, S., and Baudisch, P. Touch projector: Mobile interaction through video. *CHI* (2010), 2287–2296.
6. Cao, X., and Balakrishnan, R. Interacting with dynamically defined information spaces using a handheld projector and a pen. *UIST* (2006), 225–234.
7. Cowan, L. G. and Li, K. A. ShadowPuppets: supporting collocated interaction with mobile projector phones using hand shadows. *CHI* (2011), 2707–2716.
8. Dietz, P. H., Yerazunis, W., and Leigh, D. Very low-cost sensing and communication using bidirectional LEDs. *UbiComp* (2003), 175–191.
9. Fails, J. A., and Olsen, D. Light widgets: Interacting in every-day spaces. *IUI* (2002), 63–69.
10. Harrison, C., and Hudson, S. E. Minput: Enabling interaction on small mobile devices with high-precision, low-cost, multipoint optical tracking. *CHI* (2010), 1661–1664.
11. Hosoi, K., Dao, V. N., Mori, A., and Sugimoto, M. VisiCon: A robot control interface for visualizing manipulation using a handheld projector. *ACE* (2007), 99–106.
12. Ishii, K., Zhao, S., Inami, M., Igarashi, T., and Imai, M. Designing laser gesture interface for robot control. *INTERACT* (2009), 479–492.
13. Kemp, C. C., Anderson, C. D., Nguyen, H., Trevor, A. J., and Zhe, X. A point-and-click interface for the real world: Laser designation of objects for mobile manipulation. *HRI* (2008), 241–248.
14. Lee, J., Dietz, P. H., Maynes-Aminzade, D., Raskar, R., and Hudson, S. E. Automatic projector calibration with embedded light sensors. *UIST* (2004), 123–126.
15. Lee, J., Hudson, S. E., Summer, J. W., and Dietz, P. H. Moveable interactive projected displays using projector based tracking. *UIST* (2005), 63–72.
16. Ma, H., and Paradiso, J. The FindIT flashlight: Responsive tagging based on optically triggered microprocessor wakeup. *UbiComp* (2002), 655–662.
17. Markets and Markets. Global pico projector market. http://www.marketsandmarkets.com/ Market-Reports/ pico-projector-market-196.html, 2010.
18. Myers, B. A., Bhatnagar, R., Nichols, J., Hong, C. H., Kong, D., Miller, R., and Long, A. C. Interacting at a distance: Measuring the performance of laser pointers and other devices. *CHI* (2002), 33–40.
19. Ni, T., and Baudisch, P. Disappearing mobile devices. *UIST* (2009), 101–110.
20. Nii, H., Sugimoto, M., and Inami, M. Smart light-ultra high speed projector for spatial multiplexing optical transmission. *CVPR* (2005), 95–102.
21. Rapp, S., Michelitsch, G., Osen, M., Williams, J., Barbisch, M., Bohan, R., Valsan, Z., and Emele, M. Spotlight navigation: Interaction with a handheld projection device. *Pervasive,* (2004).
22. Raskar, R., Beardsley, P., van Baar, J., Wang, Y., Dietz, P., Lee, J., Leigh, D., and Willwacher, T. RFIG lamps: interacting with a self-describing world via photosensing wireless tags and projectors. *SIGGRAPH* (2004), 406–415.
23. Ringwald, M. Spontaneous interaction with everyday devices using a PDA. *Supporting Spontaneous Interaction in Ubiquitous Computing Workshop* (2002).
24. Summet, J., and Sukthankar, R. Tracking locations of moving hand-held displays using projected light. *Pervasive* (2005), 97–113.
25. Suzuki, G., Aoki, S., Iwamoto, T., Maruyama, D., Koda, T., Kohtake, N., Takashio, K., and Tokuda, H. u-Photo: Interacting with pervasive services using digital still images. *Pervasive* (2005), 190–207.
26. Willis, K. D. D., Poupyrev, I., Hudson, S. E., and Mahler, M. SideBySide: Ad-hoc multi-user interaction with handheld projectors. *UIST* (2011), 431–440.
27. Willis, K. D. D., Poupyrev, I., and Shiratori, T. MotionBeam: A metaphor for character interaction with handheld projectors. *CHI* (2011), 1031–1040.
28. Wilson, A. D., and Pham, H. Pointing in intelligent environments with the WorldCursor. *INTERACT* (2003).
29. Wilson, A. D., and Shafer, S. XWand: UI for intelligent spaces. CHI (2003), 545–55.