In Proc. of IEEE Conference on Pervasive Computing and Communications. Dallas Fort-Worth, Texas, March 23-26 2003. (Best paper, Mark Weiser Award)

Steerable Interfaces for Pervasive Computing Spaces

Gopal Pingali, Claudio Pinhanez, Anthony Levas, Rick Kjeldsen, Mark Podlaseck, Han Chen, Noi Sukaviriya
IBM T.J. Watson Research Center
19 Skyline Drive, Hawthorne, NY 10532 USA
gpingali@us.ibm.com

Abstract

This paper introduces a new class of interactive interfaces that can be moved around to appear on ordinary objects and surfaces anywhere in a space. By dynamically adapting the form, function, and location of an interface to suit the context of the user, such steerable interfaces have the potential to offer radically new and powerful styles of interaction in intelligent pervasive computing spaces. We propose defining characteristics of steerable interfaces and present the first steerable interface system that combines projection, gesture recognition, user tracking, environment modeling and geometric reasoning components within a system architecture. Our work suggests that there is great promise and rich potential for further research on steerable interfaces.

.

1. Introduction

Pervasive computing research has been driven by the vision [25] of numerous computational devices weaving themselves into the fabric of space. An essential element to realize this notion of "computing woven into space" is a ubiquitous interface to computing – access to pervasive computing resources should be available everywhere. This has naturally been a dominant theme in ubiquitous and pervasive computing research [2,3,4,7,20,25]. However, realizing such an interface has, thus far, meant wiring the environment and/or the people in the environment. For instance, access to computing services has been through computer monitors, touch screen panels, keyboards, mice, PDAs, cellular phones etc. – all special surfaces and devices available in the environment or carried by people.

In this paper, we pursue an alternate vision for the pervasive computing interface, especially in the context of intelligent environments. We propose that as computing disappears into the physical environment, what matters most to the user is the interface to computing. The interface should appear whenever the user needs it, wherever the user needs it, and in a form most suitable for natural interaction. In particular, both input and output interfaces should be available to the user anywhere in space, without the need for special devices. To realize this, we introduce the concept of a steerable interface – an interface to computing that can be moved around a physical environment on to ordinary objects or surfaces.

Steerable interfaces are important as they provide interaction wherever it is needed in a space. In many cases, the interface just appears when needed and where needed, as a natural extension to the physical environment, without the user having to perform any deliberate actions. And as the user neither carries nor approaches any special devices, the interaction is casual. The user can also request for the interface anywhere through natural actions such as simply asking or making a specific gesture.

Steerable interfaces have the potential to change how we access information in a number of different domains and applications. For example, during shopping, information about a product can be made available right at the product location when a shopper is in the vicinity of the product. At home, a television or a computer can appear on demand on any wall or table or countertop. Information can be made truly accessible to people with special needs in hospitals and old age homes. The design of offices and the style of working and collaborating in the office can change significantly as the dependence on a static display monitor is removed [13]. Engineering design, training, and technical support can achieve a new dimension when graphical information can be overlaid upon real products and objects and combined with spoken information. For example, a product assembly manual can take the form of dynamic audio-visual instructions that appear on the products and parts themselves, and guide the assembly process. Steerable interfaces can also result in new forms of entertainment such as games in the real world with virtual characters and hybrid theater combining the real and the virtual.

The rest of this paper is organized as follows. In Section 2, we outline the salient characteristics of steerable interfaces and give an overview of the important enabling technologies needed to realize such interfaces. Section 3 discusses related research in pervasive computing and a number of other disciplines. Section 4 gives an overview of the technologies and architecture that we have implemented thus far. In Section 5, we describe a prototype implementation of steerable interfaces for a retail store application. We conclude and discuss future research directions in Section 6.

2. Steerable Interfaces: Characteristics and Enabling Technologies

2.1 Steerable Interface Defined

We offer the following definition of a steerable interface – an interactive interface to computing that can move around a physical environment and appear on ordinary objects or surfaces or even in empty space.

2.2 Characteristics of Steerable Interfaces

While the above gives a base definition of a steerable interface, we propose that an interface should ideally exhibit several salient and desirable characteristics to qualify as a steerable interface. Below, we outline six key characteristics of steerable interfaces.

1. Moveable output interface: This is a primary characteristic of a steerable interface. By definition, the interface should be able to steer information around an environment. Examples of interfaces that satisfy this characteristic are a visual information display that moves around an environment, directional sound that can be steered to appear to be coming from different source locations, or a sound beam that can be directed to different target locations in an environment.

2. Moveable input interface: We propose that a steerable interface should serve as both input and output interface to computing. Thus, the moving display in the above example would not by itself qualify as a steerable interface. The moving display should also offer means to interact with it as it moves around a physical space. Thus, a far-field speech recognition system that allows speech-based interaction with a moving display, or a moving projected keyboard that allows the user to "type" in commands would be examples of moveable interfaces that support both input and output.

3. Adaptation to user context: One of the most important motivations for steerable interfaces is that the user can get access to computing wherever he or she wants. As opposed to the typical fixed interfaces of today, such as a monitor with a mouse and a keyboard, which enslave the user to an access point, steerable interfaces free the user to move about and carry out other functions. However, this places the burden on the steerable interface to adapt to user context. For example, an interactive display should be oriented towards the user and be close enough to the user to facilitate interaction. A related theme is that, depending on the situation, the interface should either automatically appear or provide the user mechanisms to "call up" the interface whenever desired.

4. Adaptation to environmental context: This is a closely related issue to the third characteristic above. A steerable interface should adapt to the characteristics of the environment in order to be useful. For example, a projected visual display should appear on an available surface in the environment that is of the appropriate size and orientation, has the appropriate color and texture to show the display with sufficient contrast, and is not occluded by other objects in the environment. Similarly an acoustic output should adapt to the presence of acoustic noise sources and objects blocking the acoustic signal. Thus, a steerable interface typically needs to be aware of, and reason about, the geometry and properties of the environment, and relate the user as well as input and output devices to a model of the environment.

5. Device-free interaction: We propose that a steerable interface should be able to move on to different surfaces and objects and areas in an environment without the need to place special devices in these places. Thus, it is important that the steerable interface be able to sense and support forms of interaction such as speech, hand gestures, motion of body, touch etc. which are based on the human body and do not require special devices. However, it should be stressed that steerable interfaces could very well accommodate special purpose devices in addition to providing device-free interactions.

6. Natural interaction: This is a requirement of any interface and should be specially emphasized in the context of a steerable interface. A steerable interface should be "natural" and easy to use. The end result of ubiquitous access and adaptation to user and environmental context should be an interface that is intuitive and usable rather than something that distracts, confuses, or overwhelms the user.

2.3 Enabling Technologies

Several technologies need to work in unison in order to realize steerable interfaces with the characteristics outlined above. Figure 1 shows a conceptual overview of some of the important technologies. Steerable projection [17] is a key enabler as it gives the ability to move a display on to ordinary objects and surfaces. Combining a steerable gesture recognition/computer vision system [12] with steerable projection results in interactive displays anywhere in the environment. While projection and vision can create a visual input/output interface anywhere in the environment, steerable audio and steerable microphones (with associated speech recognition) can create an audio input/output interface anywhere. Technologies for steerable sound [18] and steerable microphones [21] have begun to mature in recent years. The goal is to provide the user with an audio-visual interface anywhere and the ability to switch easily between visual and auditory modes of interaction.

A crucial aspect of a steerable interface system is the ability to provide the interface at a location close to the user. This demands that the system know the location of the user, the environment around the user, and be able to reason about where to locate the interface. Figure 1 shows additional modules to realize such intelligent steering. A user-tracking module continuously provides the location and orientation of the user(s) while a 3D model of the environment provides knowledge of the surfaces and objects in the environment. A calibration module maps data from the projection, vision, or audio modules to the 3D model of the environment, while a geometric reasoning module relates user position and the environment model to determine the location or potential locations for the interface.

A continuously updated model of context [4] is central to a steerable interface system. The model of context or the “world model” should include a 3D model of the environment, locations of users in the environment, and the state of the system including calibration parameters of the various projection, vision, and audio subsystems. As seen in Figure 1, individual components both feed in context information and feed off the context information coming in from the other components. A key challenge in steerable interface systems is to model and update context, and to coordinate the various components based on the current context. Finally, applications should be able to access the world model, and be able to control and access the services of the various components. Hence, steerable interface systems should provide an application programming interface (API) to the world model and the various components, as well as tools to simplify development of applications through the coordination of such diverse and distributed components.

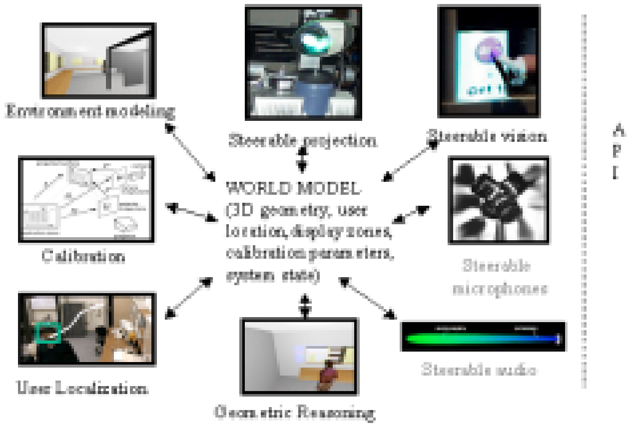

Figure 1. An overview of technologies needed to realize steerable interfaces.

3. Related Work

Considering the breadth of the research effort involved in realizing steerable interfaces, there has been significant amount of related work in a number of different disciplines. In the context of intelligent environments and pervasive computing, the EasyLiving effort [20] is close to ours in tracking people and automatically activating devices in close proximity to the user and its emphasis on the need for a geometric model [2]. However, our work significantly extends this approach by introducing the notion of steerability, developing mechanisms for steering the interface, and by formulating the geometric reasoning needed to support such steerability. Several efforts [4,7] have developed models for context in pervasive computing environments and addressed the need for deriving context from a variety of sources. However, these efforts have not addressed context and reasoning that includes a 3D model of the environment, which is crucial to realizing steerable interfaces. The work on ActiveSpaces [3] develops an infrastructure that supports the notion of a “room as a computer” and addresses how services can be provided on appropriate devices based on user context. Again, this work does not currently address the high resolution modeling of the environment and user location, which is needed to support steerable interfaces, nor true activation of devices and surfaces in close proximity to the user.

Besides these efforts in pervasive computing, the individual component technologies involved in steerable interfaces are active areas of research. Several efforts [19, 22] have looked at augmenting reality and providing large scale displays using projection systems. Gesture recognition, as a means for vision-based user interfaces, has been a very active area of research [26] although only a few efforts have involved interacting with projected images, and none has addressed the dynamic reconfiguration needed for steerable interfaces. People tracking through cameras is another area which continues to draw a lot of attention [14]. However, it is still challenging to reliably and continuously track people in the presence of dynamic projected imagery. Determining the gaze of the user [6] and the orientation of the user’s head [27] can be very useful for steerable interfaces. Geometric reasoning has a rich history in robotics and artificial intelligence [11]. 3D calibration of perspective imaging devices has been addressed in [5]. Steerable interfaces demand further advances in dynamic calibration and coordinated calibration of multimodal sensors and display devices.

On the audio side, steerable speech interfaces have drawn a lot of attention including efforts on steerable microphones [21, 9] and steerable loudspeakers [24] and steerable sound beams [18]. Much needs to be done in combining visual and acoustic steerability for both input and output. Multimodal speech localization and speech recognition efforts [15] need to be extended to function for steerable interfaces, and to take advantage of the geometric reasoning and modeling capabilities that are integral to such interfaces. Speech and gesture also need to be combined as in the initial work in [1]. Finally, the notion of steerable interfaces is related to several efforts on interactive techniques and interfaces in the computer graphics and human-computer interaction (HCI) communities [23, 10].

4. Current implementation

4.1 The Everywhere Display projector

Figure 2. The Everywhere Display projector.

Figure 2 shows a device called the Everywhere Displays projector (ED-projector) [17] which integrates an LCD projector and a computer-controlled mirror to steer a display on to any surface in an environment. As an image is steered on to a surface, it has to be pre-corrected for oblique projection distortion, in order to appear free of keystone distortion to a viewer orthogonal to the surface, as discussed in [17]. Our current implementation of the ED-projector uses standard graphics hardware to perform the distortion correction in real time. Thus, dynamic imagery and video can also be steered on to different surfaces using the ED-projector. As seen in Figure 1, a steerable camera is also combined with the steerable projector. The camera is used to recognize a user's hand gestures to allow user interaction with the projected display as discussed below.

4.2 Steerable Gesture Recognition

We have developed a dynamically reconfigurable vision-based user interface system to enable interaction with the images projected by the ED-projector. The application sends the vision system a functional description of the user interface as a configuration of widgets (describing what the interface is). Each configuration is a collection of interactive widgets, in a structure similar to how a traditional windows-based application is defined as a set of dialog windows, each containing elements such as scroll bars, buttons and menus. In the case of the vision-based user interface, each widget provides an elemental user interaction, such as detecting a touch or tracking a fingertip. Widgets generate events back to the controlling application where they are mapped to control actions such as triggering an event or establishing the value of a parameter. Based on this functional description, the vision system assembles a set of vision processing components that implement that interface, sharing computational resources when possible. Switching configurations is easily achieved by simply sending a new set of functional descriptors. The dynamically assembled vision processing components perform functions such as motion detection, region correlation, shape matching, and motion analysis. More details on how such components are used to detect finger based interactions and hand tracking can be found in [12].

In addition to supporting dynamic reconfiguration of the function of the interface, the vision system also provides for dynamic steering of the interface onto different real-world planar surfaces. In our system, the parameters of the surfaces where the interface is realized, such as size, position in space, perspective distortion, and even the user's likely position, are defined and stored independently of any particular interface. When the application requires a given interface to be active on a particular surface (that is, Where the interface should appear in the environment) the system automatically propagates the surface-specific parameters into the assembly of vision processing components that implements that interface.

This approach is powerful and lends great flexibility in developing steerable interfaces. 1) An application can easily be ported to a new environment where the surface interface is different. 2) The same surface can be used differently by different applications. 3) The same interface can be reused on multiple surfaces.

4.3 User Localization and Tracking

We adopt a real-time camera-based head tracking technique to determine the position of the user in the environment. The head tracking technique is based on motion, shape, and flesh-tone cues. We first perform a differencing operation on consecutive frames of the incoming video and threshold the result. A morphological closing operation then removes noise and fills up small gaps in the detected motion regions. A standard contour-tracing algorithm then yields the bounding contours of the segmented regions. We then smooth the contours and compute the orientation and curvature along the contour.

We then analyze the shape of each contour to check if it could be a head. We look for curvature changes corresponding to a head-neck silhouette (concavities at the neck points and convexity at the top of the head) with the head pointing up and for the circularity of a head shape. We then check for sufficient flesh-tone color within the detected head region by matching the color of each pixel within the head contour with a model of flesh tone colors in normalized r-g space. This technique detects multiple heads in real time. In the current system, we track a single user in the environment. More details can be found in [16]. We use multiple cameras with overlapping views to triangulate and estimate the 3D position of the user. Our system tracks user position to accuracy within a few inches – well within a foot, and operates at video frame rate. This high-resolution localization of the user in 3D allows us to reason about the appropriate surface for displaying the interface and about user occlusion of the display, as discussed below.

4.4 3D Environment Modeling

We have developed a modeling toolkit for creating and manipulating an environment model to support steerable interfaces. This modeling toolkit is in many ways a simplified version of standard 3D modeling software. It supports basic geometric objects built out of planar surfaces and cubes and allows importing more complex models. However, the toolkit provides additional objects such as projectors and projection display zones and annotation capabilities that are specifically required for steerable interfaces. Almost every object in the model can be annotated. This makes it possible to attach semantics to objects such as optical properties of a surface and its preferred usage. The toolkit stores the model in XML format, with objects as tags and annotations as attributes. The XML format allows the model to be easily defined and manipulated by applications as discussed in the architecture in Section 4.6. The modeling toolkit is also designed to be accessible to a geometric reasoning engine as discussed below.

4.5 Geometric Reasoning

We have developed a geometric reasoning engine that operates on a model created by the modeling toolkit described above. The main purpose of this reasoning engine is to enable automatic selection of the appropriate display and interaction zones based on criteria such as proximity of the zone to the user and non-occlusion of the zone by the user or by other objects. Applications or other modules can query the geometric reasoning engine through a defined XML interface. We currently support two types of queries. The first type of query is a property look-up action in which the geometric reasoning engine gives all the properties of a specified display zone given the location of the user in the environment. The user location is derived from the tracking module described earlier. In the second type of query, the reasoning engine receives a user position and a set of criteria, specified as desired ranges of display zone properties, and returns all display zones which satisfy the specified criteria.

The properties for a display zone include the following: 1) Physical size of the display zone in some specified units such as inches or centimeters. 2) Absolute orientation defined as the angle between the surface normal of the display zone and a horizontal plane. 3) User proximity defined as the distance between the center of the user's head and the center of a display zone. 4) Position of the user relative to the display zone, defined as the two angles to the user's head in a local spherical coordinate system attached to the display zone. This indicates, for example, whether the user is to the left or to the right of a display zone. 5) Position of the display zone relative to the user, defined as the two angles to the display zone in a local spherical coordinate system attached to the user's head. 6) Occlusion percentage, which is defined as the percentage of the total area of the display zone that is occluded with respect to a specified projector position and orientation. 7) An occlusion mask, which is a bitmap that indicates the parts of a display zone occluded by other objects in the model or by the user.

4.6 Current System Architecture

Figure 3 illustrates our current system architecture for steerable interfaces. It is a three-tier architecture composed of a Services Layer, an Integration Layer and an Application Layer. Each of the modules in the Services layer exposes a set of core capabilities through a simple HTTP/XML Application Programming Interface (API). Modules in the Services Layer have no "direct" knowledge or dependence on other modules in that layer. The modules share a common XML language called the Everywhere Display Markup Language (EDML) along with a specialized dialect for communication with each module in this layer. Applications written in any language can directly communicate with each module in this layer using EDML.

Currently the Services layer is composed of six modules. The Vision Interface (VI) is responsible for recognizing gestures and converting this information to the application using events. The Projection module (PJ) handles the display of visual information on a specified surface while the Steerable Camera module (SC) provides the video input from the surface of interest to the VI. An interaction with the steerable interface is accomplished by orchestrating the VI, PJ and SC modules through a sequence of synchronous and asynchronous EDML commands. Other modules present in the Services Layer are the 3D Environment Modeling module, the User Localization module, and a Geometric Reasoning module, as discussed in the previous sections.

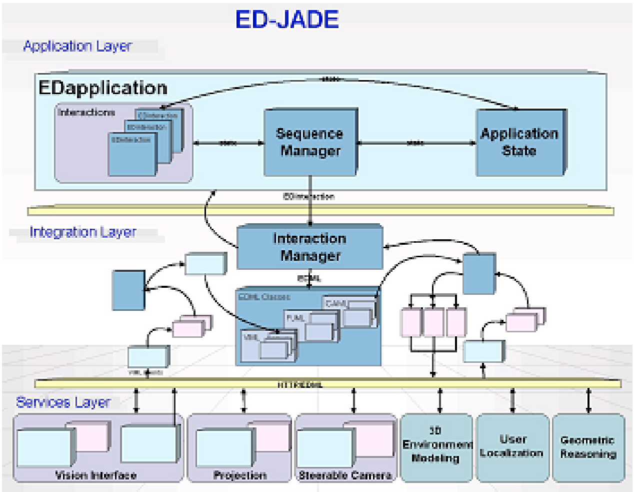

Figure 3. Overview of the current architecture for steerable interfaces.

The Integration Layer provides a set of classes that enable a JAVA application to interact with the services in an easier manner. It contains a set of JAVA wrapper objects for all EDML objects and commands, along with classes enabling synchronous and asynchronous communication with modules in the Service Layer. This layer mediates the interaction among Service Layer modules and exposes new function created through the orchestration of the capabilities available in the Services Layer. For example, through a single instruction to the Interaction Manager, a JAVA application can start an interaction that sends dozens of commands to the Vision Interface (VI), the Projection module (PJ) and the Steerable Mirror (SM) defining, instantiating, activating, and managing a complex interactive display interaction. Similarly, the Tracker class can coordinate the User Tracker, the Geometric Reasoner, and the 3D Environment Modeler in a manner that returns the current user position along with all occluded surfaces to the application at a specified interval. The Event Manager in this layer also receives Interaction events, issued by the VI module, and transforms them into JAVA widget events for the application.

The Application layer consists of a set of classes and tools for defining and running JAVA applications and a repository of reusable Interactions. ED interactions are defined using a specialized JAVA BEAN Composition editor that simplifies the task of interaction definition. Each interaction is a reusable class that is available to any application. An example of a useful reusable interaction is a commonly used yes/no confirmation dialog. The Application class is a container for composing multiple interactions, maintaining application state during execution, and controlling the sequence of interactions through the help of a Sequence Manager. We envision other tools in this layer to expedite development of steerable interface applications. One example is a Calibrator tool that allows a developer to easily calibrate the VI, PJ and SC modules for a particular application.

4.6.1 EDML. The core EDML definition includes commands for establishing communication with a module, starting and stopping their respective services, and commanding or querying the module. Basic EDML action commands fall into 3 logical pairs: Use/Release are used for definition/allocation and de-allocation of objects (e.g. Buttons, Images, etc.). Set/Get are for setting or retrieving values of objects and Activate/Deactivate for activation and deactivation of “Used” (allocated) objects.

VIML is the specialization of EDML for the VI module and consists of three basic objects. VIsurface for defining attributes of a surface, VIconfiguration for defining widgets, their spatial relationships, and for elaborating their behavior, and VIevents for communicating events, such as a button press, back to the application.

The XML example below directs the VI to use the current surface called “wall” and to use the VIconfiguration called “cfg1”. It defines the configuration as having an internal coordinate frame 500 units in x and y, and to contain a button named “done”, which is located at x=200, y=200 in that frame and to be 50 units large. A track area is also defined which is 100 units in x and y and located at 0,0 with respect to the configuration’s coordinate frame.

```
<use id="uniqueID1001">
<VIsurface name="wall"/>
<VIconfiguration name="cfg" left="0" right="0"
   top="500" bot="500">
   <VIbutton name="done" x="200" y="200" size="50" >
   <VItrackArea name="T1" left="0" right="0" top="50"
      bot="50" >
</VIconfiguration >
</use>
```

After a surface is “Used” VIML commands can be issued to “Set” or change values. A configuration is not active until an “Activate” command is issued, upon which the configuration begins to monitor its widgets and return relevant events to the application.

5. Example Implementation: The Everywhere Sales Associate

To illustrate steerable interfaces in a specific domain, we have implemented a prototype retail environment with an “Everywhere Sales Associate” (ESA) – a virtual provider of information and help throughout the store. The ESA can manifest itself in various forms, sometimes casually providing context-sensitive information and at other times appearing on demand at the customer’s behest and providing specific interactive assistance. Here we will describe three different manifestations of the ESA; all made possible by our steerable interface system.

The table of shifting perspectives: This is an ordinary table with products (books) on it. When a customer approaches the table (Figure 4), the ESA appears as a display projected on the table and provides information about the books on the table. The display is located on the part of the table that is close to the customer, is oriented towards the customer, and provides additional information about products close to the customer (e.g., highlighting the books related to “ghosts”). As the customer walks around the table, the display moves around the table to be always close to, and oriented towards the customer, and shows information relevant to the products close to the customer. In our current implementation, the displays on different parts of the table also provide different perspectives on the same

Figure 4. The “table with shifting perspectives”. A user walking around the table is presented with different perspectives on the products.

products. When the customer walks away from the table, the display disappears from the table.

This application utilizes the high-resolution tracking and geometric reasoning capabilities of the steerable interface system. It adapts the location, orientation, and the focus settings of the display based on user position.

The ubiquitous product finder: Unlike the casually appearing display in the table described earlier, the ubiquitous product finder is an example of the ESA appearing on customer demand anywhere in the store. The product finder allows a customer to look up products in a store directory, and then guides her to where the product is. This product finder is accessed in two forms. The first form is a table that is at the entrance to the store, much like the directory often found at the entrance to a mall. A customer can get help by merely touching the table, upon which an index of products appears on the table. The table has a physical slider (see figure 4.a) that the customer can manipulate to easily navigate the product index. On finding a product of choice, the customer touches a projected button to get the directions to that product. The directions appear as a sequence of images starting at the table and moving across different surfaces in the store to show the path to the selected product.

The product finder is also available to the customer as she walks in the store, in the form of paper signs located all around the store. Touching the sign (figure 4.b), which is initially blank, causes it to become an interactive display of the product guide. Product search is accomplished in much the same way as on the table, except that the customer slides her finger (figure 4.c) instead of moving a physical slider. Note that there are no physical sensors attached to the table, the slider or the paper signs. These are all activated by the steerable interface system.

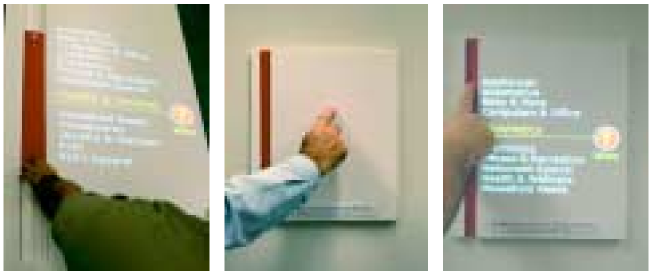

Figure 5. The Ubiquitous Product Finder. (a) projected on to a table with a physical slider; (b) user “calls” the application; (c) the paper tag becomes an interactive product finder.

In the current system, a single pan/tilt camera monitors three different sign surfaces by using the information from the person tracking system. It automatically aims the camera to the sign or table nearest to the user. Then, it activates the “call” configuration on that sign’s surface. In this way the system is always ready for the user to “call” the Product Finder. When the user touches a sign or table, the “call button” widget sends an event to the application, which then projects the “selection” graphics on the sign, while activating the corresponding configuration on the sign’s surface.

Customer-following sale signs: In a third manifestation, the ESA combines casual appearance with on-demand interaction. When the customer is in the vicinity of a certain product, the ESA appears as a sale sign to attract the attention of the customer towards that product (Figure 6 (a)). As the customer approaches the product, the sign changes to display more information, and when the customer is close enough, it offers the customer the option of interacting with the sign to obtain more details. For example, the customer can touch a button to get more information on the features of a product. A special feature of the ESA is that it automatically responds to situations where the customer occludes the projected display of information. In this example, when the customer occludes the interactive sign (Figure 6 (b)), it automatically moves to a nearby non-occluded surface and also moves the interaction button to be convenient for user interaction (Figure 6 (c)).

This example illustrates the use of user tracking information as well as geometric reasoning to determine occlusion of display zones. The application adapts the position of the display, the content of the display, and the positioning of the interaction widgets on the display based on user, environmental, and application context.

7. Concluding Remarks

In this paper we introduced the concept of steerable interface systems that augment physical environments with casual interfaces without the need for special devices on people or on surfaces. Realizing such interfaces demands new techniques for both output and input such as steerable projection and steerable gesture recognition. Steerable interfaces also demand much higher resolution in user localization and environment modeling and much more sophisticated geometric reasoning than supported in current pervasive computing architectures. We have outlined the characteristics, requirements, and challenges of steerable interfaces and presented our current implementation of a steerable interface system. This implementation exhibits the six qualifying characteristics

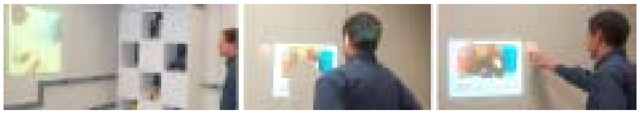

Figure 6. Customer-following sale signs. (a) A sale sign attracts the customer; (b) As the user comes closer, the sign changes and becomes interactive; (c) detecting user shadowing in (b), the system moves the display and the button.

that we proposed for steerable interfaces. However, a significant challenge is the "natural interaction" characteristic. While we have taken significant steps through hand gesture interaction and support for physical props that ease interaction, we need to conduct user studies to determine what kind of interactions users really prefer in such systems. This is an important agenda for our research.

While this paper primarily promoted the notion of interfaces without special devices, the steerable interface theme could also be complementary to the presence of special-purpose devices. For example, a wearable device such as a cell phone or a watch could provide user identity and other user context while a steerable interface can compliment this by a) providing access to information in the specific context of the physical environment (such as product information local to a product), and b) providing convenient access to information on a device (such as providing a large interactive screen for a watch computer whenever and wherever it is needed). We believe that the architecture presented here should be integrated with existing approaches [3,4] to provide more comprehensive interfaces that extend the capabilities of current systems.

We see our work as a first step towards realizing a vision of more convenient user interfaces in physical spaces. Further advances in steerable interfaces require significant interdisciplinary efforts combining expertise in graphics, acoustics, vision, speech, robotics, usability, human-computer interaction, and distributed computing architectures, to name a few! We hope to see much more research activity and effort to realize robust and usable steerable interface systems that become commonplace in the real world and give a new dimension to pervasive computing spaces.

References

1. R. Bolt. Put That There: Voice and Gesture at the Graphics Interface. ACM Computer Graphics, 14(3): 262-270, 1980.
2. Brumitt, B., et al. Ubiquitous Computing and the Role of Geometry. In IEEE Personal Communications, August 2000.
3. Renato Cerqueira, Christopher K. Hess, Manuel Román, Roy H. Campbell. Gaia: A Development Infrastructure for Active Spaces. In Workshop on Application Models and Programming Tools for Ubiquitous Computing (held in conjunction with the UBICOMP 2001), September 2001, Atlanta, Georgia.
4. Dey D. Salber, and G. Abowd. A context-based infrastructure for smart environments. In 1st International Workshop on Managing Interactions in Smart Environments (MANSE'99). 1999.
5. O. Faugeras. Three-Dimensional Computer Vision: A Geometric Viewpoint. The MIT Press, Cambridge, Massachusetts, 1993.
6. Gee, A. and Cipolla, R., Determining the gaze of faces in images. Image and Vision Computing, vol. 12, no. 10, p. 639-647, 1994.
7. K. Hendricksen, J. Indulska, and A. Rakotonirainy. Modeling Context Information in Pervasive Computing Systems. In Proceedings of Pervasive 2002, LNCS 2414, pp. 167-180, 2002.
8. Christopher K. Hess, Manuel Román, and Roy H. Campbell. Building Applications for Ubiquitous Computing Environments. In International Conference on Pervasive Computing (Pervasive 2002), pp. 16-29, Zurich, Switzerland, August 26-28, 2002.
9. Y. Huang, J. Benesty, and G. W. Elko. Microphone arrays for video camera steering. Chapter 11 in Acoustic Signal Processing for Telecommunication, S. L. Gay and J. Benesty, Eds., Boston, MA: Kluwer Academic, 2000.
10. H. Ishii and B. Ullmer. Tangible Bits: Towards Seamless Interfaces between People, Bits, and Atoms. In Proc. of CHI'97, Atlanta, Georgia, p. 234-241, 1997.
11. Kapur, D., Mundy, J. L. (eds.): Geometric reasoning. Special issue of Artif. Intell. 37(1-3), The MIT Press, Cambridge (1989).
12. Kjeldsen et al. Interacting with Steerable Projected Displays. In 5th International Conference on Automatic Face and Gesture Recognition. 2002. Washington, DC.
13. J. Lai, A. Levas, P. Chou, C. Pinhanez, M. Viveros. BlueSpace: Personalizing Workspace Through Awareness and Adaptability. In International Journal of Human-Computer Studies, vol. 57(5): 415-428, 2002.
14. T.B. Moeslund and E. Granum. A Survey of Computer Vision-Based Human Motion Capture. Computer Vision and Image Understanding, Vol. 81, No. 3, 2001.
15. C. Neti, G. Potamianos, J.Luettin, I. Matthews, D. Vergyri, H. Glotin, J. Sison, A. Mashari, J. Zhou., "Audio-visual speech recognition" Final Workshop2000 Report, CLSP Johns Hopkins University, Baltimore, 2000.
16. Pingali, G., et al. User-Following Displays. in Proc. of the IEEE International Conference on Multimedia and Expo 2002 (ICME'02). 2002. Lausanne, Switzerland.
17. Pinhanez, C. The Everywhere Displays Projector: A Device to Create Ubiquitous Graphical Interfaces. In Proc. of Ubicomp'01. 2001. Atlanta, Georgia.
18. F. J. Pompei. The Audio Spotlight. Journal of the Audio Engineering Society, Vol. 47, pp. 726-731, October 1999.
19. R. Raskar, et al. Shader Lamps: Animating Real Objects with Image-Based Illumination. In Proc. of 12th Eurographics Workshop on Rendering, London, England, 2001.
20. Shafer et al. The Easy Living Intelligent Environment System. In Proc. of CHI Workshop on Research Directions in Situated Computing, April 2000.
21. H. Silverman, W. Patterson, J. Flanagan and D. Rabinkin, "A Digital Signal Processing System for Source Location and Sound Capture by Large Microphone Arrays", In Proceedings of International Conference on Acoustics Speech and Signal Processing ICASSP 97, Munich, Germany, April 1997.
22. R. Sukthankar, et al. Smarter Presentations: Exploiting Homography in Camera-Projector Systems. In Proc. of ICCV'01, Vancouver, Canada, 2001.
23. J. Underkoffler, et al. Emancipated Pixels: Real-World Graphics in the Luminous Room. In Proc. of SIGGRAPH'99, Los Angeles, CA, p. 385-392, 1999.
24. D. B. Ward, G. W. Elko, Virtual Sound Using Loudspeakers: Robust Acoustic Cross-Talk Cancellation, Acoustic Signal Processing for Telecommuncation, Kluwer Academic Publishers, Chapter 14, pp. 303-317, March. 2000.
25. M. Weiser. The Computer for the Twenty-First Century. Scientific American, 265(3): 94-100, 1991.
26. Wu, Y. and Huang, T., Vision-Based Gesture Recognition: A Review. Lecture Notes in Artificial Intelligence 1739, 1999.
27. L. Zhao, G. Pingali, I. Carlbom. Real-time Head Orientation Estimation using Neural Networks. IEEE International Conference on Image Processing, September 2002.