

OCCLUSION DETECTION IN FRONT PROJECTION ENVIRONMENTS BASED ON CAMERA-PROJECTOR CALIBRATION

Maria Nadia Hilario

Department of Electrical and Computer Engineering
McGill University, Montréal

October 2005

A thesis submitted to McGill University
in partial fulfilment of the requirements of the degree of
Master of Engineering

© MARIA NADIA HILARIO, 2005



Library and
Archives Canada

Bibliothèque et
Archives Canada

Published Heritage
Branch

Direction du
Patrimoine de l'édition

395 Wellington Street
Ottawa ON K1A 0N4
Canada

395, rue Wellington
Ottawa ON K1A 0N4
Canada

Your file Votre référence

ISBN: 978-0-494-22647-6

Our file Notre référence

ISBN: 978-0-494-22647-6

NOTICE:

The author has granted a non-exclusive license allowing Library and Archives Canada to reproduce, publish, archive, preserve, conserve, communicate to the public by telecommunication or on the Internet, loan, distribute and sell theses worldwide, for commercial or non-commercial purposes, in microform, paper, electronic and/or any other formats.

The author retains copyright ownership and moral rights in this thesis. Neither the thesis nor substantial extracts from it may be printed or otherwise reproduced without the author's permission.

AVIS:

L'auteur a accordé une licence non exclusive permettant à la Bibliothèque et Archives Canada de reproduire, publier, archiver, sauvegarder, conserver, transmettre au public par télécommunication ou par l'Internet, prêter, distribuer et vendre des thèses partout dans le monde, à des fins commerciales ou autres, sur support microforme, papier, électronique et/ou autres formats.

L'auteur conserve la propriété du droit d'auteur et des droits moraux qui protègent cette thèse. Ni la thèse ni des extraits substantiels de celle-ci ne doivent être imprimés ou autrement reproduits sans son autorisation.

In compliance with the Canadian Privacy Act some supporting forms may have been removed from this thesis.

Conformément à la loi canadienne sur la protection de la vie privée, quelques formulaires secondaires ont été enlevés de cette thèse.

While these forms may be included in the document page count, their removal does not represent any loss of content from the thesis.

Bien que ces formulaires aient inclus dans la pagination, il n'y aura aucun contenu manquant.


Canada

Abstract

Camera-projector systems are increasingly being used to create large displays for data visualization, immersive environments and augmented reality. Front projection displays, however, suffer from occlusions, resulting in shadows and light being cast, respectively, onto the display and the user. Researchers have begun addressing the issue of occlusion detection to enable dynamic shadow removal and to facilitate automatic user sensing in interactive display applications. A camera-projector system for occlusion detection in front projection environments is presented. The approach is based on offline, camera-projector geometric and color calibration, which then enable online, dynamic camera view synthesis of arbitrary projected scenes. Occluded display regions are detected through pixel-wise differencing between predicted and captured camera images. The implemented system is demonstrated for dynamic shadow detection and removal using a dually overlapped projector display.

Résumé

Les systèmes de type projecteur-caméra sont de plus en plus utilisés pour la visualisation de données avec des affichages de grande envergure, les environnements immersifs et la réalité augmentée. Cependant, les affichages de projection frontale introduisent un problème d'occlusions, ayant pour conséquences la projection d'ombres et de lumière sur l'écran et l'utilisateur. Les chercheurs ont commencé à aborder le problème de détection d'occlusions dans le but de permettre l'enlèvement dynamique d'ombres et de faciliter la détection automatique d'utilisateurs dans les systèmes d'affichage interactifs. Un système projecteur-caméra de détection d'occlusions appliqué aux environnements de projection frontale est présenté. L'approche s'appuie sur le calibrage hors-ligne (géométrie et couleur) du système projecteur-caméra, ce qui permet par la suite la synthèse en ligne et dynamique de vues caméra qui correspondent à des scènes arbitrairement projetées. Afin de détecter les régions de l'affichage qui sont occlues, l'image caméra prévue est comparée avec celle capturée à l'aide d'une méthode de différence pixel à pixel. Ce système est utilisé dans une application de détection et d'enlèvement dynamique d'ombres, où l'affichage est assuré par deux projecteurs superposés.

Acknowledgments

I would like to thank my supervisor, Dr Jeremy R. Cooperstock, for the invaluable research opportunities while working in the SRE Lab. I greatly appreciated his technical guidance, editorial advice, funding and patience as I completed this thesis. This research was supported by Fonds de recherche sur la nature et les technologies, Fondation J. Armand Bombardier and Valorisation-Recherche Québec; this support is gratefully acknowledged. I would like to thank Daniel Sud for implementing the dual-projector display system used for shadow removal and for the technical discussions during the integration of our work. Thanks also to Jianfeng Yin, Stephen Spackman and Wei Sun, whose technical insights pointed me in the right direction, as well as to François Rioux for helping with the French version of the abstract. To Amit Sihota, many thanks for being a great friend and study buddy. To Patrick Di Nardo, I am truly grateful to have had his unconditional support and understanding. Last but certainly not least, special thanks to my parents, Gisela Villamor and Virgilio Hilario, and to my sister, Cynthia “Ate” Hilario: this thesis is dedicated to my family, to whom I am grateful for their support, encouragement and patience throughout the entire course of my studies.

TABLE OF CONTENTS

Abstract	i
Résumé	ii
Acknowledgments	iii
LIST OF FIGURES	vii
LIST OF TABLES	ix
CHAPTER 1. Introduction	1
1.1. Front Projection for Large Interactive Displays	1
1.1.1. Projection-Based Displays	1
1.1.2. Advantages of Front Projection	2
1.2. Problem Description	2
1.3. Research Overview	4
1.4. Thesis Outline	6
CHAPTER 2. Literature Review	8
2.1. Indirect Occlusion Detection Techniques and Applications	8
2.1.1. Shadow Detection and Removal	9
2.1.2. Occluder Light Suppression	11
2.1.3. Identifying the Occluded Projector in AVRP Displays	11

TABLE OF CONTENTS

2.1.4. Motivation for the Adopted Approach to AVRP	12
2.2. Direct Occlusion Detection Techniques and Applications	13
CHAPTER 3. Camera-Projector Calibration Background	16
3.1. Geometric Calibration Background	16
3.1.1. Geometry of a Camera-Projector Pair	17
3.1.2. Geometric Calibration Approaches	18
3.2. Color Calibration Background	20
3.2.1. Projector-to-Camera Color Transfer	21
3.2.2. Color Calibration Approaches	23
CHAPTER 4. Camera-Projector System Design	26
4.1. System Framework	26
4.1.1. System Setup	26
4.1.2. Algorithm Overview	28
4.1.3. Software Architecture	29
4.2. Offline Geometric Calibration	31
4.2.1. Geometric Calibration Model	31
4.2.2. Procedure for Homography Computation	34
4.3. Offline Color Calibration	35
4.3.1. Color Calibration Models	36
4.4. Online Occlusion Detection	44
4.4.1. Camera View Synthesis	44
4.4.2. Image Differencing	45
4.5. Shadow Removal Application	46
CHAPTER 5. Results and Improvements	48
5.1. Occlusion Detection	48

TABLE OF CONTENTS

5.2. Shadow Detection and Removal	51
5.3. Variable Thresholding for Image Differencing	54
5.4. Morphological Image Smoothing for Noise Reduction	56
5.5. Performance Issues and Potential Improvements	57
CHAPTER 6. Conclusion	60
Appendices	62
APPENDIX A. Projective Geometry Theory	62
A.1. Homogeneous Notation and the Projective Space	62
A.2. Projective Transformations	64
APPENDIX B. Color Theory	66
B.1. Color Perception and the Tristimulus Theory	66
B.2. Color Measurement and Representation	68
REFERENCES	70

LIST OF FIGURES

3.1 The (camera or projector) pinhole model. (based on Figure 2.8 of reference [43])	17
3.2 Geometry of a camera-projector pair.	18
3.3 Projector-to-camera color transfer for projector channel j , $j \in \{r, g, b\}$. . .	22
4.1 System framework for the single projector display case: example of the physical setup and snapshot images of an unoccluded display.	27
4.2 System framework for the case of a dually overlapping projector display: example of the physical setup and snapshot images of an unoccluded display.	28
4.3 Overview of the occlusion detection algorithm (to the left of the dashed line), extended to support shadow removal.	29
4.4 Software architecture for the camera-projector system for occlusion detection, extended to support shadow removal.	30
4.5 Camera-projector geometric calibration using planar homographies, for the single projector case.	32
4.6 Camera-projector geometric calibration using planar homographies, for the dual-projector case.	34
4.7 Projector-camera point correspondences for computing \mathbf{H}_{pic}	35

4.8 The per channel color lookup table.	37
4.9 CLUT color calibration: sample camera images captured during separate calibration of the red, green and blue projector channels.	38
4.10 Projector-camera color correspondences for computing M_{pc}	40
4.11 Auto vs. manual camera exposure (experiments with light integration time, aperture size and gain on the Sony MiniDV DCR-TRV900).	43
5.1 Occlusion detection using the CLUT model for color calibration.	49
5.2 Occlusion detection using the LLSColorMat model for color calibration. . .	50
5.3 CLUT vs. LLSColorMat color calibration for occlusion detection.	51
5.4 Shadow detection and removal process for camera frame i	52
5.5 Shadow removal process for a sequence of captured camera frames.	53
5.6 Active shadow removal using two projectors with more similar overall intensities. . .	54
5.7 Variable thresholding.	55
5.8 Smoothing the binary occlusion map through erosion-dilation.	57
A.1 Model of the projective plane \mathbb{P}^2 . (based on Figure 1.1 of reference [12]) .	63
A.2 Projective transformation via central projection. (based on Figure 1.3 of reference [12])	64
B.1 Color perception.	67
B.2 Spectral power distribution (SPD) of a color stimulus. (reproduced from Figure 8 of reference [11])	68

LIST OF TABLES

4.1 Camera color prediction using the CLUT.	38
4.2 The linear least squares problem. (based on material from reference [27]) .	41
4.3 Pixel-wise image differencing technique.	45

CHAPTER 1

Introduction

1.1. Front Projection for Large Interactive Displays

1.1.1. Projection-Based Displays. Projection-based displays are increasingly being used in data visualization, immersive environments and augmented reality applications. Projectors provide an affordable solution for large area, high resolution display, making them an appealing alternative to traditional display devices, such as CRTs, LCDs and plasma screens. In virtual environments [6][7][31][5][15], the use of projection provides a wide field of view and sense of presence, while freeing the user from cumbersome head-mounted displays. Furthermore, by exploiting computer vision techniques, significant research progress has been made in achieving scalable and flexible projected displays that are easy to configure. For instance, camera-projector calibration algorithms enable automatic geometric alignment [28][4][1][18][38] and color seamlessness [21][33][46] across multiple tiled, but casually overlapped projectors, effectively creating a single logical wall-sized display. As well, ad-hoc clusters of vision- and network-enhanced projectors can be used to create self-configuring displays [30], facilitating system installation and maintenance.

1.1.2. Advantages of Front Projection. Rear projection systems, such as the immersive CAVE [6] or the Princeton Scalable Display Wall [18], require specialized infrastructure (e.g. high-quality screens) and tend to pose inordinate space requirements, thereby motivating the use of front projection instead. With the latter technology, arbitrary surfaces, such as walls, workspaces and objects, can serve as display screens and geometric calibration used for automatic keystone¹ correction [36][29]. A new radiometric calibration technique also enables the radiometric compensation of projected displays to mask the underlying color or texture of imperfect display surfaces [23]. Front projection therefore provides a flexible solution for projection-based display and has been used for data visualization [1], augmented workspaces [47][45][39], and immersive telepresence applications [31][5][15]. More recently, there has also been growing interest in the development of novel camera-projector systems to create intelligent and interactive *ubiquitous displays* that make digital information readily available in everyday environments [24][42]. For instance, Pinhanez [24] introduced a steerable camera-projector system that follows the user in a room and transforms nearby surfaces into projected touchscreens. Raskar et al. [30] use hand-held environment-aware wireless projectors to augment the physical world with self-configuring displays.

1.2. Problem Description

An inherent problem with front projection environments is that of occlusion. When a user interacts with the display (e.g. via hand gestures) or otherwise inadvertently blocks the projector beam, distracting shadows are often cast onto the display surface, resulting in loss of information in the occluded region. Graphics projected

¹*Keystone distortion* refers to the trapezoidal display of a rendered source image due to off-axis projection.

onto the user or other occluding object are also distracting and lead to distorted imagery.

In a comparative study of projection technologies [37], it was observed that for front-projected interactive surfaces, most users develop shadow coping strategies, for example standing at the edge of the display to prevent occlusion. It was also noted that shadows did not significantly hinder user performance for simple tasks. The study was conducted with a single-user interactive display that recognized simple hand gestures for performing low-level user operations (e.g. selection, dragging, etc.). For other applications, however, the effects of shadows can be more significant. For instance, they detract from the sense of presence conveyed by virtual environments and life-size videoconference displays. In telepresence applications, they may furthermore interfere with human-human interaction by causing the local user to miss certain visual cues (e.g. sight of remote participants' gestures) that are important for successful communication, particularly when performing telecollaborative tasks.

While the occurrence of occlusion can be reduced by mounting the projector off-axis from the display surface, recent research has addressed the problem using a more active approach, called *Active Virtual Rear Projection (AVRP)*² [16][17][35][9]. Since rear projection does not suffer from shadows, AVRP simulates it using redundant overlapping front projectors, whose output is varied based on visual feedback from a camera. The camera is used to detect shadows as they occur, allowing for them to be filled in by an unoccluded projector. As well, by determining which projector is occluded, it is possible to avoid projecting distracting light on users [3]. Studies indicate that users prefer AVRP to simple front projection display [37].

²The term *Active Virtual Rear Projection* was introduced by Summet et al. [37] to encompass such techniques.

Furthermore, in an interactive front projection environment, the source of occlusion typically corresponds to the object of interest in the scene, e.g. a hand performing a gesture or user moving in an immersive space. Knowledge of occlusions is therefore also relevant to computer vision tasks that are intended for support of human-computer interaction (HCI), such as hand detection for gesture recognition, or person detection and tracking. Additionally, rather than suppressing light, an occluding object itself could potentially be augmented by customizing the projected imagery in the corresponding display region.

A computer vision technique for occlusion detection is therefore required to reduce automatically the undesirable effects of shadows, facilitate automatic sensing, and respond to user input. Addressing these issues will help make front projection technology more usable in interactive display applications.

1.3. Research Overview

We present a camera-projector system that performs occlusion detection in front projection environments. This research was motivated by the need to address the occlusion problem in the Shared Reality Environment (SRE), which is a networked multi-room research facility being developed to provide a spatially immersive virtual environment for distributed, computer-mediated human-human interaction [5]. Our primary goal was to develop an occlusion detection system and integrate it with the SRE’s multiply overlapped front projection display for the purpose of dynamic shadow detection and removal.

Ideally, however, the adopted algorithm would be one that could be applied not only for shadow detection, but also to facilitate general object detection in front projection environments, where unpredictable background and lighting changes are often induced by dynamic projected content. A generic technique could be adapted

for various HCI tasks relevant to interactive display applications. These objectives implied the need for a flexible technique that would work for dynamic displays and different display configurations, i.e. single or multiply overlapping projectors.

The adopted occlusion detection approach relies on the fact that information about the background scene, i.e. the projected imagery, is readily available. That is, projected content is known a priori for every display frame. Based on the current projector framebuffer image, it is therefore possible to predict at any instant the appearance of the display from the perspective of a monitoring camera. This applies even to dynamic displays where content often changes in unpredictable ways. In the case of multiply overlapping projectors, the camera view can be predicted by accounting for the contribution of each. Finally, occlusions can be detected by comparing captured camera images to predicted images of the display as it would appear when unoccluded.

The implemented occlusion detection algorithm is based on offline camera-projector geometric and color calibration, which estimate the image warping transform and color transfer function, respectively, between each camera-projector pair. We express the required geometric mappings as planar homographies and describe two alternative representations for the color transfer function, namely the per channel color lookup table (CLUT) and the linear least squares color transfer matrix (LLSColMat). Calibration is the enabling step for camera view synthesis of arbitrary projected scenes. Derived data is used online to generate a predicted image of the current projected background that accounts for the geometrically and color transformed appearance of the display when viewed by the camera. Occlusions can then be detected for each camera frame, by pixel-wise comparing the predicted and captured camera images to locate regions where significant color inconsistencies occur. This calibration-based detection approach is similar to that proposed by Jaynes et al. [16]. Detected occlusion

regions may represent either shadow artifacts on the display or the occluding object itself; the nature of the occlusion depends on camera-projector placement. The final output of the occlusion detection system is a binary occlusion map, i.e. a camera-sized image in which occluded display pixels are identified.

We demonstrate the performance of the implemented system when integrated with a prototype dual-projector AVRP display for shadow detection and removal. For this application, the dually overlapped projector display was configured in a way similar to that of Flagg et al. [9], such that each display pixel is illuminated by exactly one projector at any given time. Shadows identified during occlusion detection are then eliminated by instructing the unoccluded projector to illuminate the display in corresponding regions.³

In addition, we describe two simple techniques that were used to improve the results of our occlusion detection system. The first is a variable thresholding scheme for facilitating detection in dark display regions, and the other is an image smoothing step based on morphological erosion-dilation for reducing noise in the occlusion map.

Since our current implementation runs at low frame rate, we also highlight potential performance improvements that may enable the detection process to reach real-time (≥ 25 Hz) or interactive (≥ 10 Hz) rates required for interactive display applications. Suggestions include performing region-based occlusion detection, compressing occlusion data for efficient network transmission, as well as accelerating certain image processing operations using commodity graphics hardware.

1.4. Thesis Outline

The remainder of this thesis is organized as follows. We review previous occlusion detection techniques and applications in Chapter 2. We then provide background

³Work on the AVRP system was done in collaboration with SRE lab member Daniel Sud, who implemented the dually overlapped projector display system used for shadow removal.

on the camera-projector geometric and color calibration problems in Chapter 3. In Chapter 4, we describe the framework of the implemented camera-projector system, the occlusion detection algorithm, as well as integration with a dual-projector display system for shadow removal. In Chapter 5, we present our occlusion detection and shadow removal results, then describe implemented and other potential improvements to the detection algorithm. Finally, we conclude in Chapter 6 with a summary of our research and comment on possible future directions of our work.

CHAPTER 2

Literature Review

The problem of occlusion detection in front projection environments has been addressed in the context of various applications, such as shadow detection and removal, occluder light suppression, as well as hand detection and tracking for gesture recognition. In this chapter, we review previous occlusion detection techniques and provide motivation for adopting a camera-projector calibration-based approach.

Depending on the target application, the proposed detection methods segment the camera image in order to extract either the occluding object itself or its corresponding shadow artifact. We divide current techniques into two groups, namely *direct* and *indirect* occlusion detection, and discuss each approach separately. The former approach locates the occluding object directly in the scene, while the latter detects an occlusion indirectly by locating its more easily discernible shadow. Indirect methods are reviewed first.

2.1. Indirect Occlusion Detection Techniques and Applications

Locating occluding objects directly in front projection environments is a non-trivial task, largely due to the varying scene background and illumination induced by dynamic projected content. In such conditions, traditional detection techniques used

2.1 INDIRECT OCCLUSION DETECTION TECHNIQUES AND APPLICATIONS

in computer vision tasks are often unsuitable. For instance, appearance-based detection methods have generally been avoided, one reason being that illumination from front projection influences the apparent color of the occluding object, thus complicating the process of detection using color recognition [45][39][40][25]. The boundaries of an object might also appear distorted or be difficult to distinguish as it moves through projected imagery and textures, thus precluding the use of shape recognition algorithms [25]. These challenges have motivated some researchers to adopt indirect occlusion detection methods instead, since shadows are often easier to detect than the occluding object itself. Given auxiliary knowledge of scene geometry (e.g. the relative positions of the camera, projector and display surface), it is possible to infer the location of the occluding object based on the detected shadow position. Indirect occlusion detection is also the obvious approach for applications where shadows, not the occluder, are the regions of interest in the scene. We describe related techniques and applications in the following subsections.

2.1.1. Shadow Detection and Removal. As discussed in Section 1.2, front-projected displays suffer from shadows, a problem that researchers are addressing with virtual rear projection, where multiple projectors redundantly illuminate the display. Indirect occlusion detection, i.e. shadow detection, is used for *active* virtual rear projection (AVRP) to track and remove shadows dynamically while the display is in use.¹ This approach is more straightforward and accurate than detecting shadows by tracking occluders in 3D world space and then determining their position relative to the projector and display surface [3].

¹ *Active* Virtual Rear Projection (AVRP) contrasts with *Passive* Virtual Rear Projection (PVRP) in that AVRP involves the use of a monitoring camera and the adaptive control of projector output in detected shadow regions to attain the desired display intensity. In PVRP, no camera is used and overlapping projectors each illuminate the full display at all times; occluding only one projector results in “half-shadows” where output is still visible albeit at a lower contrast level. [37]

2.1 INDIRECT OCCLUSION DETECTION TECHNIQUES AND APPLICATIONS

Proposed shadow detection techniques [16][35][3][9][17] rely on the concept that a priori knowledge of projected content enables the prediction of the display appearance at any given instant, as seen by a monitoring camera. Assuming an unobstructed camera view of the projected display, shadows are detected by comparing predicted and captured images to locate display regions where significant radiometric inconsistencies occur. An image warping homography transform, recovered during an offline camera-projector geometric calibration process, is then used to map shadow regions appearing in the camera image to corresponding pixels in the projector framebuffers. The system compensates for shadows by filling them in with a second redundant projector. Sukthankar et al. [35] and Jaynes et al. [16] detect occluded regions by performing a pixel-wise comparison between predicted and captured camera images. In later work, Jaynes et al. [17] propose a new approximate region-based shadow removal approach that allows for more efficient image processing computations and network transmission of occlusion information to projectors in a distributed display system.

The shadow removal systems introduced by Sukthankar et al. and Jaynes et al. differ in that the former cannot support dynamic displays. Any image that is to be projected when the display is in use must be made available during system initialization for pre-generation of predicted camera images. This involves projecting each source image and saving a reference camera image of the display as it appears when unoccluded. The shadow removal systems described in references [3][9] are based on the same framework. This approach is of limited use in applications where the display content changes in unpredictable ways, as is the case for interactive displays or projected video in telepresence applications. Alternatively, Jaynes et al. perform an additional step during offline calibration, namely camera-projector color calibration to estimate the nonlinear color transfer function between the two devices. Each

2.1 INDIRECT OCCLUSION DETECTION TECHNIQUES AND APPLICATIONS

color channel is calibrated independently by fitting the camera response to a sigmoid function model. Geometric and color calibration data are then used to warp and color-correct the current projector framebuffer image at run-time, in order to dynamically synthesize a predicted camera image of the current projected background. This approach allows for shadows to be detected for arbitrary display images.

2.1.2. Occluder Light Suppression. Indirect occlusion detection has also been employed for the task of occluder light suppression. Cham et al. [3] use shadow detection to infer the location of the user who is occluding the display. Predicted and captured camera images of the display are compared to identify shadow regions, which are then mapped to projector space. However, besides removing shadows by selectively intensifying the output of a redundant projector, corresponding pixels in the occluded projector are also turned off to simultaneously prevent the projection of distracting light onto the user.

For applications in which only suppression of projected light on the user is required, Tan and Pausch [40] present an alternative technique using a camera fitted with an infrared (IR) filter. In their setup, both camera and projector are mounted orthogonal to the display surface and are assumed to share the same focal point. An off-axis IR LED array then reflects IR light off the display surface and onto the user, allowing for the occluder to be detected by locating the resulting shadow in IR camera images. This approach, however, cannot be used to eliminate visible shadows on the display.

2.1.3. Identifying the Occluded Projector in AVRP Displays. A further subproblem of occlusion detection pertains specifically to AVRP systems: when dealing with multiply overlapping projector displays, it is often necessary to identify

which projector is being occluded. Having such knowledge allows for better adaptive control of projection for shadow removal and occluder light suppression.

The dual-projector shadow removal system of Sukthankar et al. [35] does not perform this task. To compensate for shadows caused by a user occluding the display, all projectors are instructed to increase their output intensity simultaneously in the appropriate display regions. This invariably results in shadows being filled in by the unoccluded projector, but it also means that additional light will be projected unnecessarily onto the user by the occluded one [3]. Cham et al. [3] solve this problem by probing each projector in the event of occlusion. The authors serially instruct each projector to vary its output intensity by a small amount; if no subsequent change is detected in the camera image, then the current projector is determined to be occluded. This permits the simultaneous removal of shadow and suppression of occluder light, as explained in Section 2.1.2.

The disadvantage of cyclical probing is that additional rendering iterations are required to identify the occluded projector. One way to eliminate the need for probing is to configure a multiply overlapping display such that any display pixel is illuminated by exactly one projector at any given instant. Then, in the event of occlusion, it is immediately known which projector is occluded. This approach was proposed by Flagg et al. [9] and demonstrated in a dual-projector binary switching display system for fast shadow elimination and occluder light suppression.

2.1.4. Motivation for the Adopted Approach to AVRP. For our research on shadow detection and removal in the Shared Reality Environment, we adopted an occlusion detection approach based on camera-projector geometric and color calibration, similar to that of Jaynes et al. [16], as it inherently provides the required support for dynamic projected content. For the purpose of shadow detection and removal, we used an XOR display configuration similar to that proposed by

Flagg et al. [9], where each display pixel is always assigned to exactly one projector, to remove the need for projector probing and enable a faster system response to occlusions.

2.2. Direct Occlusion Detection Techniques and Applications

Despite the difficult scene conditions that may be induced by front projection environments, direct occlusion detection is nevertheless an important task in interactive applications. Various techniques have been proposed for the task of hand detection and tracking to support gestural interfacing with front-projected displays.

Some standard detection techniques used in computer vision, such as those relying on frame differencing² or background subtraction³, have shown a certain degree of success. For instance, Pinhanez et al. describe a projected touchscreen system that recognizes pointing gestures by using frame differencing to detect the hand moving through front projection [25]. Although fast and easy to implement, this approach to occlusion detection assumes a static background (i.e. projected image) and furthermore fails if the occluding object is stationary. Alternatively, Von Hardenberg and Brard use running image averaging⁴ to maintain an updated reference model for background subtraction, allowing for the detection of both moving and resting hands [45]. However, the adopted background updating method only accounts for small changes in background, e.g. slow variations in scene illumination. The problem remains that significant or sudden changes in projected content, such as those resulting from many interactive displays or from projected video, result in false detection of occlusion.

²*Frame differencing* detects moving object(s) in a static background by detecting changes between successive frames in an image sequence.

³*Background subtraction* detects object(s) of interest by comparing the current image with a reference background image.

⁴*Running image averaging* involves regularly computing a weighted average of the most recent frames to maintain an updated model of the scene background.

2.2 DIRECT OCCLUSION DETECTION TECHNIQUES AND APPLICATIONS

A simple way to deal with dynamic projection is to force user interaction to occur only in designated control areas outside the projected display region. While this solution may be adequate for some applications (e.g. FreeHandPresent [45], Tele-Graffiti [39]), it is likely to be overly restrictive for most interactive front projection environments. Clearly, imposing constraints on user movement is undesirable, particularly in immersive environments.

Sato et al. [32] also avoid the challenges posed by front projection through the use of a thermal infrared camera; hands are detected by sensing objects within body temperature range. Although robust to changes in background and illumination, this approach requires expensive specialized equipment. Alternatively, camera-projector synchronization can be used to detect occlusions by capturing images of the scene when projection is momentarily turned off. However, shuttering of projected light should be fast enough that it is imperceptible to the user, thus also mandating specialized and costly hardware.

We note that an occlusion detection technique based on camera-projector geometric and color calibration may be used not only for shadow detection, but also to facilitate direct occlusion detection. Unlike the previous direct methods, calibration enables the dynamic synthesis of predicted camera views through geometric and color correction of projector images, thus accounting for unpredictable changes in projected background and illumination. This allows for improved background segmentation without the need for specialized equipment such as IR cameras or camera-projector synchronization hardware. Indeed, while our implemented calibration-based detection system was intended for dynamic shadow detection and removal, early results suggested that it could well find application for other HCI tasks relevant to interaction in immersive environments, for example, hand detection for gesture recognition or person detection and tracking [13]. Calibration was performed to estimate the

2.2 DIRECT OCCLUSION DETECTION TECHNIQUES AND APPLICATIONS

geometric image warping homography between each camera-projector pair, as well as the nonlinear color mapping expressed as a per channel color lookup table. Similar independent research conducted by Licsar and Sziranyi [19] adopts a calibration-based approach to dynamic background generation and subtraction for the task of hand detection in front-projected augmented reality. The authors recover a polynomial mapping function during geometric calibration and build an intensity lookup table during color calibration.

CHAPTER 3

Camera-Projector Calibration Background

Camera-projector calibration is fundamental to our adopted occlusion detection algorithm, since geometric and color calibration are the enabling steps for dynamic camera view synthesis. We therefore provide background on the geometric and color calibration problems and briefly review available techniques.

3.1. Geometric Calibration Background

Geometric calibration involves registering the geometry of the camera-projector system by deriving the transforms between camera pixel coordinates, projector pixel coordinates and real world display coordinates. Calibration is required for various display rendering tasks, including projector image prewarping to accomplish automatic keystone correction [36][29], multi-projector display alignment on planar [28][4][1][18][38] or non-planar [41][44] surfaces, or shadow removal [16][35][3][9][17]. Calibrated camera-projector systems are also used for 3D scene reconstruction or depth extraction to recover image-based models of display surfaces [31]. For the research presented in this thesis, geometric calibration is performed to derive image

warps that are required for display configuration, camera view synthesis and shadow removal, as will be explained in the next chapter. In the following subsections, we describe camera-projector system geometry, then review common geometric calibration approaches. Projective geometry theory, on which is based the geometric model of a camera-projector system, is also provided in Appendix A.

3.1.1. Geometry of a Camera-Projector Pair. A camera is a perspective projection device, where points in the 3D physical world are projected through light rays onto the camera's 2D image plane. A projector is the dual of the camera [29], where 2D image points are projected in the opposite direction into the 3D physical world. Each device can be modeled with the common *pinhole* (or *perspective*) model [43], depicted in Figure 3.1. The model defines the *center of projection* \mathbf{O} to be the origin of the *device* (camera or projector) *reference frame*. The *focal length* f is the distance that separates \mathbf{O} from the *image plane* π . The *optical axis* (usually taken to coincide with the Z -axis) is the line through \mathbf{O} that is orthogonal to π and intersects it at the *principal point* \mathbf{o} . Thus, the 2D image point \mathbf{p} of a given 3D world point \mathbf{P} is located at the intersection of π with the ray through \mathbf{P} and \mathbf{O} . In a CCD camera and LCD projector, the CCD and LCD pixel array, respectively, are the physical instantiations of the image plane.

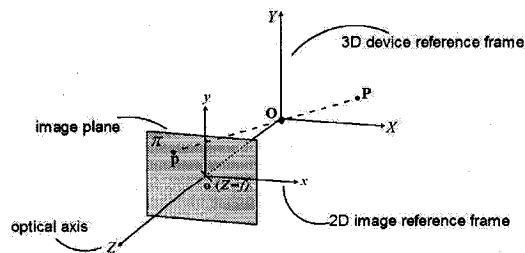


FIGURE 3.1. The (camera or projector) pinhole model. (based on Figure 2.8 of reference [43])

Figure 3.2 illustrates the geometry of a camera-projector pair, where the camera observes a display illuminated by the projector. Virtual 3D Euclidean coordinate frames are attached to the camera, projector and physical (world) display surface. Additionally, the camera and projector each have an associated 2D image coordinate frame (pixel units). This geometry is easily extended to include multiple projectors.

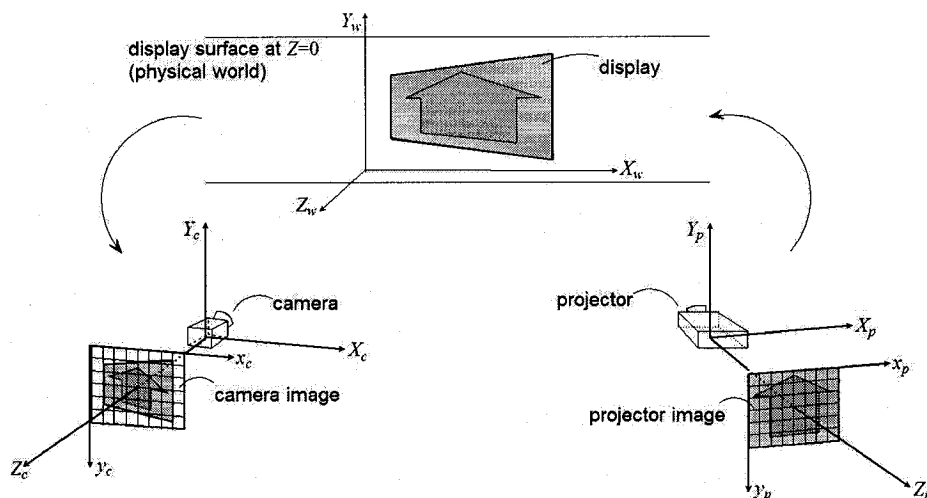


FIGURE 3.2. Geometry of a camera-projector pair.

3.1.2. Geometric Calibration Approaches. Various geometric calibration approaches are now reviewed. The required geometric transforms between the camera, projector and display frames of reference are each derived from an initial set of known point correspondences, which are obtained by detecting in a camera image feature points of a specially designed calibration pattern. For example, grid corners or the centroid of circular markers in a projected or physical calibration pattern can be detected to produce camera-projector or camera-world point correspondences, respectively. These are usually plugged into appropriate systems of equations, which are solved to estimate the geometric mapping functions.

3.1.2.1. *Explicit Calibration.* Explicit calibration involves estimating the camera or projector’s *external* and *internal parameters*, which relate the device’s image pixel coordinates to 3D world coordinates. The external parameters define the rotation and translation equations aligning the 3D world and 3D device reference frames. The intrinsic parameters relate 3D points in the device reference frame to corresponding image pixel coordinates. Intrinsic parameters include the device’s focal length (which describes the perspective projection of the optics), the x - and y -scaling factors (which define the effective size of a pixel), the pixel coordinates of the principle point (which relate the origins of the pixel and real-valued image reference frames) and the radial distortion coefficients (which is induced by the optical lens). Direct pixel correspondences between a calibrated camera and projector can then be obtained from epipolar geometry computations traditionally used in stereo computer vision.

Explicit calibration has been used to produce an accurate 3D geometric model of the camera-projector system [29]. However, explicit knowledge of the extrinsic and intrinsic parameters of each device is not strictly required. We discuss alternative geometric registration approaches in the following subsections.

3.1.2.2. *Structured Light.* Structured light techniques for geometric registration involve systematically projecting binary coded light patterns (e.g. coded vertical and horizontal bars of decreasing size) onto the display surface and locating corresponding features in captured camera images; this allows per-pixel matches to be established between the camera and projector images. This approach avoids the need for explicit calibration and 3D display surface models when projecting onto arbitrary non-planar surfaces [41]. Maintenance of fewer correspondences is possible by interpolating between known points, or by dividing the display surface into a mesh and estimating a piece-wise mapping function.

3.1.2.3. *Planar Homography Approximation.* The problem of camera-projector geometric calibration is straightforward when the display surface is planar. In this case, the geometric transforms between the camera, projector and display reference frames each can be modeled as a planar projective transformation. After estimating relevant homography matrices, mapping points between two frames is achieved easily through matrix multiplication $\mathbf{p}_2 = \mathbf{H}\mathbf{p}_1$. Radial distortion must be accounted for prior to homography estimation, although it often can be neglected without significantly affecting calibration accuracy. This is a simple and commonly-used approach to geometric calibration [16][35][4][1][30][28], which we also adopt to develop our prototype occlusion detection system.

3.2. Color Calibration Background

Color calibration involves measuring the camera color response to projected colors in order to derive the color mappings between a camera-projector pair or between two projectors. Most color calibration approaches have been proposed for color correction of projector output images, in order to minimize color variation across a large tiled multi-projector display [21][33][46]. Calibration has also been used to compensate for modulation by colored or textured display surfaces with spatially varying reflectance properties [23]. For the work on occlusion detection presented in this thesis, color calibration is performed to enable projector-to-camera color correction of synthesized camera view images, as will be explained in the next chapter. In the following sections, we describe the projector-to-camera color transfer process, then review current color calibration approaches. To provide an understanding of the concepts on which camera-projector color calibration is based, color theory is also explained in Appendix B.

3.2.1. Projector-to-Camera Color Transfer. LCD projectors and color cameras follow the tristimulus color theory. An LCD projector synthesizes colors by emitting weighted amounts of red, green and blue light from each position in the image pixel array. Color measurement in a camera parallels color perception in the human visual system, in that filters are used to split incoming light into three channels and record separately the intensity of red, green and blue light striking each pixel of the image sensor array.

Camera sensors are often designed to imitate not only the spectral sensitivity of cone cells, but also their nonlinearity by applying an artificial nonlinear gain to the linearly measured intensity of light. In addition, data projectors often apply a nonlinear relationship between input channel intensity values and resulting voltage signals to produce output light intensities that increase in a perceptually linear fashion. This process of introducing camera or projector nonlinearity is referred to as *gamma correction*.

The mapping between a desired RGB color in the source projector image, $C_p = (I_{pr}, I_{pg}, I_{pb})$ to the measured RGB color in the captured camera image, $C_c = (I_{cr}, I_{cg}, I_{cb})$, is now considered. Projector-to-camera color transfer is a complex nonlinear function that depends on several factors, including the spectral power distribution (SPD) of the projector light source, the filter and sensor characteristics of the camera and projector color channels, the hardware processing performed by the devices, as well as the radiometric properties of the display surface. The color transfer process, for a given projector channel and at a given point on the display surface, is illustrated in Figure 3.3 and explained below.

For each projector channel j , $j \in \{r, g, b\}$, the output primary color stimulus $E_{pj}(\lambda)$ corresponding to the input intensity I_{pj} is the result of several transformations: I_{pj} is first transformed by the often nonlinear response of the projector system, $\alpha_j(I_{pj})$,

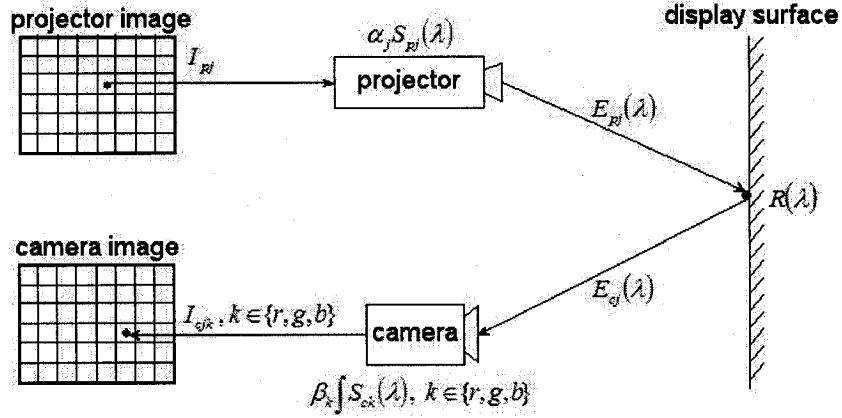


FIGURE 3.3. Projector-to-camera color transfer for projector channel j , $j \in \{r, g, b\}$.

which accounts for the hardware processing (e.g. gamma, white balancing, etc.) and electronics of both the video card and projector device. The resulting applied voltage or applied intensity value is then transformed by the projector channel's spectral response $S_{pj}(\lambda)$, which accounts for the spectral characteristics of the projector light source (light bulb) and the light filter for channel j . Thus,

$$E_{pj}(\lambda) = \alpha_j(I_{pj})S_{pj}d(\lambda) \quad (3.1)$$

$E_{pj}(\lambda)$ is then modulated by the spectral reflectance $R(\lambda)$ of the display surface to produce the input camera color stimulus $E_{cj}(\lambda)$, which represents incoming light from channel j of the projector:

$$E_{cj}(\lambda) = R(\lambda)E_{pj}(\lambda) \quad (3.2)$$

The incoming light $E_{cj}(\lambda)$ is sampled by filtering it through the red, green and blue camera channels. The corresponding camera intensities, $I_{ckj}, k \in \{r, g, b\}$, depend on the combined spectral characteristics of the camera filters and sensors, $S_{ck}(\lambda)$ and the nonlinear response of the camera imaging system β_k . The latter function accounts

for the electronics and hardware processing of both the camera and digitizer:

$$I_{ckj} = \beta_k \int S_{ck}(\lambda) \quad (3.3)$$

Physically, light entering the camera is the superposition of light from the three projector channels: $E_c(\lambda) = \sum E_{cj}(\lambda)$, $j \in \{r, g, b\}$. However, in the color transfer model, each primary projector stimulus is considered separately and their individual contributions are then added to obtain the final measured camera color $\mathbf{C} = (I_{cr}, I_{cg}, I_{cb})$. The black offset of the projector, which refers to leakage light that is emitted when projecting zero intensity from all channels, is also accounted for during summation.

Other factors exist that influence projector-to-camera color transfer. For instance, ambient illumination (e.g. due to other light sources or surface interreflections) also contributes to the measured camera color response. The reflectance properties of a display surface may also vary with position and direction of light. It is common to assume the use of a *Lambertian* surface, which is one that reflects light such that it appears equally bright from any viewing angle. Furthermore, depending on camera and projector position, orientation, and lens shape, measured luminance may vary within a single projector's field of view [21] due to distance attenuation of light. Finally, camera and projector controls, such as projector brightness, gamma correction, camera exposure and gain, and white balance, influence color transfer.

3.2.2. Color Calibration Approaches. Camera-projector color calibration is typically accomplished by projecting known color samples and measuring the resulting camera color response. These color correspondences are then used to derive the projector-to-camera color mapping based on a given transfer model.

Majumder et al. [21] present an algorithm that uses a digital camera to perform color calibration of a multiprojector display, in order to achieve photometric (luminance) uniformity all projectors. High dynamic range imaging [8], which involves taking several images of a scene at different exposures, allows the nonlinear luminance response of the camera and each projector to be recovered almost as accurately as if a spectroradiometer were used.

Alternatively, Nayar et al. [23] introduce a new calibration technique intended for correct display on colored or textured surfaces. Camera-projector calibration is carried out by first accurately recovering the nonlinear response of the camera using calibration charts or high dynamic range imaging [22][8]. Per-pixel projector-to-camera 3×3 color mixing matrices, as well as the nonlinear response of the projector, are then estimated from known color correspondences. The concatenation of these transformations yields the color transfer function between the two devices, which is used to color-compensate projector images a priori and thus account for the spatially varying reflectance properties of the underlying display surface.

To obtain perceptually acceptable results for the above applications, it is necessary to accurately recover the color mappings between each camera-projector pair. For the task of occlusion detection, however, the intended goal of calibration is to perform projector-to-camera color correction during camera view prediction. While the previous calibration techniques are in theory applicable, even a rough estimate of the color transfer function may be sufficient for successful occlusion detection [16]. Through experiments, Jaynes et al. [16][17] observed that the projector-to-camera color transfer function can be modeled by assuming channel independence and approximating it with a separate sigmoid equation for each channel (similarly in other work, Majumder found the transfer function to be roughly *S-shaped* [20]). The sigmoid models the black offset of the projector, the saturation point of the camera and the

intermediate portion of the camera response that increases in a roughly linear fashion. Nonlinear least squares optimization is used to derive the function parameters from a set of color correspondences. Calibration results are then used to synthesize color-corrected predicted camera images for shadow detection.

For the occlusion detection work presented in this thesis, we also assume that a rough estimate of the camera-projector color transfer function suffices for color calibration. Our experimental results, presented in Chapter 5, indicate that successful occlusion detection can be achieved despite this simplifying assumption.

CHAPTER 4

Camera-Projector System Design

In this chapter we present the system framework for the implemented camera-projector system and describe our adopted occlusion detection algorithm. We also describe the system's integration with a dually overlapped projector display to form a prototype AVRP system for shadow detection and removal.

4.1. System Framework

4.1.1. System Setup. The implemented camera-projector system for occlusion detection is comprised of the following physical components:

- *Projector Display System:* Graphics rendering system composed of a single or multiple projectors with overlapping fields of view. Each projector renders the same source image onto the target display surface.
- *Effective Display:* The illuminated display image on the display surface, resulting from the superposed and aligned contribution of each projector. The effective display is a quadrilateral subregion of all projectors' intersection area. A planar display surface is assumed. This assumption suffices for our research on the SRE, where the immersive display is rendered by six overlapping projectors installed to illuminate three walls of the room; occlusion

detection can be performed with a piecewise planar surface by considering quadrilateral subregions of the display on each plane separately.

- *Camera:* A single monitoring video camera whose field of view fully encloses the effective display region.
- *Processing System:* Computer system that drives the projector display system and camera, and runs the occlusion detection program. Application processing of occlusion data, for example shadow removal, is also performed.

Figure 4.1 illustrates the system setup for the case of a single projector display. In this example, the camera observes the display, which was rendered by texture mapping the source image onto a projector image-sized rectangle. Due to nonorthogonal placement of the camera and projector with respect to the display surface, the display and camera images may be affected by perspective foreshortening (i.e. keystone distortion).

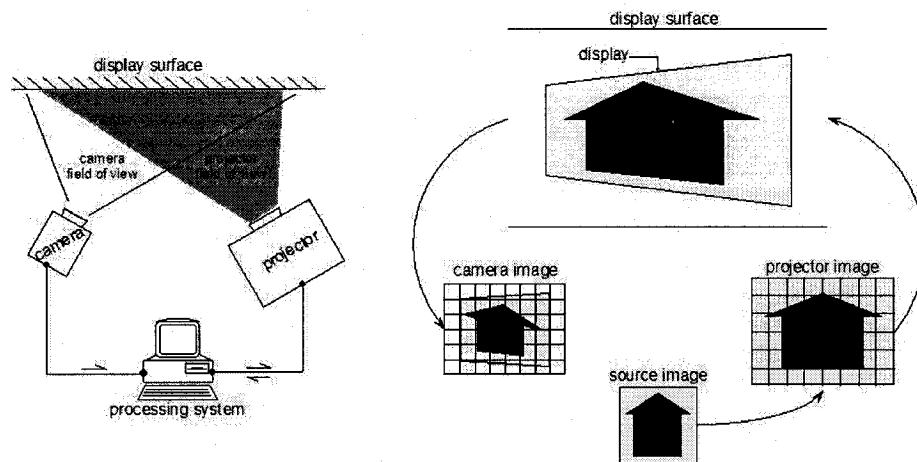


FIGURE 4.1. System framework for the single projector display case: example of the physical setup and snapshot images of an unoccluded display.

The system setup for a multiply overlapping display is depicted in Figure 4.2, for the dual projector case. As shown, each projector must prewarp the source image to produce a seamless display in which the projected content is geometrically aligned on

the display surface. Prewarping can also be used to correct for keystone distortion such that the display appears rectangular from the viewer's perspective; this applies to both single and multi-projector displays.

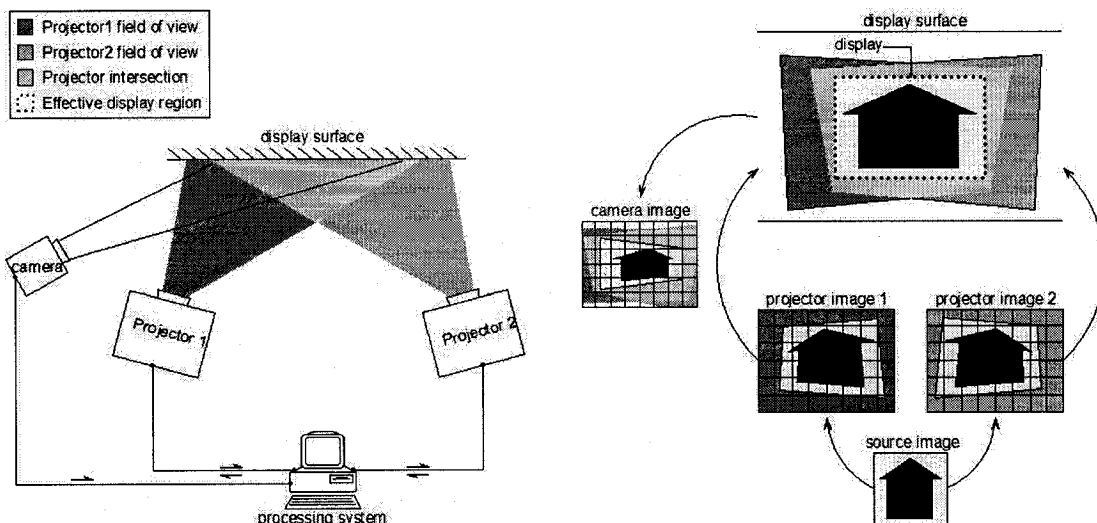


FIGURE 4.2. System framework for the case of a dually overlapping projector display: example of the physical setup and snapshot images of an unoccluded display.

4.1.2. Algorithm Overview.

4.1.2.1. Occlusion Detection Algorithm. The implemented occlusion detection algorithm, summarized in Figure 4.3 to the left of the dashed line, consists of offline camera-projector calibration, followed by online occlusion detection that occurs for each camera frame. Offline calibration is performed in two steps, namely geometric and color calibration, to compute the image warping homography transform and color transfer function, respectively, between each camera-projector pair. During on-line occlusion detection, camera view synthesis is performed to predict the appearance of the projected display as it would appear, unoccluded, from the perspective of the monitoring camera. Pixel-wise comparison is then performed between corresponding predicted and captured camera images to locate significant color inconsistencies,

which correspond to occluded display regions. Depending on camera-projector placement, these regions may represent shadow artifacts on the display or the occluding object itself.

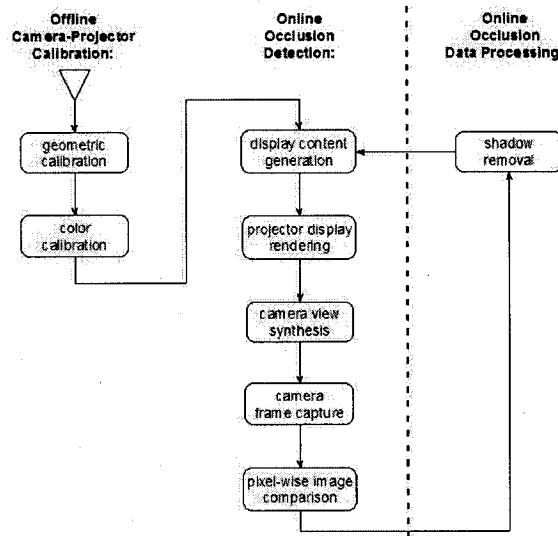


FIGURE 4.3. Overview of the occlusion detection algorithm (to the left of the dashed line), extended to support shadow removal.

4.1.2.2. Shadow Detection and Removal Algorithm. As discussed earlier, occlusion data can be used for various applications, including AVRP displays, as well as user detection and tracking to support HCI. As per our primary research goal in the SRE Lab, we focus on the former and demonstrate the implemented occlusion detection system as part of a prototype dually overlapping projector system for shadow detection and removal. The algorithm described previously was extended to include an occlusion data processing step, which involves adaptive control of projector output intensity to compensate for detected shadows. This extension is shown in Figure 4.3.

4.1.3. Software Architecture. Figure 4.4 illustrates the software architecture and data flow for the implemented camera-projector system. The *projector graphics engine* renders the display, by texture mapping the source image, then prewarping

the imagery and adjusting the display intensity as required for each projector. Source images originate from the *occlusion detection system* during offline camera-projector calibration, or from the *user application* during regular online display operation. The occlusion detection system receives input frames from the projectors and camera for use during offline calibration and online detection. The final output of this system is a camera image-sized, binary occlusion map that identifies occluded display pixels. This occlusion data can then be processed by the *shadow removal system*.

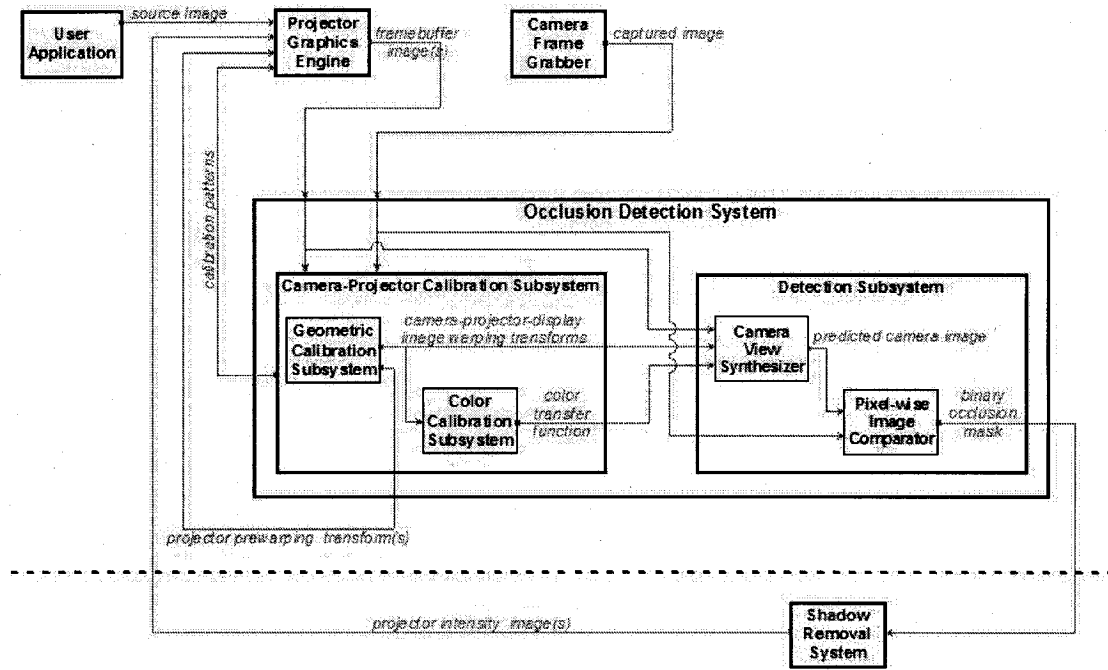


FIGURE 4.4. Software architecture for the camera-projector system for occlusion detection, extended to support shadow removal.

Our camera-projector software is written in C, except for the geometric calibration process, which is implemented using Matlab. The OpenGL graphics library is used for real-time projector display rendering. The occlusion detection and shadow removal programs run on a Linux-based machine.

The remainder of this chapter details each step of the occlusion detection process, as well as the shadow removal method. Offline camera-projector geometric calibration is discussed in Section 4.2, and offline color calibration in Section 4.3. Sections 4.4.1 and 4.4.2 respectively describe the camera view synthesis and pixel-wise image differencing steps that comprise the online occlusion detection process. Finally, Section 4.5 explains the implemented shadow removal method.

4.2. Offline Geometric Calibration

Offline geometric calibration is performed for two reasons. For display configuration during initial system setup, corrective projector prewarps must be computed and applied to align multiple overlapping projectors and optionally also eliminate keystone distortion. During the occlusion detection process, camera-projector image warps are required for camera view synthesis and, in the case of shadow removal, for mapping occlusion regions detected in camera space to corresponding projector pixels.

4.2.1. Geometric Calibration Model. For our occlusion detection system, we adopt the planar homography approach to camera-projector geometric calibration. The required pixel mappings are computed as 3×3 homography matrices between camera, projector and display image planes. The following assumptions are made regarding system geometry:

- Static display configuration: Geometric calibration is performed offline during system initialization and must be repeated after any change in camera or projector placement (position and orientation with respect to the display surface). Adjusting the manual keystone control of the projector or the zoom factor of any device also requires re-initialization.
- A planar display surface is used, as mentioned earlier.

- Radial lens distortion is negligible.

Figure 4.5 illustrates the adopted geometric calibration model for the case of a single projector display.

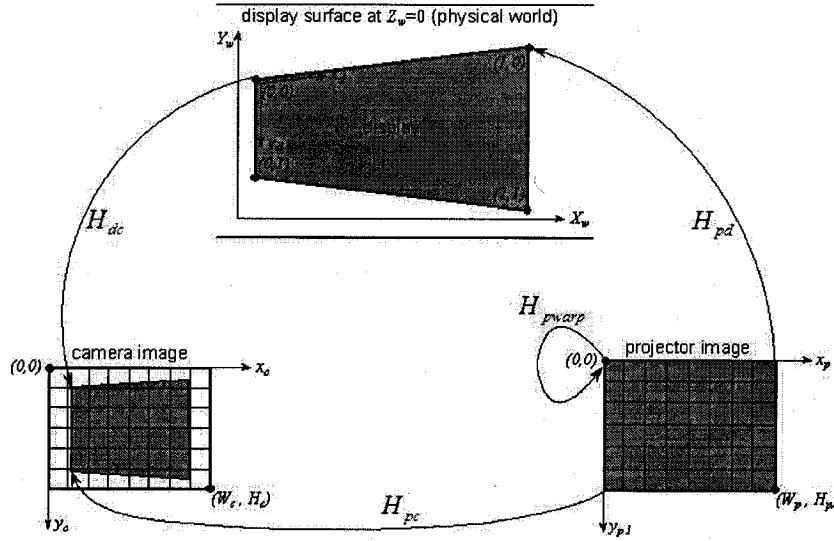


FIGURE 4.5. Camera-projector geometric calibration using planar homographies, for the single projector case.

As shown, we distinguish between *physical* and *conceptual* display frames of reference. The display surface is assumed to be aligned with the $Z_w = 0$ plane of the physical world; the directions of the world X - and Y - axes are user-defined. While the effective display region is a quadrilateral on the physical surface, we define the conceptual model of the display to be a unit square which, depending on the extent of keystone distortion, might be viewed in perspective. This distinction facilitates computations as this model may be used regardless of the physical shape of the projected display quadrilateral. For example, the amount of image processing can be reduced by considering only those camera pixels that fall within the effective display region, i.e. that map to points inside the display unit square. The metric dimensions and world coordinates of the display need not be known for occlusion detection; these

need only be recovered during initial system configuration if aligning the edges of the display with the world reference frame (e.g. for keystone correction). The latter is accomplished by prewarping the projected imagery in software.

The following homographies are derived (i identifies a given projector in a multi-projector system):

- $\mathbf{H}_{p_i c}$: projector-to-camera image warping homography, used during online occlusion detection to predict the camera view based on projector framebuffer contents.
- \mathbf{H}_{dc} : warping homography between the display unit square and the camera image. As mentioned, \mathbf{H}_{dc} is used to reduce the amount of image processing by considering only camera pixels inside the effective display region.
- \mathbf{H}_{pwarpi} : projector prewarping homography, used only during display configuration, typically to accomplish multiprojector alignment or keystone correction. \mathbf{H}_{pwarpi} is used to prewarp the projected imagery such that it occupies a quadrilateral subregion of the projector framebuffer. If no prewarping is to be performed, as in the single projector display example of Figure 4.5, \mathbf{H}_{pwarpi} equals the Identity matrix.
- \mathbf{H}_{dpi} : warping homography between the display unit square and each projector framebuffer image. The computation of \mathbf{H}_{dpi} is an intermediate step in the derivation of \mathbf{H}_{pwarpi} , as will be explained in Section 4.2.2.

The geometric calibration model supports multiple overlapping projectors, as depicted in Figure 4.6 for the dual projector case.

Projector prewarping is performed in real-time. For each projector, after texture mapping the source image onto a projector framebuffer-sized rectangle, prewarping occurs by applying \mathbf{H}_{pwarpi} as the OpenGL projection matrix.

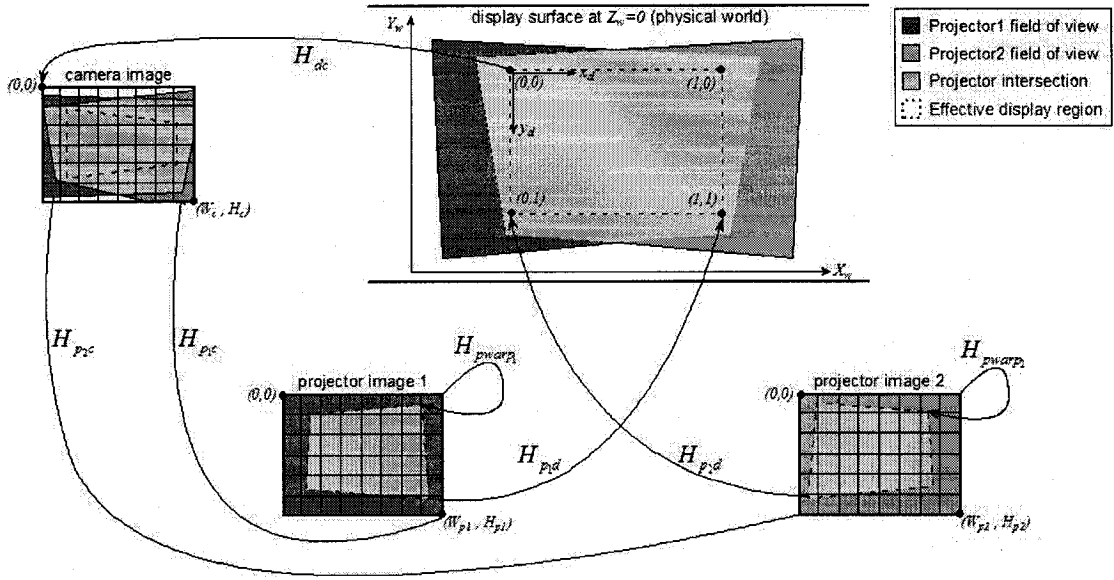


FIGURE 4.6. Camera-projector geometric calibration using planar homographies, for the dual-projector case.

4.2.2. Procedure for Homography Computation. In this subsection, we describe the steps used to compute the required homographies. We note that alternative calibration procedures are possible. Although the implemented camera-projector system for occlusion detection has integrated OpenGL graphics rendering functionality, it was configured, for the shadow removal application, to operate as a client interfacing with a dual-projector display server. In this case, the server performs, in addition to graphics rendering, geometric calibration using an alternative procedure [34].

The $\mathbf{H}_{p_i c}$ matrix for each projector i is computed first. Point correspondences between projector i and the camera are obtained by projecting a calibration grid of known size and detecting corresponding grid corners coordinates in the camera image, as depicted in Figure 4.7. These point correspondences are used to estimate $\mathbf{H}_{p_i c}$. While our occlusion detection software is written mostly in C, geometric calibration is performed in Matlab and our implementation uses routines from Bouguet's *Camera*

Calibration Toolbox for Matlab [2] to perform corner detection with subpixel accuracy and homography estimation.

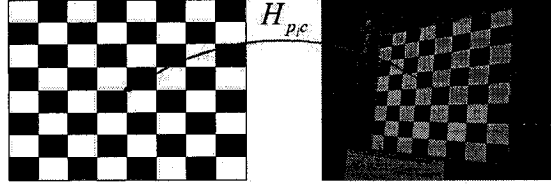


FIGURE 4.7. Projector-camera point correspondences for computing \mathbf{H}_{pc} .

The effective display region is then defined interactively by the user, by specifying in the camera image the desired corners of the display quadrilateral. The selected display region must be inside the fully overlapped projection area. The resulting camera coordinates are coupled with the corner coordinates in (conceptual) display space, namely $(0,0)$, $(1,0)$, $(1,1)$ and $(0,1)$, to generate four point correspondences used to compute \mathbf{H}_{dc} . Alternatively, with the help of computer vision techniques, display region definition may be performed in software to automatically rectify and optimize the effective display area. This research is being conducted by SRE Lab member Daniel Sud [34].

For each projector i , \mathbf{H}_{dp_i} is computed through matrix multiplication: $\mathbf{H}_{dp_i} = \mathbf{H}_{pc}^{-1} \mathbf{H}_{dc}$. The recovery of this homography is an intermediate step in the derivation of \mathbf{H}_{pwarpi} , where \mathbf{H}_{dp_i} is used to obtain the desired display corner coordinates in projector space. The resulting coordinates are coupled with the projector framebuffer corner coordinates, $(0,0)$, $(W_{p_i}, 0)$, (W_{p_i}, H_{p_i}) and $(0, H_{p_i})$, to produce four point correspondences that are used to determine \mathbf{H}_{pwarpi} .

4.3. Offline Color Calibration

Offline color calibration is performed to enable projector-to-camera color correction of the synthesized camera image when predicting the camera view of a projected

display. However, we only recover a rough estimate of the complex nonlinear color transfer function between each camera-projector pair. We assume that this simplification suffices for our occlusion detection tasks.

4.3.1. Color Calibration Models. We propose two alternative color transfer models, namely the per channel color lookup table and the linear least squares color transfer matrix. We discuss these in the following subsections.

The following assumptions are made:

- A planar white Lambertian surface is used.
- Intra-projector color variation is negligible.
- Ambient illumination is not explicitly modeled, but accounted for in the measured black offset of the projectors.
- The camera’s exposure and white balance functions are set to manual mode. The need for manual camera adjustment is discussed briefly in Section 4.3.1.3.
- Static display configuration: Color calibration is performed offline during system initialization and must be repeated after adjusting camera exposure or white balance settings, or projector color controls (e.g. brightness, contrast, gamma factor). Recalibration is also required after significant changes in camera-projector geometry or ambient illumination (e.g. room lights are switched on or off).

4.3.1.1. Per Channel Color Lookup Table. As discussed in Section 3.2.1, LCD projectors synthesize colors by combining weighted amounts of red, green and blue light, and cameras split incoming light into three color channels to record separate red, green and blue intensities for each pixel. Given this, in our early work on occlusion detection, we expressed the nonlinear color transfer function between each camera-projector pair as a per channel color lookup table (CLUT) that stores the measured

camera response to increasing intensities of projected primary colors. However, since projectors have a black offset and the spectral sensitivities of the camera sensors may overlap, projecting a primary color may generate a response from all three channels of the camera. For instance, in a color calibration test run, the camera measured an RGB color of $\mathbf{C} = (16, 181, 11)$ in response to the projection of green light $\mathbf{P} = (0, 204, 0)$. Thus, each entry in the lookup table for a given projector channel has three values that specify the measured intensity for each camera channel.

The CLUT for one camera-projector pair, given 8-bit color channels (or 256 intensity levels per channel), is therefore comprised of three 256×3 lookup tables, as shown in Figure 4.8. It is noted that all stored camera response values include the effect of the cumulative black offset of the projectors, $\mathbf{C}^0 = (C_r^0, C_g^0, C_b^0)$.

lookup table for the projector red channel				lookup table for the projector green channel				lookup table for the projector blue channel			
C_r, C_g, C_b				C_r, C_g, C_b				C_r, C_g, C_b			
P_0	C_r^0	C_g^0	C_b^0	P_0	C_r^0	C_g^0	C_b^0	P_0	C_r^0	C_g^0	C_b^0
P_1	C_r^1	C_g^1	C_b^1	P_1	C_r^1	C_g^1	C_b^1	P_1	C_r^1	C_g^1	C_b^1
P_2	C_r^2	C_g^2	C_b^2	P_2	C_r^2	C_g^2	C_b^2	P_2	C_r^2	C_g^2	C_b^2
	\vdots	\vdots	\vdots		\vdots	\vdots	\vdots		\vdots	\vdots	\vdots
P_{255}				P_{255}				P_{255}			

FIGURE 4.8. The per channel color lookup table.

The CLUT is built by iterating through the projection of primary colors of increasing intensity, measuring the RGB camera response for each and storing this data. Since this involves projecting 3×256 color samples, we speed up the process by projecting calibration color grids instead of separate images for each calibration color, as shown in Figure 4.9. The pre-computed homography \mathbf{H}_{pc} is used to map

a rectangular subregion of each color patch to camera space. The stored RGB camera response is the average color measured over corresponding patch pixels and over multiple camera images.

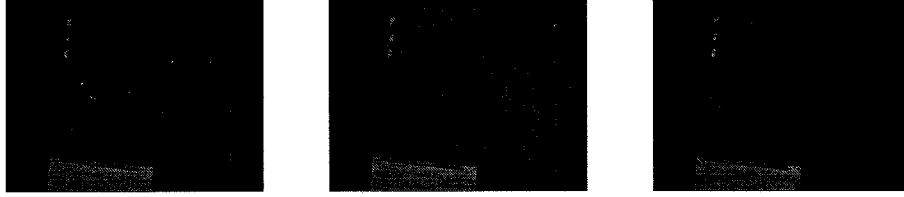


FIGURE 4.9. CLUT color calibration: sample camera images captured during separate calibration of the red, green and blue projector channels.

The camera RGB response, $\mathbf{C} = (C_r, C_g, C_b)$, to a projected RGB color $\mathbf{P} = (P_r, P_g, P_b)$ can be predicted by summing the camera responses to each of the projected primary components in isolation, and then accounting for the black offset. These computations are described in Table 4.3.1.1.

For a camera-projector pair:

Given:

- the projected RGB color, $\mathbf{P} = (P_r, P_g, P_b)$, and
- the cumulative black offset $\mathbf{C}^0 = (C_r^0, C_g^0, C_b^0)$
- the per channel color lookup table (CLUT)

The predicted RGB camera response to projector channel $j \in \{r, g, b\}$ only is obtained by using the intensity P_j as in index into the CLUT:

$$\mathbf{C}^j = (C_r^j, C_g^j, C_b^j)$$

The overall predicted camera RGB response is $\mathbf{C} = (C_r, C_g, C_b)$, where each channel intensity $C_i, i \in \{r, g, b\}$ equals:

$$C_i = \sum C_i^j - 2C_i^0, j \in \{r, g, b\}$$

TABLE 4.1. Camera color prediction using the CLUT.

In the above computations, the black offset is subtracted twice because its value is multiply included, i.e. in C^r , C^g and C^b . An alternative scheme for the CLUT is to subtract the black offset from the measured camera values before storage. When predicting the response to a projected color, the black offset must be added once, instead of being subtracted, from $\sum C_i^j$.

While the CLUT models the nonlinearity of the projector-to-camera color transfer function, the iterative approach to building it and the associated storage requirements may render this approach impractical, especially in a multiprojector system. A potential modification is to reduce the size of the CLUT by projecting fewer calibration colors and using interpolation to derive the camera response for intermediate projector intensities.

4.3.1.2. Linear Least Squares Color Transfer Matrix. In addition to the per channel color lookup table, we propose an alternative model for color calibration, namely the linear least squares color transfer matrix (LLSColormat). The LLSColormat is a linear approximation, for each camera-projector pair, of the projector-to-camera color transfer function. Linear estimation of color mapping functions have demonstrated some success for the application of color correction between multiple cameras [50]; the idea was proposed to investigate whether a similar approach could also be applied to camera-projector color calibration. As will be seen in Chapter 5, our experimental results indicate that the proposed LLSColormat model produces acceptable occlusion detection results.

The adopted linear model for the color transfer function, relating a projected RGB color, $\mathbf{P} = (P_r, P_g, P_b)$, to a predicted RGB camera response, $\mathbf{C} = (C_r, C_g, C_b)$, is expressed with Equation 4.1. The coefficients m_{11} , m_{21} and m_{31} represent the cumulative black offset of the projectors.

$$\begin{pmatrix} C_r \\ C_g \\ C_b \end{pmatrix} = \mathbf{M}_{\text{pc}} \begin{pmatrix} 1 \\ P_r \\ P_g \\ P_b \end{pmatrix} \quad \text{where } \mathbf{M}_{\text{pc}} = \begin{bmatrix} m_{11} & m_{12} & m_{13} & m_{14} \\ m_{21} & m_{22} & m_{23} & m_{24} \\ m_{31} & m_{32} & m_{33} & m_{34} \end{bmatrix} \quad (4.1)$$

The matrix \mathbf{M}_{pc} is referred to as the LLSColorMat and is derived by linear least squares optimization from a set of known projector-camera color correspondences. These are obtained by iteratively projecting a number of calibration colors and measuring the RGB camera response to each. As depicted in Figure 4.10, this process is performed using color calibration grids in a manner similar to that used for the CLUT, although the projected color samples need not be only primary colors. In our experiments, we project 216 RGB colors, which sample the RGB color cube at regular intervals. The set of projector-camera color correspondences is then used to formulate a linear least squares fitting problem. The general linear least squares fitting problem is summarized in Table 4.3.1.2.

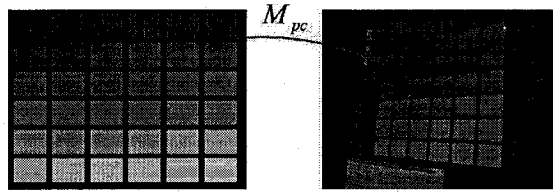


FIGURE 4.10. Projector-camera color correspondences for computing \mathbf{M}_{pc} .

To solve for \mathbf{M}_{pc} , the projector-camera color correspondence values are plugged into a system of linear equations of the form $\mathbf{Ax} = \mathbf{y}$. The 12×1 solution vector \mathbf{x} (i.e. $M = 12$) represents the twelve unknown coefficients of \mathbf{M}_{pc} that are to be computed. The design matrix \mathbf{A} is an $N \times 12$ matrix, where N is the total number of intensity data points. For example, 216 projected calibration RGB colors results

Given a linear system of equations, represented in matrix form as $\mathbf{Ax} = \mathbf{y}$, solve for x in the *least-squares sense*, i.e., solve $\min \|\mathbf{y} - \mathbf{Ax}\|$, where:

\mathbf{A} = an $N \times M$ matrix, where:

N = number of data points

M = number of *basis functions* in the linear model equation

e.g. In Equation 4.1, the model equation for each camera channel intensity $C_i, i \in \{r, g, b\}$ is a linear combination of four basis functions, namely 1, P_r , P_g and P_b .

Matrix \mathbf{A} , termed the *design matrix*, is a matrix of coefficients; it contains basis function values at each data point.

\mathbf{x} = an $M \times 1$ vector

Vector \mathbf{x} contains the parameters (i.e. the coefficients of the basis functions) to be fit to the linear model.

\mathbf{y} = an $N \times 1$ vector

Vector \mathbf{y} contains the *right-hand side* values, i.e. the value of the linear model equation at each data point.

TABLE 4.2. The linear least squares problem. (based on material from reference [27])

in 216 measured camera RGB colors, which correspond to $3 \times 216 = 648$ measured channel intensities; hence, $N = 648$ and \mathbf{y} is a vector containing these intensities.

Each row in the design matrix \mathbf{A} contains the values of the basis functions (i.e. the channel intensities for the projected calibration colors). This linear least squares problem is expressed with Equation 4.2:

$$\begin{bmatrix}
1 & P_{1r} & P_{1g} & P_{1b} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & P_{1r} & P_{1g} & P_{1b} & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & P_{1r} & P_{1g} & P_{1b} \\
1 & P_{2r} & P_{2g} & P_{2b} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & P_{2r} & P_{2g} & P_{2b} & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & P_{2r} & P_{2g} & P_{2b} \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
1 & P_{Nr} & P_{Ng} & P_{Nb} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & P_{Nr} & P_{Ng} & P_{Nb} & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & P_{Nr} & P_{Ng} & P_{Nb}
\end{bmatrix}
\begin{pmatrix} m_{11} \\ m_{12} \\ m_{13} \\ m_{14} \\ m_{21} \\ m_{22} \\ m_{23} \\ m_{24} \\ m_{31} \\ m_{32} \\ m_{33} \\ m_{34} \end{pmatrix} = \begin{pmatrix} C_{1r} \\ C_{1g} \\ C_{1b} \\ C_{2r} \\ C_{2g} \\ C_{2b} \\ \vdots \\ \vdots \\ C_{Nr} \\ C_{Ng} \\ C_{Nb} \end{pmatrix} \quad (4.2)$$

The unknown coefficients of $\mathbf{M}_{\mathbf{p}c}$ are estimated by solving this system of equations using linear least squares fitting by singular value decomposition. Our implementation makes use of linear least squares fitting routines from *Numerical Recipes in C* [27].

In a multiply overlapping projector system, the black offset is accounted for only in the LLSColormat of the first projector. That is, given n projectors, the matrix elements m_{11} , m_{21} and m_{31} are set only for $M_{p_{1c}}$ of the first projector, and reset to zero for $\mathbf{M}_{\mathbf{p}_x c}$, $2 < x < n$. The reason for maintaining a cumulative black offset instead of per projector black offsets has to do with automation of the calibration process. Each projector is calibrated separately and measuring per projector black offsets would require turning off the other projectors manually to avoid leakage light. An alternative scheme is to maintain the cumulative black offset as a separate tri-vector $C^0 = (C_r^0, C_g^0, C_b^0)^T$, and estimate $\mathbf{M}_{\mathbf{p}_k c}$, $2 < k < n$ as a 3×3 matrix, by subtracting the value of the black offset from the sample camera RGB responses prior to linear least squares estimation:

$$\begin{pmatrix} C_r \\ C_g \\ C_b \end{pmatrix} = \sum_{k=1}^n \mathbf{M}_{\mathbf{p}_k c} \begin{pmatrix} P_r \\ P_g \\ P_b \end{pmatrix} + \begin{pmatrix} C_r^0 \\ C_g^0 \\ C_b^0 \end{pmatrix} \quad \text{where } \mathbf{M}_{\mathbf{p}_k c} = \begin{bmatrix} m_{11} & m_{12} & m_{13} \\ m_{21} & m_{22} & m_{23} \\ m_{31} & m_{32} & m_{33} \end{bmatrix} \quad (4.3)$$

4.3.1.3. *Manual Camera Adjustment.* Many cameras are equipped with internal light metering systems used for adjusting exposure automatically to produce optimal image brightness for a given scene. Exposure can be controlled by varying light integration time, aperture size and electrical gain. While producing more visually pleasing images, these automatic exposure features reduce the reliability of our camera-projector color calibration process and hence make it difficult to predict the camera color response during camera view synthesis.

Under automatic exposure mode, the overall brightness of the captured camera image varies depending on the color content of the projected scene, as shown in Figure 4.11(a). Switching the camera to manual mode results in better stability in terms of brightness (see Figure 4.11(b)).

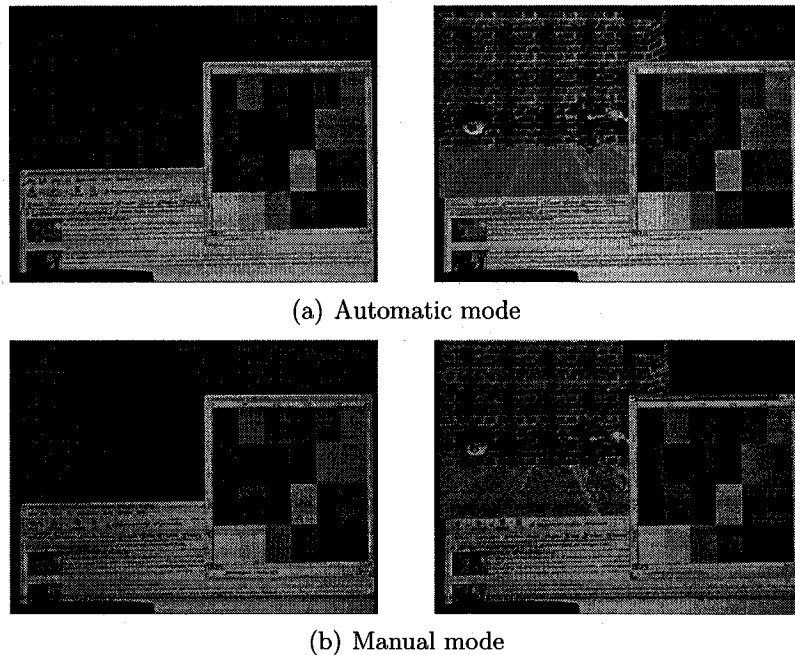


FIGURE 4.11. Auto vs. manual camera exposure (experiments with light integration time, aperture size and gain on the Sony MiniDV DCR-TRV900).

Thus, manual camera adjustment should be employed for more reliable camera-projector color calibration and camera view prediction, although intensity fluctuations in each color channel cannot be completely eliminated due to sensor noise. As well, the supported set of manual exposure controls differs with camera model, and many devices perform internal image processing operations that cannot be deactivated. In informal experiments with manual mode on the Sony MiniDV DCR-TRV900, the measured channel intensity variance at a single pixel was between 10 and 20 levels (or 4–8%), given 8-bit color channels (i.e. max 256 levels for each channel).

4.4. Online Occlusion Detection

Camera-projector geometric and color calibration data are used during online occlusion detection. For each camera frame, occlusions are detected by synthesizing a predicted camera image of the projected scene and comparing it with the actual captured camera image. We discuss camera view synthesis and image differencing in the following sections.

4.4.1. Camera View Synthesis. For a single projector display, the predicted camera image is synthesized by using the \mathbf{H}_{pc} homography to warp the projector framebuffer image to camera space. However, as discussed in Section 3.2.1, due to the many factors influencing the camera-projector system, a camera viewing a projected display is unlikely to produce an image whose colors match exactly those from the projector image. Color correction using the projector-to-camera color transfer function is therefore performed after image warping, to predict the camera view of the projected scene more accurately. Color correction of the synthesized camera image is performed by applying the color transfer function (either the CLUT or the LLSColorMat) on a per pixel basis. Additionally, to reduce the amount of image processing, the camera view is only predicted in image regions corresponding to the display. That is, the \mathbf{H}_{dc} homography is used to determine which camera pixels fall

within the effective display region; those that do not remain unprocessed in the final synthesized image.

In the case of multiply overlapping projectors, a predicted camera image is generated for each projector in isolation; the final synthesized camera image is the color superposition of all projectors' contribution, obtained by summing the predicted camera color response values at each display pixel and accounting for the cumulative black offset.

4.4.2. Image Differencing. Following camera view synthesis, pixel-wise color differencing is conducted between the predicted and captured camera images to detect occlusions. When comparing the two images for one camera frame, a given display pixel is considered to be occluded if the observed color, $\mathbf{C}_{\text{obs}} = (C_{\text{obs}_r}, C_{\text{obs}_g}, C_{\text{obs}_b})$ differs from the predicted color, $\mathbf{C}_{\text{pred}} = (C_{\text{pred}_r}, C_{\text{pred}_g}, C_{\text{pred}_b})$ by more than a specified threshold T . The difference between the two colors is measured as the total Euclidean distance between predicted and observed RGB intensities. The threshold is expressed in intensity units and is specified by the user during system initialization. During online occlusion detection, the following fixed thresholding scheme is used to evaluate occlusion at each pixel:

For a given display pixel in the camera image:

```

if (  $\sqrt{(C_{\text{pred}_r} - C_{\text{obs}_r})^2 + (C_{\text{pred}_g} - C_{\text{obs}_g})^2 + (C_{\text{pred}_b} - C_{\text{obs}_b})^2} > T$  )
    occluded = true;
else
    occluded = false;

```

TABLE 4.3. Pixel-wise image differencing technique.

Once again, the \mathbf{H}_{dc} homography is used to limit image processing to be performed only inside the effective display region. The final output of the occlusion

detection system is a camera image-sized binary occlusion map that identifies occluded display pixels in camera space.

Depending on camera-projector geometry and the position of the occluding object, color inconsistencies may represent a shadow on the display surface or the occluding object itself blocking the camera view. Generally, additional processing has to be performed to determine the nature of the occlusion. A potential solution is to analyze the overall color of each occluded region; shadows may be identified by regions whose average color is gray. Alternatively, this task sometimes can be solved easily by imposing constraints on camera-projector system geometry. For instance, if the camera is placed at an extreme angle to the display surface, it may be assumed to have a clear light of sight of the display at all times; detected occlusions can then be identified automatically as being shadows on the display. Of course, the feasibility of such constraints would depend on the user application.

4.5. Shadow Removal Application

For the shadow removal application, a prototype AVRP display was constructed by integrating the implemented occlusion detection system with a dually overlapped projector display server developed by Daniel Sud [34]. In this configuration, graphics rendering, geometric calibration and shadow removal is performed by the server, as mentioned earlier.

The overlapping dual-projector system was configured such that each display pixel is illuminated by exactly one projector at any given time, thus providing a straightforward solution to the occlusion detection subproblem of identifying which projector is being occluded. For this prototype application, the camera is assumed to maintain an unobstructed view of the display, as described in the previous section. However, this constraint will no doubt have to be relaxed for practical use in interactive environments such as the SRE.

During display operation, shadows are identified by the occlusion detection system. The output binary occlusion map is then provided to the projector display server, which uses it in an *exclusive-OR (XOR) shadow removal process* to transfer the responsibility of illuminating shadowed display pixels to the unoccluded projector [34]. In short, this technique involves warping the binary occlusion map to each projector's frame of reference and applying per-pixel XOR logic with the given projector's *intensity mask*. The projector intensity mask specifies whether each projector pixel is currently on or off.

CHAPTER 5

Results and Improvements

In this chapter, we present the results of occlusion detection and shadow removal experiments, then describe improvements made to the detection algorithm. These include a simple variable thresholding technique for improving shadow detection in darker display regions, and the addition of a final smoothing step after image differencing to reduce noise in the output binary occlusion map. We also discuss performance issues related to the current implementation and other systems, highlighting various potential improvements.

5.1. Occlusion Detection

We first present our early experimental results for occlusion detection using the CLUT color calibration model that was described in Section 4.3.1.1. Figure 5.1 provides an example of occlusion detection for a simple front-projected scene, rendered by a single projector. The source image (Figure 5.1(a)) is first texture mapped to occupy the full area of the projector framebuffer. During online display operation, the pre-computed \mathbf{H}_{pc} homography is used to warp the projector framebuffer image for camera view synthesis. Color correction is then applied using the CLUT to obtain the final predicted camera image. This is compared to the captured camera image in

order to detect occlusions. We also note that the \mathbf{H}_{dc} homography is used to limit processing to the effective display region.

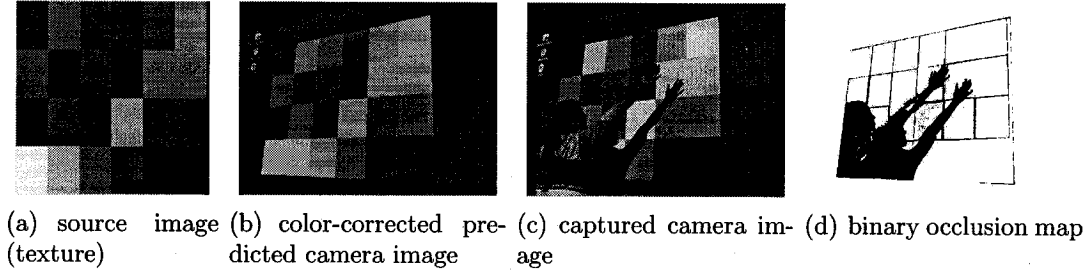


FIGURE 5.1. Occlusion detection using the CLUT model for color calibration.

This example shows that a calibration-based approach to detection may be used for direct or indirect occlusion detection in front projection display environments, i.e. to detect either the occluding object itself, or its shadow cast onto the display. Two common situations are illustrated in which occlusion data is useful. The first involves a shadow being cast onto the display surface because the user is blocking the projector. Occluded display regions are detected by the system, as shown in the output binary occlusion map of Figure 5.1(d), where occluded pixels are marked in black. However, display pixels identified as being occluded do not only represent shadows, but also regions in the camera image that correspond to the occluding object itself, i.e. in this case, the user's body. Occlusion data can be used for shadow removal using redundant illumination by a second projector, as will be demonstrated later, or for general object detection to enable detection and processing for human-computer interaction purposes. However, further processing must first be performed to distinguish between the two types of occlusion data.

We also present, in Figure 5.2, similar occlusion detection results obtained using the LLSColormat model for color calibration. We show that acceptable results are still obtained despite recovering only a *linear* estimate of the projector-to-camera

color transfer function. In this example, occluded display pixels are marked in green in Figure 5.2(d).

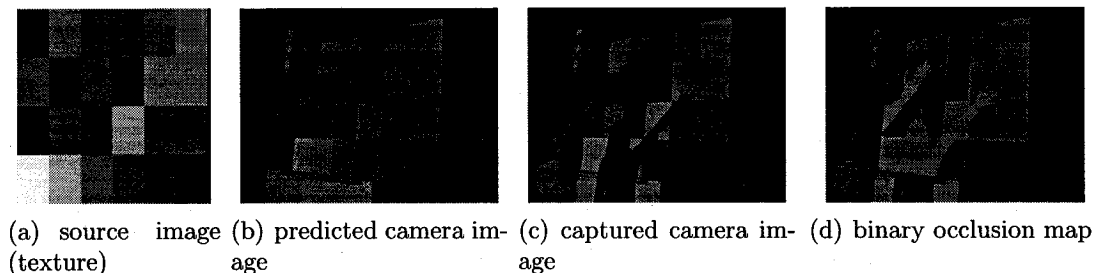


FIGURE 5.2. Occlusion detection using the LLSColorMat model for color calibration.

Figure 5.3 compares the CLUT and LLSColorMat color calibration approaches by illustrating occlusion detection results obtained for each, given the same source texture and captured camera input. As shown, both approaches allow for color correction of the predicted camera image such that its color content resembles more closely that of the captured image. However, since the CLUT models the nonlinearity of the projector-to-camera color transfer function which was observed to be roughly *S-shaped* [20][16], it may better approximate the camera response to projected intensities near the low or high end of the range. This can be seen in the predicted camera images of Figure 5.3, where the camera response to dark regions of the projected display (e.g. at the bottom-right or top-left corners of the color grid) was estimated better using the CLUT than with the LLSColorMat. Nevertheless, both color calibration models yielded similar occlusion detection results (we address the issue of noise in the binary occlusion map in Section 5.4). We also recall that both the CLUT and LLSColorMat are only rough estimates of the complex projector-to-camera color transfer function. Recovering a highly accurate nonlinear model may be unfeasible in certain environments due to various factors such as unpredictable illumination changes. This is the case for our research on the SRE, where despite manually adjusting camera exposure

settings, we observed that the camera response to projected colors is still influenced by surface interreflections off the white walls of the room.

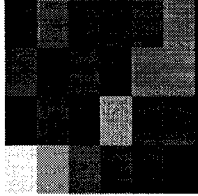
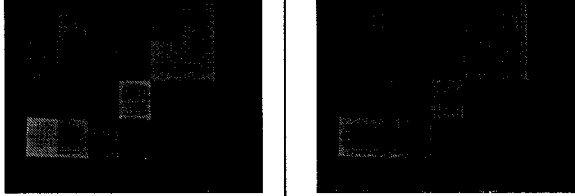

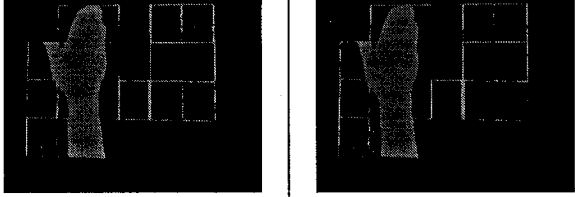
	Color Calibration Approach	
	<u>CLUT</u>	<u>LLSColorMat</u>
 source image (texture)	 predicted camera image	
 captured camera image	 binary occlusion map	

FIGURE 5.3. CLUT vs. LLSColorMat color calibration for occlusion detection.

5.2. Shadow Detection and Removal

For our research on the Shared Reality Environment, we demonstrate the performance of the implemented occlusion detection system when operating as part of the prototype dual-projector AVRP system described in Section 4.5. Experimental results are provided in Figure 5.4, which depicts the details of the XOR shadow removal process for one camera frame. As shown, the second projector compensates for the shadow that resulted from occluding the first one. We note that although the second projector is operating at full intensity in the corresponding region, its display

is dimmer than that of the first. During occlusion detection, these intensity differences are accounted for by per projector color calibration, allowing for the synthesis of a more accurate, color-corrected predicted camera image (see Figure 5.4(g)).

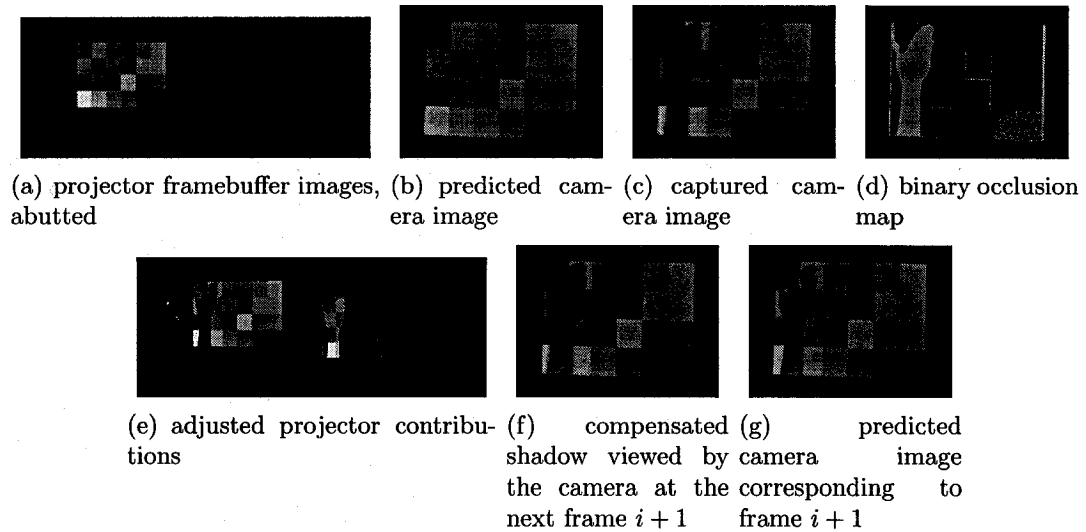


FIGURE 5.4. Shadow detection and removal process for camera frame i

Shadow detection and removal results over a sequence of frames are also illustrated in Figure 5.5, where the entire display is illuminated initially only by the first projector. Subsequent occlusions are detected and the second projector is instructed to fill in shadows selectively as they occur (Frames i to $i + 6$). In Frames $i + 6$ and onward, it is the second projector that is being occluded and shadowed display pixels are re-assigned to the first projector.

The above examples demonstrate that a calibration-based approach to occlusion detection provides support for dynamic projected content through camera view synthesis, as well for single or as multiply overlapping projector displays by superposing the contribution of each projector in the system. However, we note that despite having demonstrated the relative success of the implemented AVR system, visible shadow artifacts remain even after compensation, similar to the “half-shadows”

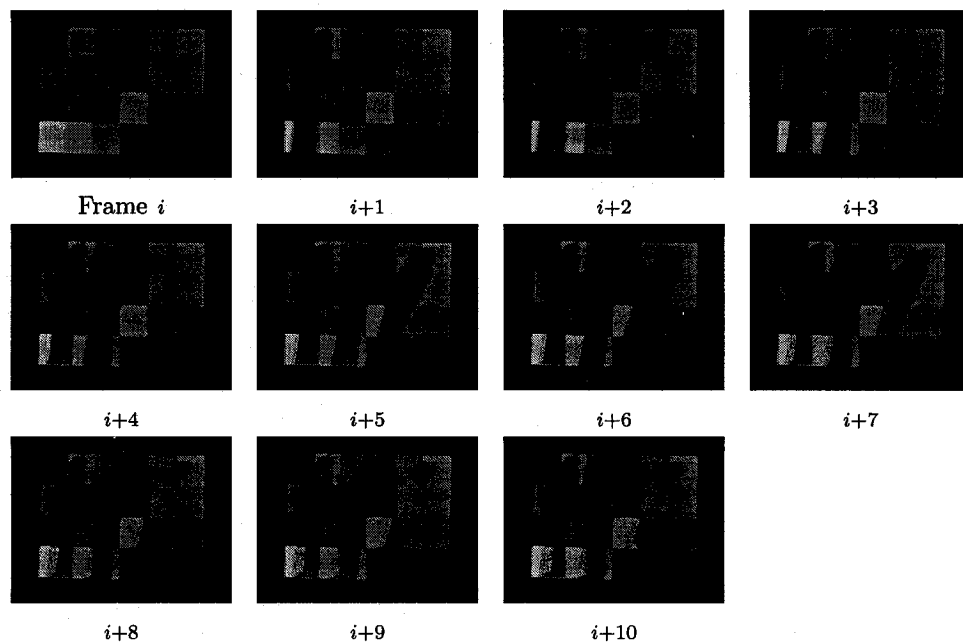


FIGURE 5.5. Shadow removal process for a sequence of captured camera frames.

produced by PVRP systems (as discussed in Section 2.1.1); these are due to the significant difference in overall intensity between the two test projectors. In the future, true color seamlessness between multiple projectors can be achieved by performing inter-projector color calibration and adjusting the intensity of each projector automatically. In Figure 5.6, however, we provide the results of shadow removal using two projectors that were manually color-calibrated to have more similar overall intensities. Ignoring noise, this example illustrates better the advantages of an active occlusion detection approach (i.e. AVR_P) over a passive one (i.e. PVR_P), in that the former allows for the possibility of fully eliminating shadows.

Finally, we also note that while the presented camera-projector system framework allows for multiple overlapping projectors, the current implementation is assumed to run on a single machine with dual-head graphics and is therefore limited to a maximum of two projectors.¹

¹This single-machine limitation is being addressed by ongoing research in our lab and has already been solved by other researchers [16].

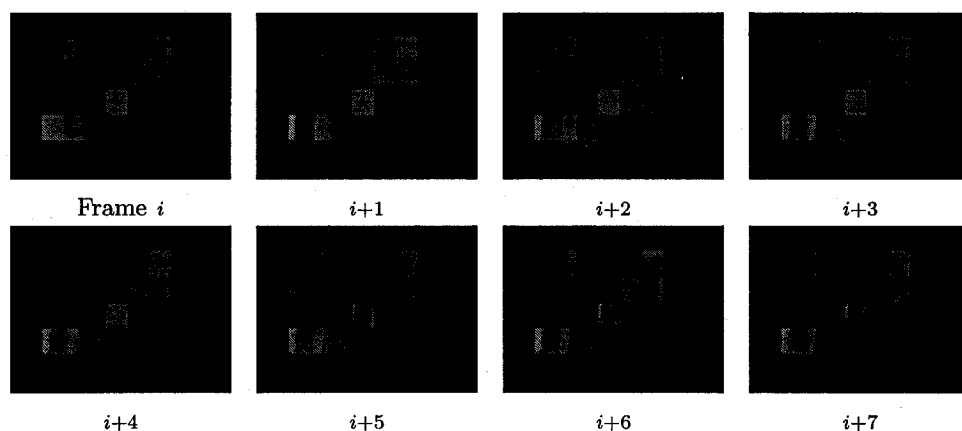


FIGURE 5.6. Active shadow removal using two projectors with more similar overall intensities.

5.3. Variable Thresholding for Image Differencing

We observed that detection of occlusion can fail in low-intensity display regions. With fixed thresholding, whether an occlusion will be detected depends in part on the color that is being projected in that location. During shadow detection, for example, a shadow cast over a projected white region on the display will result in greater radiometric inconsistencies than if it were cast over a projected dark gray region. Since shadow is detected only if the distance between the predicted and observed colors is greater than the fixed threshold, successful detection in the latter case requires a lower threshold value to increase system sensitivity to occlusion. We address this issue with a simple variable thresholding technique that accounts for the varying degree of measured color difference when occlusions occur.

Instead of using a fixed threshold intensity, its value is dynamically computed for each pixel to equal a specified percentage of the predicted color intensity, thus increasing the sensitivity of the detection algorithm in darker display regions. The percentage value used for thresholding is specified by the user during system initialization. Experimental results are provided in Figure 5.7, where shadows cast over

5.3 VARIABLE THRESHOLDING FOR IMAGE DIFFERENCING

dark display regions are more easily detected with variable rather than fixed thresholding. However, the system is also more sensitive to errors in the predicted camera image.

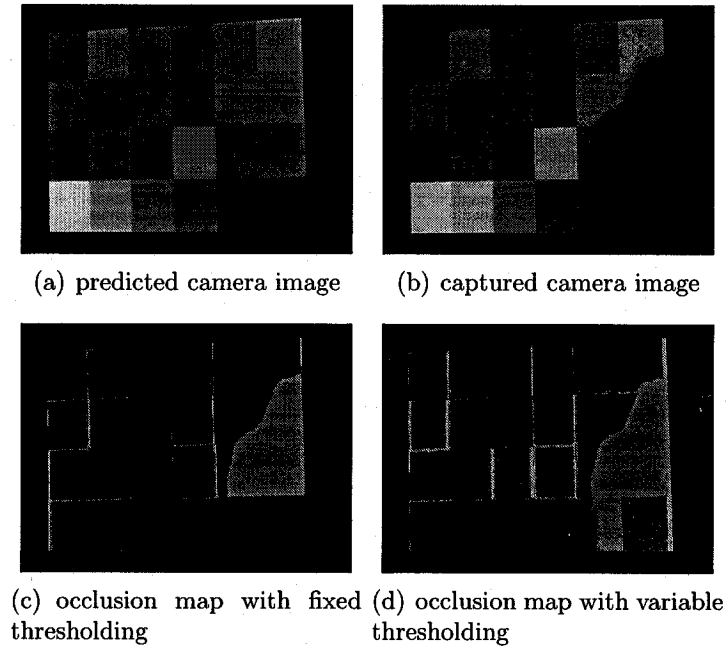


FIGURE 5.7. Variable thresholding.

For projected black or near black regions, however, two situations may occur. First, a low threshold increases sensitivity to errors in the predicted image, as well as to intensity fluctuations in the captured camera image due to sensor noise. More false positives then occur. For shadow detection, this may cause display illumination to repeatedly switch between two projectors, although this effect is not very noticeable in the case of projected black light. If the occlusion detection system is used for direct object (e.g. hand) detection, false positives can be handled by considering complementary data from parallel detection techniques, such as motion detection and tracking.

A second situation that may occur in projected black display regions is the failure to detect any occlusion at all. For instance, a projected black region on the display may be virtually indistinguishable from a shadow, although for shadow removal purposes, the amount of lost information is likely insignificant.

5.4. Morphological Image Smoothing for Noise Reduction

As shown in the previous examples, noise in the output binary occlusion map is an issue. False positives may occur because geometric and color calibration inaccuracies respectively result in image warping and color correction errors during camera view synthesis. In particular, the latter type of errors occur since we only recover a rough estimate of the projector-to-camera color transfer function. False positives may also be caused by sensor noise.

This problem, however, can be addressed by performing image smoothing of the binary occlusion map. We reduce noise by applying a simple *erosion-dilation* operation. Erosion and dilation are morphological operations that apply a *structuring element* to each pixel to determine whether its value should be set or reset. Erosion shrinks objects and is commonly used to eliminate detail from a binary image, while dilation expands objects and is used to bridge gaps [10].

A 3×3 structuring element is used for erosion of the binary occlusion map, where a pixel's value is reset if any pixel in its surrounding 8-neighborhood is not tagged as being occluded. To help eliminate gaps caused by erosion, dilation is then performed using a 3×3 structuring element that marks a pixel as being occluded if any of its neighbors is also occluded. Figure 5.8 shows the improved results of occlusion detection after image smoothing by erosion-dilation.

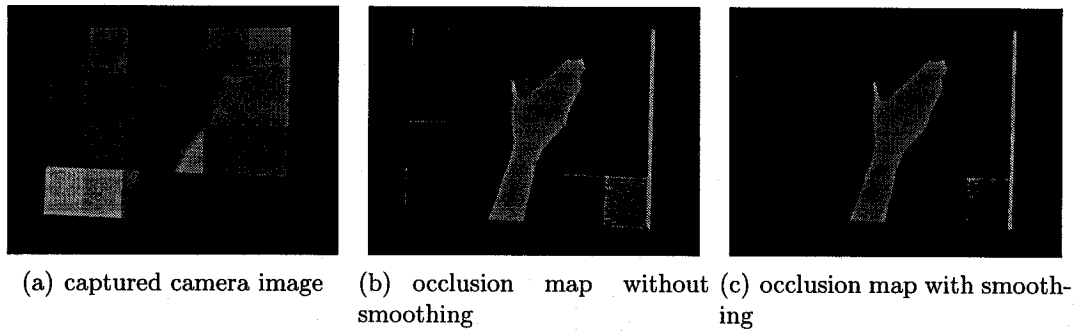


FIGURE 5.8. Smoothing the binary occlusion map through erosion-dilation.

5.5. Performance Issues and Potential Improvements

While the proposed calibration-based approach provides a flexible solution to occlusion detection, suboptimal performance of current implementations remains an issue. The main problem is that low frame rates render such techniques impractical for use in interactive display applications. For instance, the per-pixel shadow removal system introduced by Jaynes et al. runs at approximately 2 Hz [17]; the employed resolutions of the camera and projectors were unspecified. Similarly, our current per-pixel occlusion detection system runs at low frame rate. Simple detection for a single projector, of which 768×576 screen pixels were used as the effective display region, viewed by a 640×480 resolution camera, runs at approximately 4–5 Hz. Combined with the shadow removal algorithm, the frame rate drops to 1 Hz; this assumes the same camera resolution and the use of a $2 \times 1280 \times 1024$ dual projector system where the effective display is a quadrilateral subregion of the projector intersection area. It is also noted that the binary-switching projector system proposed by Flagg et al. performs both shadow removal and occluder light suppression at approximately 8 Hz [9], however it does not, as mentioned earlier, support dynamic projected content.

Evidently, current detection techniques must be optimized to enable detection and processing operations to reach a nominal frame rate of 10 Hz as required for

interactive display applications. We observed from code profiling results that performance hits in our implementation occur primarily during the camera view prediction process. Performance is first limited by the characteristically slow AGP bus read speed of the graphics card, as this is used to obtain the required projector framebuffer images from which the predicted camera image is synthesized. System response is also slowed because projector-to-camera image warping and color correction are currently performed in software. Our code profiling results account for the significant performance difference between our implementation, which supports dynamic projected content, and that of Flagg et al., in which all predicted camera images are generated offline, as discussed earlier in Section 2.1.1.

One obvious improvement is to perform region-based rather than per-pixel occlusion detection. For instance, Jaynes et al. achieved shadow removal at approximately 9 Hz by computing approximate shadow-bounding rectangular regions using low resolution images. This reduced overhead due to image processing and network communication in a distributed multi-projector display system [17]. Furthermore, since the effective display region may not occupy the full-resolution projector screen area, potential speed improvements during camera view prediction can be obtained by reading only the relevant sub-image of each projector framebuffer.

It is also noted that per-pixel detection can still be performed while reducing network communication overhead with the results transmitted in compact form, by run-length encoding the occlusion image [16] or by simply denoting pairs of (*start*, *stop*) columns on each scan line where an occlusion region is detected.

Another interesting optimization is to employ graphics hardware to accelerate certain image processing tasks. We consider using the hardware-accelerated OpenGL rendering capabilities of commodity graphics cards for implementing parts of our occlusion detection algorithm. For example, camera view synthesis can be performed through projector-to-camera image warping and color correction using the OpenGL

projection matrix and color matrix, respectively. In the case of multiple overlapping projectors, the final predicted camera image can be synthesized by superposing each projector's contribution using the OpenGL accumulation buffer. This way, a single (typically smaller) camera-sized image is read from the graphics card rather than multiple projector framebuffer-sized images.

Recently, other research areas have demonstrated growing interest in taking advantage of the programmability of commodity graphics hardware for real-time implementation of various image processing tasks, in particular, by using programmable vertex and pixel shaders. The latter can be used to apply per-pixel color correction in real-time. Moreover, such shaders can be used to perform image segmentation and morphological image operations, such as erosion and dilation in real-time [49]. Similarly, real-time image differencing and noise smoothing operations can be carried out in hardware to compare predicted and observed camera images during online occlusion detection. These suggestions highlight the possibility of implementing online occlusion detection largely, if not completely, in hardware, using commodity graphics cards. Unfortunately, it was not feasible to implement these hardware optimizations in sufficient time. However, further performance improvements to the implemented system will likely be addressed in future work within our research lab.

CHAPTER 6

Conclusion

In this thesis, we have addressed the problem of occlusion detection in front projection display environments by using an approach based on camera-projector calibration. We have developed a camera-projector system that first performs offline geometric and color calibration. These are the enabling steps for dynamic camera view synthesis from a priori known projector images. Display occlusions are then detected by comparing predicted and captured camera images on a pixel-wise basis to locate regions where radiometric inconsistencies occur. This approach facilitates the occlusion detection process in dynamic front-projected scenes.

Our experimental results have indicated the success of the implemented camera-projector system in performing direct or indirect occlusion detection. As well, we have demonstrated the use of our occlusion detection technique for the application of shadow removal, through redundant illumination with a dually overlapped projector display system.

Given the suboptimal performance of current implementations, we have highlighted potential improvements, emphasizing the need to speed up the camera view synthesis step, as well as the the interesting possibility of accelerating image processing tasks through hardware programmability of commodity graphics cards. We have

also described our implementation of simple variable thresholding and morphological image smoothing techniques for improving the accuracy of the presented occlusion detection algorithm. Currently, we are implementing another technique for reducing edge noise during the shadow removal process; this involves performing intensity blending between overlapping projectors at detected shadow edges, after having compensated for occluded display regions using the described per-pixel XOR shadow removal method. Addressing these issues would increase the feasibility of using front projection technology in various mixed reality and interactive display applications. Also of immediate importance is the need to scale the algorithm to support multiple cameras and projectors, in order to realize a true shadow-free front-projection display environment. Other future directions for research include improving the color calibration model to treat nonlinearities and to enable, in addition to occlusion detection, intra- and inter-projector color correction for a seamless tiled multiprojector display.

APPENDIX A

Projective Geometry Theory

We describe projective geometry notation and basic concepts in the 2D planar case. The 3D case is a straightforward extension of these ideas. Presented theory is based on material from references [12] and [43]. For in depth theory, Hartley and Zisserman's authoritative textbook, *Multiple View Geometry in Computer Vision* [12], is recommended.

A.1. Homogeneous Notation and the Projective Space

A *homogeneous* vector represents an equivalence class of vectors, where two nonzero vectors are considered equivalent if they differ only by an overall scale factor. An example is the homogeneous 3-vector $w(x_1, x_2, x_3)^T$, where w is an arbitrary real scalar. The 2D projective space \mathbb{P}^2 , or the *projective plane*, is the set of all equivalence classes of vectors in 3D Euclidean space \mathbb{R}^3 , excluding the vector $(0, 0, 0)^T$. \mathbb{P}^2 can be modeled as a plane (e.g. the plane $z = 1$) cutting a set of 3D lines through the origin but not itself passing through the origin. Points and lines in \mathbb{P}^2 respectively correspond to lines and planes through the origin in \mathbb{R}^3 (see Figure A.1).

A line through the origin in \mathbb{R}^3 can be interpreted as having been formed by the set of nonzero position vectors $w(x, y, z)^T$, where w is a varying real scalar. Its

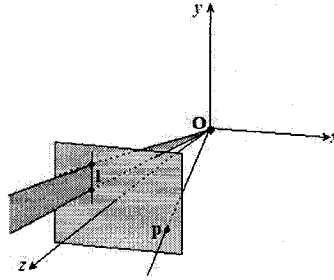


FIGURE A.1. Model of the projective plane \mathbb{P}^2 . (based on Figure 1.1 of reference [12])

corresponding point in \mathbb{P}^2 therefore can be represented by the homogeneous vector $\mathbf{p} = w(x, y, 1)^T = (xw, yw, w)^T = (x', y', w)^T$. Alternatively, the homogeneous notation for points can be explained by considering one that lies on a line l in the plane expressed by the equation $ax + by + c = 0$. A real point $(x, y)^T$ in the plane lies on line l if and only if $(x, y, 1)^T(a, b, c) = 0$. The position vector $(x, y)^T$ can thus be augmented to a 3-vector by adding a trailing coordinate of value 1, but it is clear that for any nonzero factor w , the homogeneous vector $\mathbf{p} = (xw, yw, w)^T$ represents the same point $(x, y)^T$. Furthermore, a line l in \mathbb{P}^2 , which is characterized by the a, b and c coefficients in the equation $ax + by + c = 0$, can also be expressed as a homogeneous vector $\mathbf{l} = (wa, wb, wc)^T$, since any nonzero w factor results in the same line. Finally, points and lines are duals in the projective plane, i.e. dual theorems may be obtained simply by interchanging the roles of points and lines in the statement. For example, in the symmetric incidence equation $\mathbf{p}^T \mathbf{l} = \mathbf{l}^T \mathbf{p} = 0$, one can consider \mathbf{p} to denote a point on the line l or a line through the point l .

The described homogeneous coordinate notation can be extended to general projective n -space \mathbb{P}^n : a point in Euclidean space \mathbb{R}^n is represented in \mathbb{P}^n by an augmented homogeneous $(n+1)$ -vector. In the 3D case, a point $(x, y, z)^T$ in \mathbb{R}^3 is denoted in \mathbb{P}^3 by the homogeneous 4-vector $(xw, yw, zw, w)^T = (x', y', z', w)^T$. The scale factor w can be any nonzero value but is typically chosen to equal $w = 1$ to avoid division when recovering coordinates in real space.

A.2. Projective Transformations

A *projective transformation* is a “linear transformation between two projective spaces” [43]. A common point called the *center of projection* is defined and mappings between points in the two spaces are obtained by projecting connecting rays through this center. This process is called *central projection* and is illustrated in Figure A.2 for the case of projective transformation between two planes. It is shown that projective transformations preserve straight lines but not necessarily parallelism. Parallel lines in one plane map to converging lines in the other plane. This explains the visual effect of *perspective foreshortening*, where distant objects (or lines) appear smaller (or shorter).

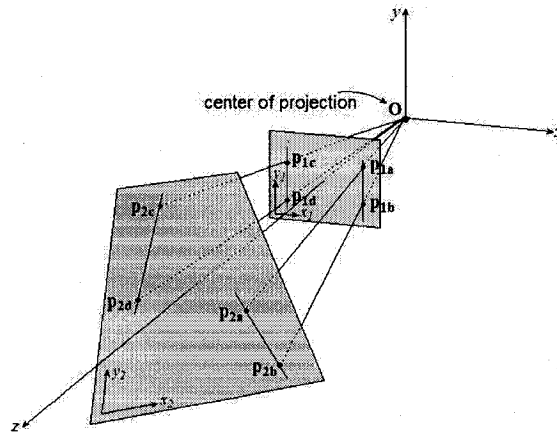


FIGURE A.2. Projective transformation via central projection. (based on Figure 1.3 of reference [12])

Projective transformations exist between projective spaces of the same or differing dimensions. As explained in Section 3.1.1, mappings between \mathbb{P}^3 and \mathbb{P}^2 are used to model the geometry between the 3D world scene and 2D image during camera image formation and projector image display.

Also of interest is a projective transformation of \mathbb{P}^2 into itself, called a planar *homography* or *collineation*. It is a linear invertible mapping between homogeneous

3-vectors that can be represented by a non-singular 3×3 matrix:

$$\begin{pmatrix} x'_2 \\ y'_2 \\ w_2 \end{pmatrix} = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix} \begin{pmatrix} x'_1 \\ y'_1 \\ w_1 \end{pmatrix} \quad (\text{A.1})$$

In short, the above equation is written as $\mathbf{p}_2 = \mathbf{H}_{12}\mathbf{p}_1$. The 9 parameters of a homography can be estimated, up to an unknown scale factor, from a minimum of four known point correspondences of which no three points are collinear. A linear system of equations is formed that can be solved using various linear algebra methods [12]. If more than four correspondences are available, the best fit is computed using least squares optimization.

Moreover, the composition of two homographies is also a homography, whose corresponding matrix is the multiplication of the former two matrices, e.g.:

$$\begin{array}{llll} \text{If:} & \mathbf{p}_2 = \mathbf{H}_{12}\mathbf{p}_1 & \text{and} & \mathbf{p}_3 = \mathbf{H}_{23}\mathbf{p}_2, \\ \text{then:} & \mathbf{p}_3 = \mathbf{H}_{13}\mathbf{p}_1, & \text{where} & \mathbf{H}_{13} = \mathbf{H}_{23}\mathbf{H}_{12}. \end{array}$$

APPENDIX B

Color Theory

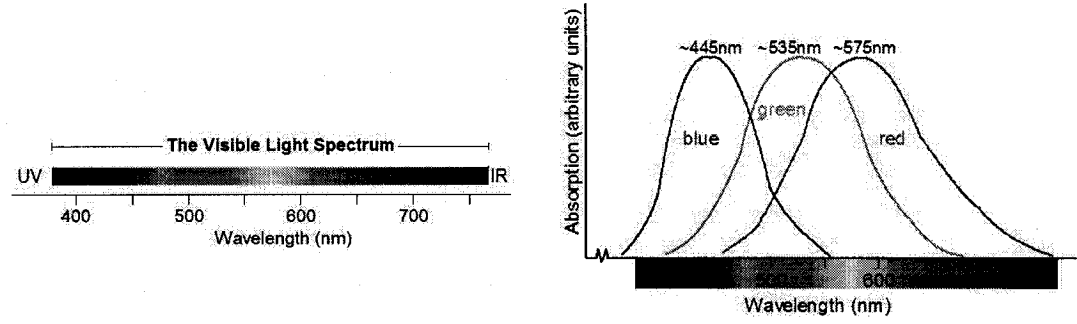
Presented color theory is based primarily on material from the textbooks of references [10], [26] and [14].

B.1. Color Perception and the Tristimulus Theory

Light is electromagnetic radiation composed of energy particles called *photons*, the contained amount of energy in each being proportional to the frequency (or inversely proportional to the wavelength) at which it travels. Color is a perceived sensation in response to excitation of the human visual system by *visible light*, i.e. light from the wavelength band of the electromagnetic spectrum between approximately $390nm$ to $780nm$ (see Figure B.1(a)). The visible light spectrum is partitioned into six broad color regions, namely violet, blue, green, yellow, orange and red, with each region blending into the next.

The human eye has two types of photo-receptors, namely rods and cones. Rods respond to the overall intensity (flux) of light striking them, while cones are sensitive to the wavelength of incident photons and are responsible for color vision. Three types of cone cells exist, each of which responds selectively to wavelengths around either the red, green or blue regions of the visible spectrum (see Figure B.1(b)). It

B.1 COLOR PERCEPTION AND THE TRISTIMULUS THEORY



(a) The visible light spectrum. (based on Figure 6.2 of reference [10]) (b) Spectral sensitivity curves of cone cells. (reproduced from Figure 6.3 of reference [10])

FIGURE B.1. Color perception.

is also noted that cone cells have a nonlinear perceptual response to the intensity of light which can be modeled by a logarithmic curve.

The multitude of colors that can be perceived, not only limited to those found in the visible spectrum, is due to varying spectral composition of light entering the eye. A color stimulus is the result of light from an optical source being reflected off or transmitted through a surface. However, different optical sources emit light with different wavelength mixtures, and different surfaces when illuminated absorb certain wavelengths while reflecting or transmitting others. A light source is represented by its *spectral power distribution (SPD)*, $E_s(\lambda)$, which describes how emitted light power is distributed across the wavelengths. The *spectral reflectance or transmittance* of a surface, $R(\lambda)$, describes the percentage of incident light that is reflected or transmitted as a function of wavelength. The SPD of the resulting color stimulus, $C(\lambda)$, is the product of the SPD of the light source and the reflectance (or transmittance) of the surface [48], as expressed by the following equation and illustrated in Figure B.2:

$$C(\lambda) = E_s(\lambda)R(\lambda).$$

Different color stimuli therefore excite the three types of cone cells to varying degrees based on their SPD. The excitation or intensity levels registered by the cones can be expressed as: $I_r = \int S_r(\lambda)C(\lambda)d\lambda$, $I_g = \int S_g(\lambda)C(\lambda)d\lambda$ and $I_b = \int S_b(\lambda)C(\lambda)d\lambda$,

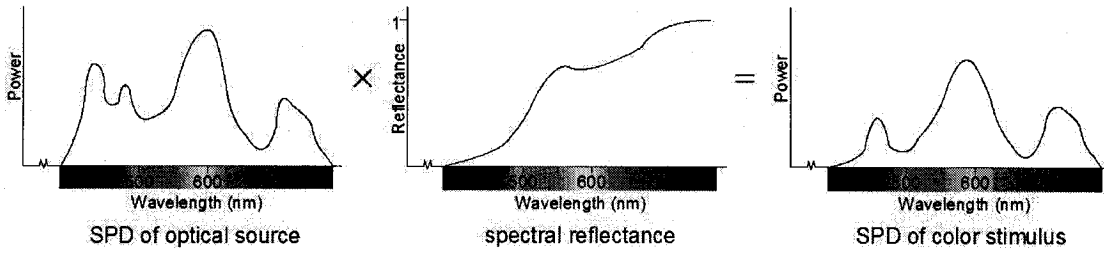


FIGURE B.2. Spectral power distribution (SPD) of a color stimulus. (reproduced from Figure 8 of reference [11])

where $S_r(\lambda)$, $S_g(\lambda)$ and $S_b(\lambda)$ are the spectral sensitivities of the cones. A particular combination of red, green and blue intensities then translates into a perceived color. Two stimuli that have different SPDs but result in the same perceived color are called *metamers*. Considering this process of color perception, Young's *tristimulus color theory* was proposed in 1801, suggesting that most colors can be reproduced by mixing appropriate proportions of three primary color stimuli, typically chosen to be red $C_r(\lambda)$, green $C_g(\lambda)$ and $C_b(\lambda)$ light. Thus, a color stimulus $C(\lambda)$ generally can be matched with the additive mixture: $C(\lambda) = w_r C_r(\lambda) + w_g C_g(\lambda) + w_b C_b(\lambda)$.

As well, if two color stimuli are combined, $C_1(\lambda) = w_{r1} C_r(\lambda) + w_{g1} C_g(\lambda) + w_{b1} C_b(\lambda)$ and $C_2(\lambda) = w_{r2} C_r(\lambda) + w_{g2} C_g(\lambda) + w_{b2} C_b(\lambda)$, the resulting color stimulus will be $C_3(\lambda) = (w_{r1} + w_{r2}) C_r(\lambda) + (w_{g1} + w_{g2}) C_g(\lambda) + (w_{b1} + w_{b2}) C_b(\lambda)$.

B.2. Color Measurement and Representation

Physically, a color stimulus is characterized by its SPD. Expensive precision-instruments such as spectroradiometers or spectrophotometers are used to measure emitted, reflected or transmitted radiant light power as function of wavelength. Numerically, a color C can be represented by its tristimulus values, i.e. the relative amounts or intensities of the three primary color components required to match the desired color C . Thus, given red, green and blue primaries, a color C can be expressed as $C = (I_r, I_g, I_b)$.

B.2 COLOR MEASUREMENT AND REPRESENTATION

Color can also be represented in terms of perceptual attributes, namely *luminance* and *chrominance*. Luminance (Y) is the measured amount of perceived energy in incident light and is associated with color brightness. Chrominance is composed of hue (H) and saturation (S) of a color. Hue indicates the dominant wavelength (or the dominant perceived color, e.g. a red or orange hue) in the incident mixture of photons (or colors). Saturation is the amount of white light (light composed of an equal distribution of all wavelengths) mixed in with the hue; it indicates color purity (e.g. pink is less saturated or pure than red). Luminance and chrominance of a color can be computed from its tristimulus values.

REFERENCES

- [1] A. Andrew, G. Gill, A. Majumder, H. Towles, and H. Fuchs, *Pixelflex2: A comprehensive, automatic, casually-aligned multi-projector display*, ICCV Workshop on Projector-Camera Systems (PROCAMS-2003) (2003).
- [2] J.-Y. Bouguet, *Camera calibration toolbox for matlab*, http://www.vision.caltech.edu/bouguetj/calib_doc/index.html (2004).
- [3] T.-J. Cham, J. Rehg, R. Sukthankar, and G. Sukthankar, *Shadow elimination and occluder light suppression for multi-projector displays*, Computer Vision and Pattern Recognition (2003).
- [4] H. Chen, R. Sukthankar, G. Wallace, and K. Li, *Scalable alignment of large-format multi-projector displays using camera homography trees*, IEEE Visualization (2002).
- [5] J.R. Cooperstock, *Interacting in shared reality*, HCI International, Conference on Human-Computer Interaction (2005 (to appear)).
- [6] C. Cruz-Neira, D.J. Sandin, and T.A. DeFanti, *Surround-screen projection-based virtual reality: the design and implementation of the cave*, SIGGRAPH '93, ACM Press (1993), 135–142.

- [7] M. Czernuszenko, D. Pape, D. Sandin, T. DeFanti, G.L. Dawe, and M.D. Brown, *The immersadesk and infinity wall projection-based virtual reality displays*, ACM SIGGRAPH Computer Graphics **31** (1997), no. 2, 46–49.
- [8] P.E. Debevec and J. Malik, *Recovering high dynamic range radiance maps from photographs*, ACM Siggraph (1997), 369–378.
- [9] M. Flagg, J. Summet, R. Somani, J.M. Rehg, R. Sukthankar, and T.-J. Cham, *Shadow elimination and occluder light suppression for switched multi-projector displays*, International Conference on Computer Vision, Demo Session (2003).
- [10] R.C. Gonzalez and R.E. Woods, *Digital image processing, 2nd edition*, Prentice Hall, 2002.
- [11] C. Harlow, *Image processing, lecture notes, chapter 4*, <http://www.engr.panam.edu/~caharlow/6399/notes/ch4.colora.pdf> (2005).
- [12] R. Hartley and A. Zisserman, *Multiple view geometry in computer vision*, Cambridge University Press, 2000.
- [13] M.N. Hilario and J.R. Cooperstock, *Occlusion detection for front-projected interactive displays*, 2nd International Conference on Pervasive Computing (2004).
- [14] R. Jain, R. Kasturi, and B.J. Schunck, *Machine vision*, McGraw-Hill Inc., 2000.
- [15] C. Jaynes, B. Seales, K. Calvert, Z. Fei, and J. Griffioen, *The metaverse - a collection of inexpensive, self-configuring, immersive environments*, 7th International Workshop on Immersive Projection Technology/Eurographics Workshop on virtual Environments (2003).
- [16] C. Jaynes, S. Webb, R. Steele, M. Brown, and W. Seales, *Dynamic shadow removal from front projection displays*, IEEE Visualization (2001).

- [17] C. Jaynes, S. Webb, R. Steele, and R.M. Steele, *Camera-based detection and removal of shadows from interactive multiprojector displays*, IEEE Transactions on Visualization and Computer Graphics **10** (2004), no. 3, 290–301.
- [18] K. Li, H. Chen, Y. Chen, D.W. Clark, P. Cook, S. Damianakis, G. Essl, A. Finkelstein, T. Funhouser, T. Housel, A.Klein, Z.Liu, E. Praun, R. Samanta, B. Shedd, J.P. Singh, G. Tzanetakis, and J. Zheng, *Building and using a scalable display wall system*, Computer Graphics and Applications **20** (2000), no. 4, 29–37.
- [19] A. Licsar and T. Sziranyi, *Hand gesture recognition in camera-projector system*, International Workshop on Human-Computer Interaction (2004).
- [20] A. Majumder, *Properties of color variation across multi-projector displays*, SID Eurodisplay (2002).
- [21] A. Majumder and R. Stevens, *Color non-uniformity in projection based displays: Analysis and solutions*, IEEE Transactions on Visualization and Computer Graphics **10** (2004), no. 2, 177–188.
- [22] T. Mitsunaga and S.K. Nayar, *Radiometric self-calibration*, Computer Vision and Pattern Recognition **1** (1999), 374–380.
- [23] S.K. Nayar, H. Peri, M.D. Grossberg, and P.N. Belhumeur, *A projection system with radiometric compensation for screen imperfections*, ICCV Workshop on Projector-Camera Systems (PROCAMS-2003) (2003).
- [24] C. Pinhanez, *The everywhere displays projector: A device to create ubiquitous graphical interfaces*, Ubiquitous Computing 2001 (UbiComp'01) (2001).
- [25] C. Pinhanez, F. Kjeldsen, A. Levas, G.S. Pingali, J. Hartman, A. Levas, M.E. Podlaseck, V. Kwatra, and P.B. Chou, *Transforming surfaces into touch-screens*, Tech. report, IBM Research Division, December 2001.

- [26] K.N. Plataniotis and A.N. Venetsanopoulos, *Color image processing and applications*, Springer, 2000.
- [27] W.H. Press, S.A. Teukolsky, W.T. Vetterling, and B.P. Flannery, *Numerical recipes in c, 2nd edition, the art of scientific computing*, Cambridge University Press, 1992.
- [28] R. Raskar, *Immersive planar displays using roughly aligned projectors*, IEEE Virtual Reality (2000).
- [29] R. Raskar and P.A. Beardsley, *A self-correcting projector*, IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR) **2** (2001), 504–508.
- [30] R. Raskar, J. van Baar, P. Beardsley, T. Willwacher, S. Rao, and C. Forlines, *ilamps: Geometrically aware and self-configuring projectors*, ACM Transactions on Graphics (TOG) **22** (2003), no. 3, 809–818.
- [31] 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*, Computer Graphics, Annual Conference Series (1998), 179–188.
- [32] Y. Sato, Y. Kobayashi, and H. Koike, *Fast tracking of hands and fingertips in infrared images for augmented desk interface*, Fourth International Conference on Automatic Face and Gesture Recognition (2000).
- [33] M.C. Stone, *Color and brightness appearance issues in tiled displays*, IEEE Computer Graphics and Applications **21** (2001), no. 5, 58–66.
- [34] D. Sud, *Design of a multi-projector display system*, Master’s thesis, Electrical and Computer Engineering Department, McGill University, 2005.
- [35] R. Sukthankar, T.-J. Cham, and G. Sukthankar, *Dynamic shadow elimination for multi-projector displays*, Computer Vision and Pattern Recognition (2001).

- [36] R. Sukthankar, R. Stockton, and M. Mullin, *Automatic keystone correction for camera-assisted presentation interfaces*, Proceedings: ICMI (2000).
- [37] J. Summet, G.D. Abowd, G.D. Corso, and J.M. Rehg, *Virtual rear projection: A comparison study of projection technologies for large interactive displays*, Tech. report, Georgia Institute of Technology, 2003.
- [38] R. Surati, *Scalable self-calibrating display technology for seamless large-scale displays*, Master's thesis, PhD thesis, Department of Computer Science, Massachusetts Institute of Technology, 1998.
- [39] 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*, International Journal of Computer Vision **53** (2003), no. 2, 115–133.
- [40] D.S. Tan and R. Pausch, *Pre-emptive shadows: Eliminating the blinding light from projectors*, Interactive poster at CHI 2002 Conference on Human Factors in Computing Systems (2002).
- [41] J.-P. Tardif, S. Roy, and M. Trudeau, *Multi-projectors for arbitrary surfaces without explicit calibration nor reconstruction*, The 4th International Conference on 3-D Digital Imaging and Modeling (3DIM'2003) (2003), 217–224.
- [42] Y. Tokuda, S. Iwasaki, Y. Sato, Y. Nakanishi, and H. Koike, *Ubiquitous display for dynamically changing environments*, ACM Conference on Human Factors in Computing Systems (CHI2003) Extended Abstract (2003).
- [43] E. Trucco and A. Verri, *Introductory techniques for 3-d computer vision*, Prentice Hall, Inc. 1998.

- [44] J. van Baar, T. Willwacher, S. Rao, and R. Raskar, *Seamless multi-projector display on curved screens*, isbn: 3-905673-00-2, Eurographics Workshop on Virtual Environments (2003), 281–286.
- [45] C. von Hardenberg and F. Brard, *Bare-hand human computer interaction*, ACM Workshop on Perceptive User Interfaces (2001).
- [46] G. Wallace, H. Chen, and K. Li, *Color gamut matching for tiled display walls*, Eurographics Workshop on Virtual Environments (2003), 293–302.
- [47] P. Wellner, *Digitaldesk*, Communications of the ACM **36** (1993), no. 7, 87–96.
- [48] S. J. Williamson and H. Z. Cummins, *Color in nature and art*, Wiley, 1983.
- [49] R. Yang and G. Welch, *Fast image segmentation and smoothing using commodity graphics hardware*, Journal of Graphics Tools **7** (2002), no. 4, 91–100.
- [50] J. Yin and J.R. Cooperstock, *Color correction methods with application to digital projection environments*, Journal of WSCG (2004).

Document Log:

Manuscript Version 0

Typeset by $\mathcal{A}\mathcal{M}\mathcal{S}$ - $\mathcal{L}\mathcal{A}\mathcal{T}\mathcal{E}\mathcal{X}$ — 4 October 2005

MARIA NADIA HILARIO

CENTRE FOR INTELLIGENT MACHINES, MCGILL UNIVERSITY, 3480 UNIVERSITY ST., MONTRÉAL
(QUÉBEC) H3A 2A7, CANADA, *Tel.* : (514) 398-8200

E-mail address: `mnhilar@cim.mcgill.ca`

Typeset by $\mathcal{A}\mathcal{M}\mathcal{S}$ - $\mathcal{L}\mathcal{A}\mathcal{T}\mathcal{E}\mathcal{X}$