BlueTable: Connecting Wireless Mobile Devices on Interactive Surfaces Using Vision-Based Handshaking

Andrew D. Wilson Raman Sarin

Microsoft Research
One Microsoft Way
Redmond, WA
awilson@microsoft.com, ramans@microsoft.com

ABSTRACT

Associating and connecting mobile devices for the wireless transfer of data is often a cumbersome process. We present a technique of associating a mobile device to an interactive surface using a combination of computer vision and Bluetooth technologies. Users establish the connection of a mobile device to the system by simply placing the device on a table surface. When the computer vision process detects a phone-like object on the surface, the system follows a handshaking procedure using Bluetooth and vision techniques to establish that the phone on the surface and the wirelessly connected phone are the same device. The connection is broken simply by removing the device. Furthermore, the vision-based handshaking procedure determines the precise position of the device on the interactive surface, thus permitting a variety of interactive scenarios which rely on the presentation of graphics co-located with the device. As an example, we present a prototype interactive system which allows the exchange of automatically downloaded photos by selecting and dragging photos from one cameraphone device to another.

CR Categories: H5.2 [User Interfaces]: Input Devices and Strategies

Keywords: mobile devices, interactive tabletops, computer vision, ubiquitous computing, Bluetooth

1 INTRODUCTION

Associating and connecting wireless mobile devices is often a cumbersome process. Transferring a file from a phone using Bluetooth, for example, involves a pairing process in which the name of the target device must be known or recognized by the user, the navigation of a number of dialog boxes, and possibly the entry of passwords.

Part of the difficulty of this process lies in the lack of physical grounding during the connection process: for example, physical feedback plays no role in establishing or breaking the connection. Unlike wired connections, wireless connections have no obvious representation in the real world.

Previous works have explored the idea of detecting synchronous events such as touching or bumping devices together [5, 6], or pressing the same key simultaneously [11] as ways of associating two wireless devices in a physically-grounded manner. This approach not only restores some of the tangible feedback of establishing a wired connection, but also allows appropriately equipped devices to find each other automatically. For example, two devices that are equipped with accelerometers may associate based on the observation that "bump" events were recorded at nearly the same point in time [7]. That such a co-occurrence in time might happen accidentally is thought to be extremely unlikely, particularly when near-field wireless communication techniques such as Bluetooth are used.

Other works have demonstrated techniques to connect a mobile device to a large display by having the display present a randomly chosen key which the user then enters on the mobile device. Because the key is only visible to the user as they are viewing the display, the display system has some guarantee that it is connected to the correct device [3]. This key can take the form of an alphanumeric string to be entered on a device keypad, a sequence of motions that are then matched by an accelerometer-equipped device [9], or a visual pattern shown on the display which is then captured and decoded by cameraphone [1, 14].

Established RFID technologies and upcoming near field communication (NFC) techniques can support device association if the device is equipped with the appropriate RF tags [4, 10, 13]. Typically this will entail the use of short range (0 to 3 inches) RFID readers which require the placement of the tagged device on a small reading surface, thus limiting their application to small physical configurations. Longer range RFID readers, on the other hand, present the opposite problem: as a user it is difficult to judge whether a given tag is within reading range. This uncertainty can lead to unintended connections. Worse, it raises obvious privacy and security concerns. While it is possible to tile a large area with multiple short range RFID readers [16], such an approach may be prohibitively expensive.

We explore the use of interactive surfaces (e.g., tables) to facilitate connecting mobile devices. Unlike vertical displays, horizontal interactive displays naturally afford the placement of multiple devices and the display of co-located graphics. This paper presents a device association technique which establishes connections between multiple devices and an interactive surface. A device is connected automatically when the user places it on the interactive surface. This connection is then broken when the user removes the device from the surface. This interaction is natural and easily understood by users.

In particular, our device association technique uses computer vision techniques to bootstrap and verify Bluetooth connections to devices placed on the interactive surface. An important feature of our technique is that it determines the precise position of the devices on the surface, thus enabling a wide variety of interactive scenarios which RFID and NFC technologies do not readily support. This is accomplished without adding hardware requirements to most mobile devices.

For example, we present an interactive system which applies our handshaking technique to initiate the automatic transfer of cameraphone photos: after taking a number of snapshots, the user places their phone on the table, and the photos automatically spill out around the phone, onto the interactive surface. A second user may then place their phone on another part of the surface, resulting in the display of their photos. Photos may be exchanged

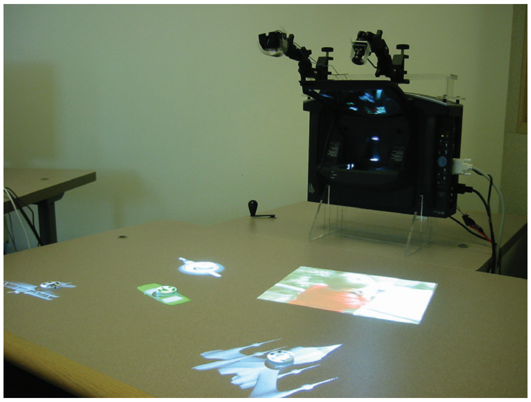

Figure 1. PlayAnywhere projection-vision interactive surface system uses a single camera, infrared illuminant, and a short-throw projector on a regular table surface.

from one device to the other by simply dragging a photo from one device to other. These interactions benefit substantially from the position information determined in the handshaking process.

In this paper we detail the technique and our prototype, and outline other interaction scenarios enabled by the technique.

2 PROJECTION-VISION SYSTEM

Before introducing our device association technique, we introduce as background the vision-based interactive surface system used in this work.

Interactive surfaces are often equipped with sensors to detect the user's touch, and some also have the ability to detect the presence of other physical objects. Computer vision is one technique that is particularly useful in the location, tracking and identification of objects placed on a surface. In the present work, we build on the PlayAnywhere system [17] which combines a projector and vision-based sensing for compact interactive surface applications. PlayAnywhere uses an infrared-sensitive camera and infrared illuminant to monitor the user's gestures and the placement of objects on an everyday surface such as an office desk (see Figure 1). The camera is positioned and calibrated to monitor exactly the active display area of the projector, such that when objects are detected by vision processes, the system may then display graphics that are precisely registered with the physical object. Because the camera senses only near-infrared light (approximately 900nm, just beyond visible, and not impacted by heat sources), the sensed image does not include the projected image composed of only visible light.

Vision systems such as PlayAnywhere can be applied to a variety of sensing tasks. For example, it is straightforward to detect the placement of most objects on the surface by comparing the most recent image of the surface with a reference image of the empty surface: first we compute the pixel-by-pixel difference between the reference image and current image. Pixel differences that are greater than some threshold are marked in a new image as '1' while all others are marked as '0'. Following this binarization step, *connected components analysis* [2] groups all sets of adjacent pixels with value '1', yielding a rough list of all the spatially distinct objects on the surface. This process is repeated for every frame at video rate (30Hz).

In conjunction with the connected component analysis algorithm, various statistics about the shape of each object may be computed. For example, geometric moments (mean, covariance) of the connected component pixel locations may be computed to determine an ellipse fit to the connected component, giving a rough indication of object size and orientation.

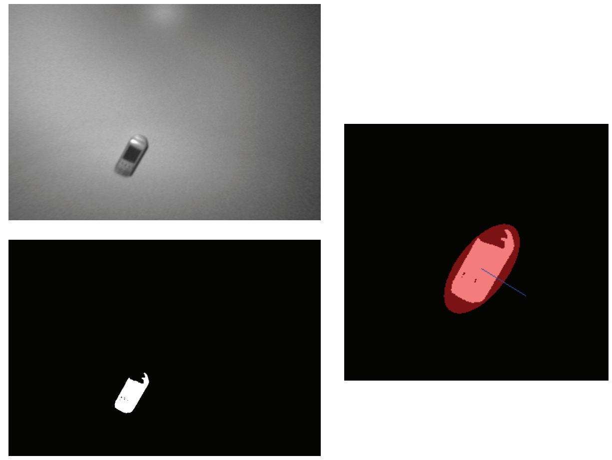

Figure 2. Upper left: input image with phone; Lower left: binarized image; Right: binarized image with ellipse fit (red).

While the field of computer vision offers much more sophisticated object recognition techniques, we use the ellipsoidal model to detect when a mobile phone is placed on the surface, simply by detecting the stable observation of an object with an ellipsoidal model with phone-like dimensions, with wide tolerances to accommodate a variety of phone models (see Figure 2). This simple approach works well in the interactive scenarios we explore here, and allows interactive performance. While more sophisticated object recognition techniques may be able to recognize a given mobile device model, they provide no help in distinguishing two devices of the same model.

An alternative approach to uniquely identifying a physical object by computer vision is to use visual markers or codes that are applied directly to the object. However, such codes are often visually unappealing, require a significant amount of space on the device, and, because they are generally static, are easily copied.

3 HANDSHAKING OVER BLUETOOTH AND VISION

We contribute a technique by which our vision-based interactive surface system is able to identify and precisely position mobile devices as they are placed on the surface. The technique leverages the capabilities of existing mobile devices. The general idea is to for the interactive surface system ("host" system) to connect to each available Bluetooth device and have each in turn make a signal that is visible to the computer vision system. If the signal is seen, the system will know that it is connected to the device on the surface at a given location.

The basic handshaking algorithm is summarized as:

1. Detect the placement of a new phone-shaped object by visual means.
2. For each switched-on Bluetooth device:
    a. Attempt to connect to the device over Bluetooth. Continue if the device advertises our globally unique identifier (GUID), else move on to the next device.
    b. Command the device to blink its IRDA (infrared) port.
    c. If the blink is detected at the position of the object go to step 3, else move on to the next Bluetooth device.
3. Determine the exact orientation of the device (optional).
4. Begin application-specific communications via Bluetooth, passing on position and orientation information to application running surface interactions.

Like long range RFID, Bluetooth is ill-suited to determine if a device has been placed on the table surface, but it can be used to suggest a list of mobile devices that might be on the surface. Furthermore, a connected Bluetooth device may be programmed to identify itself visually to the host system by modulation of light sources that the vision system is able to detect (e.g., infrared light from the IRDA port, light from the mobile device display, or controllable notification LEDs). Once discovered by the vision system, the same Bluetooth connection may be used to exchange application data such as photos, contact information, remote user interfaces, and so on. In the following, we detail the process by which devices identify themselves to the host system.

The host system is equipped with a Bluetooth transceiver. Once a phone-shaped object is detected, the host invokes the Windows Bluetooth API to collect a list of all Bluetooth devices in range (the devices need not be “paired” in the Bluetooth sense). Because Bluetooth has a range of about 30 feet, this list will include devices that may not be on the surface, or even in the same room, so it necessary to determine which of these wireless devices correspond to the phone-shaped object just placed on the surface (if any).

Each mobile device that will participate with the interactive surface system is running a piece of custom software designed to handle the following handshaking process: the interactive system initiates a Bluetooth connection to each available Bluetooth device. If a device is found to be running our custom software (as indicated by particular Bluetooth service GUID), the connection will succeed and the interactive surface system is then able to exchange data and commands with the mobile device via a socket connection. The host system then requests that the custom process trigger the infrared (IRDA) port to blink for a short period of time. If the connected device is on the surface, PlayAnywhere’s infrared vision system will be able to detect the blinking of the IRDA port at nearly the same time it requests the blink, thus confirming that the connected device is on the surface. If no blink is detected, the host moves on to the next Bluetooth device and repeats the process.

The use of a synchronized signal in the image is motivated by the observation that a signal in one modality and a signal in another modality are very unlikely to be precisely synchronized in time. This depends largely on the scope of what is being sensed: because the monitored space of the table top is rather small, matching signals are unlikely to happen by chance.

The vision system detects the blinking of the IRDA port by continuously computing the pixel-wise difference of the input image with a like image collected at the beginning of the identification process. Such difference images highlight areas that change during the blinking process, and as such may be used to detect the blink from the image. As before, connected components are found from the difference image. Such connected components correspond to blinking infrared light either from IRDA port directly, or from the same light reflected off the surface if the IRDA port is oriented out the side of the device (see Figure 3).

Our technique supports placing more than one connected device on the surface at time. This enables a wide variety of interactions on interactive surfaces. For example, projected icons of documents or user interface controls for each device may be co-located with the device on the surface, and multiple devices may be connected to exchange data. For such applications it is important to confirm that the observed blinking pattern matches up with the newly acquired surface object. This is accomplished by confirming that the blinking pattern coincides with the position of the candidate object by computing the distance between the blinking connected component and the object’s connected component.

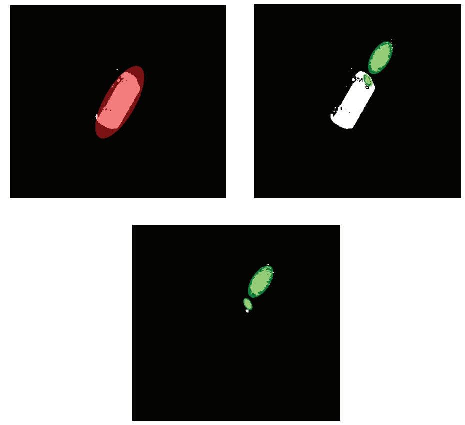

Figure 3. IRDA port blinking. Top left: phone shape detected, with fit ellipsoid model (red). Top right: infrared light detected at IR port and on surface (green). Bottom: infrared light is detected apart from phone shape by image differencing.

Additionally, it is useful for applications to know the orientation of the device on the surface. For example, user interface graphics can match the orientation of the device to maintain readability from the user’s own orientation, or transferred documents can be shown to emanate from the front of the device. The ellipsoidal model gives the precise orientation of the object, but only up to a 180 degree ambiguity. Because IRDA ports are typically located at an end of the device, the position of the blinking connected component can be used to overcome the ambiguity in orientation. The handshaking process determines which end of the object connected component is nearest to the blinking connected component, and updates the orientation accordingly (see Figure 4).

Once connected and visually verified, the host is free to exchange application data such as cameraphone pictures, contact information, etc. During this time the shape of the device may be tracked frame-to-frame by the vision system, such that projected graphics associated with the device may follow its movement. When the user wishes to revoke the connection to the host, they may simply remove the device from the surface. At this point the vision system will lose track of the phone. The above IRDA port blinking procedure may be run again to verify that the device has been removed from the surface if necessary. In the case of top-down projection vision systems, where users may occlude device objects by their hands and arms, it is helpful to build in tolerances for long dropouts in the appearance of the object, and to drop the Bluetooth connection only after an extended period of time.

4 ELABORATIONS

4.1 Exchanging Codes

The vision-based handshaking procedure as outlined is not

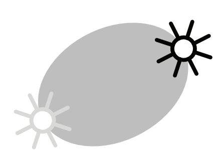

Figure 4.180 degree ambiguity of ellipsoidal model is removed by determining at which end of the ellipse the infrared signal appears.

foolproof. For example, spurious "blinking" may be detected if the user moves their hands over the phone at the moment that the host requests some connected phone to blink its IRDA port. This may result in an association with the incorrect device. Furthermore, it raises security concerns. Cooperative users that understand that the handshaking process takes time and requires a line of sight are likely to place the phone down and wait until the transfer is complete before continuing. But a maleficent user may wilfully trigger an incorrect association by waving their hands over the phone at the right moment, thus spoofing the simple change-detection blinking pattern.

A natural extension of the handshaking process is for the IRDA device to modulate the IR LED using a unique pattern, such that the vision can verify the device connection by recovering the same pattern. For example, the device can communicate a binary code [8]. A further elaboration would be for the host to pick a random key value each time it requests a device to blink. This key is then transmitted via Bluetooth to the device with the blinking request, possibly encrypted. When the host recovers the same key value from the blinked code, it can be reasonably certain that the visual signal it just saw was non-accidental.

Our present prototype system uses an 8 bit key which is communicated to the host by pulse-width modulation (PWM) of the IR LED on the mobile device. The software on the device obtains direct control over the IRDA port in order to modulate the IR LED directly. "On" and "off" pulses are generated by holding the IR LED on for 300ms and 150ms respectively, with a 150ms break between each pulse. The average total time per pulse is thus 375ms, yielding a transmission rate of 2.67 bits per second, and an average time of 3s to transmit an 8 bit code. An example of the PWM signal as recovered from the vision system is illustrated in Figure 5.

An 8 bit code length is sufficient to prevent confusion with extraneous motion on the interactive surface, as well as prevent spoofing attempts made by the less determined maleficent users. Longer code lengths are possible, of course, but only at the expense of lengthening the handshaking process.

Because the handshaking technique requires testing each device in turn, the total time to complete the handshaking process is proportional to the number of candidate devices. In the present prototype, the initial computation of the list of available Bluetooth devices and the code transmission time dominates the total time to complete the handshaking process. If the code is made longer, or the number of candidate devices is large, the time to complete the handshaking will be too great to be useful in interactive systems. There are a number of strategies available to speed the process. We leave these as future work. First, the host system can terminate the code transmission process early if the first bits of the code are not seen immediately after the request to transmit the code is made. The host may then move to the next device.

Secondly, the host vision system can employ cameras with higher frame rates. Our current prototype uses a video camera running at standard NTSC video rates (30Hz), but higher frame rates are available. Most notably, today's common CMOS image sensors include the ability to set a region of interest (ROI) over which light levels are measured. Only pixels belonging to the ROI are transmitted to the host. In the case of CMOS imagers, the resulting frame rate is proportional to the size of the ROI. Our initial experiments with CMOS cameras and IR LEDs show that it is possible to obtain frame rates of 500Hz with a small ROI around the LED. This suggests a way to speed the code transmission process: first, set the IR LED on for a pulse long enough to enable the host vision system to find it with no ROI set (whole frame visible, 30Hz). Then set the position of a small ROI centered on the recovered location of the LED. The camera may then match a high bit rate transmission from the device.

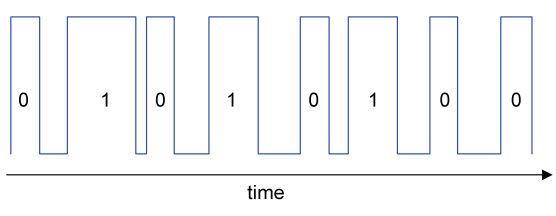

Figure 5. Pulse width modulation, least significant bit first, for the 8 bit value 42, using direct control of IRDA port. Timings were collected on the host interactive surface system, from image analysis. Limits on the control of timing and sampling error contribute to inaccuracies in pulse width.

4.2 Visible Spectrum Blinking

The use of the IRDA port in the handshaking process has a number of distinct advantages. Near infrared is directly supported by our vision-based interactive surface system, where its use allows the vision system to ignore projected (visible) graphics, and to work in a variety of indoor ambient light conditions. Also, because infrared is invisible to the naked eye, the vision-based handshaking process is invisible to users, and thus does not intrude on the overall interaction. This contrasts with related synchronization approaches that use visible codes or markers. Such visible patterns require space, are visually distracting and are often difficult to make visually appealing.

Unfortunately, while IRDA ports have been very popular standard features on smartphone and other mobile devices for years, many of the latest devices do not include IRDA functionality. In an ironic twist for the present work, this perhaps can be explained by the success of the Bluetooth standard, which fulfils many of the same roles as IRDA on mobile devices.

To address this issue, we have explored visible light-based strategies which involve the modulation of the device's display. This approach works in much the same way as with the IR LED, except that rather than turning the IR LED on and off, a region of the display may be turned on (painted white) or turned off (painted black) (see Figure 6). Our current prototype uses the same PWM scheme to modulate the display as to modulate the IR LED, and the same code is used to recover both types of signals from the image. However, because today's transmissive displays use backlights that are invisible in the infrared, our prototype system includes a second camera sensing in the visible domain expressly for the purpose of recovering visible light modulation. This second camera is identical to the primary infrared camera except that no IR-pass filter is applied.

During the handshaking process the host can look for the appropriate modulation in both the visible and infrared channels simultaneously, or the host system can query the connected device for its preferred method. Our prototype uses the latter strategy, and examines only one channel at a time to avoid unnecessary image processing. Because animated projected graphics can confuse vision processes working in the visible channel, it is important to limit the analysis of the visible channel to the region corresponding to the device's display, and possibly control the presentation of any graphics placed directly on the device itself.

Modulating some portion of the screen is an attractive alternative to modulating the IR LED because most mobile devices will have a display, and thus the approach is compatible with most mobile devices. However, modulating the display can be visually distracting, and requires temporarily taking over the user interface shown on the display. Other related visible channel approaches include modulating notification LEDs that are placed

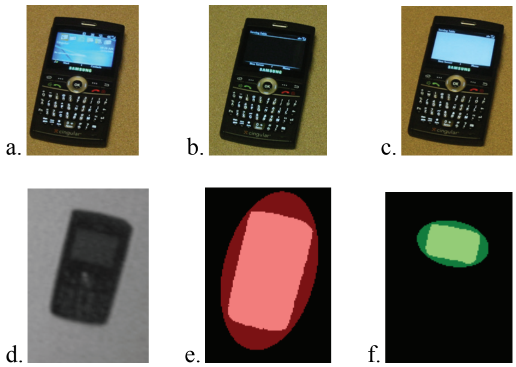

Figure 6. Visible light modulation. Samsung SGH-i607 smartphone (a) is not equipped with an IRDA port so BlueTable modulates onscreen graphics instead. Pulses are generated by changing the onscreen graphics from black (“on” pulse) (b) to white (c). The device is initially detected from the IR signal as before (d and e), but the pulse is detected in the visible spectrum using a second camera (f).

on the front of a device, typically next to the display, and modulating backlight levels. Both of these approaches may be less obtrusive.

5 BlueTable: Bluetooth Photo Synching on an Interactive Surface

To demonstrate the device-to-surface connection technique we implemented an application that allows the transfer of photos from a cameraphone to the interactive surface and/or another mobile device. After the user has taken a number of photos with their phone, they may simply place the phone on the interactive surface to initiate the transfer to the host device.

During the Bluetooth/vision handshaking process, graphical feedback in the form of a pulsating blue halo is projected directly on the phone. This indicates to the user that the connection is in progress. As pictures are transferred over Bluetooth, they appear to fly out of the top of the phone, and are arranged in a messy stack, as if they were casually tossed on the surface of the table (see Figure 7).

The user may move, rotate and resize each of the photos by directly placing their hands on the photo and using natural gestures (e.g., rotating the hand to rotate the photo, pulling two hands apart to enlarge). The user may bring a given photo the top of a stack of photos by simply touching any part of the photo. The free transform of the photos is accomplished by computing the optical flow corresponding to the hand motion over the currently selected (topmost) photo and computing the best rotation, translation and scaling factors that fit the optical flow field (see Figure 8). We employ the touch algorithm and the optical flow-based movement described in [17]. Related manipulation of photos on interactive surface is presented in [15].

While BlueTable shows the transfer of data from a single device to the host, it is natural to consider placing multiple devices on the surface. Because the image processing and handshaking procedure sees each device as a distinct object on the table, the approach supports multiple simultaneous devices. This suggests a number of interesting applications: for example, two people can meet over an interactive surface, place their mobile devices on the surface and “drag” documents from one device to the other. When either user wishes to terminate the interaction, they merely remove their device from the surface.

BlueTable supports the exchange of photos by dragging a photo graphic onto the blue halo of any connected device (other than the

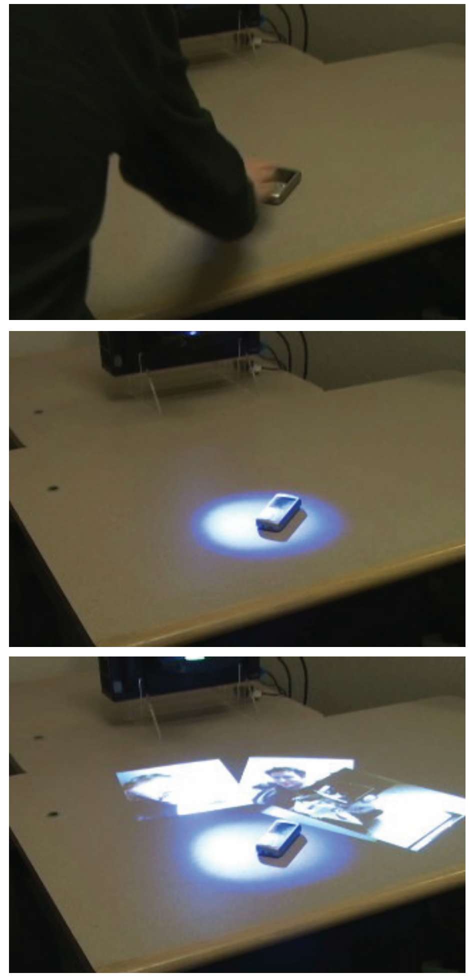

Figure 7. BlueTable prototype connects automatically to cameraphone via Bluetooth, transfers photos, and spills them onto interactive surface.

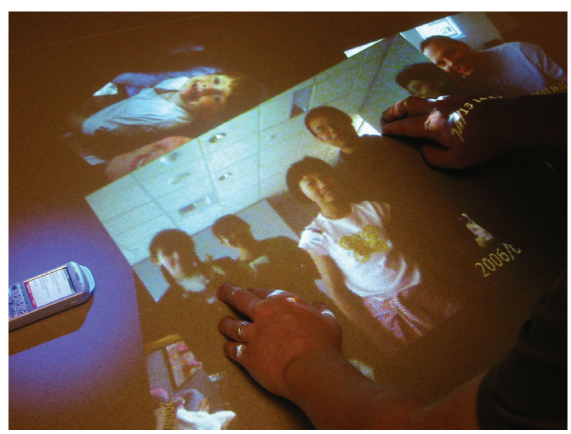

Figure 8. Photos may be scaled, rotated and moved on the desk by using one or both hands. Optical flow techniques are used to recover the motion of the hands. This motion is then modeled as a combination of simultaneous rotation, scaling and translation movements. An individual photo is brought to the top of the stack of multiple photos by touching any part of the photo with an outstretched finger.

Figure 9. Support of multiple devices and transferring documents. Photos taken with the left cameraphone may be transferred via Bluetooth to the phone on the right by dragging photos onto its blue halo. The transfer is depicted visually by an animation (red) from the source phone to target phone.

one that contributed the photo). When the photo is close enough to the target device, the transfer of the file on the host system is initiated over Bluetooth, and the transfer process is depicted graphically by an animation of a moving graphic (red dot) from the device which originally contributed the photo to the target device (see Figure 9). This animation makes it clear to the user that the transfer is underway, and from which device the photo originally came. This multiple device interaction also demonstrates the use of infrared-based handshaking and visible light handshaking: where one of the phones does not have an IRDA port, visible light modulation as discussed above is used.

6 EXTENSIONS AND FUTURE WORK

6.1 Interactions

Our Bluetooth photo synchronization demonstration hints at the range of possible interactions and applications. It is useful to consider a few categories of applications where the technique may be useful in future work.

6.1.1 Mixing display and user interface

The co-located presentation of a connected device and the larger display around it suggests applications in which UI is moved from the device to the surface display (and back again) as appropriate [12]. For example, interactions that are difficult on small-screened devices might be replicated on the larger screen, where precise interaction, and reading large bodies of text may be easier. User interfaces for such interactions might spill from the device onto the surface (see Figure 10).

Likewise, interactions could migrate from the surface to the device while it is on the surface. This redirection could be useful when the device display and user interface has higher fidelity than that of the interactive surface: consider an interactive system with no touch capability. In this case user interface elements such as buttons might be captured to the device while it is on the surface. Button presses and stylus events could be forwarded to the surface system, while the function and onscreen presentation of the device might depend on where the device is placed on the surface.

Also, this exchange between devices of differing display sizes can support the choices based on users' privacy needs: documents on the large surface are more likely to be seen publicly, while documents on the mobile device might be more private in nature. In a group or collaborative setting, it might be useful to quickly move documents on the mobile device to the public large display.

6.1.2 Connecting multiple devices

In the case of the BlueTable demonstration system, cameraphone devices are connected by virtue of having been placed on the surface. The sensing system and handshaking process also allows the precise tracking of the devices on the surface. In the case of BlueTable, this is used to display photos next to the device which provides them, such that the transfer of photos from one device to the other can be done naturally by dragging a photo from one device to another. The ability to connect to and track multiple simultaneous devices leads to a number of scenarios involving multiple devices.

For example, a mechanical keyboard and a mobile device might be automatically connected if they are placed close to one another on the interactive surface, so that keystrokes are directed to the mobile device. The orientation of each of the devices may also play a role in whether devices are connected. For example, two devices that are pointed at each other might be connected, mimicking the manner in which two IRDA devices must be arranged to "see" each other's signal. Or devices that are oriented along the same general direction might be considered a connected group, perhaps if they belong to the same user, while devices at other orientations are assumed to be from another user and are so grouped separately.

The surface display can make the various connections clear by drawing the connections as they are made and broken. Such illustrations can address the problem that it is often difficult to determine active wireless connections. Activity along the connection can be illustrated by appropriate animation and color, as used in the BlueTable transfer process. Furthermore, if the surface is capable of sensing touch, users could establish, break and move connections by directly manipulating their graphical representation on the surface.

When multiple devices are placed on the surface to be connected, should each device maintain a separate connection to the host, or should the devices establish direct connections to each other, with the host system coordinating the connections? In the former case data from a device is forwarded on to a second device by the host (in which case the two devices can be said to be "logically" connected). While the latter may be desirable in some scenarios, we note that during the time that the devices are on the surface, a connection to the host will be necessary to provide some feedback in the form of the projected display. Establishing a direct connection to the device may be a useful strategy in cases where the users are more likely to trust each other than the host system (as in a public kiosk), in which case sensitive data would

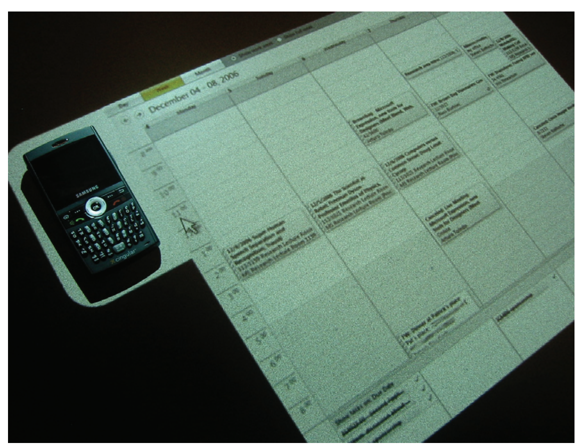

Figure 10. Mockup of a complex user interface (calendar application) that spills out of a mobile device onto BlueTable.

be transferred over the direct connection, and only enough information to construct useful feedback (or anonymized data) would be sent to the host.

7 OBSERVATIONS WITH USERS

While we have conducted no formal user study to evaluate user performance with BlueTable, we have demonstrated an early version of the system to several hundred technically-inclined adults and a number of children. In these sessions usually one of the authors took a few photos with a camera phone, placed the phone on the surface to show those photos appearing on the surface, and enlarged and rotated a photo using their hands. In some cases the participants took the pictures and put the device on the surface themselves. A few observations may be drawn from conducting this large number of demos.

First, it is interesting to note that the complete interaction of taking a picture with the cameraphone to the moment of interacting with the photos on the interactive surface is so effortless that it does not disrupt any ongoing conversation with the operator of the phone and any observer. In general, the only time the operator's attention is "captured" by the device is during the process of framing the shot to take. This stands in stark contrast to most mobile device interactions which command a great deal of attention to small buttons and user interface controls.

Secondly, observers have no trouble understanding that the connection of the device to the system is a consequence of having placed the phone on the device, and that the displayed photos were subsequently transferred from the phone. More conventional means of connecting devices wirelessly are far more difficult to discern as an observer. The comparatively transparent nature of BlueTable's interaction may positively impact collaborative interactions: because an observer can see the connection being physically made and broken, we argue that collaborative interactions will be easier. For example, an observer may be more likely to add their device to the surface in order to exchange documents if they can see their partner connect, and also if they know that they can revoke the connection at any time by removing their device from the surface. The fact that even children can use the system suggests that people unfamiliar with complex new technologies such as wireless communication would be able to use BlueTable.

Part of the appeal of BlueTable may be the ease with which documents locked up on mobile devices are moved to a more public form. Small groups of friends for example, enjoyed the experience of taking pictures and seeing them immediately appear on a large display.

Not surprisingly, while observers easily understand how to use BlueTable, very few are able to discern the technical mechanics of the sensing and handshaking process. Some people tried to put their own phone on the surface, expecting something to happen. Many observers are wowed by the BlueTable demonstration. While such reactions encourage engagement with this technology, this excitement for novelty is difficult to overcome for the purposes of objective evaluation.

8 CONCLUSION

A technique to automatically associate and connect a wireless mobile device with a vision-based interactive surface is presented. The technique relies on a visual handshaking procedure to determine which of the available Bluetooth devices are on the interactive surface. Devices are connected to the host system (and possibly to each other) by virtue of having been placed on the surface by the user. We believe this interaction grounds the act of connecting devices in a way that is immediately obvious to users who would otherwise find conventional association methods too difficult to use. Furthermore, because the combination of the handshaking technique and interactive surfaces allows the tracking and precise positioning of multiple devices on the surface, numerous interactive scenarios are enabled, such as the display of documents and user interface elements that are associated with the device, and the connection and interaction with multiple devices.

REFERENCES

[1] Ballagas, R., Rohs, M., Sheridan, J. and Borchers, J., Sweep and Point & Shoot: Phonecam-Based Interactions for Large Public Displays. in *CHI '05 Extended Abstracts on Human Factors in Computing Systems*, (2005), 1200-1203.

[2] Ballard, D., and C. Brown *Computer Vision*. Prentice-Hall, 1982.

[3] Barton, J.J., Nylander, S., Folowosele, F. and Harrison, B., Dialing for Displays: Session Initiation Protocol for Opportunistic Augmentation. in *Pervasive Computing and Communications, 2006 (Percom 2006)*, (2006).

[4] Corner, M. and Noble, B., Zero-interaction authentication. in *MobiCom 2002*, (2002).

[5] Hinckley, K., Synchronous Gestures for Multiple Users and Computers. in *ACM UIST 2003 Symposium on User Interface Software & Technology*, (2003), 149-158.

[6] Holmquist, L., Mattern, F., Schiele, B., Alahuhta, P., Beigl, M. and Gellersen, H., Smart-Its Friends: A Technique for Users to Easily Establish Connections Between Smart Artefacts. in *Ubicomp 2001*, (2001), 116-122.

[7] Lester, J., Hannaford, B. and Borriello, G., "Are You With Me?" -- Using Accelerometers to Determine if Two Devices are Carried by the Same Person. in *2nd International Conference on Pervasive Computing*, (Linz, Austria, 2004).

[8] Moore, D.J., Want, R., Harrison, B.L., Gujar, A. and Fishkin, K., Implementing Phicons: Combining Computer Vision with Infrared Technology for Interactive Physical Icons. in *ACM Symposium on User Interface Software and Technology (UIST)*, (1999), 67-68.

[9] Patel, S., Pierce, J. and Abowd, G., A Gesture-Based Authentication Scheme for Untrusted Public Terminals. in *Proceedings of the 17th annual ACM symposium on User interface software and technology*, (2004), 157-160.

[10] Pering, T., Ballagas, R. and Want, R. Spontaneous marriages of mobile devices and interactive spaces. *Communications of the ACM*, *48* (9). 53-39.

[11] Rekimoto, J., Ayatsuka, Y. and Kohno, M., SyncTap: An Interaction Technique for Mobile Networking. in *Mobile HCI 2003*, (2003), Springer, 104-115.

[12] Rekimoto, J. and Saitoh, M., Augmented Surfaces: A Spatially Continuous Workspace for Hybrid Computing Environments. in *Proceedings of CHI 99*, (1999), 378-385.

[13] Rukzio, E., Leichtenstern, K., Callaghan, V., Schmidt, A., Holleis, P. and Chin, J., An Experimental Comparison of Physical Mobile Interaction Techniques: Touching, Pointing and Scanning. in *Eighth International Conference on Ubiquitous Computing (Ubicomp 2006)*, (2006).

[14] Scott, D., Sharp, R., Madhavapeddy, A. and Upton, E., Using Visual Tags to Bypass Bluetooth Device Discovery. in *ACM SIGMOBILE Mobile Computing and Communications Review*, (2005), 41-53.

[15] Shen, C., Everitt, K.M. and Ryall, K., UbiTable: Impromptu Face-to-Face Collaboration on Horizontal Interactive Surfaces. in *ACM International Conference on Ubiquitous Computing (UbiComp)*, (2003).

[16] Sugimoto, M., Kusunoki, F. and Hashizume, H., Supporting Face-to-Face Group Activities with a Sensor-Embedded Board. in *CSCW Workshop on Shared Environment to Support Face-to-Face Collaboration*, (2000).

[17] Wilson, A., PlayAnywhere: A Compact Tabletop Computer Vision System. in *Symposium on User Interface Software and Technology (UIST)*, (2005).