The Smart Bookshelf: A study of camera projector scene augmentation of an everyday environment

Danny Crasto, Amit Kale and Christopher Jaynes
Department of Computer Science and
Center for Visualization and Virtual Environments
University of Kentucky, Lexington KY*

Abstract

Recent research in projector-camera systems has overcome many of the obstacles to deploying and using intelligent displays for a wide range of applications. In parallel with these developments, projector costs continue to decline with corresponding increase in resolution, brightness and contrast ratio. In light of this trend, we are exploring the unique capabilities that camera-projector systems can offer to intelligent environments and ubiqutous computing.

Our initial step towards environments that are intelligently augmented by projector-camera devices, is a smart bookshelf application. The system utilizes a camera pair and a projector to monitor the state of a real world library shelf. As books are added to the shelf a foreground detection algorithm which takes into account the projected information yields new pixels in each view that are then verfied using a planar parallax constraint across both cameras to yield the book spine. Using a simple calibration scheme, the homography induced by the world plane in which book spines approximately lie is estimated. Users are then able to query for the presence of a book through a user interface and book spines are highlighted by transforming image pixels to their corresponding points in the projector's frame via the known homography. The system also can display the state of the bookshelf at any time in the past. Projected information can also be used to enhance the image-processing tasks at hand and we briefly explore this in this work.

1 Introduction

Augmented reality research focuses on the issues related to modifying or enhancing a view by seamless integration of computer generated imagery with the physical world. Techniques include real-time editing/modification of live video that is played directly to the user thorough head worn displays or rendering of graphical images onto a see-through display in such a way as to augment the physical world. Recently, researchers have been exploring how the physical environment can be augmented with one (or more) digital light projectors. In this approach, projected light is cast onto objects directly in the scene in order to change their appearance, or to provide information to a viewer in the scene [1, 2].

Using a projector-camera system, augmentation of the scene can be accomplished without a head worn display and the augmented scene can be observed by a large number of viewers without replication of the displays. Although what should be projected into the scene may be different for different viewpoints (i.e. perspectively correct insertion of three-dimensional objects), in many cases artificial texture and surface markings are sufficient for a variety of augmentation applications.

In addition to these advantages, cost of commodity digital light projectors, and their increasing prevalence in our everyday environments, makes the projector-camera approach to dynamic augmentation of the scene attractive. Likewise, significant progress has been made in projected displays that utilize a camera to monitor projected imagery as well as the scene into which it is being projected. These projector-camera systems have address many significant problems related to projector-based augmented reality including correct color production [3], geometric warping and blending [4], multi-projector cooperation [4, 5], resolution requirements, and blending of projected imagery with underlying surface characteristics [2]. Much of this work is being conducted within the computer vision and graphics communities and as progress is made, projector-based augmentation is becoming more feasible.

This work explores how projector-camera systems can be used to continuously monitor and appropriately augment a changing scene. We introduce a simple application in order to better explore the unique capabilities that a projector-camera system can offer to intelligent and ubiquitous computing environments. In particular, we demonstrate that the presence of a camera can be utilized to monitor scene state (geometry, presence/absence of objects, etc.) in ad-

*This work was funded by NSF CAREER Award IIS-0092874 and by The Kenucky Office of the New Economy

Proceedings of the Seventh IEEE Workshop on Applications of Computer Vision (WACV/MOTION'05)
0-7695-2271-8/05 $ 20.00 IEEE

dition to the traditional utilization of a camera in order to calibrate and correctly render information via the projector. These issues are explored in the context of the "Smart Bookshelf", a system that passively monitors a shelf of books to automatically maintain a database of what books are present, their appearance models, and their position both in the monitoring cameras, and the frame buffer of the projector. The Smart Bookshelf supports the following tasks; 1) User query for the presence of a books responded to by automatic highlighting of the book on the shelf or projecting its last known location if it has moved, 2) Projection of auxiliary information such as author, last person to use the book, etc.. into the scene, and 3) re-projection of the bookshelf appearance at any given previous moment. These three straightforward tasks, if implemented, can allow users to more effectively locate books, discover when they have been removed, and track down titles in a large library.

The system initially builds a database that contains information related to book position, time of addition, and the pixel values (texture information) on the book spine. The database is updated as books are removed, added or replaced. Upon request, the system highlights the book spine using a digital light projector. One difficulty in detecting books is that the sides of the book are visible along with the spine. In subsequent frames, then, this portion of the book will be occluded by neighboring books potentially leading to false positives in foreground images. Furthermore, it is important that these regions are not highlighted by the projector in this situation because neighboring books will be highlighted at inappropriate times.

This problem is overcome by noting that the book spines on a shelf all lay in an approximate plane. Using two cameras, then, the homography between the two cameras induced by this plane can be estimated and used to segment spine regions from regions outside of the plane. The effects of deviation from this assumption are addressed in Section 2. Using a combination of background subtraction and this homography, spine regions of the books can be reliably identified. Color histograms of extracted book spines are stored in a database for appearance indexing.

Finally, the location of book spine pixels, in each camera can be mapped to their corresponding location in a projector using the same observation about the approximate spline plane. In doing so, objects can be highlighted by the projector upon user request. Additional information about the book can be rendered into the scene, and the previous location of books, and their movement history can be replayed by the projector for a user who is looking for a particular book.

1.1 Previous Work

An early system for augmenting an environment by combining a projected information and computer vision was the "digital desk" described by Wellner and MacKay [6]. The idea was to augment a regular working desk by projecting electronic information as well as to monitor paper documents to facilitate a seamless integration of information. Over the last few years, several augmented environments had been designed using single or multiple video. In [7], Pinhanez et al. propose the use of steerable projector-camera systems employing computer vision and their applications including a ubiquitous product finder for retail environments. Raskar et al.[4] propose the use real-time computer vision techniques to dynamically extract per-pixel depth and reflectance information for the visible surfaces in the office including walls, furniture, objects, and people, and then to either project images on the surfaces, render images of the surfaces, or interpret changes in the surfaces. More recently, Crowley [8], proposed a steerable camera-projector system with two degrees of freedom which can be used to detect and track a planar surfaces for display.

Our work is most similar in spirit to the ubiquitous product finder in [7], in the sense that it facilitates easy access of the requested book for the user. However along with this task, it automates the process of maintaining book records through a continuous monitoring of the contents of the bookshelf and allowing for a easy playback of its state at any time in the past.

2 Proposed Methodology

Our goal is to create a bookshelf that can detect the presence of books as they are added to its shelves, monitoring books as they are removed and replaced, and then projecting information about books directly onto the bookshelf using a corresponding projector.The technique uses two cameras to monitor the bookshelves and one projector to augment the scene with projected imagery. The cameras and projectors are oriented so that their frustums provide coverage of the shelves to be monitored. The system is then calibrated in a two phased process. First, the pixel mappings between all cameras and projectors are discovered. We assume that the book spines lie approximately in a world plane, π that passes through the front of the bookshelf. The geometric calibration phase, then, consists of discovering the homographies that map pixels from one device to another that correspond to this plane. Also, a set of color transfer functions that describe, for a given color in the projector, the color that will be observed in each camera are computed. This color mapping information is used during the image processing and scene augmentation process. In particular, it is important that image processing routines, such as back-

Proceedings of the Seventh IEEE Workshop on Applications of Computer Vision (WACV/MOTION'05)
0-7695-2271-8/05 $ 20.00 IEEE

(a) (b)

Figure 1: Two views of the bookshelf seen from monitoring cameras. Color characteristics of the book spines are used for recognition. Multiple views of the plane containing the spines help segment book spines as they are added and removed from the scene.

ground modelling, are able to take into account projected imagery that may be generated at run-time. Figure 5(a) shows a picture of our physical setup as well as a schematic view of the situation.

From common experience, we know that books are stacked on bookshelves with their spines facing the user as shown in Figure 1. It can be seen that the image of the spine of the book can provide a unique characterization of the book via the text, color and intensity patterns. Hence the first step towards building a smart bookshelf is to be able to isolate the spine regions of each book reliably. However, since the position of the cameras is somewhat unconstrained, sides of the books may be visible based on the location of the book in the bookshelf. Non-spine regions can be a significant problem if non-spine pixels are extracted as part of the appearance model of a book for several reasons. First, these pixels are more likely to be in shadow and will be perspectively distorted with respect to the monitoring camera. This can lead to unstable object color models. In addition, the goal of book segmentation is to augment books on the shelf with the projector. Pixels off the book spine should not be included in the augmentation process because of the potential for occlusion and erroneous augmentation of neighboring books. By way of example, consider a book added to an empty bookshelf and the resulting foreground detection via background differencing as shown in Figure 2. When a second book is added to the bookshelf, the side of the first book is no longer visible due to occlusion by the second book. Subsequent augmentation of the scene(shown in Figure 2) results in a highlighted region significantly larger than the visible region of the book.

We filter pixels that are not on the spine of the book by noting that the spines of the books are more or less coplanar. Under this assumption a very strong constraint exists for the corresponding spine points in the left and right images of the stereo pair [9]. Specifically, if $\tilde{\mathbf{m}}_L$ and $\tilde{\mathbf{m}}_R$ denote homogeneous coordinates of a spine point in the left and

(a) (b)

Figure 2: (a) and (b) show a single book added to the shelf and the corresponding foreground pixels detected by a background subtraction algorithm. Notice that sides of the book are included in the detected foreground.

right views respectively then

$$\tilde{\mathbf{m}}_L = H_{LR}\tilde{\mathbf{m}}_R \tag{1}$$

where H_{LR} is a collineation which is a function of the location of the spine plane and the camera internal and relative external parameters.

One way of extracting the book spines is to consider the foreground images (i.e. as in Figure 1) obtained in each of the cameras and then performing cross-validation for potential correspondences using (1) to extract small image regions and measuring local color similarity. This technique has been successfully used in other contexts [10]. However, it requires color calibration of the different cameras because of potentially dramatically different radiometric response functions that the cameras might exhibit. In addition, the approach is sensitive to specular highlights and other artifacts that lead to nonlinear (and different) responses in each view. For the problem at hand, we employ commodity cameras and do not want to burden the user with sophisticated camera calibration.

Since books are approximately rectangular blocks, the binarized left foreground image, when transformed by the relevant homography to the right camera, will overlap the corresponding right camera image only in the region of the spine. The same holds when the right foreground image is transformed to the left camera. In other words pixels on the book spine obey a planar parallax constraint across cameras. This “homography” filtering provides a robust way of extracting the region of interest without taking recourse to ad-hoc image processing operations. Furthermore this approach eliminates the need for carrying out a cross-camera color calibration. Using this straightforward geometric constraint, book spines can be reliable detected and segmented in the view of each camera as books are added to the shelf. One concern that may arise is as regards what would happen if the book spines did not in fact lie along the same plane as may be expected in the normal usage of the bookshelf. To see the effect of deviation from this assumption,

Proceedings of the Seventh IEEE Workshop on Applications of Computer Vision (WACV/MOTION'05)
0-7695-2271-8/05 $ 20.00 IEEE

consider the homographies H and $H + \Delta H$ and induced by the planes $n^T X + d = 0$ and a plane parallel to it but at a slightly different depth $n^T X + (d + \Delta d) = 0$ respectively. Following [9] we can show that

$$\Delta H = K'(tn^T)K^{-1}\frac{\Delta d}{d} \quad (2)$$

where K, K' represent the internal parameters of the cameras n, d represent the world plane coordinates of the assumed spine plane Π, t represents the translation between the cameras. As $\frac{\Delta d}{d} \rightarrow 0$ $\Delta H \rightarrow 0$. We evaluated this hypothesis in our experiments and it turned out that the degradation of performance in fact was not significant.

2.1 System Calibration

It is necessary to perform a two stage calibration of the the system namely : the pixel mapping between the left to the right camera and to the projector relative to the spine plane . Secondly, a set of color transfer functions that describe the mapping of the color gamuts of the projector and cameras must be carried out. Geometric calibration is used directly during the scene augmentation phase to correctly illuminate books and other parts of the scene (detected in the camera frame) using pixels in the frame-buffer of the projector. Color calibration is required to improve the robustness of image processing routines in the presence of projected information (see Section 2.2).

In order to obtain the homography between the two cameras, several corresponding points on the spines of the books in the two cameras were collected and the DLT method [9] was used to obtain the homographies H_{RL} and H_{LR}. In order to compute the homography between the projector and the cameras, a white board is placed against the bookshelf. Random dots are projected onto the screen and their positions in the camera frame are recorded. Using these point correspondences the homographies H_{LP} and H_{RP} are computed as before.

In order to compute the color transfer function between the projector and the camera a technique described in [11] is used. Essentially this involves projecting uniform color images of increasing intensity and observing the corresponding intensity in the corresponding observed image. This can be computed for each color channel, holding the other two color values constant at zero. Parameters of a sigmoidal function are fit to the mean measured data points using the nonlinear optimization.

2.2 Foreground Detection in the Presence of Projected Information

A unique aspect of this work is the close coupling of image processing and scene augmentation tasks. Because augmentation takes place in the environment, rather than on a see-through display for example, image processing algorithms must be able to take projected information into account. For example, when detecting foreground pixels against a background model, changing information projected onto the bookshelf for augmentation purposes can produce significant false positives. Here we describe how this can be taken into account during a foreground detection process. In practice the effect of projected information must be modeled so that it can be removed from the background prior to foreground segmentation. This is a two-phased process. First the contents of the projector must be mapped to their appropriate positions and colors in each of the camera frames. For a given frame-buffer each projector pixel is mapped to each camera position via the known homographies, $x_c = H^1_{pc} x_p$, to determine what pixels in the background image for camera c must be modified. The color value of projector pixel, x_p is then added to the color values already stored at x_c. The resulting color value is mapped through the sigmoid function recovered during the color calibration phase. This new image represents a synthetic background to which the current foreground image can be compared. Although we are studying how more sophisticated background models that involve multi-modal statistical descriptions of expected pixel color, currently we only update a single mean background model that can be directly subtracted from foreground images. Figure 2.2 depicts the foreground detection process in the presence of changing augmented information. The scene was captured as a book as being added to the shelf. At the same time, the results of a previous book search still reside in the frame-buffer of the projector and highlight the scene. By combining the geometrically corrected and color mapped values in the frame-buffer into the space of a monitoring camera (shown in Figure 2.2, this information can be used to construct a synthetic background image by overlaying information with the current background model (Figure 2.2c). This synthetic background yields foreground detection results that contain real (and not augmented) regions. Calibration error, unknown scene geometry (the top of the bookshelf cuts off some of the projected information), as well as unmodeled material surface characteristics such as the white book highlights can lead to false positives. However,foreground detection using synthetic backgrounds improves system robustness overall.

The foreground detection subsystem is a key component in the overall algorithm and is used to both detect the presence of new objects as well as objects that have been removed from or moved on the shelves. By taking into account projected information in the background model, the system can continuously monitor the scene while the augmentation system is in operation.

Proceedings of the Seventh IEEE Workshop on Applications of Computer Vision (WACV/MOTION'05)
0-7695-2271-8/05 $ 20.00 IEEE

(a) (b) (c) (d)

Figure 3: Foreground segmentation in the presence of augmented imagery. (a) Scene containing overlayed projected information as well as "real" change. (b) Framebuffer contents of the projector after they have been warped and color corrected for the left camera. (c) Synthetic expected image constructed after combining warped and corrected framebuffer contents with background model. (d) Detected differences.

2.3 Adding books to the bookshelf

For simplicity of explanation let us consider that the bookshelf is empty at first and the user has chosen the "book add" option. Let ESL and ESR denote the images of the empty shelf in the left and right cameras respectively. After the first book is added, background subtraction based on thresholding the norm of the difference in the RGB values of the corresponding foreground and background pixels is used to obtain a foreground image as shown in Figure 2(b). If projected information is present, foreground images are generated using the process described in Section 2.2 considering the frame buffer contents. Let FL_1 and FR_1 denote the foreground images for the left and right camera respectively, H_{LR} and H_{RL} denote the collineations transforming the right camera pixels to the left one and vice-versa respectively. In order to extract the spine in the left camera we do the following: The ON pixels in the left camera are transformed to the right camera via H_{RL}. The source pixel in FL_1 is turned off unless the corresponding pixel in FR_1 is ON to get SPL_1. To summarize:

$$SPL_1(\tilde{\mathbf{m}}_L) = \begin{cases} 1 & \text{if} \quad FL_1(\tilde{\mathbf{m}}_L) = 1 \text{ and} \\ & \quad\quad FR_1(H_{RL}\tilde{\mathbf{m}}_L) = 1 \\ 0 & \quad\quad \text{otherwise} \end{cases} \tag{3}$$

The largest bounding box $BL_1 = \{xL_1, yL_1, sxL_1, syL_1\}$ containing the resulting "homography filtered" image is then computed, where (xL_1, yL_1) denote the coordinates of the top left corner of the bounding box and sxL_1 and syL_1 denote the width and height of the bounding box. The color histogram HL_1 of the foreground image FL_1 in the region marked by BL_1 is computed and stored. Similar information for the right camera is computed as well. Also the time stamp of the book addition event is recorded. The user is then prompted to enter the name of the book. The book along with the associated bounding box, time of capture and the color histogram is entered into the database.

If a second book is added to the shelf, the spine information for the first book must be taken into account so that the sides of the first book are not included in the background when detecting the new foreground. To do this base background ESL is augmented by the pixels inside bounding box BL_1 superimposed on the new foreground

$$EL_1(i,j) = \begin{cases} FL_1(i,j) & \text{if} \quad (i,j) \in BL_1 \\ ESL(i,j) & \quad\text{otherwise} \end{cases} \tag{4}$$

The augmented EL_1 is an element of the current state $S(1)$ of the bookshelf. The state $S(t)$ of the bookshelf also includes other information such as identities of books currently present in the shelf, their locations and bounding boxes in both cameras.

After the user chooses "book add" option again and adds a new book, the updated background EL_1 is used to obtain the foreground images $F_L(2)$. The planar parallax technique, described before, is used to obtain the bounding box for the second book and the second book along with the associated bounding box, time of capture and the color histogram is entered into the database. The new augmented background EL_2 is computed as

$$EL_2(i,j) = \begin{cases} FL_2(i,j) & \text{if} \quad (i,j) \in BL_2 \\ FL_2(i,j) & \text{if} \quad (i,j) \in BL_1 \\ ESL(i,j) & \quad\text{otherwise} \end{cases} \tag{5}$$

The state of the bookshelf is updated to include EL_2 as also information about the second book. In this way by knowing the state of the bookshelf at any time t, new books can be added to the database. Figure 4 shows the steps in extracting the spine region of a newly added book.

2.4 Tracking book removal and replacement

We assume that a single change happens over a fixed time interval. When a book is removed from or replaced back into the bookshelf, the last known state of the bookshelf

Proceedings of the Seventh IEEE Workshop on Applications of Computer Vision (WACV/MOTION'05)
0-7695-2271-8/05 $ 20.00 IEEE

(a) (b) (c) (d)

Figure 4: Images of a bookshelf taken when a book is added. (a) Foreground image detected using the previous state of the bookshelf . (b) Foreground image after eliminating non-spine pixels using (3) (c) Bounding box BL_j computed from (c) superimposed on the foreground image for computation of the histogram HL_j for the book j. (d) Updated state of the bookshelf as described by (5).

$EL_t \in S(t)$ is used to obtain the foreground FL_{t+1} similar to the way it was done during building of the bookshelf. For the case when a book is removed the image of the region where it was removed from will contain parts of the sides of the other books or the background.

Once again the planar parallax technique described by (3) can be used to extract the "spine"portion of the foreground with the implicit understanding that it corresponds to the region where a book used to be if the change event was a "Book Remove". A bounding box $BL = (x, y, sx, sy)$ is computed for the resulting region and a color histogram $HL(t+1)$ is computed for the corresponding foreground image. If the change event was a "Book Replace" event, then $HL(t+1)$ is expected to be close to HL_i if book i was the book that was replaced. On the other hand if the change event was a "Book Remove" event then $HL(t+1)$ would correspond to a foreground region which includes the sides of the neighboring books and $HL(t+1)$ wont be close to any of $\{HL_1, \cdots, HL_N\}$. Specifically we compute

$$d^* = min_{i=1,\cdots,N} d(HL, HL_i) \quad (6)$$

If $d^* < T_{hist}$ then $i^* = argmin_i d^*$ is declared to be the book added. If $d^* > T_{hist}$, then the event is identified as a book removal event. In the absence of noise if the book m located at (x_j, y_j) was removed, then $x = x_j$ and $y = y_j$. To robustly detect which book was removed, we find the books $p \sim x_i$ and $q \sim x_{i+1}$ such that $x_i < x < x_{i+1}$. If book p was the one that was removed, then the region given by (x_i, y_i, s_x, s_y) in EL_t and in FL_{t+1} will be significantly different when compared using an appropriate histogram distance, while the region given by $(x_{i+1}, y_{i+1}, s_x, s_y)$ in EL_t and FL_{t+1} would be nearly identical. If we denote the histogram distance in the first case by d_i and the second case as d_{i+1}, then the book removed can be declared as

$$r \sim \text{argmax}_{i,i+1}(d_i, d_{i+1}) \quad (7)$$

where any appropriate histogram distance measure such as Bhattacharya distance or χ-squared distance [12] can used for computing the histogram distance.

In both cases the state of the bookshelf $S(t)$ is updated by adding appropriate information as regards replacement or removal.

2.5 Augmenting the smart bookshelf

Information about the past states of the bookshelf as well as information regarding a specific book can be made available easily to the user using the projector. The most common task that the system should perform is to facilitate access to a book requested by the user. For instance, suppose a particular book i is requested by the user, then the system checks the current state of the bookshelf $S(t)$ to check if the book i is present. If it is present, knowing the homography induced by the spine plane between the projector and the left camera H_{LP}, the bounding box BL_i corresponding to the ith book is prewarped and pixels contained inside it are turned ON

$$P(i,j) = \begin{cases} 1 & \text{if} \quad (i,j) \in H_{PL} BL_i \\ 0 & \text{otherwise} \end{cases} \quad (8)$$

If the book is not present then its "checked out" is displayed. Apart from highlighting the book requested by the user (Figures 5 (c),(d),(e)) several other visualization tasks regarding the library can also be performed. For instance, the user may request the state of the bookshelf or a particular book at an arbitrary time t (Figure 5 (f)). Furthermore auxiliary information about a book may also be projected into the scene. The projected information is taken into account during the subsequent image processing tasks as described in Section 2.2.

3 Experimental Results

The smart bookshelf was implemented in our lab using a pair of tripod mounted cameras and a single projector. The baseline between the cameras was about one meter and the projector was approximately between the two cameras as shown in Figure 5. We first estimated the homographies

Proceedings of the Seventh IEEE Workshop on Applications of Computer Vision (WACV/MOTION'05)
0-7695-2271-8/05 $ 20.00 IEEE

induced by the spine plane between the cameras and projectors as discussed in Section2.1. Users were told to add books to the bookshelf at different times as a part of the library building stage described in Section 2.3 and to query for the presence of a book and to remove or replace books back into the shelf. The system updates the states of the bookshelf and logs the time of capture providing a means of visualizing the state of the book shelf at any time.

The system monitors the bookshelf for changes by periodically capturing images of the shelf every five minutes. Using background subtraction incorporating the projected information as described in Section 2.2 followed by homography filtering (3), a bounding box is computed for each foreground image. The threshold for the background subtraction was chosen to ensure that there were few missed detections, since the planar parallax constraint is relied upon to deal with false positives. A "change" event is declared only when the bounding box is larger than a set threshold. In order to enter a new book into the bookshelf the user chooses a "Book Add" option. After a book is detected, the user is then prompted to enter the title of the book. The time of the "Book Add" event is logged also. The color histogram of the foreground image within the bounding box is then computed and stored. We used normalized (r, g) histograms as color models for the incoming book spines since they have greater illumination invariance. With each added book the state of the bookshelf is updated. As noted in Section 2, we assume that the book spines lie on the same world plane and the homography is estimated under this assumption. In our experiments the placement of books did not always follow this constraint. Nevertheless as explained in Section 2 and (2) if the depth variation Δd is small as compared to d the error introduced by deviations from the coplanarity assumption are small. In our setup the distance of the spine plane from both cameras was about one meter and we found that for small depth variations from the assumed book spine plane, the detection of spine regions was not adversely affected.

The other class of change events are the "Book Remove" and "Book Replace" events. Again the planar parallax constraint is applied to the detected foreground pixels. The normalized (r, g) histogram H for the changed region is computed. The histogram is compared to the histograms of the books present in the previous state of the bookshelf. Based on the smallest distance d^* between H and the histograms of the books present in the previous state of the bookshelf, the decision between "Book Remove" event or a "Book Replace" event as discussed in Section 2.4. We used the χ-squared histogram distance measure for comparing the histograms of the changed region with the book histograms. The computed histogram distance for the "Book Remove" event was found to be an order of magnitude higher than for the "Book Replace"event. Based on the detected change the state of the bookshelf is updated along with the time of a change event.

One drawback in directly employing the bounding box to represent the book spine is that the camera coordinate system may not be exactly aligned with the edges of the shelf. This results in the spine regions obtained by the homography filtering operation to be tapered. As a result the computed bounding box can enclose several "OFF" pixels. If this effect is not accounted for, some regions from the spine of the next book may be included in the model for the current book causing an error that accumulates with each successive addition. One way to deal with this is to estimate the skew of the edges of the shelf and to align the two coordinate systems before computing the bounding box. An alternative approach is to simply keep a record of the "ON" pixels within the bounding box. This bitmap together with the bounding box information prevents the error from being propagated with the addition of successive books and obviates the need for computing the skew of the shelf edges with respect to the camera coordinate system. The bitmap was used along with (4)to update the background model. The bit-map is also taken into account when computing the color histogram for each change region.

To access a particular book the user enters its title. By performing a lookup of the database, the bounding box information along with the bitmap related to that book is obtained. The coordinates of the ON pixels are warped using the homography between the projector and the camera and the desired book is highlighted as shown in the Figure 5(a). Additional information about the book can be projected on the upper edge of the shelf as well.

Since our system maintains a record of camera images representing the states of the bookshelf, we can rewind and display the state of the bookshelf at any time in the past. The spine information along with the bitmasks can be warped and projected to get an easy access to the desired state of the bookshelf as shown in Figure 5(f).

4 Conclusions and Future Work

In this paper we presented an initial step towards environments that are intelligently augmented by a camera-projector systems. The "smart bookshelf" uses two cameras and a projector, to provide coverage of the bookshelf to be monitored. The system builds a database that contains information related to book position, time of addition and the texture information on the book spine through normalized (r, g) histograms and continuously monitors the book shelf to detect addition, removal and replacement of books. A foreground detection algorithm which takes into account the projected information along with a planar parallax constraint is used to extract the spine of the book. Based on the detected change, the database is updated. Knowing the

Proceedings of the Seventh IEEE Workshop on Applications of Computer Vision (WACV/MOTION'05)
0-7695-2271-8/05 $ 20.00 IEEE

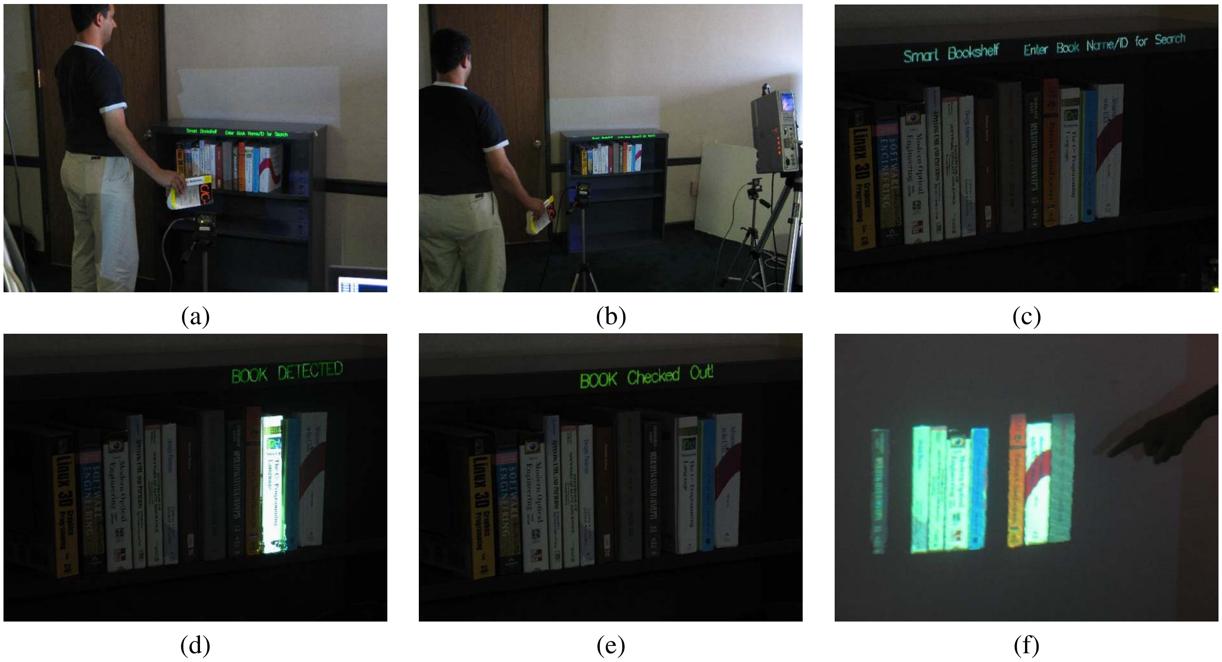

Figure 5: Images of the smart bookshelf (a) and (b) the setup (c) (d) and (e) show different states of the bookshelf in response to a query by the user (f) shows the projection of the requested past state of the bookshelf.

homography induced by the spline plane between the projector and the cameras, the smart bookshelf supports user query for the presence of a book, by automatic highlighting of the book on the shelf or indicating its absence if has been removed. Furthermore the system maintains a record of the state of the bookshelf which can be replayed as desired.

At present our approach assumes that a single change event occurs over a fixed time interval. One of our goals is to relax this constraint and allow for more robust change detection by making use of optical character recognition [13]. Furthermore we wish to explore the possibility of monitoring a space for object position and recording their appearances for dynamic playback.

References

[1] R. Raskar et al., "ilamps: geometrically aware and self-configuring projectors," *ACM Transactions on Graphics*, 2003.

[2] S.K Nayar, H. Peri, M. D. Grossberg, and P. N. Belhumuer, "A projection system with radiometric compensation for screen imperfections," *Proceedings of PROCAMS*, 2003.

[3] A. Majumder and R. Stevens, "Color non-uniformity in projection based displays," *IEEE Transactions on Visualization and Computer Graphics*, 2003.

[4] R.Raskar et al, "The office of the future : A unified approach to image-based modeling and spatially immersive displays," *Proc. of the ACM SIGGRAPH 1998*, 1998.

[5] 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.

[6] P. Wellner, W. Mackay, and R. Gold, "Computer-augmented environments: back to the real world," *Communications of the ACM*, 1993.

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

[8] S. Borkowski, O. Riff, and J. L. Crowley, "Projecting rectified images in an augmented environment," *Proceedings of PROCAMS*, 2003.

[9] R. I. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*, Cambridge University Press, 2000.

[10] P. Batavia and S. Singh, "Obstacle detection using color segmentation and color stereo homography," *Proceedings of ICRA*, 2001.

[11] C. Jaynes, S. Webb, M. Steele, M. Brown, and B. Seales, "Dynamic shadow removal from front projection displays," *Proc. of IEEE Visualization*, 2001.

[12] A.Kale, K. Kwan, and C. Jaynes, "Epipolar constrained user bushbutton selection in projected interfaces," *Proceedings of the 1st CVPR workshop on real time vision for human computer interaction*, 2004.

[13] V. Wu, R. Manmatha, and E.M. Riseman, "Textfinder: An automatic system todetect and recognize text in images," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1999.

Proceedings of the Seventh IEEE Workshop on Applications of Computer Vision (WACV/MOTION'05)
0-7695-2271-8/05 $ 20.00 IEEE