In Proc. Pervasive Computing and Advances in Pervasive Computing, Austrian Computer Society, 2004

Occlusion Detection for Front-Projected Interactive Displays

Maria Nadia Hilario*, Jeremy R. Cooperstock

Abstract

Advances in projector-camera technology allow for the transformation of virtually any surface into a display screen, leading to increased opportunities for interactive ubiquitous displays and mixed reality environments. While rear-projection environments are often prohibitively space-consuming, front-projection display systems suffer from occlusions. When a user interacts with the display or inadvertently blocks the projector, distortions appear in the projected image and shadows are cast on the display surface. However, sufficient knowledge of these occlusions allows for a corrected projection display in which overlapping projectors can fill in the occluded region, thereby producing an apparently unoccluded display. As a starting point to this objective, an occlusion detection system for a front projection display environment is presented. The approach is based on a camera-projector color calibration algorithm that estimates the RGB camera response to projected colors, allowing for predicted camera images to be generated for each projected scene. Pixel-wise color differencing between observed and predicted camera images is then employed to locate occluded display regions.

1. Introduction

Significant progress has been made in addressing the challenges associated with creating high-quality projected displays, often by exploiting computer vision techniques. For instance, camera-projector calibration algorithms have been developed to achieve geometric and color seamlessness [8][13][4] across multiple overlapping projectors. This effectively creates a single, possibly non-planar, high-resolution display. Other camera-projector systems have been developed for projecting compensated images onto textured surfaces [5] or texture onto three-dimensional objects [1]. As advances in projector-camera technology allow virtually any surface to be transformed into a display screen, several opportunities emerge for innovative use of video projection in interactive ubiquitous displays [6][12][16] and mixed reality environments [7][10]. For instance, Pinhanez [6] introduces a steerable camera-projector system to follow users across a space and transform nearby surfaces into projected touch screens. As well, Raskar et al. [10] use hand-held environment-aware wireless projectors to augment the physical world with self-configuring displays.

Rear projection systems unfortunately tend to pose inordinate space requirements, thereby motivating the use of front projection instead, in interactive display environments. However, a computer vision challenge related to front-projected displays is that of occlusion detection. Occlusions occur when a user interacts directly with the display (e.g. via gestures or tangible bits), or inadvertently blocks

*Centre for Intelligent Machines, McGill University, 3480 University Street, Montreal, QC, Canada H3A 2A7, mnhilar@cim.mcgill.ca

the projector. Differences in depth, surface shape and reflectance properties between the display surface and the occluding object can lead to distortions in the projected image. As well, shadows are often introduced on the display, resulting in loss of information in the occluded region. Knowledge of occlusions is therefore imperative in order to reduce the undesirable effects of shadows and to avoid projecting distracting light on users. It can also be used to assist in such tasks as body or hand tracking. As another interesting application, an occluding object itself could potentially be augmented by customizing the projected imagery in the corresponding display region.

Our first step towards achieving this goal was to build a prototype camera-projector system for occlusion detection in a front-projected display environment. Our approach is based on a camera-projector color calibration algorithm that estimates the camera response to projected colors. Camera images can then be predicted for each projected scene and compared with observed images to locate occluded display regions. Such regions correspond either to the occluding object itself or to its shadow.

2. Related Work

Since illumination from front projection can severely distort the inherent surface color of an occluding object [6][14][15][17], this complicates the task of locating the source of occlusion based solely on its appearance (e.g. with the use of color recognition algorithms). This problem has been addressed in the context of specific applications, for instance, hand detection or tracking in augmented reality environments. Takao et al. [14] use background segmentation with the constraint that the user's hand is only detected when proximal to the projection region. Pinhanez [6] employs motion segmentation, assuming that occlusions are caused only by moving objects. The weakness of the previous method is addressed by Von Hardenberg and Brard [17], who combine motion segmentation and image differencing with a reference image to track moving and resting hands. However, dynamic projected imagery can result in false identification of occluding objects. Sato et al. [11] detect hands using a thermal infrared camera that senses objects within body temperature range. While robust to changes in background and skin color, this approach requires expensive specialized equipment.

Other occlusion detection techniques have been applied to shadow elimination [2][3], where multiple projectors provide redundant illumination in shadowed display regions. Rather than track the user directly, Cham et al. [2] and Jaynes et al. [3] detect occlusions by tracking shadows on the display. Assuming an unobstructed camera view of the projected scene, shadows are identified via pixel-wise differencing between predicted and observed camera images; they are then filled in by a second unoccluded projector. Cham et al. also use their shadow detection algorithm to infer the location of, and avoid projecting light on the user. The two shadow elimination methods [2][3] differ in that the former does not support dynamic displays. Images to be projected are known at system startup, allowing a reference camera image of each unnocluded scene to be pre-generated. Occlusions cannot be detected if projected images are constructed on the fly. Alternatively, Jaynes et al. [3] perform camera-projector color calibration to estimate the color transfer function between the two devices, which is used to dynamically generate a predicted camera image for the current projected scene.

Both shadow elimination methods impose the same constraint regarding system setup: they both shadow elimination methods require the camera to be positioned so as to maintain an unobstructed view of the projection surface. For applications in which only suppression of projected light on the user is required, Tan and Pausch [15] employ a camera with an infrared (IR) filter. Both camera and projector are aligned with the projection surface and are assumed to share the same focal point. An off-axis IR LED array then reflects IR light off the surface, allowing occluders to be detected by locating their shadow in IR camera images. This approach, however, cannot be used to eliminate visible shadows on the display.

Many of the algorithms described above were designed for specific applications. Assumptions concerning the characteristics of occluding objects (e.g. hand position, motion or temperature), as well as constraints imposed on the placement, either of cameras or IR arrays, may reduce possibilities for user interaction. However, by detecting occluded regions directly in the scene, By detecting all occluded regions directly in the scene, and determining whether each represents the occluding object itself or its shadow on the display, we hope to obtain a flexible algorithm that can be applied as a preliminary step to various image processing tasks, as required for generic front projection applications.

3. Algorithm Overview

Our occlusion detection algorithm consists of offline camera-projector calibration, followed by online occlusion detection for each video frame. Calibration is performed in two steps, namely geometric registration to compute the transformation from projector to camera frames of reference, and color calibration, to determine the mapping from projected colors to their corresponding camera response. Calibration results are used to construct predicted camera images of each projected scene. Occlusion detection occurs by pixel-wise differencing between predicted and observed camera images.

As an initial simplification, our current setup employs only a single camera and a single projector in a static configuration. We also assume a planar Lambertian display surface of uniform color, constant lighting conditions, and negligible intra-projector color variation. In future systems, we plan to scale our algorithm to support multiple cameras and projectors, as well as arbitrary display surface shapes.

3.1. Offline Geometric Registration

To maintain a flexible system setup, we minimize constraints on camera and projector placement. Each device may be mounted at an arbitrary position and orientation with respect to the display surface. Furthermore, except during offline calibration, the camera need not maintain an unobstructed view of the display, permitting the user to have direct contact with the projected display.

We derive the projector-camera pixel mapping and use it when generating predicted camera images. The system can also compute a transformation that prewarps the projector image to correct keystone distortion due to off-centered projection. The resulting projected image is aligned with the edges of a specified world coordinate frame. Numerous methods have been proposed for automating both geometric registration and keystone correction for camera-projector systems [9][12]. We adopt an approach based on the work of Sukthankar et al. [12], and model the camera-projector geometric transformation as a planar homography, given by:

$$[x_p, y_p, w_p]^T = H_{pc}[x_c, y_c, w_c]^T$$

Pixel correspondences required to compute the 3x3 matrix H_{pc} and the projector prewarping transformation are obtained by respectively detecting the corners of a projected and printed grid in the camera view. For the scope of this paper, we omit the detailed description of these computations. Although we currently do not model radial distortion due device optics, our model produces acceptable results for simple projected textures.

3.2. Offline Color Calibration

Due to the many factors influencing the camera-projector system, including sensor and filter characteristics, display surface properties, as well as geometric configuration, a camera viewing a projected display is unlikely to produce an image whose colors match exactly those from the source image. In order to generate predicted camera images for each projected scene correctly, we must ascertain the color mapping from projected colors to camera colors. We estimate the color transfer function by iterating through the projection of primary colors of varying intensities, measuring the RGB camera response for each and storing this data as a color lookup table (CLUT). This approach is similar to that of Jaynes et al. [3]. A small difference is that they complete calibration more quickly by projecting fewer color samples and computing a parametric fit; however, more color deviations might occur as a result of curve fitting.

We speed up our algorithm by projecting color calibration grids rather than separate images for each calibration color. We use the pre-computed homography H_{pc} to map a rectangular subregion of each color patch to camera space. The recorded camera response is the average RGB color over corresponding patch pixels, measured over multiple camera images. As the stored CLUT values include the camera response to the projector's black offset we must also record the RGB response to an assumed pure black color $(R = 0, G = 0, B = 0)$. The predicted camera response, $\mathbf{C}_c$ to a projected color $\mathbf{C}_p = (R_p, G_p, B_p)$ can then be computed by summing the predicted camera responses to each of the projected color components, (R_p, G_p, B_p) in isolation, and accounting for the black offset. This predicted value is then compared to the observed camera response at run-time.

3.3. Online Occlusion Detection

Camera-projector calibration results are used in the online process of occlusion detection. For each video frame, a predicted camera image is constructed by using the H_{pc} homography to map the projector framebuffer image to camera space. Colors in this image are defined on a per pixel basis by using the original projected colors as an indices into the CLUT. To account for the (typically) lower camera resolution than projector resolution, the color for each pixel in the predicted camera image is a color combination of a group of pixels in the projector input image. Pixel-wise color differencing can then be conducted between the expected and observed camera colors. Significant discrepancies between expected and observed values indicate the presence of distortions in the projected display, which are assumed to be caused by occlusions.

4. Preliminary Results and Discussion

Preliminary results of our occlusion detection method were obtained using a simple front-projected scene. Figure 1 illustrates the input projector image, the predicted camera image of the unoccluded scene, as well as the corresponding observed camera image for comparison. We illustrate two common situations where occlusions occur in a front-projected interactive display. In Figure 2(a), the observed camera image on the left depicts the shadow cast on the display surface as a result of the user blocking the projector. Occluded display regions detected by the system are indicated in the right image, in which occluded pixels are set to green. The second example, depicted in Figure 2(b), places the occluding object in direct contact with the display surface. The corresponding occluded display regions are also detected.

For projected scenes where both of the above situations occur simultaneously, further analysis of each

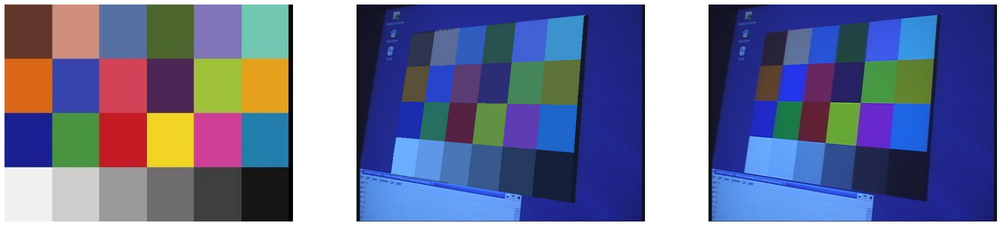
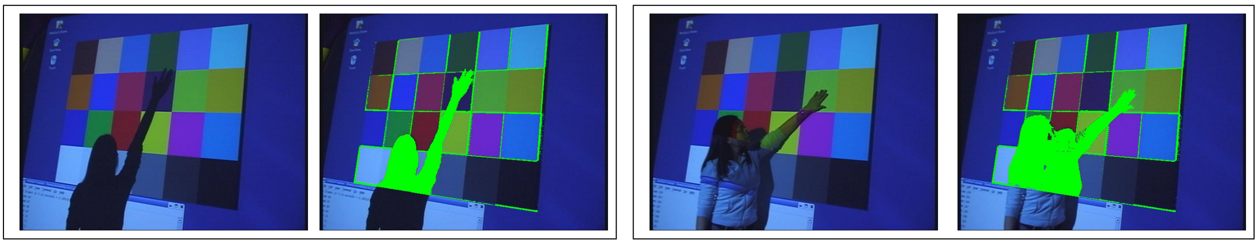


Figure 1. Camera view prediction; projector image (left), predicted (middle) and observed (right) camera images of the unoccluded scene.



(a) (b)

Figure 2. Camera view of occluded scenes and detected occlusions; (a) shadow on the display surface, (b) occluding object in direct contact with the display surface.

occluded display region must be performed to determine the source of the occlusion, i.e. whether the color distortion is due to the occluding object itself or to its shadow on the display surface. Other important considerations include handling cases where the reflectance properties of the occluding object closely match those of the display surface, or where the projected light overwhelms the color of the object. In this case, the object may blend in with the display.

In interactive ubiquitous display environments employing front projection, processed occlusion information can be useful for various tasks. As mentioned, redundant illumination by a second projector can eliminate shadows, producing a seemingly unoccluded display. For occluded display regions corresponding to the occluding object itself (e.g. the user), distracting projected light can be suppressed. Alternatively, a very interesting application would be to augment an occluding object with projected light. Undesirable display distortions could be replaced with customized projection in that region, thus achieving the effect of *painting* light on the user. This could be useful in the design of flexible projected interfaces. As a simple example, different colors or shapes could be projected on the user's hand to convey system feedback. A more difficult task would be to project interface components (e.g. menus) onto the user's arm and body for easy accessibility.

5. Summary

Preliminary results of our camera-projector algorithm indicate its potential to perform general occlusion detection in a front projection display environment. Different forms of occlusion (i.e. both shadows and the occluding object itself) can be detected, as necessary for interactive applications and shadow-corrected displays. In order to simplify the initial task of occlusion detection, our prototype makes certain assumptions, such as constant lighting and planar Lambertian display surfaces. To improve robustness for ubiquitous displays, these constraints will, no doubt, need to be relaxed. However, of more immediate importance is the need to scale the algorithm to support multiple cameras and projectors, in order to realize a true occlusion-free front-projection display environment. As well, the system must be able to recognize the user in order to respond to interaction by customizing projected information. An effective algorithm for occlusion detection and analysis would increase the feasibility of using front projection for ubiquitous interactive displays.

6. Acknowledgements

Our thanks to Christopher Jaynes of the University of Kentucky for helpful discussions regarding the shadow removal approach described in reference [3].

References

[1] D. Bandyopadhyay, R. Raskar, and H. Fuchs. Dynamic Shader Lamps: Painting on Movable Objects. *IEEE and ACM International Symposium on Augmented Reality (ISAR'01), New York, New York*, pages 207–216, October 2001.

[2] T.-J. Cham, J. Rehg, R. Sukthankar, and G. Sukthankar. Shadow Elimination and Occluder Light Suppression for Multi-Projector Displays. In *Proceedings: Computer Vision and Pattern Recognition*, 2003.

[3] C. Jaynes, S. Webb, R. Steele, M. Brown, and W. Seales. Dynamic shadow removal from front projection displays. In *Proceedings: IEEE Visualization*, 2001.

[4] A. Majumder and R. Stevens. Color Non-Uniformity in Projection Based Displays: Analysis and Solutions. In *IEEE Transactions on Visualization and Computer Graphics*, Vol. 10, No. 2, pages 177–188, To appear March/April 2004.

[5] S. K. Nayar, H. Peri, M. D. Grossberg, and P. N. Belhumeur. A Projection System with Radiometric Compensation for Screen Imperfections. In *Proceedings: ICCV Workshop on Projector-Camera Systems (PROCAMS-2003), Nice, France*, October 2003.

[6] C. Pinhanez. The Everywhere Displays Projector: A Device to Create Ubiquitous Graphical Interfaces. In *Proceedings: Ubiquitous Computing 2001 (Ubicomp'01), Atlanta, Georgia*, September 2001.

[7] R. Raskar, G. Welch, M. Cutts, A. Lake, L. Stesin, and H. Fuchs. The Office of the Future: A Unified Approach to Image-Based Modeling and Spatially Immerse Displays. In *Proceedings: Computer Graphics, Annual Conference Series*, pages 179–188, 1998.

[8] R. Raskar. Immersive Planar Displays using Roughly Aligned Projectors In *Proceedings: IEEE Virtual Reality*, 2000.

[9] R. Raskar and P.A. Beardsley. A Self-Correcting Projector. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, Vol. 2, pages 504–508, December 2001.

[10] R. Raskar, J. van Baar, P. Beardsley, T. Willwacher, S. Rao, and C. Forlines. iLamps: Geometrically Aware and Self-Configuring Projectors. In *ACM Transactions on Graphics (TOG)*, Vol. 22, No. 3, pages 809–818, July 2003.

[11] Y. Sato, Y. Kobayashi, and H. Koike. Fast tracking of hands and fingertips in infrared images for augmented desk interface. In *Fourth International Conference on Automatic Face and Gesture Recognition*, 2000.

[12] R. Sukthankar, R. Stockton, and M. Mullin. Automatic keystone correction for camera-assisted presentation interfaces. In *Proceedings: ICMI*, 2000.

[13] H. Chen, R. Sukthankar, G. Wallace, and K. Li. Scalable Alignment of Large-Format Multi-Projector Displays Using Camera Homography Trees. In *Proceedings: IEEE Visualization*, 2002.

[14] N. Takao, J. Shi, and S. Baker. Tele-Graffiti: A Camera-Projector Based Remote Sketching System with Hand-Based User Interface and Automatic Session Summarization. In *International Journal of Computer Vision*, Vol. 53, No. 2, pages 115–133, July 2003.

[15] D.S. Tan, and R. Pausch. Pre-emptive Shadows: Eliminating the Blinding Light from Projectors. In *Interactive poster at CHI 2002 Conference on Human Factors in Computing Systems, Minneapolis, MN*, 2002.

[16] Y. Tokuda, S. Iwasaki, Y. Sato, Y. Nakanishi, and H. Koike Ubiquitous display for dynamically changing environments. In *ACM Conference on Human Factors in Computing Systems (CHI2003) Extended Abstract*, 2003.

[17] C. von Hardenberg and F. Brard. Bare-Hand HumanComputer Interaction. In *Proceedings: Proceedings: ACM Workshop on Perceptive User Interfaces, Orlando, Florida*, November 15-16 2001.