Epipolar Contrained User Pushbutton Selection in Projected Interfaces

Amit Kale, Kenneth Kwan and Christopher Jaynes
UK Center for Visualization and Virtual Reality
1 Quality St. Suite 857
Lexington KY 40507

Abstract

An almost ubiquitous user interaction in most HCI applications is the task of selecting one of out of a given list of options. For example, in common desktop environments, the user moves the mouse pointer to the desired option and clicks it. The analog of this action in projector-camera HCI environments involves the user raising her finger to touch one of the different virtual buttons projected on a display surface. In this paper, we discuss some of the challenges involved in tracking and recognizing this task in an projected immersive environment and present a hierarchical vision based approach to detect intuitive gesture-based "mouse clicks" in a front-projected virtual interface.

Given the difficulty of tracking user gestures directly in a projected environment, our approach first tracks shadows cast on the display by the user and exploits the multi-view geometry of the camera-projector pair to constrain a subsequent search for the users hand position in the scene. The method only requires a simple setup step in which the projector's epipole in the camera's frame is estimated. We demonstrate how this approach is capable of detecting a contact event as a user interacts with a virtual pushbutton display. Results demonstrate that camera-based monitoring of user gesture is feasible even under difficult conditions in which the user is illluminated by changing and saturated colors.

1 Introduction

In the recent past there has been a significant research focused on camera projector systems. This is partly due to the observation that camera-based calibration of projected displays allows very-large, cost-effective immersive displays with very little setup or maintenance burden placed on the user [1, 2, 3]. This research has spawned many smart projector applications such as scalable alignment of large multi-projector displays [2, 4], smarter interfaces for controlling computer based presentations [5, 6], and dynamic shadow elimination [7, 8]. Perhaps most importantly, camera-projector research has begun to explore the development of very flexible visually immersive environments e.g. "Office of the Future"[9] that offer completely new applications.

Given the scientific and commercial interest in these emerging technologies, a natural next step is to exploit the the camera-projector system to support human-computer interaction (HCI) [10, 11]. The system must be able to detect human gesture, interpret the context of the action, and respond appropriately. Understanding human actions is an active area of research in computer vision. However, when this task is transfered from the domain of an ambient (or controlled) environment to a situation in which the user may be illuminated by the projected imagery, the problem takes on a new dimension. For instance, traditional approaches to tracking may fail when the user is illuminated by varying (and saturated) colors. Surprisingly, this situation is likely to occur in many of the new display environments that are emerging from the multi-projector display community.

Although the work presented here assumes the presence of a front-projected display (and cast shadow of the gesture), the assumption is not overly restrictive. In addition, some of the principles used to track and recognize gesture in a front-projection environment can be used to alleviate some of the same problems with tracking user gestures against a changing back-projected display. Front-projected displays are recently used in favor of back-projected and controlled display walls due to lower cost, space savings, and ease of maintenance. Immersive environments that emphasize reconfigurability, and rapid deployment [7], almost certainly cannot assume the presence of backprojection screens. Finally, new applications that emphasize display on everyday surfaces, anywhere [10, 12], by definition cannot support controlled backprojection display. Given these new applications, it is important that camera-based HCI methods are developed that do not degrade when users are illuminated by a projector.

One approach to camera-based HCI in a projected display is to opportunistically capture and process imagery while the projectors are synchronously turned off. This is the approach taken by the the blue-c project [13] that acquires a volumetric model of the user within a projected display. Given the 3D reconstruction of the subject in an

Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops (CVPRW'04)
1063-6919/04 $ 20.00 IEEE

immersive environment, event detection can be achieved by directly analyzing the three-dimensional configuration of the user and determining if it corresponds to a particular event. This and similar approaches address the problem of projected illumination by shuttering the projected light to remove its effects from the user [13], detecting and eliminating light projected on the user altogether [8], or simply by disallowing HCI to occur in the frustum of a projector. Although these approaches have met various levels of success, they require specialized hardware (gen-lock and expensive high shutter rate projectors), or make assumptions about the environment (i.e. that the projected image is known or fixed).

An almost ubiquitous user input in most HCI applications is the task of selecting one of out of a given list of options. For example, in common desktop environments, the user moves the mouse pointer to the desired option and clicks it. The analog of this action in the case of an immersive environment involves the user raising her finger to touch one of several virtual buttons projected on the display surface. In this paper, we discuss some of the challenges involved in performing this task in an immersive projected environment and present a hierarchical vision based approach to detect this "mouse click" or contact event. This work is motivated by the following observation: shadows cast by users interacting within an immersive environment are often simpler to detect than the occluder. Detected shadows can constrain the location of the occluder and are often sufficient to recognize simple gestures. Rather than viewing shadows as an obstacle, we can exploit information given by the shadow to expedite the detection of a contact event. Segen and Kumar [14] have used joint shadow and hand information for gesture recognition. However their approach relies on using hue values of skin for detection of the hand region. In projected interfaces or immersive environments detection of skin region (as we shall see in Section 2) can be quite difficult.

Initially, the epipole of the projector in the camera's frame is estimated using a novel approach that requires very little user input. The shadow of the hand is detected and tracked using a mean-shift tracker. Using appropriate histogram metrics, the onset of the contact event is detected. The tracked shadow and the projector epipole define a constrained region that could contain the occluding object (hand). Background subtraction is used to extract the hand from the restricted epipolar swath region. The Euclidean distance between the hand centroid and the tracked hand-shadow is computed to detect the contact event. Because we employ geometric constraints, the computational burden normally associated with tracking and monitoring can be reduced and real-time rates can be achieved. Experimental results are presented for the case where a user interacts with three virtual buttons on the screen. Initial results demonstrate that contact approach, and the contact event itself can be measured robustly using our method.

The paper is organized as follows. In Section 2 we discuss challenges in gesture recognition in immersive environments. In Section 3 the details of the algorithm are covered. Section 4 presents the experimental results and Section 5 concludes the paper with speculation about how constrained tracking of user gesture via detected shadows my be applied to a wider range of gestures common to user interfaces.

2 Challenges for HCI in immersive environments

Most current automated approaches for recognizing hand gestures [15] rely on detection and tracking of skin regions. In order to detect skin regions the raw RGB color values are usually transformed to a color-space where hue is measured against known target values. A comparison of different color-space transformations for skin detection is discussed in [16]. As an example, consider the transformation to the HSV space. Independent of ethnicity, skin regions are restricted to either very low or very high values of hue under ambient lighting and a simple algorithm for skin detection can be obtained by setting appropriate thresholds on hue values in the scene. These settings are fairly robust for a particular (non-changing) lighting scenario.

An immersive environment or even projected interface is fundamentally a constantly changing, interactive display. The changing radiometric characteristics may be approximated and taken into account [7], by underlying image processing algorithms, but these approximations are often insufficient to support straightforward skin detection or are far too complex to estimate and then use at real-time rates. As a user is illuminated with projected information the hue of the skin is transformed based on the color being projected.

One of the ways to deal with this problem is to perform automatic white balancing [17] under a given colored lighting. Assuming that a certain region viewed by the camera is white, we can compensate its color values to remove the bias introduced by non-white illumination. However, white balancing may not correctly restore the hue of the skin regions to their ambient values. Furthermore if more than one color is projected white balancing may become complicated and expensive. This is an issue when real time performance is desired. Another approach is to model skin appearance under different illuminations by building histograms of skin pixels under different illuminations [18]. Figure 1 shows the RGB and hue images of the hand for ambient and color illuminations. As can be seen, under color illumination some backgrounds can attain hue characteristics of skin. Detection and tracking of skin regions under varying illumina-

tion is thus a hard problem. To circumvent this difficulty, it would then be necessary to simply shutter off the projected information [13, 8], or require that the user does not enter the frustum of any projector. Fast shuttering of projected energy requires additional expensive hardware and may not be feasible for large scale multi-projector environments. Turning off projected information has been explored for situations in which users may be "blinded" by projected energy, but requires an accurate model of what is projected at each frame. This information is simply unavailable to an interactive display.

3 Proposed Methodology

The work presented here is motivated by the observation that shadow regions are relatively easy to detect and track even under widely varying illumination. Detection and tracking of the hand regions is a rather formidable problem as we saw in Section 2. Ultimately, interface gestures are performed by the user and his/her hands and not the shadows on the display surface. However, by tracking the shadow we can infer the appropriate search region for the hand in the scene. Moreover, the position of *both* the shadow and the casting object can yield information to a gesture recognition system. Here we detail how we track both regions (when appropriate) and how the measured distance between the hand and its shadow is a robust image-based measure of detecting the contact even.

Considering the non-rigid nature of the hand, a mean-shift tracker is used to track the centroid of this hand-shadow region. It is necessary to track the hand centroid only when the hand is close to the screen containing the virtual buttons. This proximity of the hand to the screen can be detected by the occlusion of the shadow of the hand by the hand itself. After detecting the onset of contact, the estimated epipolar geometry between the camera and projector can be used to restrict the search region for the hand. Additional information about the approximate color-mapping between the camera and projector as well as the contents of the projector frame buffer at any given instant is then used to detect the presence of the user's hand within this small search region. The Euclidean image distance between the centroid of the tracked hand shadow and the centroid of the hand region is measured and when this distance drops below a threshold, the color in the neighborhood of the hand shadow centroid is declared to be the corresponding virtual button "pressed" by the user.

3.1 Projector Epipole Estimation

A data projector is a dual of a camera and the projection process can be modeled using the standard pinhole camera model. Given a shadow on the display surface and detected in a camera image, the corresponding occluding object must lie along an epipolar line in the image that relates the multi-view geometry of the projector-camera pair. Our approach is to estimate the position of this projector epipole in the frame of the camera and the constrain the search for the user hand via the implied epipolar lines that emanate from the detected shadow.

One way of determining the epipole for a camera-projector pair is to compute a pair of homographies between them by determining matchpoints between the devices on two different world planes. The epipoles in the two images can then be computed solving the generalized eigen-value problem for the two homographies [19]. This idea was used by Raskar and Beardsley [20] as an intermediate step in the camera-projector calibration problem. In order to obtain the homographies it is necessary to vary the configuration of the system with respect to the plane which can prove to be cumbersome. For our problem all we need to restrict the search space for the hand is the location of the projector epipole. A much simpler approach can be used in order to do this. Presence of an occluder in the frustum of the projector will cast a shadow on the screen. In an image of the object and its shadow, the line joining a point on the object and its shadow will pass through the projector epipole. Thus, given two pairs of corresponding object-shadow points, the join of their lines determines the epipole. More formally, let (o_1, s_1) and (o_2, s_2) be the image plane coordinates (expressed in homogeneous coordinates) of two distinct world points. Then the epipole e can be determined as

$$e = l_1 \times l_2 \tag{1}$$

where $l_1 = o_1 \times s_1$ and $l_2 = o_2 \times s_2$ where $\times$ represents the cross product. In practice, it is necessary to consider more points when estimating the epipole. A simple way to do this is to simply move a suitable object in the field of view of the camera. Pairs of points on the object and their corresponding shadows can be used to generate the lines passing through the projector epipole. Using these lines and (1) estimates of the epipoles can be obtained. During this bootstrapping process, the projector is instructed to project white to alleviate the tracking problems that this paper addresses.

3.2 Mean-shift Tracking

In order to track the shadow it is necessary to compute the location of the hand shadow in the first image. Prior shape information about the hand shadow regions and the location of the shadow can constrain the initial tracking system. One way to compute the location of the hand shadow is to use the chamfer system [21]. Alternatively, if the approximate areas where shadows are likely to emerge on the display, simpler search techniques can be used.

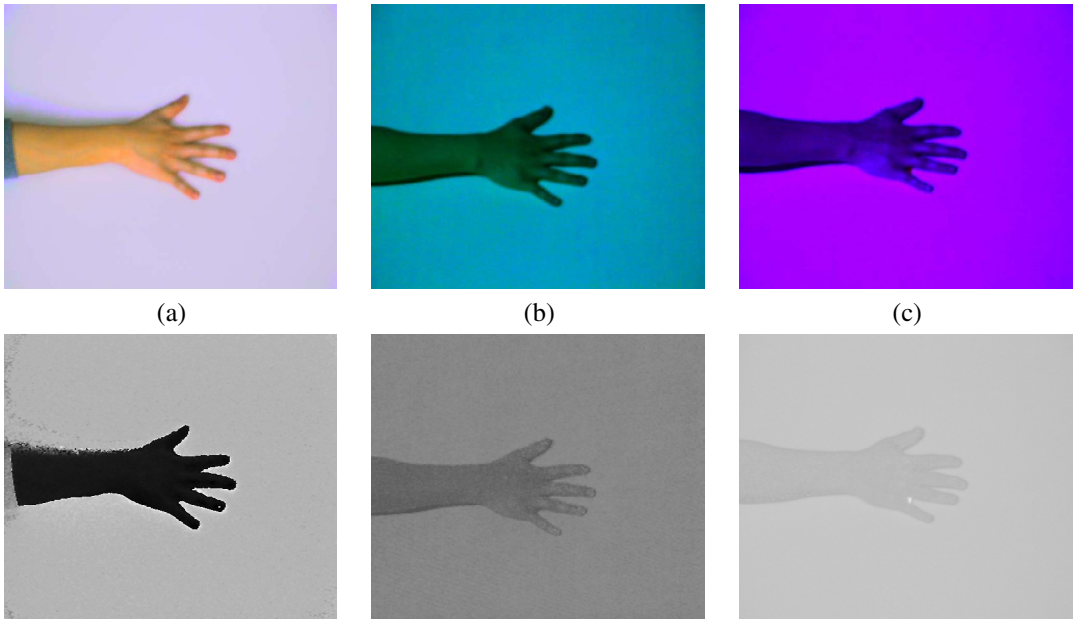

Figure 1: Images of the hand taken under different illuminations; RGB images for (a) Ambient lighting (b) Saturated Blue Color (c) Saturated Magenta Color; Hue images for (d) Ambient lighting (e)Saturated Blue Color (f) Saturated Magenta Color ;

The intensity histogram of the image patch around the detected centroid of the hand-shadow region as it emerges from the edge of the monitoring camera defines the target histogram. Taking into account the non-rigid nature of the hand, we use the mean-shift tracking algorithm of Comaniciu et al.[22] to robustly update the estimated position of the cast shadow. Mean-shift tracking is based on maximizing the likelihood of the model (hand shadow) intensity distribution and the candidate intensity distribution using the Bhattacharya coefficient.

$$\rho(\mathbf{m}) = \sum_{u=1}^{n} \sqrt{q_u p_u(\mathbf{m})} \tag{2}$$

where $\mathbf{m}$ is the center of the hand region, n is the number of bins in the distribution, and q_u and p_u are the weighted histograms of the model and candidate respectively. The weights for the histograms are obtained using the Epanechnikov kernel. The center of the hand region in the next frame is found using

$$\mathbf{m}_{new} = \frac{\sum_{i=1}^{n_h} \mathbf{x}_i w_i g(||\mathbf{m}_{old} - \mathbf{x}_i||)}{\sum_{i=1}^{n_h} w_i g(||\mathbf{m}_{old} - \mathbf{x}_i||)} \tag{3}$$

where $\mathbf{x}_i$ are the pixels in the image patch and g is the derivative of the Epanechnikov kernel. The weights w_is are computed as

$$w_i = \sum_{u=1}^{n} \delta[b(\mathbf{x}_i) - u] \sqrt{\frac{q_u}{p_u(\mathbf{m})}} \tag{4}$$

where $\delta(.)$ is the Kronecker delta function and $b(.)$ is a function that associates to a pixel the index of the histogram bin corresponding to the intensity value associated with the pixel. As the hand starts making contact with the virtual buttons the shadow of the hand starts getting occluded by the hand. This occlusion of the hand-shadow indicates the onset of contact. In order to detect this, it is necessary to compare the tracked shadow region in the neighborhood of its centroid in the present frame to the target histogram. One measure could simply be simply the number of shadow pixels. This measure is not scale invariant however. It is more appropriate to consider scale invariant metrics e.g. the Bhattacharya distance which is also used for the mean-shift tracker.

$$d_{Bhattacharya}(p, q) = \sqrt{1 - \sum_{u=1}^{n} \sqrt{q_u p_u(\mathbf{m})}} \tag{5}$$

Another scale-invariant histogram distance metric is the chi-squared distance defined by (6) which has been used in [23] for scene change detection in digital video sequences.

$$d_{\chi^2} = \sum_{u=1}^{n} \frac{(q_u - p_u(\mathbf{m}))^2}{(q_u + p_u(\mathbf{m}))} \tag{6}$$

3.3 Detection of the contact event

A simple way of detecting contact is to consider the value of this metric as a function of time. When the metric is

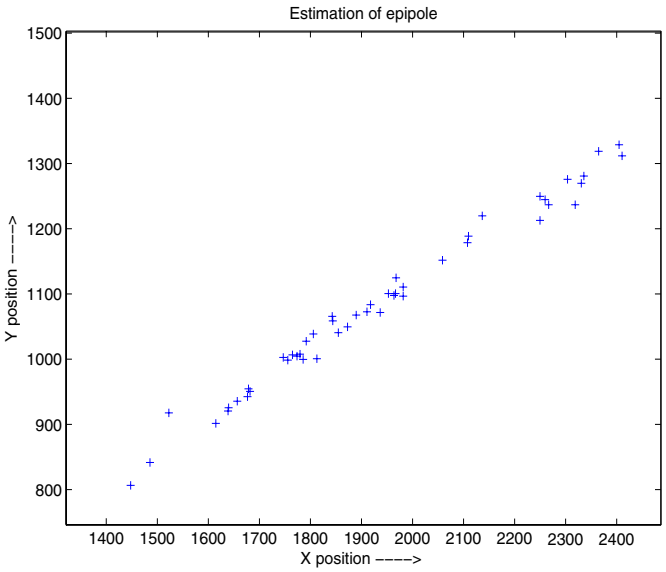

Figure 3: Estimates of epipoles obtained by considering pairs of object-shadow points from several images of a rectangular board

sufficiently large, contact can be assumed to have occurred. However this simple scheme has the disadvantage that the extent to which the hand occludes the shadow varies based on the location of the user with respect to the display. Hence it is more appropriate to use this temporal metric information only to signal the onset of contact. Once the onset of the contact has been detected it is necessary to detect the true hand position in the image plane. Knowing the projector epipole, a simple way to limit the search region is to construct a swath region starting at the corners of the window enclosing the hand shadow centroid. Furthermore, since the hand is assumed to be close to the shadow when the shadow begins to be occluded it is possible to limit the depth of this epipolar swath.

Given this implied search area on the image plane, there are several options to determining the location of the users hand. One way is to compute an edge-map within the swath region and compute its centroid. On nearing contact the centroid of this edge-map would be expected to merge with the centroid of the hand shadow. Alternatively if the image displayed by the projector does not change too rapidly and the color transfer function between the camera and projector is known, a simple background differencing between the swath region in the current image and the reference image can be used to detect the presence of the hand.

4 Experimental Results

The approach was tested using a single- ceiling mounted projector p while a camera, mounted approximately 20-degrees off-axis also on the ceiling monitors the scene. Three different colored buttons where projected and the subject was instructed to touch each button sequentially. Figure 2 shows a few images from the dataset. This section discusses the implementation details of our approach and explores the robustness of the virtual touchbutton detection system.

In order to use this method it is necessary to compute the epipole. As discussed in Section 3, this requires several object-shadow point correspondences. In order to simplify the task of establishing correspondence, a rectangular board was moved around in the field of view of the camera. About 40 images were captured and the estimates of epipole locations were obtained using (1). Figure 3 shows the estimated epipoles. Since the camera and projector axes are almost parallel to each other there is considerable variance in the estimated epipoles. The epipole used in our experiments is the mean of this cloud of points. Clearly, the approach is unable to provide accurate information about the epipole position for traditional multi-view calibration tasks. However, only a rough estimate is required to constrain the subsequent search of the user's hand position.

Assuming that the person has an outstretched finger for the touchbutton gesture, the initial shadow region in the image is detected by thresholding the intensity values and analyzing the shape characteristics of the cast shadow. Of course the shadow changes shape according to a perspective projection of the hand to the display surface so these shape constraints must be quite weak. A rectangular binary mask is translated horizontally and its correlation with the shadow regions is computed. Since we assume an outstretched finger, the correlation in the finger (shadow) region will be smaller than that in the hand (shadow) region. The first instance of a large change in the correlation value can be used to approximately detect the hand shadow region. Given the estimated hand shadow location in the first frame, the histogram of the intensity values around the location is computed. This histogram is used as the target histogram. Note that unlike [22] the target histogram is one dimensional. Mean shift iterations as described in [22] are used to track the centroid of the hand shadow in each frame.

In order to detect the onset of contact, the distance between the histogram of the hand-shadow region around the tracked centroid and the target histogram for each frame is computed. Note that at the end of each mean shift iteration the Bhattacharya distance (computed using (5)) between the target histogram and the histogram for the image-patch around the centroid is available. However we wanted to explore if a different histogram distance e.g. chi-squared distance (computed using (6)) would be more suitable for this task. Both the Bhattacharya distance and the chi-square histogram distance measures were tested for detection of onset of contact. Figure 4 shows a comparison of the dis-

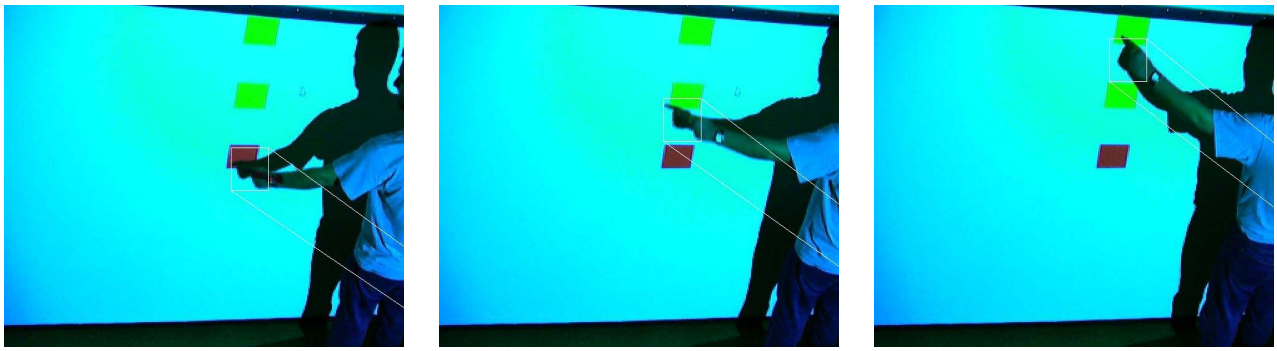

Figure 2: Few images taken from our experimental setup. The white lines connect the corners of the window enclosing the hand shadow region to the epipole.

tance measures for a video sequence of the person touching the virtual buttons and then withdrawing. The peaks in the plot correspond to the person making contact with the virtual buttons while the valleys correspond to the persons hand being far away from the virtual buttons. The solid red curve shows the chi-squared distance while the blue dash-dotted curve shows the Bhattacharya distance as a function of time. As can be seen from this figure, for a fixed threshold, the chi-squared distance exceeds the threshold less frequently than the Bhattacharya distance. This is because the chi-square distance, since it uses the square of the difference in the histogram values, penalizes differences more when they are large, whereas small differences are penalized less. For the case shown in Figure 4 and for a threshold chosen to be 0.2, the chi-squared distance exceeds the threshold for 30% of the frames while the Bhattacharya distance does so for 60% of the frames. Furthermore none of the true contact onsets were missed by the chi-squared distance for the chosen threshold. Since crossing the threshold implies that the hand centroid must be computed, using the chi-squared distance leads to a reduction in the amount of computation as compared to the Bhattacharya distance. When the chi-squared distance exceeds the specified threshold, it is necessary to look for the hand. Given the approximate location of the epipole (obtained as discussed earlier) the search region is restricted appropriately. In particular we consider a window around the tracked position of the hand shadow. The epipolar swath region is determined by the lines joining the opposite corners of the window with the epipole. Figures 2 shows the epipolar swath region constructed for a few images in our dataset. Furthermore, since the onset of contact has been detected by the histogram distance, it is not necessary to consider the entire epipolar swath. It is sufficient to traverse a limited distance along the epipolar swath direction. Figure 5(a) shows the intensity image within the delimited swath region. Since the camera and projector axes are almost parallel the lines connecting the corners of the window to the epipole are almost parallel to

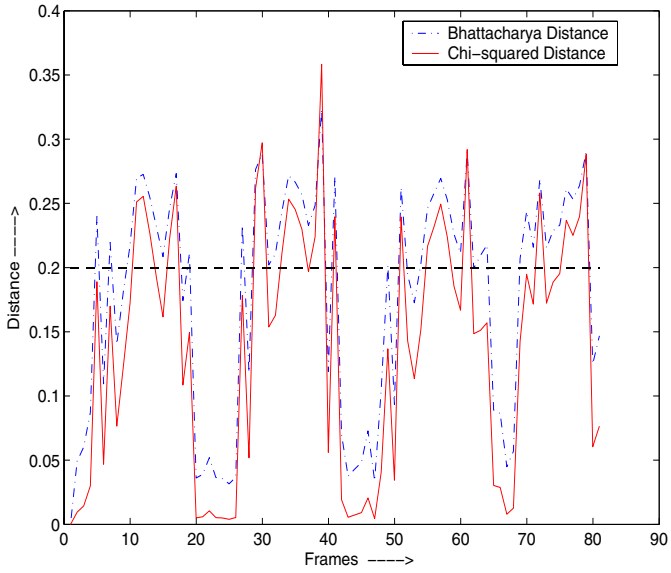

Figure 4: Comparison between Bhattacharya and chi-squared distances for detection of onset of contact. Observe that for a given threshold the chi-squared distance has fewer crossovers

each other. Given this delimited epipolar swath region it is necessary to compute the hand centroid in this region. One approach to this problem is to consider the edge-map within this region. This edge map would consist of edges from the hand as well as the shadow regions. As the hand starts making contact with the screen the centroid of the combined edge region would be expected to merge with the centroid of the hand shadow. However the drawback of this approach is that as the person makes contact with the middle and top buttons, his/her hand passes through at least one other button. This results in detection of spurious edges which causes the centroid computation to be unstable and resulting in false positives. Hence a more robust approach must be sought. Assuming that the display does not change very rapidly and that the color calibration between the camera and projector is known, one simple solution to this problem is to consider a simple pixel-wise background subtraction within the delimited swath region. As the number of pixels in this region is significantly smaller than that of the entire region (less than 4% of the total number of pixels in most cases)the added computational burden is not as significant as compared to an approach that uses background subtraction for the entire image. In particular the hand region is computed as

$$I_{hand}(i,j) = \begin{cases} 1 & \text{if} \quad (i,j) \in ES \text{ and} \\ & |I_{hand}(i,j) - I_{ref}(i,j)| > T_1 \text{ and} \\ & I_{hand}(i,j) > T_2 \\ 0 & \text{otherwise} \end{cases}$$

where ES denotes the epipolar swath, T_1 denotes a threshold to determine if the pixel is a foreground pixel,and T_2 is a threshold to determine if the pixel is a shadow pixel. The estimated hand region corresponding to the intensity image in Figure 5(a) is shown computed from the above equation is shown in Figure 5(b). The Euclidean distance between the tracked shadow centroid and the hand region is then computed. When the distance falls below a certain threshold, contact is declared. The color of region in the neighborhood of the contact region can be inspected to take the appropriate course of action.

Figure 6 shows the Receiver Operating Characteristics (ROC) plots for the contact event (hypothesis H_1) versus no contact (hypothesis H_0). The ROC plots the probability of detection of the contact event (P_D) against the probability of a false alarm (P_F). The video was analyzed manually to detect which frames had the contact event happen in them. The threshold for the histogram distance was set at a value so that no contact event was missed. The plot was generated by varying the threshold on the Euclidean distance between the hand and shadow regions and counting the number of times the contact event gets detected when no contact has occurred (for P_F) and when contact has occurred (for P_D), for a given threshold. The total number of frames was 243 out of which 31 frames had the contact event happen.

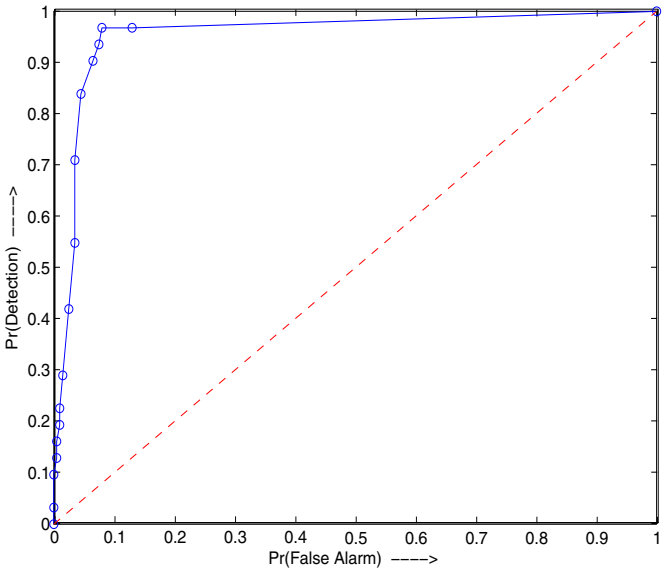

Figure 6: Receiver Operating Characteristics for the contact event (hypothesis H_1) versus no contact (hypothesis H_0).

The ROC can be used to choose a threshold to get a good tradeoff between P_D and P_F.

5 Conclusions and Future Work

In this paper we presented a method for detecting contact events which can work under the arbitrary lighting conditions typically encountered in an interactive, projected display. Instead of using skin tone detection (which can be unreliable under varying lighting conditions), the approach focuses on the shadow cast by the hand. The location of the hand shadow is detected and tracked using a mean shift tracker. Since the hand occludes the shadow before contact happens, the deformation of the hand-shadow region is then used to detect the onset of contact. A novel method which required very little user input was introduced to estimate the projector epipole. After detecting the onset of contact, the epipole was used to define a restricted search region for the hand. Background subtraction was then used to extract the hand from the restricted epipolar swath region. The Euclidean distance between the hand centroid and the tracked hand-shadow was computed to detect the contact event. The experimental results showed that the contact approach and the contact event itself can be measured robustly using our method.

Our approach used a single-camera projector pair. Future work would focus on achieving greater view invariance. For instance, in the experimental setup we have considered, there are certain positions in the camera's field of

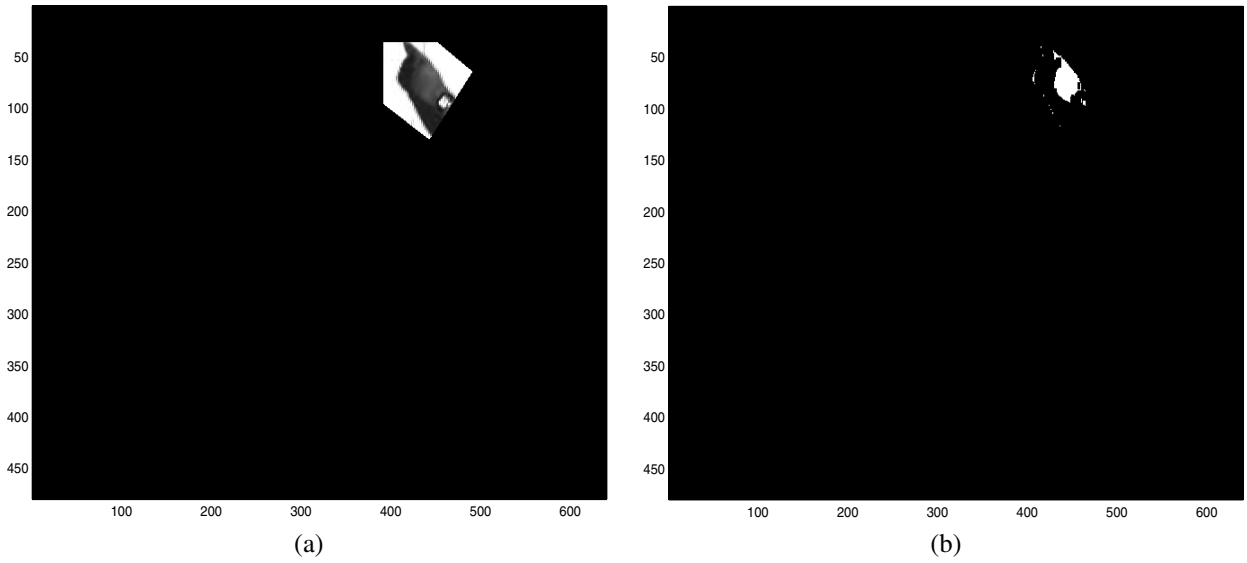

Figure 5: Delimited Epipolar swath region constructed after the onset of contact has been detected. (a)Intensity Image (b) Binary Image showing the hand region after background subtraction

view in which the person completely occluded the hand-shadow.One possible approach to remedy this situation would be to use multiple cameras. It would also be interesting to use 3D information about the hand by using the shadow and a full calibration between the camera-projector pair similar to Segen and Kumar [14].

References

[1] C. Jaynes, W. B. Seales, K. Calvert, Z. Fei, and J. Griffioen, "The metaverse- a networked collection of inexpensive, self-configuring immersive environments," *7th International Workshop on Immersive Projection Technology, 9th Eurographics Workshop on Virtual Enviroments*, 2003.

[2] Ramesh Raskar, Michael S. Brown, Ruigang Yang, Wei-Chao Chen, Greg Welch, Herman Towles, Brent Seales, and Henry Fuchs, "Multi-projector displays using camera-based registration," *Proceedings of IEEE Visualization*, 1999.

[3] H. Chen, R. Sukthankar, G. Wallace, and K. Li, "Scalable alignment of large-format multi-projector displays using camera homography trees," *Proc. of IEEE Visualization*, 2002.

[4] C. Jaynes, S. Webb, and R. M. Steele, "A scalable framework for high-resolution immersive displays," *International Journal of the IETE*, vol. 48, 2002.

[5] R. Sukthankar, R. Stockton, and M. Mullin, "Smarter presentations: exploiting homography in camera-projector systems," *Proc. of the ICCV*, 2001.

[6] Claudio Pinhanez, "Creating ubiquitous interactive games using everywhere displays projectors," *Proc. of the International Workshop on Entertainment Computing*, 2002.

[7] C. Jaynes, S. Webb, M. Steele, M. Brown, and B. Seales, "Dynamic shadow removal from front projection displays," *Proc. of the ACM SIGGRAPH 1998*, 1998.

[8] T. Cham, J. Rehg, R. Sukthankar, and G. Sukthankar, "Shadow elimination and occluder light suppression for multi-projector displays," *Proc. of CVPR*, 2003.

[9] Ramesh Raskar, Greg Welch, Matt Cutts, Adam Lake, Lev Stesin, and Henry Fuchs, "The office of the future : A unified approach to image-based modeling and spatially immersive displays," *Proc. of the ACM SIGGRAPH 1998*, 1998.

[10] N. Sukaviriya, M. Podlaseck, R. Kjeldsen, A. Levas, G. Pingali, and C. Pinhanez, "Augmenting a retail environment using steerable interactive displays," *Proc. of CHI*, 2003.

[11] R. Sukthankar, R. Stockton, and M. Mullin, "Self-calibrating camera-assisted presentation interface," *Proceedings of International Conference on Control, Automation, Robotics and Computer Vision*, 2000.

[12] Ruigang Yang and Greg Welch, "Automatic projector display surface estimation using every-day imagery," *9th International Conference in Central Europe on Computer Graphics, Visualization and Computer Vision 2001*, 2001.

[13] M. Gross, "Blue-c: A spatially immersive display and 3d video portal for telepresence," *Immersive Projection Technology and Virtual Environments*, 2003.

[14] J. Segen and S. Kumar, "Shadow gestures: 3-d hand pose estimation using a single camera," *Proc. of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, 1999.

[15] C. Rao, A. Yilmaz, and M. Shah, "View-invariant representation and recognition of actions," *International Journal of Computer Vision*, vol. 50, no. 2, 2002.

[16] M.C. Shin, K. I Chang, and L.V Tsap, "Does colorspace transformation make any difference on skin detection?," *Proc. of the Workshop on Applications of Computer Vision*, 2002.

[17] D. A. Forsyth, "A novel algorithm for color constancy," *International Journal of Computer Vision*, vol. 5, no. 1, pp. 5–36, 1990.

[18] M. Soriano, B. Martinkauppi, S. Huovinen, and M. Laaksonen, "Skin detection in video under changing illumination conditions," *Proc. of ICPR*, pp. 839–842, 2000.

[19] B. Johansson, "View synthesis and 3d reconstruction of piecewise planar scenes using intersection lines between the planes.," *Proc. of ICCV*, pp. 54–59, 1999.

[20] R. Raskar and P.A. Beardsley, "A self-correcting projector," *Proc. of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, 2001.

[21] D. M. Gavrila, "Pedestrian detection from a moving vehicle," *Proc. of European Conference on Computer Vision*, 2000.

[22] D. Comaniciu, V. Ramesh, and P. Meer, "Real time tracking of non-rigid objects using mean shift," *Proc. of CVPR*, 2000.

[23] R. M. Soriano, C. Robson, D. Temple, and M. Gerlach, "Metrics for scene change detection in digital video sequences," *Proceedings of the IEEE International Conference on Multimedia Computing and Systems*, 1997.