# Applications of Steerable Projector-Camera Systems

Claudio Pinhanez, Rick Kjeldsen, Anthony Levas, Gopal Pingali, Mark Podlaseck, Noi Sukaviriya
*IBM Research, T.J. Watson*
*pinhanez,fcmk,levas,gpingali,podlasec,noi@us.ibm.com*

## Abstract

*How can interactive computer interfaces be created anywhere in a space without wiring or modifying objects or people? We propose using steerable projector-camera systems employing computer vision to realize such "steerable interfaces." In this paper, we illustrate the potential of the new kinds of applications enabled by steerable interfaces and discuss the challenges imposed on computer vision through the presentation of four application prototypes: a collaborative assembly task coordinator; a multi-surface presentation viewer; a ubiquitous product finder for retail environments; and an interactive merchandise shelf.*

## 1. Introduction

The main goal of our research is to create devices capable of realizing the *ubiquitous computing* (*ubicomp*) vision of everywhere computer access and seamless integration of computers with the real world [1]. However, unlike the most commonly proposed devices that have to be either worn or carried by the user (such as cell phones or PDAs), or embedded in the environment, we are investigating devices that can create interactive displays with minimal physical change to the user or to the environment.

In this context, using projectors to create interfaces becomes a natural candidate, especially when combined with cameras and computer vision systems able to directly detect user interaction with the projected surface. The combination projector-camera is thus able to create *projected interfaces* on most planar surfaces without requiring any modification on it.

However, if the goal is to provide multiple points of seamless computer access in an environment, it is not practical to install pairs of projector-cameras facing every possible surface. Even in the unlikely case that projectors become extremely cheap, installation of these devices will still be costly. Even worse, any change in the configuration of the environment will require re-installation and re-wiring of many devices.

To cope with those problems, we have been advocating the use of *steerable projected interfaces* [2, 3], referred in this paper simply as *steerable interfaces.* Steerable interfaces can be created, for example, by fitting a pan/tilt mirror into a LCD projector and combining it with a pan/tilt/zoom camera.

We have described the basic structure of the vision systems used in our prototypes in previous publications [4, 5]. In this paper we focus on four different application prototypes that explore the potential of steerable interfaces. These applications employ vision systems that detect point-and-click interaction, toolbar scrolling, and simple user activity recognition. However, unlike traditional vision-based interfaces, these vision systems are required to work seamlessly on distinct surfaces under very different observation angles and lighting conditions. They also have to cope with the changes in the pattern of user interaction caused by moving the interface to a new surface, which alters the relative size and orientation of the gesture to the interface. In some ways, these applications require computer vision systems much more robust and adaptable to contextual changes than currently used vision systems.

This paper presents these new computer vision-based applications that are only possible due to use of steerable projectors and cameras. As we present these applications, we analyze and discuss the challenges created by steerable interfaces to computer vision and the ways we have addressed them. We conclude by suggesting possible ways to improve the performance of steerable vision systems.

## 2. Related Work

Deviceless interaction with projected images based on video processing was pioneered in the 1980s by Krueger [6] in systems that often used a backlit white background to simplify segmentation. Real-time static background segmentation became a reality in the 1990s, notably in the ALIVE system [7] and the technique has been a cornerstone of most segmentation work in vision-based gesture recognition.

However, only a little of that work has attended to the problems of segmentation in the context of a projected display. Because the projected image drastically changes the appearance of the hand, we tend to agree with Hardenberg and Berard [8] in that background subtraction does not give reliable results in the presence of strong projected images.

Different proposals have been made to use hand or body gestures to interact with the projected images. In general, gesture-based interaction methods tend to fall into one of two categories: either variations of a point-and-click paradigm [8, 9] or application-specific pose or motion gestures [10]. Both these approaches have disadvantages. While the point-and-click approach is often not well suited to hand pointing due to limited resolution and the lack of a natural “click”, a large number of application-specific gestures makes interfaces complex: users have to learn and execute these gestures largely without feedback, as discussed in [11].

## 3. Computer Vision for Steerable Interfaces

The applications described in this paper use a device to create steerable interfaces called the *Everywhere Displays projector (ED-projector)* [2]. Figure 1 shows a prototype of this device, made of off-the-shelf components: a LCD projector, a pan/tilt mirror used in theatrical lighting, and a videoconference pant/tilt/zoom camera.

The system is currently run on two different computers, one responsible for the application software and the projector software; the other controlling the positioning of the camera and runs the computer-vision system responsible for detecting user interaction. The projector software stores and manages the calibration parameters to correct the distortion caused by oblique projection as well as the projector focus and zoom parameters. In the current implementation, these parameters are determined manually as described in [2].

Our methods to detect user activity involve the interpretation of the path of the user’s hand. Because the projection can radically change the appearance of objects moving through it, appearance-based techniques were ruled out. Instead motion is detected by looking at the differences between adjacent images in the video stream. Hands are tracked by looking for fingertips in the filtered motion data. Candidate fingertips are identified based on shape and evaluated using spatial heuristics such as orientation, distance from the user’s body, and the previous fingertip locations.

To build an interface, application designers assemble “widgets” that specify areas where users can interact. Each type of widget responds differently to the user’s motions. A touch widget looks for an in-pause-back motion of the hand. A continuous motion widget reports the relative location of a fingertip in one or two dimensions. The system can also track objects of distinct colours. As we see, our system employs traditional computer vision methods in the implementation of the basic widget set (see [4] for details).

Most real-life vision interface systems incorporate the vision system as a module that is hard-coded to operate under a fixed set of circumstances. Our system, on the other hand, requires the application to send the vision system a description of the user interface as a configuration of widgets (describing *what* the interface is). Based on these specifications, the vision system assembles a set of image processing components that implement the interface, sharing computational resources when possible. To change the interaction, a new interface description can be sent to the system at any time.

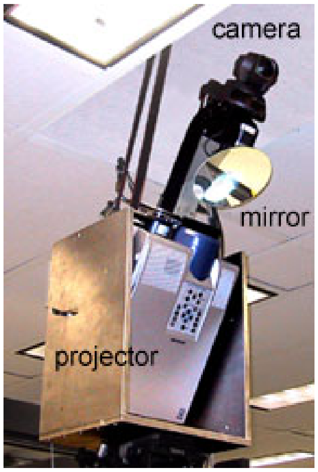

**Figure 1. Prototype of the ED-projector.**

The system, described in detail in [5], also provides for the deployment of an interface onto different real-world planar surfaces. Like Zhang et al. [12], we dynamically map the interactive interface based on the four corners of the contour of the surface. However, in our system the parameters of the surfaces where the interface can be realized are represented and referenced independently of any particular interface. These include the size, location and perspective distortion within the image and characteristics of the physical environment around that surface, such as the user’s likely position while interacting with it. When the application requests an interface to be activated on a particular surface (that is, defines *where* the interaction should happen in the environment), the system retrieves the surface parameters and propagates them through the assembly of image processing components that implements that interface.

By explicitly decoupling the information describing the characteristics of *where* an interface happens in an environment, i.e., surface-specific information, we enable: (1) the porting an application to a new environment where the interaction surfaces could be different; (2) the use of one surface by multiple applications; and (3) the use of the same interface on multiple surfaces.

## 4. Applications Using Steerable Interfaces

Steerability considerably expand the potential of projected interface systems by making them devices much more adaptable to environmental changes, especially when compared to embedded displays and monitors. For example, if the furniture of an office is rearranged, it is

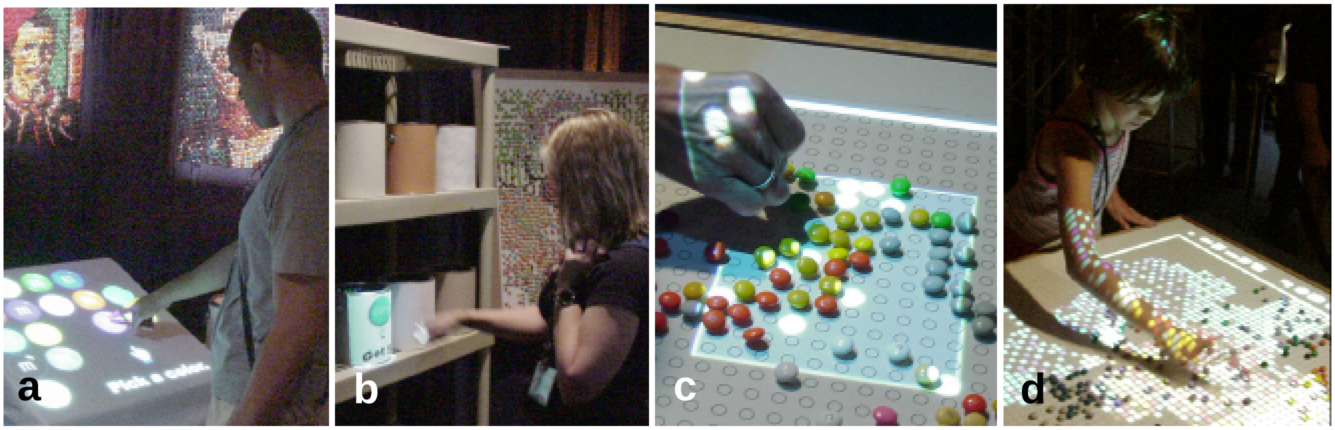

**Figure 2. The M&M picture assembly task: a) choosing a color; b) "clicking" a bucket after getting the M&Ms; c) placing the M&Ms on the highlighted areas; and d) "finger painting" to reveal the complete picture.**

necessary to rewire monitors and, many times, remove and reinstall embedded displays. Instead, if a steerable interface system is used, adaptation is simply a matter of recalibrating the position and parameters of the interactive surfaces by software.

However, the addition of steerability to a projected interface system goes beyond easy and flexible deployment of interfaces on different surfaces. It also enables a new set of applications that explore more direct connections to real-world objects. In particular, it allows the creation of augmented reality applications where information about objects is projected directly onto or around them, freeing the users from the inconvenience of wearing head displays and cumbersome input devices and avoiding the difficulty of registering the computer generated imagery to the real-world view.

The possibilities opened up by the use of steerable interfaces to create different and exciting applications come with a price. On the display side, there are issues such as reduction of perceived resolution, constraints on environment brightness, and occlusion, as discussed in [2].

### 4.1. Collaborative Assembly Task Coordinator

Steerable interfaces can augment reality without requiring users to wear goggles or other cumbersome devices. To demonstrate this premise, we developed for a the Emergent Technologies Exhibit of SIGGRAPH'01 a whimsical demonstration of a system that uses steerable interfaces to coordinate a collaborative assembly task. In this prototype application, the object to be assembled is a picture made of 3,000 M&M's (multi-colored sugar-coated chocolates) where each M&M is regarded as a pixel of the picture. Each user contributes to the task by placing 10 to 15 pixels. Here is a description of the main characteristics of this application:

**Context:** a collaborative assembly task in a factory.

**Application Goal:** to guide users to find the components to be assembled and to help placing them in the appropriate position.

**Technology:** the application uses an ED-projector overlooking the assembly area. The vision system implements widgets that detect touch events and that report continuous motion of the user's hand on a surface.

**Interaction:** as the user enters the environment, she is asked to touch a picture projected on a board to activate the system. The system responds by projecting an instruction asking the user to go to the selection area. The interface is then steered towards a table. Figure 2.a shows this table as it is transformed into an interactive menu for color selection. The user simply touches the color of his choice to select it. Then the user receives a projected instruction prompting him to look for a highlighted bucket. In Figure 2.b, the system highlights the bucket that contains the M&M's of the selected color and provides additional information on how many M&M's to extract from the bucket. The interface also includes a touchable button, where the user indicates that the retrieval of M&Ms is completed. When this button is touched, the system projects on the bucket an instruction telling the user to walk to the picture board. In Figure 2.c, the system is pointing to the exact places on the picture board where the M&M's should be placed. After the M&Ms are put in place, the user can signal the end of this task by touching a button projected on the side of the picture board. The user is then invited to wave her hand over the surface to progressively reveal the rest of the picture being built (see also the video, from the system's camera point of view, at http://www.research.ibm.com/people/p/pinhanez/download/video1.avi).

The computer vision system faced many issues in this application. The first difficulty involved the interactive menu for color selection. As shown in Figure 2.a, this menu presented a choice among 18 buttons, arranged in a

**Table 1: Performance of the vision system (130 users)**

| | success | false positives | false negatives |
|---|---|---|---|
| Entrance | 88% | 9% | 3% |
| select board | 94% | 2% | 4% |
| buckets | 55% | 33% | 12% |
| board (place) | 91% | 8% | 2% |
| board (paint) | 80% | 2% | 18% |
| total w/o buckets | 89% | 6% | 5% |
| **total** | **81%** | **12%** | **7%** |

grid configuration closely juxtaposed to each other. Correct selection was possible because the system looked for the complete gesture of touching and not simply by the presence of the hand in a particular spot. Therefore, the user could select a color in the top row without triggering the buttons on the rows below. As we see here, simple tracking without gesture recognition is often insufficient to implement even basic projected interfaces.

A second major challenge for the vision system was to realize the bucket interface dialog onto the many different buckets. At the time of this demonstration, the automatic mapping of configurations of widgets onto surfaces described in section 3 was not implemented, so each interface had to be painstakingly mapped to each surface that realized it. In fact, the difficulties experienced during this demonstration were a major factor to develop the description-based interface system described in section 3..

More than 650 novice users participated in the collaborative assembly task demonstration. Considering that the vision system and the projection system were integrated only weeks preceding the exhibition, the combined system worked remarkably well. As shown in Table 1, a sample with 130 consecutive users with 621 button touch events (touching gestures or false detections) yielded correct detection of touching gestures in 81% of the events, with 12% false negatives and 7% false positives. If the bucket events are excluded from the count, the performance exceeds 89%.

The buckets yielded a high number of errors for several reasons. The biggest problem was that while picking M&M's from the bucket, the user often partially or completely occluded the display with their head or back, accidentally triggering the button projected on the bucket. A possible way to handle this is to include in the vision system improved methods to distinguish between the shapes of hands and other body parts. Currently, the system only looks for fingertip-shaped patterns on the motion image. However, if view-based methods are used to identify body parts larger than a fingertip, the apparent shapes can vary dramatically from surface to surface due to the different relative positioning of camera.

Although we choose to use the M&Ms placement task to demonstrate steerable interfaces in augmented reality, it should be evident the applicability of this technology in real-world assembly tasks and similar augmented reality scenarios.

**Figure 3. Anywhere presentation viewer.**

## 4.2. Anywhere Presentation Viewer

Another important area of applications of steerable interfaces involves the access to desktop applications from different surfaces. A typical scenario occurs when people assemble in an office or meeting room to review electronic documents. According to the number of participants and their roles, documents are preferably viewed by individuals on monitor-like displays, by large groups of people on vertical screens, or by a small number of people in configurations that better facilitate collaborative work such as the one shown in Figure 3.

Trying to address these problems, we developed a presentation viewer that can move to different surfaces:

**Context:** office or meeting room.

**Application Goal:** to create a viewer for presentations that can be accessed on walls or tables to facilitate collaborative work.

**Technology:** an ED-projector overlooks the working area. The vision system utilizes only touch widgets; the interfaces are dynamically defined by the application and mapped automatically onto the calibrated surfaces.

**Interaction:** using a touch-screen LCD monitor, the user selects through an icon a surface in the environment to launch the application. The presentation is projected on the selected surface, topped by buttons allowing the users to select the next, the previous, or the first slides, or to dismiss the viewer. The viewer can be moved to other surface by selecting the corresponding icon on the LCD monitor.

This application clearly demonstrates the need of a vision system that can automatically map any application interface onto any given surface. In real offices and meeting rooms, tables move to different areas, walls are set up and removed, etc. Easily coping with these environmental changes is an essential feature of the vision system here.

A difficulty with this application, particularly in collaborative scenarios such as the one depicted in Figure 3, is that the context invites multi-user interaction. If that is the case, buttons have to respond to activation patterns from different directions corresponding to where each user is positioned relatively to the interface. Such ability is not currently supported by our system: it assumes that all touch widgets in an interface are accessed from the same direction. Supporting multi-directional access requires either tracking and differentiating multiple fingertip paths simultaneously, which is currently very error prone, or monitoring the location of every user. In the second case, the vision system may use contextual information provided by an environmental user positioning tracking system. We are also evaluating more sophisticated hand path representation and recognition methods.

## 4.3. Ubiquitous Product Finder

Computerized access to information in public spaces such as malls and stores has been primarily deployed through kiosks. A major advantage of using virtual display systems such as the ED-projector in such spaces is that they can be installed on the ceiling, thereby preventing accidental damage and discouraging vandalism and theft.

We are exploring such ubicomp applications in public spaces in the context of retail environments. The first application developed in this framework is a system to help customers to find products in a large store. A major requirement for this application is ubiquity, i.e., the product finder has to be easily accessible from anywhere in the store. Another requirement is that it has to be simple to learn and use.

Figure 5 shows a conceptual view of the *ubiquitous product finder*. ED-projectors are installed in every aisle, and small white boards placed in strategic locations on the shelves. These boards can be easily moved to different positions to respond to marketing and customer needs. A large table is place at the entrance to the store (on the right side of Figure 4). On it, an enlarged, intuitive-to-use version of the interface is projected aiming to make customers acquainted to the system. Signage boards are hung up from the ceiling or placed on top of the shelves, allowing the system to project directions to the product requested by the customer or to display advertisements when the system is idle.

A full-scale version of this application has not been deployed yet. Here, we describe a laboratory prototype of the system we have built and tested.

**Context:** a large retail environment.

**Application Goal:** to help customers everywhere in the store to find products by pointing directly to where they are located.

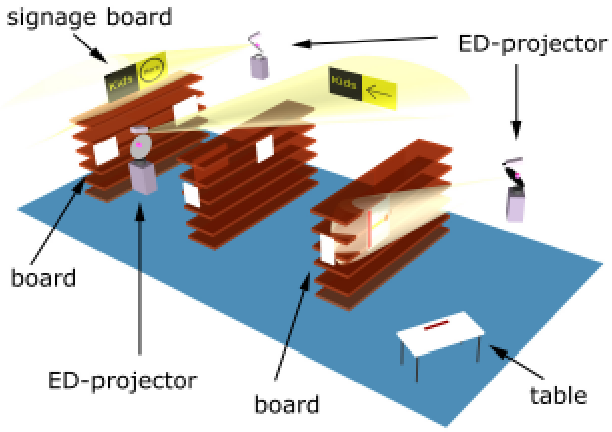

Figure 5 Concept of ubiquitous product finder.

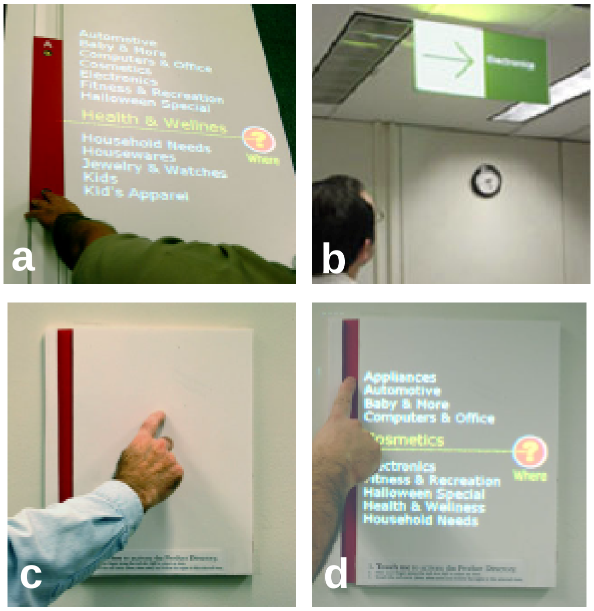

Figure 4. Prototype of ubiquitous product finder.

**Technology:** (laboratory prototype) an ED-projector is mounted on the ceiling. A table with a wooden red slider is used for projecting a larger introductory version of the product finder interface. Two smaller foam boards, visually similar to the table with the slider, are placed on the walls representing the smaller replicas of the same interface used in other areas of the store. The vision system is able to detect touch events, hand position and movement over the foam boards, and the position of the wooden slider relative to the table.

**Interaction:** the product list is projected on the product finder table located at the entrance of the store. The user can move the wooden red slider on the left of the table to find a product as illustrated in Figure 4.a. As he does this, the product list projected to the right of the slider scrolls up and down mirroring the motion of the slider. Once the user touches the “where” symbol on the right, arrows guiding the user to the location of the highlighted product are projected on signage boards hanging from the ceiling as illustrated in Figure 4.b (see also video at http://www.research.ibm.com/people/p/pinhanez/download/video2.avi). Alternatively, the user may initiate the product finder from the foam boards at other locations of the store by touching the boards as shown in Figure 4.c. This action triggers the system to project the product finder interface onto these foam boards as shown in Figure 4.d. In the wall version, the user slides his finger up and down a red velvet strip to scroll the product list (see http://www.research.ibm.com/people/p/pinhanez/download/video3.avi).

Our laboratory prototype of this application currently does not address two important issues present in real stores. First, we use a camera-based user tracking system (based on [13]) to detect proximity to the foam boards. If a user is close to one of them, we aim the ED-projector camera to the board surface and, without projecting any information, wait for the user to “call” the system by touching the board. This is not a practical solution in real store where multiple customers may be close to different foam boards at the same time. We envision here the use of an array of steerable cameras that can be dedicated exclusively to monitor the foam boards. The second issue is the inability of the system to help more than one customer at a time. In particular, if many boards share an ED-projector, we may have situations where users will have to wait for their turn to use the product finder. The severity of this is yet to be tested in a real-world deployment, but we expect that customers may tolerate reasonable waiting times as the situation is similar to when they consult an information desk or a sales assistant.

The flexible structure of our computer vision system simplified significantly the implementation of the laboratory prototype. To detect the movements of the wooden slider of Figure 4.a, a widget to detect linear movements of blocks of solid color was implemented. In the case of the foam boards, fingertip tracking is used to determine its position on the velvet strip as shown in Figure 4.d. Switching between the two different widgets is accomplished by the application simply sending the vision system the description of the appropriate interface.

We initially had some difficulties with the interaction on the velvet strip. After scrolling through the list and finding the desired product, users move away to touch the “where” button on the right, sometimes causing unwanted scrolling before they exit the tracking region. We solved the problem heuristically, ignoring quick motions after a pause that immediately exit the tracking region

This problem is common to many gesture-based interfaces: the segmentation of continuous gestures into distinct “phrases”. In our case the problem is simplified somewhat by using the spatial adjacency of gestures to interaction widgets. As with the problem of continuous speech, there is some amount of “co-articulation” that affects the ends of adjacent phrases, complicating their recognition. Gestural phrasing has been an issue for many years but, despite many different proposals (such as [14]), it is still largely unsolved.

## 4.4. Interactive Clothing Bins

Another application developed in the context of a retail environment uses a checkerboard arrangement of bins containing women’s pants and projectable surfaces as shown in Figure 6. It explores steerable interfaces as a way to augment the functionality of everyday objects and furniture. In this particular application, the system detects user interest with the product in a particular bin and automatically brings information designed to assist the customer.

**Context:** a retail environment.

**Application Goal:** to assist customers interested in a product by displaying information about it.

**Technology:** an ED-projector is aimed at the clothing bins. In our particular implementation, the system is unable to project an image large enough to cover all projectable surfaces on the bins. Instead, it selects groups of juxtaposed surfaces and steers the interface towards them according to the user hand’s position, i.e., the top parts of the bins, the bottom part of the bins, etc. The vision system can detect user activity in each of the clothing bins and touch events. No surface contains any sensor or wire. A camera-based user tracking system (similar to [13]) is used to determine the position of the user relative to the clothing bins.

**Interaction:** When the user is at a distance from the bins, a circulating series of advertisements for women’s clothing are projected on the panels, trying to attract customers to the area. When the user approaches the bins, the display is changed to advertise the store’s credit card and special promotions. Once the user starts examining pants in a bin, information pertaining to the pants in that bin is displayed in the proximity of the bin. Additional interactive panels are projected allowing the user to find information about similar pants styles, to check the size chart, or to check whether the desired size and color is available in stock (see demonstration video at http://www.research.ibm.com/people/p/pinhanez/download/video4.avi).

The major difference between this application and the ones previously examined is that this system proactively tries to engage and help the customer. By recognizing user activity in a particular bin (by simple hand tracking) and interpreting it as an interest in the products stocked there, the application takes the “liberty” of augmenting the environment with information. This example clearly demonstrates the potential of steerable interface systems to create new models of interaction by directly engaging people while they execute everyday activities in the real world. Since application interfaces can be created on areas very close to where the activities are happening, user attention can be more easily grabbed. This kind of engaging application involving physical objects cannot be easily implemented with traditional embedded displays such as monitors or kiosks.

This example also shows how computer vision methods and algorithms for recognition of human activity and behavior can be used for the development of applications with steerable interfaces. Our system determines user interest in a particular bin by detecting long duration hand movements within the bin. The performance achieved with this simple measurement is adequate for a research prototype but we clearly see opportunity here for more sophisticated methods of action and behavior recognition.

## 5. Improving the Performance of the Vision System of a Steerable Interface

By decoupling the functional definition of the interface from the specification of its location in the physical environment and in the camera image, the ED-projector vision system supports a great deal more flexibility than most vision-based interaction systems. This flexibility requires significant architectural complexity to dynamically realize an efficient recognition process from the task and surface descriptions.

Up to this point, our efforts have mostly concentrated on developing this flexible internal architecture [5], while the interaction widgets themselves have been comparatively simple. We are now beginning to concentrate on developing widgets that are capable of supporting more interesting interactions. This work is proceeding along several dimensions.

People have difficulty seeing the exact shape of a moving object, so natural gestures most often consist of a sequence of motions and static poses, where a hand movement is followed by a static pose, possibly followed by another movement, and so on [9]. In order to recognize hand poses, it will be necessary to either incorporate techniques to recognize shape in the sparse and noisy motion data or to incorporate hand segmentation techniques that do not rely on motion. However, as mentioned above, to avoid problems caused by the change in appearance of the user's hands as they pass through the projected image, the current interactions have all been based on detection and analysis of motion in the video stream. One possible alternative is to use a segmentation method that uses knowledge about the image being projected, the physical surface where the interface is being projected on, and how they interact, so that it can estimate the appearance of the projection to the camera. By using techniques such as those presented in [15], it may be possible to estimate the appearance of the projection in the digitized video stream. This will also allow interactions to take place in the presence of a changing projection, such as a projected video clip.

**Figure 6. Interactive clothing bins.**

Interestingly, the flexibility provided by computer vision allows interactions that are more abstract than the manipulative hand gestures implemented to date. One line of investigation concerns intentional control actions that the user takes with parts of the body other than their hands. These include interactions where whole body movements such as waves or jumps are used to control the interaction [16].

In addition to extending the gesture recognition capabilities of the system, we feel performance can be significantly improved by enabling the system to adapt to its current context. There are several areas where adaptation may be important. For instance, the current widget set requires information about the expected size and orientation of the user's hand. By making better use of user tracking information and observing user movements as they approach a widget, it is possible to determine this type of information automatically.

Every person moves with different rhythms and patterns. When users interact with a system using physical devices like a keyboard or mouse, the device standardizes and masks many of these differences, but because our vision system directly senses and interprets a user's movements, it can be sensitive to them. Techniques similar to [17] can allow adaptation to an individual user's movement patterns.

One constraint on any interaction system is that it be able to respond to the user smoothly and in a timely manner. For a vision-based system this implies it runs at a sufficiently high frame rate. Research with other interaction methods [18] has shown that tracking rates on the order of 10 Hz and response times less than 200ms are needed for a system to feel comfortable, except in the case of manipulation of virtual objects. If the frame rate falls below what is required to support these response times there are several trade-offs that can be adjusted to compensate. For example the incoming resolution may be reduced, trading off accuracy for speed. We are exploring techniques to do this automatically.

## 6. Conclusion

We have shown in this paper how possible ways to use computer vision in applications based on steerable projected interfaces. Our basic approach involves using motion data to identify the user's hand and path analysis to identify touch events, hand movement, and simple manipulative actions.

A major factor enabling the development of three of the prototype systems was the application-driven dynamic definition of interfaces and their automatic mapping to any surface known by the vision system. With this architecture (defined in detail in [5]), surface and viewpoint-specific characteristics are abstracted from the development process of the application.

Some shortcomings of the current system and challenges for computer vision have also been identified as a result of our work developing these applications. In general, projected interfaces limit the use of appearance-based techniques. Also, steerability requires gesture recognition algorithms to be viewpoint independent or, at least, adaptable to new viewpoints if basic contextual information is provided. Gesture segmentation based on spatial location and greater use of contextual information such as user position contributed to the success of the system. As the complexity of the underlying widgets increases, a challenge will be to ensure that the additional complexity does not spill over to affect the application developer or the end user.

Finally, in this paper we demonstrate that ubicomp applications can be implemented based on the steerable interface paradigm. The extension that this paradigm will become a major part of the ubiquitous computing reality substantially depends, in our view, on the development of adequate and robust computer vision systems that satisfy the constraints imposed by projection and steerability.

## References

[1] M. Weiser, "The Computer for the Twenty-First Century," *Scientific American*, vol. 265, pp. 94-100, 1991.

[2] C. Pinhanez, "The Everywhere Displays Projector: A Device to Create Ubiquitous Graphical Interfaces," in Proc. of Ubiquitous Computing 2001 (Ubicomp'01), Atlanta, Georgia, 2001.

[3] G. Pingali, C. Pinhanez, A. Levas, R. Kjeldsen, M. Podlaseck, H. Chen, and N. Sukaviriya, "Steerable Interfaces for Pervasive Computing Spaces," in IEEE International Conference on Pervasive Computing and Communications - PerCom'03, Dallas-Fort Worth, Texas, 2003.

[4] F. Kjeldsen, C. Pinhanez, G. Pingali, J. Hartman, A. Levas, and M. Podlaseck, "Interacting with Steerable Projected Displays," in Proc. of the 5th International Conference on Automatic Face and Gesture Recognition (FG'02), Washington, DC, 2002.

[5] R. Kjeldsen, A. Levas, and C. Pinhanez, "Dynamically Reconfigurable Vision-Based User Interfaces," in 3rd International Conference on Vision Systems (ICVS'03), Graz, Austria, 2003.

[6] M. W. Krueger, *Artificial Reality II*: Addison-Wesley, 1990.

[7] C. Wren, A. Azarbayejani, T. Darrell, and A. Pentland, "Pfinder: Real-Time Tracking of the Human Body," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 19, pp. 780-785, 1997.

[8] C. v. Hardenberg and F. Berard, "Bare-hand human-computer interaction," in Proc. of Workshop on Perceptive User Interfaces, PUI'01, Orlando, Florida, 2001.

[9] F. Quek, T. Mysliwiec, and M. Zhao, "Finger mouse: A freehand pointing interface," in Proc. of International Workshop on Automatic Face- and Gesture-Recognition, Zurich, Switzerland, 1995.

[10] J. Segen, "GestureVR: Vision-Based 3D Hand Interface for Spatial Interaction," in Proc. of ACM Multimedia Conference, Briston, England, 1998.

[11] F. Kjeldsen, "Visual Recognition of Hand Gesture as a Practical Interface Modality." New York, New York: Columbia University, 1997.

[12] Z. Zhang, Y. Wu, Y. Shan, and S. Shafer, "Visual Panel: Virtual Mouse, Keyborad, and 3D Controller with an Ordinary Piece of Paper," in Proc. ACM Perceptual/Perceptive User Interfaces Workshop (PUI'01),, Florida, USA, 2001.

[13] A. Senior, A. Hampapur, Y.-L. Tian, L. Brown, S. Pankanti, and R. Bolle, "Appearance Models for Occlusion Handling," in Proc. of Workshop on Performance Evaluation of Tracking and Surveillance (PETS2001), 2001.

[14] J. W. Deng and H. T. Tsui, "An HMM-Based Approach for Gesture Segmentation and Recognition," in International Conference on Pattern Recognition (ICPR'00), Barcelona, Spain, 2000.

[15] L. B. Wolff, S. K. Nayar, and M. Oren, "Improved Diffuse Reflection Models for Computer Vision," *International Journal of Computer Vision*, vol. 30, 1998.

[16] W. Freeman, K. Tanaka, J. Ohta, and K. Kyuma, "Computer Vision for Computer Games," in Proc. Second Intl. Conf. on Automatic Face and Gesture Recognition (FG'96), Killington, Vermont, 1996.

[17] M. A. Giese and T. Poggio, "Synthesis and Recognition of Biological Motion Patterns Based on Linear Superposition of Prototypical Motion Sequences," in Proceedings of the MVIEW 99 Symposium at CVPR'99, Fort Collins, Colorado, 1999.

[18] S. Card, T. Moran, and A. Newell, "The Keystroke-Level Model for User Performance Time with Interactive Systems," in *Readings in Human-Computer Interaction: A Multidisciplinary Approach*, R. Baecker and W. Buxton, Eds.: Morgan Kaufmann, 1987.