

UC Irvine

UC Irvine Electronic Theses and Dissertations

Title

A Unified Approach for Consistent Calibration of Spatially Augmented Reality Systems

Permalink

<https://escholarship.org/uc/item/9195w7pd>

Author

Abbaspour Tehrani, Mahdi

Publication Date

2018

Supplemental Material

<https://escholarship.org/uc/item/9195w7pd#supplemental>

Peer reviewed|Thesis/dissertation

UNIVERSITY OF CALIFORNIA,
IRVINE

A Unified Approach for Consistent Calibration of Spatially Augmented Reality
Systems

DISSERTATION

submitted in partial satisfaction of the requirements
for the degree of

DOCTOR OF PHILOSOPHY
in Computer Science

by

Mahdi Abbaspour Tehrani

Dissertation Committee:
Professor Aditi Majumder, Chair
Professor Gopi Meenakshisundaram
Professor Shuang Zhao
Professor Brian Cummings

2018

Portion of Chapter 3 © 2017 IEEE
All other materials © 2018 Mahdi Abbaspour Tehrani

DEDICATION

To my parents and my sister for all their supports and sacrifices

TABLE OF CONTENTS

| | Page |
|--|-------------|
| LIST OF FIGURES | v |
| LIST OF TABLES | x |
| LIST OF ALGORITHMS | xi |
| ACKNOWLEDGMENTS | xii |
| CURRICULUM VITAE | xiii |
| ABSTRACT OF THE DISSERTATION | xvi |
| 1 Introduction | 1 |
| 1.1 Challenges | 2 |
| 1.2 Goal | 5 |
| 2 Geometric Registration and Calibration | 10 |
| 2.0.1 Main Contributions | 11 |
| 2.1 Related work | 13 |
| 2.1.1 Registration vs. Parameter Estimation | 13 |
| 2.1.2 One-time vs. Continuous Registration | 17 |
| 2.1.3 Scalability, Accuracy and Robustness | 19 |
| 2.2 System Overview | 21 |
| 2.3 Offline Calibration Method | 24 |
| 2.3.1 Connectivity Graph Construction | 26 |
| 2.3.2 Partial 3D Reconstruction and Camera Calibration | 27 |
| 2.3.3 Initial Localized Projector Calibration | 30 |
| 2.3.4 Radially Cascading Surface Integration | 31 |
| 2.4 Online Image Correction | 33 |
| 2.5 Continuous Registration | 34 |
| 2.5.1 Projector Movement | 39 |
| 2.5.2 Changing Display Surface Geometry | 39 |
| 2.6 Implementation and Results | 40 |

| | |
|--|------------|
| 3 Camera Intrinsic Calibration | 45 |
| 3.1 Background and Related Work | 47 |
| 3.2 Device Setup | 51 |
| 3.2.1 Defocus Blur | 52 |
| 3.2.2 Pixel Shift | 53 |
| 3.3 Proposed Calibration Method | 54 |
| 3.3.1 Correspondence to Masking Layer | 55 |
| 3.3.2 Correspondence to Background Layer | 57 |
| 3.3.3 Camera and Lens Parameter Estimation | 59 |
| 3.4 Hardware Prototype | 62 |
| 3.5 Evaluation and Discussion | 63 |
| 3.6 Conclusion | 66 |
| 4 Photometric Calibration | 68 |
| 4.1 Background and Related Work | 69 |
| 4.1.1 Main Contribution | 71 |
| 4.2 Notations | 72 |
| 4.3 Shape-Aware 3D Gamut Morphing | 77 |
| 4.3.1 Per Projector White Point Balancing | 78 |
| 4.3.2 Chrominance Gamut Morphing | 79 |
| 4.3.2.1 Graph Based Spatial Sampling | 80 |
| 4.3.2.2 Spherical Space Transformation | 83 |
| 4.3.2.3 Finding per pixel scale factor | 84 |
| 4.3.3 Brightness Smoothing | 85 |
| 4.4 Online Image Correction | 88 |
| 4.5 Implementation and Results | 89 |
| 5 User Interaction | 90 |
| 5.1 Related Work | 92 |
| 5.2 Main Contributions | 95 |
| 5.3 Projector Tracking | 96 |
| 5.3.1 Dynamic Projector Calibration | 97 |
| 5.3.2 Shadow Based Interaction | 100 |
| 5.4 Interaction Modalities | 100 |
| 5.5 Conclusion | 104 |
| 6 Conclusion and Future Work | 105 |
| Bibliography | 107 |

LIST OF FIGURES

| | Page |
|--|------|
| 1.1 Left: Showing height map on a 3D relief using sandpit covered by 4 projectors denoted as P1 to P4. Right: high resolution imagery on a small table top object illuminated by 6 projectors denoted by P1 to P6. | 2 |
| 1.2 Left: Showing a dome illuminated by three overlapping projectors. Notice the difference of color in areas covered by a single projector and multiple projectors. Right: This figure shows a relief map illuminated by two tiled projectors (top) and superimposed with two more tiled projectors to increase the brightness (bottom) | 3 |
| 1.3 This figure shows a multi-projector display made of 6 tiled projectors. Note the ghosting in the overlap of projectors before geometric calibration (left) and the seamless image after geometric calibration (right). | 4 |
| 1.4 This figure shows the comparison of proposed unified calibration (right) with result of registration by estimating the device parameters independently and using this estimation for 3D reconstruction without cross-validation between devices (middle) on a display made of 4 projectors on a sandpit observed by 4 cameras (left). | 5 |
| 1.5 Left: Showing the projected grid lines on a spherical surface with only geometric registration. Right: result of geometric registration with shape conforming geometric warp enabled by complete geometric calibration. | 6 |
| 1.6 This figure shows different classes of color variation. Top left: shows color variation between different projectors. Top right: variation of color in the overlap of projectors. Note the overlap area is brighter as it is illuminated by two projectors. Bottom left: shows intra-projector color variation. Note that the brightness falls off at the periphery of projectors. Bottom right: this image shows the variation of color across the display surface caused by geometry of the display. | 7 |
| 1.7 This figure shows result of geometric registration of multi-projector display before (left) and after (right) photometric calibration. Note that before photometric correction the brightness of the display is much higher in the overlap of projectors. | 8 |

| | | |
|-----|---|----|
| 2.1 | Left: Our setup made of 6 projectors and 4 cameras illuminating a hand-modeled sand pit, creating a surface of arbitrary shape, each projector shown with a different color. Middle: recovered 3D shape of the sand-pit and parameters of the projectors and cameras for two different shapes. Right: A seamless elevation map projected on the arbitrary shape. Note that the warmer colors light the higher regions and the cooler colors light the lower regions demonstrating the accuracy of our surface geometry reconstruction. | 11 |
| 2.2 | The system with m cameras and n projectors connected to the p computers via two independent interconnection networks. | 21 |
| 2.3 | (a): The configuration of a system with 6 projectors 4 cameras. Each blue region shows the projection area of a projector and dashed colored lines show the FOV of the cameras. (b and c): local adjacency graph graph for P_1 and C_1 constructed by C_1 . The edges of partial A_p and A_c are shown in blue and green respectively. The cross edges are shown with red dashed lines. The nodes of the A_c are color coded to be consistent with the camera field of views shown in (a). (d): The connectivity graph between projectors and cameras. Note that many regions of the surface is seen by only one camera. For example, most of P3 is seen only by C2. Similarly, most of P1 is seen only by C1. | 22 |
| 2.4 | This figure shows the pipeline of our method. Our method has two components. First an offline calibration algorithm provides the displacement and attenuation map, then our online correction algorithm uses these maps to create a seamless imagery on the arbitrary shape. | 25 |
| 2.5 | This figure shows steps of our algorithm. (a): Our setup with 3 projectors and 3 cameras on a dome. (b): Adjacency graph for projectors and cameras. (c,d and e): Camera calibration and partial 3D reconstruction for 3 projectors after Step 2. (f): Partially reconstructed display surface of P_1 , P_2 and P_3 in red, blue and green respectively after Step 2. Each of these 3D reconstructions are in different coordinate systems with different rotation, translation and scale factors. (g) Complete 3D reconstruction of the display surface of P_1 , P_2 and P_3 in red, blue and green respectively after step 3. (h): The 3D reconstruction of the display surface, and the calibrated projectors after Step 4 of radially cascading refinement. | 27 |
| 2.6 | This figure shows the order that projectors added to set V from our previous example in Figure 2.3. The gradient of the color from red to yellow shows the order of the projectors - P_5 , P_6 , P_3 , P_2 , P_1 , P_4 . . . | 29 |
| 2.7 | (a): Projected image with 3 projectors before correction. (b): Geometrically registered image with these 3 projectors. (c): Distorted image after moving a projector. (d): Corrected image after projector recalibration. | 31 |

| | | |
|------|---|----|
| 2.8 | Top: Our setup with 2 projector (left) and 4 projector for super-imposed projection (right). Middle: Connectivity graph of the setups. Bottom: Result with two projectors (1x2 array) (left) and super imposed projection with 4 projectors (2x1 array on a 1x2 array) (right). | 35 |
| 2.9 | Final result of projecting on a dome with 3 projector and 3 cameras (top) and zoomed in image (bottom). | 36 |
| 2.10 | This figure shows the reprojection error (in pixel) of the 3D points on the projectors image plane in x and y direction in our simulator. . . . | 36 |
| 2.11 | This figure shows our result projecting on a sand pit with 4 projectors and 4 cameras. Left: Connectivity graph for projectors and cameras. Middle: Recovered 3D geometry of the display surface and parameters of the projectors and cameras. Right: Final result. | 37 |
| 2.12 | This figure shows our result projecting on a sand pit using same setup as Figure 2.1. (a and b): View-dependent projection from the incorrect view point(a) and correct view position(b). (c and d): Zoomed in image to show the accuracy of geometric registration. (e and f): Shows the result of changing the appearance of display geometry by projecting different contents. | 37 |
| 2.13 | Top: Showing the overlap of projectors for our setup with 10 projectors and 4 cameras on a cylindrical display surface. Bottom: Final result of projecting a graffiti on the cylindrical wall. | 38 |
| 2.14 | This figure shows our result on small objects. First row: Our setup with 6 projectors and 8 cameras in tiled configuration around the vase. The projectors illuminate the vase and other objects around it. Second row: Our final result from different view points. Third row: Showing a checker board pattern on the vase to demonstrate the accuracy of geometric registration(left) and the result of superimposed projection on the vase (right). | 41 |
| 2.15 | This figure shows the comparison of our method (right) with result of registration by estimating the devince paramters by standard checkerboard calibration method and using this estimation for 3D reconstruction without cross-validation between devices (middle) on a display made of 4 projectors on a sandpit observed by 4 cameras (left). | 44 |
| 3.1 | Sketches of the proposed setups. <i>Left</i> : Minimal configuration using two layers: the first layer is an emissive planar display panel and the second layer, a thin opaque sheet with a grid of holes, is placed slightly angled in front. <i>Center</i> : Using two tilted hole planes enables a better coverage for wide angle lenses and an additional calibration plane. <i>Right</i> : Three layered hole planes consisting of yellow, cyan and magenta filters allow to capture correspondences to three planes with one configuration. . . . | 50 |

| | | |
|------|--|----|
| 3.2 | Defocus blur of a point source. The size and shape of the CoC depends on the lens' aperture size and shape. <i>Top</i> : A closed physical aperture creates a smaller image of the hole. <i>Bottom</i> : A large aperture size creates bigger images of the hole and a larger area of the display panel is visible through the hole. | 52 |
| 3.3 | Captured images with closed (<i>Left</i>) and open aperture (<i>right</i>). The CoC of the holes is influenced by the aperture but also by its shape as well as the spatial location on the image plane. | 53 |
| 3.4 | Captured images of a masking layer. Gray code patterns of two varying frequencies (<i>Left,Right</i>) are displayed on the background layer. Note how the pixel shift described in Sec. 3.2.2 leads to slightly shifted lines when observed through different holes, best seen in the left image at the green line. | 53 |
| 3.5 | <i>Top</i> : Different light rays from a (defocused) pixel on the display will intersect with camera's sensor at different position. Only the ray which passes through the optical center of the lens (blue ray) resides on the same line. <i>Bottom</i> : We only consider the rays passing the optical center of the lens for the calibration. | 54 |
| 3.6 | Color coded height field plot of a sample hole of one of the captured images of the blocking layers (<i>Left</i>) and the center found by fitting a 2D Gaussian function (<i>Right</i>). | 56 |
| 3.7 | Color coded pixel coordinates sensed by the camera through the holes on the display. Note that the projection of the holes on the display is free of distortions and only contains a small amount of keystoneing. . . . | 57 |
| 3.8 | <i>Left</i> : Prototypical setup with two masking layers. The background layer is an emissive LCD panel and the other layers consist of thin opaque sheets perforated by a grid of holes. <i>Right</i> : Three different hole layer configurations: (a) v-shaped planes to get better coverage when calibrating wide angle lenses as shown on the left, (b) cyan, magenta and yellow layers to generate superimposed correspondences using color sensors, and (c) a minimal, single plane configuration. | 58 |
| 3.9 | Extending the rays which intersect on the camera's sensor allows to estimate the focus distance. | 61 |
| 3.10 | Estimated aperture sizes compared to ground truth values. | 65 |
| 3.11 | <i>Left</i> : 3D printed sample all-in-one prototype with lens mount and smartphone as display panel. <i>Right</i> : According CAD images. | 66 |
| 4.1 | This figures shows the color gamut of 4 different projectors (left) and projecting white color with these projectors (right). Notice a small variation in color gamut creates a noticeable color variation in user's perception. | 74 |
| 4.2 | <i>Left</i> : This figure shows the effect of incoming light direction on brightness. Right: showing the effect of surface geometry of projector's orientation on size and density of pixels. | 76 |

| | | |
|-----|---|-----|
| 4.3 | This figure show the variation of 3D color gamut of two projectors. We smoothly morph the color gamut in the overlap of projectors. The color of each 3D color gamut in this figure indicates the color of white. | 78 |
| 4.4 | This figure shows the projection of white color using four projectors on a relief map before (left) and after (right) white point balancing. | 79 |
| 4.5 | This figure shows projection of white color using four projector before (left) and after (right) 2D chrominance gamut morphing. As you can see despite the fact that the chrominance color is smoothly morphed across the display surface, still there is visible brightness variation in overlap areas. | 86 |
| 4.6 | This figure shows our result before (left) and after (right) Brightness smoothing and Bezier curve fitting. | 86 |
| 4.7 | This figure shows the result of our photometric registration method in a system made of four overlapping projectors projecting on a arbitrary shaped relief map. | 88 |
| 4.8 | This figure shows a system made of three projectors projecting on a dome before photometric correction (left), result of our method (middle) and result of RGB blending (right). | 89 |
| 5.1 | This figure shows (a) four overlapping projectors projecting on the sand-box (b) result of geometric registration of the multiple stationary projectors by automatically projecting environment texture (grass on high elevations and water on low elevations) and (c) result of 3D reconstruction of the display surface. | 96 |
| 5.2 | This figure shows the result of extraction of the projected ball by hand-held projector before (left) and after (right) removing environment background. | 98 |
| 5.3 | This figure shows (a) single or multiple user interacting with the system by moving the ball using handheld projector (first column),(b) Iteraction of the projected content with geometry of the display surface (second column) and (c) interaction of the user with the projected content using hand shodow (third column). First row shows the initial position of the ball and second row shows the position of the ball after interaction with shadow and geometry of the surface. Note that in these figures the projector does not move and only the projected content moves. | 101 |
| 5.4 | This figure shows the interaction of the ball with dynamic overlaying content in our Live Content Prototype. The color of the ball is changed by hitting other moving projected contents. | 101 |
| 5.5 | This figure shows the result of our flash light system. The user can see different parts of the content by moving the handheld projector. For instance in this example we created a prototye for small kids that they can find the position of different animals living in different elevations of the terrain | 103 |

LIST OF TABLES

| | Page |
|--|------|
| 2.1 Comparison of our work with previous works | 19 |
| 2.2 System running time. | 43 |
| 2.3 Comparison of the estimated projector intrinsic (in pixel) using check-board and our system. | 44 |
| 3.1 Comparison of the calibration results using the prototypes of the proposed method (cf. Fig. 3.8) and the algorithm proposed by Zhang: f_x and f_y are focal length of the camera in horizontal and vertical direction. (c_x, c_y) is the coordinate of the principal point of the camera, and $k_1 \dots k_4$ are the four estimated distortion parameters | 63 |
| 3.2 Estimated distances of the camera's focal plane in comparison with the actual, i.e. measured, distance. | 65 |
| 3.3 Reprojection error comparison [pixels] | 66 |
| 4.1 This table shows the comparison of our method with existing works. . . | 72 |
| 5.1 Categorized of handheld projector based interaction paradigms. | 92 |
| 5.2 Comparison of the HIP paradigm with prior work | 93 |

LIST OF ALGORITHMS

| | Page |
|---|------|
| 1 Partial 3D Reconstruction | 29 |
| 2 Radialy Cascading Surface Integration | 33 |
| 3 Intrinsic Calibration Method | 55 |

ACKNOWLEDGMENTS

First I would like to thank my adviser, Prof. Aditi Majumder for wonderful advises and helping me through all these years by teaching me how to do research. Special thanks to Prof. Gopi Meenakshisundaram for helping me through my PhD studies. I would also thank Prof. Shuang Zhao and Prof. Brian Cummings for accepting to be in my dissertation committee. I would also thank Link Foundation Fellowship for their one year financial supports of my research.

The intrinsic camera calibration method has been done in collaboration with Disney Research Zurich. I would like to thank Dr. Anselm Grundhofer for this work. I also want to thank Alexander Sidenko for helping me to develop the interaction paradigms in our system.

CURRICULUM VITAE

Mahdi Abbaspour Tehrani

EDUCATION

University of California Irvine, Irvine, CA, 2012 - 2018
PhD. in Computer Science **GPA: 3.86/4**

Sharif University of Technology, Tehran, Iran, 2007 - 2012
B.S. in Computer Engineering - Software **GPA: 3.97/4**,

RESEARCH INTEREST

Computer Graphics, Computer Vision, Human computer interaction

PUBLICATION

Mahdi Abbaspour Tehrani, Aditi Majumder, *Color seamlessness of multi projector displays on arbitrary shaped surfaces*, In progress to be submitted.

Mahdi Abbaspour Tehrani, Alexander Sidenko, Aditi Majumder, *HIP: Handheld Interactive Projectors Via Continuous 3D Tracking*, In progress to be submitted to IEEE Computer Graphics and Applications 2019.

Mahdi Abbaspour Tehrani, M. Gopi, Aditi Majumder, *Automated Projector Calibration and Registration for Arbitrary-Shaped Multi-Projector Systems*, Submitted to IEEE Transactions on Visualization and Computer Graphics, 2018.

Mahdi Abbaspour Tehrani, Thabo Beeler, Anselm Grundhofer, *A Practical Method for Fully Automatic Intrinsic Camera Calibration Using Directionally Encoded Light*, Computer Vision and Pattern Recognition (CVPR), Honolulu, Hawaii, July 2017. (Selected as Spotlight Paper)

Mahdi Abbaspour Tehrani, Aditi Majumder, M. Gopi, *Correcting perceived perspective distortions using object specific planar transformations*, International Conference on Computational Photography (ICCP) , Evanston, IL, USA, May 2016.

Behzad Sajadi, **Mahdi Abbaspour Tehrani**, Mehdi Rahimzadeh, Aditi Majumder, *High-Resolution Lighting of 3D Reliefs Using a Network of Projectors and Cameras*, 3DTV, 2015. (**Best Paper Award**)

Mahdi Abbaspour Tehrani, Aditi Majumder, M.Gopi, *Undistorting Foreground Objects in Wide Angle Images*, IEEE International Symposium on Multimedia (ISM), Anaheim, CA, USA, 2013.

Soheil Hassas Yeganeh, Jafar Habibi, Hassan Abolhassani, **Mahdi Abbaspour Tehrani**, Jamshid Esmaelnezhad, *An approximation algorithm for finding skeletal points for density based clustering approaches*, Proceedings of the IEEE Symposium on Computational Intelligence and Data Mining, CIDM, part of the IEEE Symposium Series on Computational Intelligence, pp. 403-410, Nashville, TN, USA, 2009.

M.Abbaspour, E.Asgari, S.Bagheri, M. N. Do, P.Khanipour, J.Lu, S.MahAbadi, A.Vakilian, *Automatic pill identification*, ADSC technical report, pp. 1 - 16, Oct. 2010.

M.Abbaspour, E.Asgari, S.Bagheri, M. N. Do, P.Khanipour, J.Lu, S.MahAbadi, A.Vakilian, *Indoor positioning and navigation with a camera phone*, ADSC technical report, pp. 1 - 10, Oct. 2010.

POSTERS

Mahdi Abbaspour Tehrani, Aditi Majumder, *Auto-Calibration of Multi-Projector Systems on Arbitrary Shapes*, Applied Imagery Pattern Recognition (AIPR), Washington DC, October 2016.

Behzad Sajadi, **Mahdi Abbaspour Tehrani**, Mehdi Rahimzadeh, Aditi Majumder, *Auto-Registration of Distributed Network of PPPs on Non-planar Surfaces*, ICCP, Evanston, IL ,May 2016.

DEMOS

Aditi Majumder, Duy-Quoc Lai, **Mahdi Abbaspour Tehrani**, *A multi-projector display system of arbitrary shape, size and resolution*, SIGGRAPH, 2015.

Aditi Majumder, Duy-Quoc Lai, **Mahdi Abbaspour Tehrani**, *A multi-projector display system of arbitrary shape, size and resolution*, IEEE Virtual Reality Conference, 2015.

AWARDS AND HONORS

Best Demo Award, Computer science research showcase, University of California Irvine, 2017.

Recipient of Link Foundation Fellowship including one year financial support, 2016-2017.

Best Demo Award, Computer science research showcase, University of California Irvine, 2016.

Recipient of David Feign Fellowship Award, July 2015.

Best Paper Award, 3DTV 2015.

Recipient of Deans Fellowship, including four years of full financial support, from University of California, Irvine, 2012-2016

Awarded summer internship grant from **ADSC (Advance Digital Sciences Center)**, joint research center between **University of Illinois at Urbana-Champaign** and **A*STAR**, July 2010 - September 2010.

Ranked 195th in the annual nationwide universities entrance exam (Konkoor) over more than 300,000 participants, 2007.

Honorary Admitted to Graduate Study at Sharif University Computer Engineering Department, 2010.

Selected as an outstanding student for Computer Engineering Olympiad team of Sharif University of Technology, April 2011.

TEACHING

Teaching Assistant, University of California Irvine.

- Computer Graphics, Visual Computing, Artificial intelligence, Boolean Algebra, Introduction to Programming, Advance Graphics, Linear Algebra

Teaching Assistant, Sharif University of Technology.

- Signals and Systems, Design and Analysis of Algorithms, Theory of Languages and Automata, System Analysis and Design and Advanced Programming.

ABSTRACT OF THE DISSERTATION

A Unified Approach for Consistent Calibration of Spatially Augmented Reality Systems

By

Mahdi Abbaspour Tehrani

Doctor of Philosophy in Computer Science

University of California, Irvine, 2018

Professor Aditi Majumder, Chair

Spatially augmented reality (SAR) systems illuminate objects or environments of different shapes or size, starting from table top objects to room-sized environments, using multiple projectors that are casually aligned creating full or partial overlaps with their neighbors. Full overlaps or superposition is used to increase brightness and partial overlaps or tiling is used to increase resolution and scale. Such systems enable applications such as visualization, entertainment and education by projecting data right on the physical 3D model.

Creating a seamless display on complex shapes using multiple projectors entail two primary challenges. Geometric registration which addresses pixel alignment in the overlap of multiple projectors and warping to conform to underlying 3D shape and color registration addresses the spatial variation of color within and across different projectors and also color variation introduced by the complex non-planar shape of the 3D surface.

My thesis presents un uniform approach for consistent calibration of SAR systems enabling easy deployment of spatially augmented reality environments of different sizes

(from small objects to large immersive environments), different shapes (inside-looking-out or outside-looking in), and different configurations (tiled or superimposed). First We show that a unified approach that relies on cross-correlation and validation between the devices (projectors and cameras) and display properties achieves a consistent system reconstruction resulting in a highly accurate, robust and scalable registration in multi-projector systems that is not possible by independent calibration of devices and reconstruction of surface properties. Second we present the first method to achieve complete color seamlessness (both luminance and chrominance) for multi-projector displays on completely arbitrary shapes using a novel shape-aware 3D color gamut morphing method. Finally we present multiple interaction modalities, namely interaction of the handheld projector (a) with the surface geometry only, (b) with overlaid digital content only (from other static or dynamic handheld projectors), and (c) with both the digital content and the surface simultaneously.

Chapter 1

Introduction

The size of data that needs to be visualized is increasing rapidly in terms of resolution and dimensions. Therefore, it is vital to develop new techniques for data visualization and user interaction to enable better data exploration which in turn enables better data understanding and interpretation. In this context, traditional displays cannot provide an adequate data visualization experience, both in terms of scale and resolution, specially when dealing with three dimensional data. This is primarily due to fact that 3D nature of the data is not adequately captured or explored in a 2D flat screen.

Advent of Virtual Reality (VR) systems started a new generation of data visualization techniques where the user can view and interact with 3D content in a virtual environment. However, these head-mounted displays encumber the users and disconnect them from the real world. Augmented Reality (AR) tries to solve this problem by augmenting the real world with digital or virtual data. This can be done in two ways. A picture of the real world in a device (e.g. tablets or phones) can be augmented with digital information. However, this still does not break free from the confines of a 2D flat

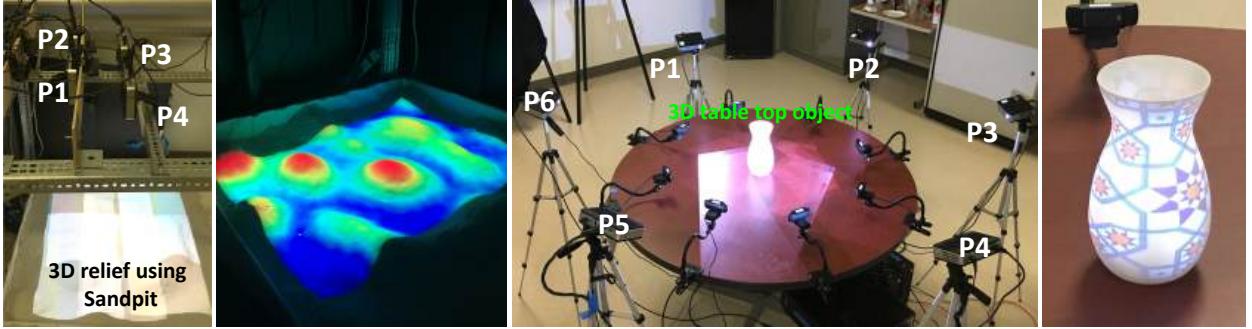


Figure 1.1: Left: Showing height map on a 3D relief using sandpit covered by 4 projectors denoted as P1 to P4. Right: high resolution imagery on a small table top object illuminated by 6 projectors denoted by P1 to P6.

screen display. The second way is to use a see-through headset (e.g. hololens) where the digital 2D or 3D content is augmented with the 3D physical world seen through the headset. However, both AR and VR systems encumber the users with wearables, inhibits natural interaction and cannot provide multi-user shareable experiences.

Spatially Augmented Reality (SAR) systems address the aforementioned shortcomings by projecting light on real world objects enabling shareable visualization of the data right on the 3D object itself. Such systems entail using multiple projectors to create high-resolution imagery on physical objects (e.g., dome, relief map or table top object) as shown in Figure 1.1. The popularity of SAR systems has progressively increased with the advent of consumer projectors. SAR systems today are being used in several applications such as visualization, entertainment and education.

1.1 Challenges

More often than not, every part of the 3D object cannot be illuminated at the desired resolution using a single projector. Therefore, multiple projectors are usually tiled to

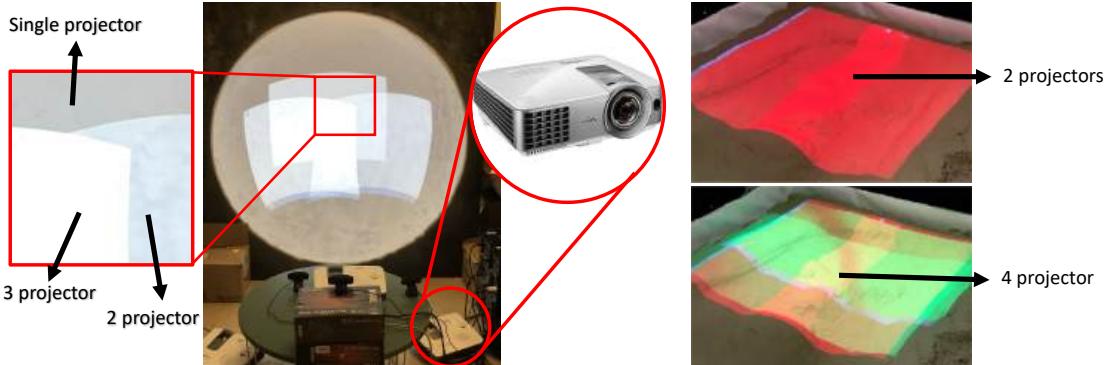


Figure 1.2: Left: Showing a dome illuminated by three overlapping projectors. Notice the difference of color in areas covered by a single projector and multiple projectors. Right: This figure shows a relief map illuminated by two tiled projectors (top) and superimposed with two more tiled projectors to increase the brightness (bottom)

light all parts of the surface to achieve high pixel density in all different regions of a 3D surface. In this case, the projectors overlap in their periphery to create brighter overlap region as shown in Figure 1.2. Consumer projectors often cannot produce enough brightness to operate in the presence of significant ambient light. In such cases, projectors are also superimposed to increase the brightness of the SAR system as shown in Figure 1.2. Using multiple projectors to thus illuminate a 3D surface leads to two problems as follows.

(a) The same point on the surface when lit by multiple different pixels coming from multiple projectors should get the same content. If not, we have ghosting as shown in Figure 1.3. This is called the problem of *geometric misregistration*. The second issue is warping the image to conform to the underlying geometry of the 3D shape. When creating shareable multi-user experience, view independent (a image that is not correct for any one user) rendering of content on the 3D surface is required. This is achieved by mapping conformally (e.g. wall papering) the images on to the 3D surface and is called *shape conforming warping*.

(b) The color can show spatial variation within a single projector for consumer devices

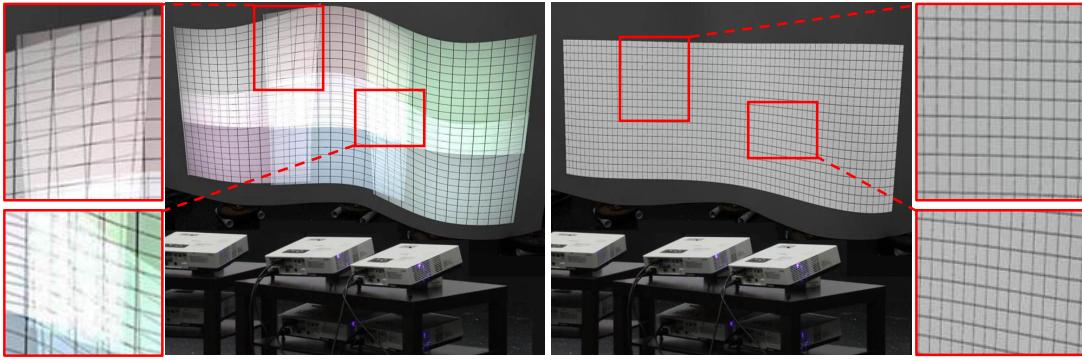


Figure 1.3: This figure shows a multi-projector display made of 6 tiled projectors. Note the ghosting in the overlap of projectors before geometric calibration (left) and the seamless image after geometric calibration (right).

and also can vary across devices. The brighter overlap regions adds to this color variation. This is called the problem of *photometric variation*.

The problems of geometric misregistration and photometric variation were addressed manually taking tens of man hours even for systems with only two projector overlaps. Calibrating large number of projectors with three or four projector overlaps were practically impossible manually. Therefore, it was only possible to use simple surface geometries (e.g. flat or cylindrical surfaces). More importantly, it was almost impossible to get perfect registration which can register images at a subpixel level. To alleviate these problems, automated calibration procedures were designed by embedding cameras in the system along with the projectors. This allowed cameras to observe projected patterns, a process often referred to as structure light, and gather information from these captured patterns that can automatically address geometric misregistration and photometric variation and achieve highly accurate calibration for complex geometries and large number of projectors.

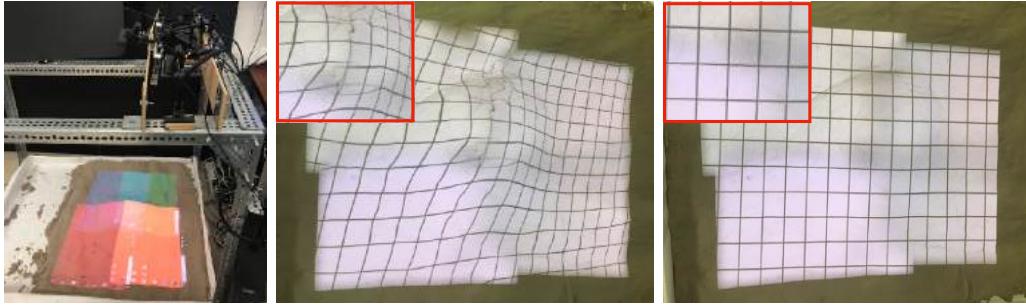


Figure 1.4: This figure shows the comparison of proposed unified calibration (right) with result of registration by estimating the device parameters independently and using this estimation for 3D reconstruction without cross-validation between devices (middle) on a display made of 4 projectors on a sandpit observed by 4 cameras (left).

1.2 Goal

The goal of this thesis is to show that a unified approach for geometric and photometric color calibration that depends on cross-correlation and cross-validation result in a consistent system calibration instead of the traditional approach that independently calibrates every device and surface properties (e.g. geometric shape) and then tries to put them together in one system. This is illustrated in Figure 1.4. Therefore, this thesis enables a generic, completely automatic and scalable geometric and photometric calibration method that enables automated and accurate deployment of seamless content on arbitrary 3D objects and surfaces using multiple projectors and cameras.

In the context of camera based automated solutions for geometric misregistration and photometric variation, it is important to explain a few subtleties. Geometric registration that addresses alignment of pixels to remove ghosting entails finding, for every pixel of a projector, the set of pixels from other projectors overlapping with it on the display surface. This set is non-empty only for pixels in the region of the display surface where more than one projector overlaps. Geometric registration can be performed using *a single* observing camera *without* a full fledged estimation of the intrinsic and

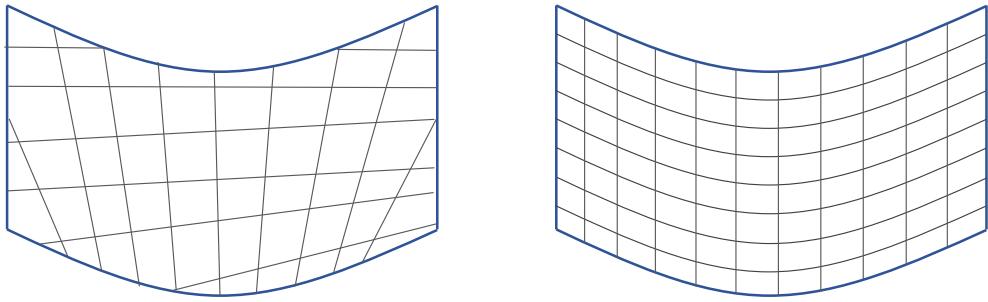


Figure 1.5: Left: Showing the projected grid lines on a spherical surface with only geometric registration. Right: result of geometric registration with shape conforming geometric warp enabled by complete geometric calibration.

extrinsic projector/camera parameters and reconstruction of the 3D surface geometry. In this case the registration will be correct from the viewpoint of the camera and show distortions from other viewpoints, as shown in Figure 1.5. Therefore, in this case, the system is registered, but the image is not *warped to conform to the underlying shape of the display*. Shape conforming geometric warp is only possible when all the device parameters and the 3D surface geometry is accurately reconstructed, called complete geometric calibration of the system. Therefore, geometric calibration, provides geometric registration as a by-product, but not vice versa. Only a full geometric calibration of system parameters can allow us to correct the distortion of projected content resulting in a *shape conforming warp* of the content. As is evident, geometric calibration entailing both alignment of pixels and shape conforming warp is the general solution to project any kinds of content on any surface. Figure 1.5 shows the difference between achieving only geometric registration and both geometric registration and warp. Most previous work focus on geometric registration while this thesis focuses on geometric calibration that enables the additional geometric warp. In addition, this is the first work that achieves complete projector calibration for a multi-projector-camera system. Details of related work is presented in Chapter 2.

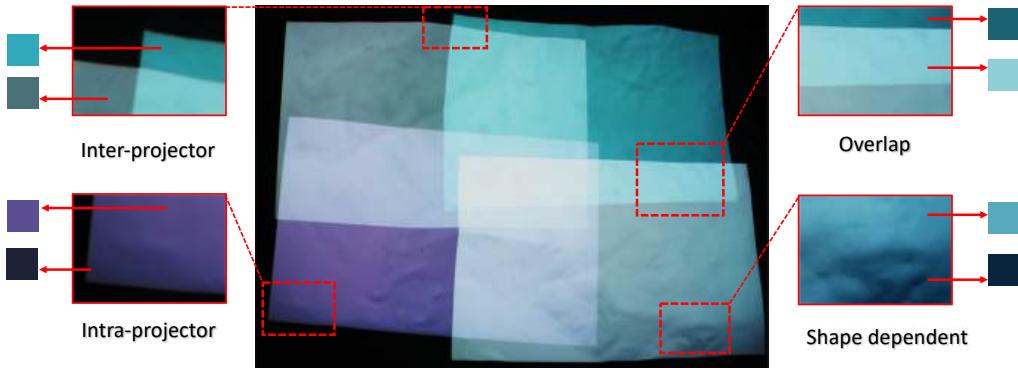


Figure 1.6: This figure shows different classes of color variation. Top left: shows color variation between different projectors. Top right: variation of color in the overlap of projectors. Note the overlap area is brighter as it is illuminated by two projectors. Bottom left: shows intra-projector color variation. Note that the brightness falls off at the periphery of projectors. Bottom right: this image shows the variation of color across the display surface caused by geometry of the display.

Photometric calibration addresses the spatial variation of color within and across different projectors to create a seamless display. A comprehensive color calibration is critical when using off-the-shelf inexpensive projectors that often show severe spatial color variation, even when using projectors of same make and model. Color entails 2D chrominance and 1D luminance. Previous works categorize the color variation in three different classes: intra-projector, inter-projector, overlap variation. Intra-projector variation entails the variation of color within each single projector. Inter projector variation entails the difference in both chrominance and luminance across the different projectors. finally Overlap variation refers to the differently colored higher-brightness regions created by the overlap of multiple projectors. Figure 1.6 demonstrates different kinds of color variation in a four projectors display surface projecting on a arbitrary shaped relief map.

However this categorization completely ignores the effect of the display surface shape on color variation problem which can be rather significant. This thesis introduces the formalization of this shape based color variation (Figure 1.6). All prior works only

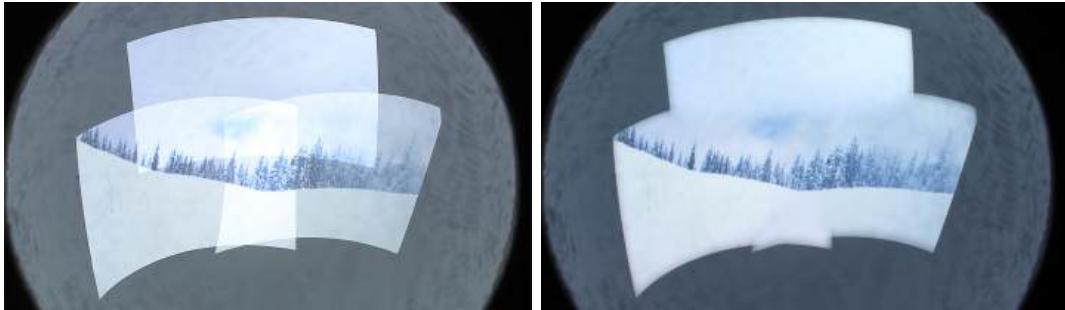


Figure 1.7: This figure shows result of geometric registration of multi-projector display before (left) and after (right) photometric calibration. Note that before photometric correction the brightness of the display is much higher in the overlap of projectors.

assume planar surfaces and therefore do not address shape dependent color variation. Even for planar surfaces, initial works assumed intra-projector uniformity and applied only a color balancing. Later works show that for consumer devices spatial variation has to be addressed. [53, 88] show that achieving color uniformity in consumer devices that show high spatial variation can make the quality of the display so bad to render it useless. Instead morphing the color gamut can achieve seamlessness. Our work presents an unified approach that considers all kinds of color variation across the display surface including color variation due to complex 3D surface geoemtry. Figure 1.7 shows a multi-projector display made of three projectors before and after photometric calibration.

Finally we show that the projectors them self can be used as the interaction device. Projectors are getting smaller every day and they can be used as a handheld device. By dynamic calibration of the projector we can track the projectors position and orientation and used the movement of the projector as an interaction paradigm. We show several interaction methods including

- (a) The projected content can interact with the surface geometry by implementing the rules of physics. For example projected object can move on the surface based on the magnitude and direction of the gradient of the surface.

- (b) The projected content from the handheld projector can interact with the content projected by other projectors including handheld projectors and stationary projectors.
- (c) We show the combination of both interactions with the overlaid digital content and the surface geometry simultaneously.
- (d) Finally as the projectors project the digital content by illuminating the surface we can use the shadows of the user's hand to interact with the digital content.

The reminder of this thesis will start with our geometric calibration method for accurately estimating all system parameters including intrinsic and extrinsic parameters of projectors and extrinsic parameters of camera and the surface properties (chapter 2). Then in chapter 3 we will show our practical method for fully automatic camera intrinsic calibration using directionaly encoded light. This is followed by our photometric calibration method explained in chapter 4. Finally we will show all the user interactions in our system enabled by dynamic calibration of devices (chapter 5).

Chapter 2

Geometric Registration and Calibration

In this chapter we present a completely automated and scalable multi-projector registration system that allows multiple completely uncalibrated projectors and cameras on arbitrary shape surfaces. Our method enables illumination of arbitrarily shaped surfaces using m uncalibrated projectors run by p machines and observed by n cameras where the projectors can be either tiled or superimposed. Our method starts with uncalibrated non-linear projectors and unknown surface geometry and recovers intrinsic and extrinsic parameters of projectors and cameras, projector non-linearities and the surface geometry. Note that geometric registration (assuring that the same point on the surface receives identical content from multiple overlapping projectors) does not entail estimating all the device parameters and surface geometry. Most prior systems achieve geometric registration without estimation of all system parameters. Estimating projector parameters allows for quick recalibration of the system in the face of projector movements, by re-estimating only the parameters of the moved projector and not

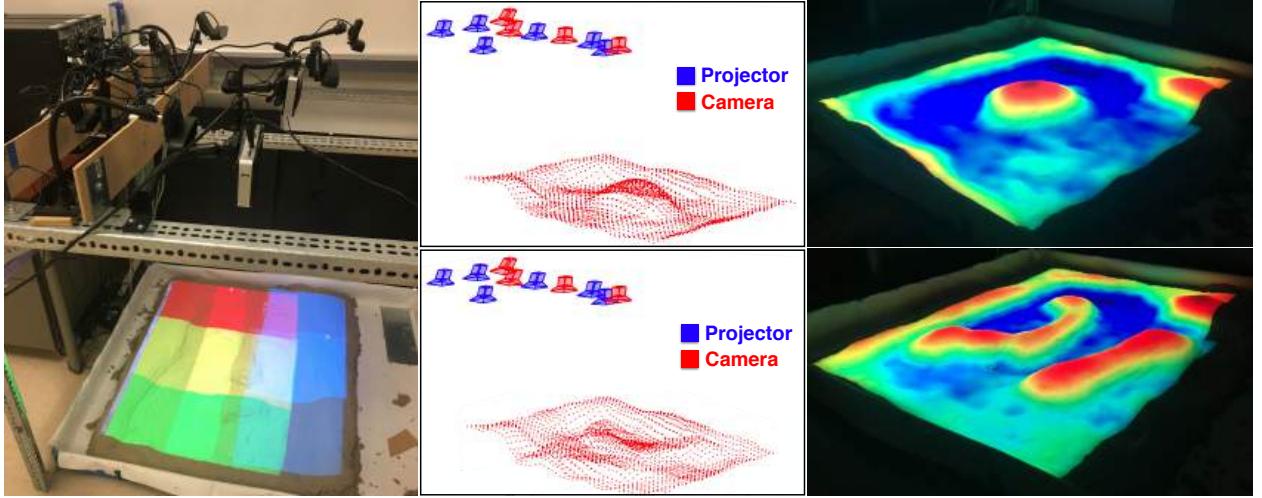


Figure 2.1: Left: Our setup made of 6 projectors and 4 cameras illuminating a hand-modeled sand pit, creating a surface of arbitrary shape, each projector shown with a different color. Middle: recovered 3D shape of the sand-pit and parameters of the projectors and cameras for two different shapes. Right: A seamless elevation map projected on the arbitrary shape. Note that the warmer colors light the higher regions and the cooler colors light the lower regions demonstrating the accuracy of our surface geometry reconstruction.

the entire system. Thus, our work can enable easy deployment of spatially augmented reality environments of different sizes (from small table top objects to large immersive environments), different shapes (inside-looking-out or outside-looking in), and different configurations (tiled or superimposed) using the same proposed method.

2.0.1 Main Contributions

The main contributions of this work are as follows.

1. Our main contribution lies in a complete consistent recovery of all the parameters of the m projectors and n cameras, the shape of the illuminated 3D surface, and subsequent geometric registration in a multi-projector system using uncalibrated devices without using any fiducials. More importantly, unlike any prior method that need at

least two cameras to observe *every* surface point, we can achieve this registration even when parts of the surface are observed by just one camera as long as it is illuminated by a projector. Of course, we do need a minimum of 2 cameras, i.e $n \geq 2$. This significantly simplifies the deployment of the system. Since we recover all the device and surface parameters, we can map any content on the arbitrarily shaped surface in both view-dependent and view-independent (e.g. conformal mapping) manner.

2. Unlike any prior work, we can estimate the parameters for *non-linear* projectors recovering their non-linear distortion parameters, and their intrinsic and extrinsic parameters as well. Recovering intrinsic parameters of a projector are harder than a camera due to the presence of an offset parameter to account for offset projection. This parameter is usually assumed to be zero when recovering camera intrinsic parameters. Further, the difficulty of estimating projector parameters has been well-acknowledged and attributed to the assumption that projectors are linear devices when in reality they, especially commodity ones [79], are usually non-linear. Our method achieves this successfully in a multi-projector system.
3. The novelty of our technique lies in using multiple levels of cross validation across multiple devices and surface geometry to recover the numerous system parameters robustly. Therefore, existence of a large number of devices, which is often viewed as a challenge for any multi-projector system, is exploited effectively to achieve high robustness and accuracy. However, the estimated device parameters may be different than the parameters recovered when calibrating each device separately and independently using physical fiducials. Instead, the estimated parameters provide a consistent model of the entire system that inherently assures scalability. Further, it allows interactive projector movements via tracking of features embedded in the content via the same cameras observing the system (chapter 5). Simultaneously achieving robustness, inter-

activity accuracy, and scalability when using multiple uncalibrated devices on unknown 3D shapes has been an elusive goal for all existing works on multi-projector displays and we address this effectively in this chapter.

2.1 Related work

This chapter focuses on geometric registration and not color registration (chapter 4) that deals with spatial variation in chroma and brightness across the display [54, 88]. We compare the proposed method with prior work from three different perspective.

2.1.1 Registration vs. Parameter Estimation

Geometric registration of multi-projector system means, for every pixel of a projector, finding the set of pixels from other projectors overlapping with it on the display surface. This set is non-empty only for pixels in the overlapping region of more than one projector’s image on the display surface. Geometric registration can be performed using an observing camera without a full fledged estimation of the intrinsic and extrinsic projector/camera parameters and reconstruction of the 3D surface geometry. Therefore, estimation of all system parameters provides geometric registration as a by-product, but not vice versa.

Only a full estimation of system parameters can allow us to harness the complete potential of two-pass rendering [76] when interfacing with 3D content delivery on multi-projector displays. The estimated projector intrinsic and extrinsic parameters are crucial to compute precisely the images to be projected from each projector.

Closely related to the aforementioned issue of registration is the kind of content mapping that can be provided. View dependent mapping maps the content on the display surface from particular viewpoint. Therefore, view-dependent mapping is only correct for a single static vantage point and may result in visible distortions from other viewpoints. View independent mapping, on the other hand, provides a mapping of the content on the surface that is suitable for multi-user viewing (e.g. wallpapering via conformal mapping).

Geometric registration (instead of a calibration) is sufficient for a view-dependent mapping from a single vantage point. The camera used for registration is placed at the vantage point and the image to be viewed by the viewer is considered to be in the camera's coordinate system. Mapping the image seamlessly from this viewpoint on to the display then entails just picking the content for a pixel of any projector from the registered camera pixel location. But, for view-independent mapping the shape of the surface needs to be recovered or known, which in turn requires recovering or knowing the device parameters of the uncalibrated projectors or cameras seeing the surface. The same is required for creating view-dependent images from multiple viewpoints as is necessary in a system where the position of a tracked user changes dynamically. Computing these parameters require estimating all the parameters of the multi projector system.

Comparison with Prior Work: Most prior work propose geometric registration and not full fledged estimation of all the system parameters. Since *planar surfaces* enable relating (linear) projectors via homography, most prior works that use multiple projectors on a flat surface essentially achieve a geometric registration [75, 74, 112, 7]. [6] allows projector non-linearities while still achieving only geometric registration. Finally, [19] introduces a distributed radially cascaded method that breaks away from

tree-based approach to significantly increase the accuracy, scalability and robustness of registering multiple projectors on a planar display.

When considering *non-planar surfaces*, a large number of methods try to avoid the problem of recovering the surface geometry explicitly thereby achieving only geometric registration [59]. Other works try to use fiducials on cylindrical surface and create a piecewise linear 2D display parametrization in the camera space to achieve the registration [36, 102]. This allows a wall-papering on the 3D surface whose quality is completely dependent on the density of the piecewise linear parametrization dictated by the number of fiducials used. Though these systems are motivated by the objective of generating a simple system with minimal use of cameras, they introduce perceivable distortions when viewed from a different vantage point than the camera, the distortions being more perceivable as the distance between the camera position and the view position increases. Further, only view dependent mapping from a single vantage point can be achieved.

A body of work [89, 91, 93, 87, 92] shows that for a class of non-planar shapes whose category and approximate measurements are known apriori (e.g. vertically extruded surfaces whose height and aspect ratio can be measured, domes whose radius can be measured), the simplicity of a single uncalibrated camera can be retained (without using stereo cameras) and the geometry of the display surface can be recovered at the same time. This 3D geometry recovery enables both view-independent content mapping and view-dependent content mapping from any view point. Later works break free from single camera and use stereo cameras for reconstructing the geometry of special kind of surfaces called quadric surfaces (e.g. spheres, cylinders, ellipsoids and paraboloids) [77], or a corner of a room [41]. However, since projection parameters are not estimated in any of the aforementioned works, two-pass rendering and easy

reconfiguration is still not possible.

Our work is closest to multi-projector systems on arbitrary non-planar shapes that are observed by multiple cameras. One of the earliest works in this direction [79] uses tightly coupled centralized method to achieve geometric registration using custom-built 3D fiducials (e.g. checkered cubes) and has the following steps. (a) It is assumed that every point on the surface is seen by at least two cameras. These cameras are calibrated using the 3D fiducial. (b) Structure light patterns from the projectors are then used by each calibrated camera to reconstruct a part of the display surface. (c) These partial reconstructions are then stitched together with respect of one camera’s coordinate system to create the entire 3D geometry. (d) The known 3D geometry and the correspondences recovered via structured light is then used to achieve the multiple projector registration. Since no cross-validation is applied, the model created by the independently estimated device parameters is not consistent with each other or the recovered 3D surface resulting in severe accumulation of error across different devices and poor accuracy of registration. Most recently, [94] uses a technique similar to [79] while surface reconstruction is improved via a centralized technique using cross-validation across multiple cameras yielding a more accurate reconstruction. More recently, [2, 1, 3, 45] have developed similar centralized 3D reconstruction method to achieve registration on complex 3D shapes, but without using any fiducials. To constrain the system sufficiently, these methods use *completely superimposed projectors* and validate the results from photometric and geometric stereo, resulting in self-calibrating systems. However, note that all these aforementioned works merely address geometric registration and does not strive to estimate the projector intrinsic and extrinsic properties, the camera parameters and the surface geometry in a consistent manner.

It has been identified that commodity projectors cannot be modeled as a dual of a linear

camera [81]. Some prior work achieve projector calibration in small single projector-camera systems [30, 97] and most of these works use planar or 3D calibration patterns [100, 48, 73, 47, 49, 103, 69, 82, 50, 67]. [90] is the only work in the multi-projector display domain that addresses the projector calibration issue, but only when projecting on a cylindrical geometry while assuming a linear projector model. Therefore, [90] cannot be used for varied surface geometries. [83] estimates the parameters of multiple projectors using a planar sheet used as calibration target and this calibration target has to be placed at differentnt position and orientations during the calibration process. [31] estimates the extrinsic parameters of multiple projectors by having the intrinsic parameters of all projectors and cameras.

Most prior work focus on specific configurations or setups [79, 116, 90, 77] and fail to provide a method that can scale to a general system of m projectors, n cameras where precalibrating each of the $m+n$ devices is cumbersome and difficult. Unlike any earlier systems, we demonstrate the scalability of our system to a large number of devices and very general display shapes. Since we recover the unknown display shape and all the parameters of the uncalibrated devices (including non-linear distortion parameter for projectors) together, we can create very large multi-projector systems easily.

2.1.2 One-time vs. Continuous Registration

Both geometric registration and calibration require pixel correspondences between projectors and cameras that are obtained by projecting a sequence of structured light patterns from the projectors while capturing them with the cameras. Therefore, any change in the system such as a projector movement or a change in the shape of the 3D surface requires computing new correspondences by either (a) disrupting content

projection by reprojection and recapture of the structured light patterns; or (b) using the features embedded in the content thereby causing no disruption.

Comparison with Prior Work: All the aforementioned works so far have focused on one-time calibration which is usually repeated every time anything changes in the system. Continuous calibration has been addressed so far only on smaller *single projector* systems. Yang and Welch [113] assumes a single calibrated projector-camera pair and monitor the projected content (as opposed to special structured light patterns) continuously to estimate the change of the shape of the surface (e.g. a moving cloth) and compensate the projected image to account for it. Using a projector augmented by two stereo cameras, Cotting et al. [24, 25, 40] embeds imperceptible structured light patterns in the projected imagery to estimate the shape of the display surface and the pose of a single projector continuously. Zhou et al. [116] achieves the same by tracking displayed image features. Johnson et al. [42] shows that multiple such single-projector-two-camera units can be used in a distributed framework to achieve continuous calibration in a multi-projector setup if the geometry is known. Zollman et al. [118] uses optical flow techniques to present a hybrid technique that can compensate for small changes in display configuration for small changes in optical flow and resort to active structured light projection only when optical flow results are unreliable due to large changes in the display configuration. Their system is designed for correcting the image from a view of a moving observer, but the single projector is static. In our system we show the result of moving the projector itself in a multi-projector system.

Table 2.1: Comparison of our work with previous works

| Method | Projector Calibration | Surface Type | Camera requirement | Scalability | accuracy | Physical Pattern |
|----------------------------------|-----------------------|--------------------------------------|----------------------------|-------------|-------------|------------------|
| Raskar et. al. 1999[79] | ✓ | Arbitrary | 2/point | ✗ | multi-pixel | Yes |
| Raskar et. al. 2003[80] | ✗ | Quadric | 1 camera and 1 tilt sensor | ✗ | multi-pixel | No |
| Harville et. al. 2006[36] | ✗ | Developable | 1 | ✓ | sub-pixel | No |
| Zhou et. al. 2008[116] | ✓ | Arbitrary | 1/projector | ✗ | multi-pixel | Yes |
| Sun et. al. 2008[102] | ✗ | Cylindrical | 1 | ✓ | sub-pixel | Yes |
| Johnson et. al. 2009[42] | ✗ | Known | 2/projector | ✓ | sub-pixel | No |
| Sajadi et. al. 2009,2010[89, 91] | ✗ | Smooth vertically extruded | 1 | ✓ | sub-pixel | No |
| Sajadi et. al. 2010[87] | ✗ | Swept | 1 | ✓ | sub-pixel | No |
| Sajadi et. al. 2011[93] | ✗ | Domes | 1 | ✓ | sub-pixel | No |
| Sajadi et. al. 2010,2011[90, 92] | ✓ | Piecewise smooth vertically extruded | 1 | ✓ | sub-pixel | No |
| Sajadi et. al. 2015[94] | ✗ | Arbitrary | 2/point | ✓ | sub-pixel | No |
| Our method | ✓ | Arbitrary | 1/point | ✓ | sub-pixel | No |

2.1.3 Scalability, Accuracy and Robustness

When considering multi-projector systems, effectiveness and ease in deployment of a method is crucially dependent on the accuracy, scalability and robustness of the method. This is where we make the most significant contribution.

Accuracy: Unlike other computer vision systems that reconstruct 3D geometries for visualization purposes, the accuracy requirement of 3D reconstruction for the purpose of projecting on top of it from multiple projectors is much more challenging. This is because even a single pixel error in the reconstructed surface or the intrinsic and extrinsic device parameters manifests themselves as highly visible mis-registrations in the displayed imagery.

Scalability: Scalability of a method is the ability to produce accurate results on a large number of devices, large size of display and large variability of display shapes.

Robustness: Robustness refers to the ability of a system to deliver accurate results *consistently* over time in face of changes in number of devices and shape of the display.

Comparison with Prior Work: Several complex procedures, most often involving multiple fiducials on the display surface, are used to achieve camera calibration and 3D display reconstruction [79, 24, 25, 40, 116, 42, 36, 102]. However, these methods usually demonstrate their registration result for relatively simpler system configurations avoiding intersection of more than two projectors in the overlap area or a cycle formed via a relationship of overlap between the projectors that can cause inconsistencies (e.g. the concatenated transformations across a loop is not identity) and therefore large accumulated errors. Some prior work [94] cross validate the 3D reconstruction across multiple cameras thereby increasing the robustness of the surface reconstruction process without addressing projector parameter estimation at all. In contrast, our method leverages multiple rounds of cross-validation in a distributed fashion to achieve a hitherto unseen robustness. All prior systems operate under the constraint that every point on the arbitrary 3D shape is observed by at least two cameras to allow the global centralized multi-view reconstruction of 3D shape [94]. Further, these methods only recover the mapping between the projector space and the 3D surface instead of recovering the intrinsic and extrinsic parameters of the projector. Therefore, recalibration, if the projector is moved, requires running the whole process of 3D reconstruction and projector registration again even if the projection surface is unchanged and only projectors are moved. Unlike [79, 77, 2, 1, 41, 24], our method works as long as a display surface point is observed by at least one camera, making our system unconstrained, scalable, easy to deploy, and robust.

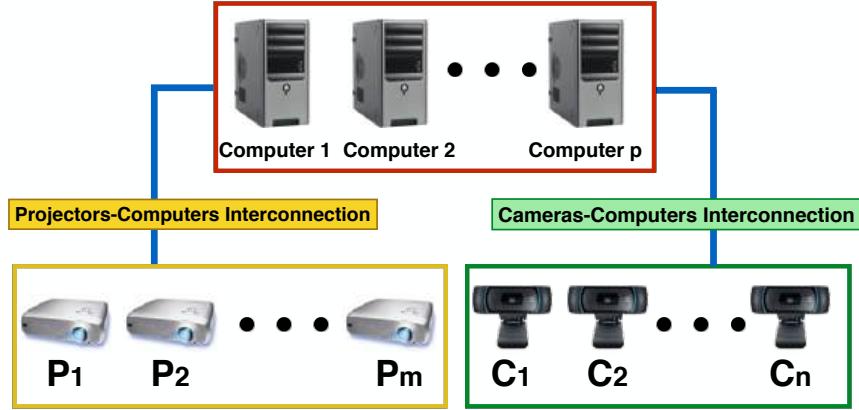


Figure 2.2: The system with m cameras and n projectors connected to the p computers via two independent interconnection networks.

2.2 System Overview

Our system consists of m projectors tiled or superimposed to illuminate an unknown 3D surface S observed by n cameras connected in a distributed network (Figure 2.2). P_i denotes projector i , $1 \leq i \leq m$ and C_j denotes camera j , $2 \leq j \leq n$. The m projectors and n cameras are connected to the computers via interconnection network I_p and I_c respectively. They illuminate an arbitrary shaped surface S whose 3D coordinates are denoted by (X, Y, Z) . We assume that every point on S is illuminated by at least one projector and observed by *at least one* camera and only a small portion of the projection area of each projector (at least 0.2 of projection area) has to be seen by at least two cameras. One common configuration that easily enforces this constraint is pairing every projector with a camera (i.e. $m = n$) assuring a baseline between them. Since the camera field of view (FOV) is usually wider than a projector, this assures that every point on S is seen by at least one camera as long as every point on S is illuminated by at least one projector. However, since today's camera FOV are usually much larger than that of projectors', it is fairly easy to set up a system where $n < m$. Also, note that we do not impose any constraint against completely

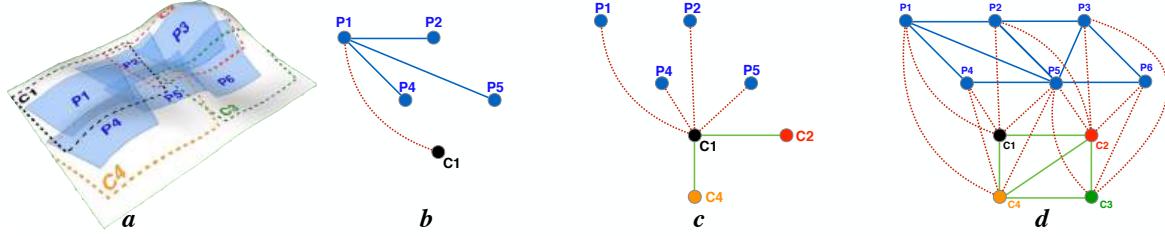


Figure 2.3: (a): The configuration of a system with 6 projectors 4 cameras. Each blue region shows the projection area of a projector and dashed colored lines show the FOV of the cameras. (b and c): local adjacency graph graph for P_1 and C_1 constructed by C_1 . The edges of partial A_p and A_c are shown in blue and green respectively. The cross edges are shown with red dashed lines. The nodes of the A_c are color coded to be consistent with the camera field of views shown in (a). (d): The connectivity graph between projectors and cameras. Note that many regions of the surface is seen by only one camera. For example, most of P_3 is seen only by C_2 . Similarly, most of P_1 is seen only by C_1 .

superimposing projectors. Figure 2.3 shows a system with $m=6$, $n=4$, that satisfies these assumptions.

We assume that we know the focal length f_j of camera C_j . In practice, this is achieved by procuring f_j from the image meta-data. This is a standard practice today with almost any computer vision technique [90, 58, 26]. Further, as in [99, 90], we assume the principal center of the camera to be coincident with the center of the image plane, no skew between the image axes, and square pixels. Such assumptions are practical in almost all commodity cameras. Therefore, the intrinsic matrix K_j^c for camera j is given by

$$K_j^c = \begin{pmatrix} f_j & 0 & 0 \\ 0 & f_j & 0 \\ 0 & 0 & 1 \end{pmatrix}, \quad (2.1)$$

where f_j is the focal length of the camera C_j in pixels.

Since most practical projectors come with small non-linearities [64, 114] we use a non-linear model for the projector to account for the radial and tangential distortions that

is given by combining a non-linear model with a linear model. Let us assume E_i is the extrinsic matrix for projector P_i . Then the normalized pinhole projection of 3D point (X, Y, Z) , using a linear camera model, is given by $(x_i, y_i) = (\frac{X'}{Z'}, \frac{Y'}{Z'})$ where

$$\begin{pmatrix} X' \\ Y' \\ Z' \end{pmatrix} = E_i \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix}, \quad (2.2)$$

Augmenting the non-linear model to the above equation to account for the nonlinearities, we get

$$\begin{pmatrix} x'_i \\ y'_i \end{pmatrix} = (1 + \alpha_i^1 r_i^2 + \alpha_i^2 r_i^4 + \alpha_i^3 r_i^6) \begin{pmatrix} x_i \\ y_i \end{pmatrix} + \begin{pmatrix} 2\alpha_i^4 x_i y_i + \alpha_i^5 (r_i^2 + 2x_i^2) \\ \alpha_i^4 (r_i^2 + 2y_i^2) + 2\alpha_i^5 x_i y_i \end{pmatrix} \quad (2.3)$$

where α_i^1 , α_i^2 and α_i^3 are radial distortion coefficients and α_i^4 and α_i^5 are tangential distortion coefficients for projector P_i and $r_i = x_i^2 + y_i^2$. Therefore, the final coordinate of the 3D point in projector's image plane is given by $(p_i, q_i) = (\frac{x''_i}{z''_i}, \frac{y''_i}{z''_i})$ where

$$\begin{pmatrix} x''_i \\ y''_i \\ z''_i \end{pmatrix} = K_i^p \begin{pmatrix} x'_i \\ y'_i \\ 1 \end{pmatrix}, \quad (2.4)$$

where K_i^p is the intrinsic parameter matrix of the projector P_i given by

$$K_i^p = \begin{pmatrix} f_i^p & 0 & u_i \\ 0 & f_i^p & v_i \\ 0 & 0 & 1 \end{pmatrix}, \quad (2.5)$$

where f_i^p , v_i and u_i are the focal length, vertical and horizontal offset in pixels respectively. These parameters are required to model the offset projection feature available in most commodity projectors.

Our method comprises of an *offline calibration* followed by an *online image correction* (Figure 2.4). The offline calibration recovers display surface and geometric parameters of each device (projector and camera) which are used to create a displacement map for each projector. These displacement maps are then used in the online image correction phase (Section 2.4) to warp the geometry of the images projected by each projector in real-time. For color calibration, the parameters recovered in the offline calibration are used to create per projector attenuation maps that are applied after the displacement map in the online image correction phase to achieve simple edge blending [79] in the overlap regions to smooth color transition between the projectors. The result is seamlessly registered imagery on an arbitrarily shaped surface S .

2.3 Offline Calibration Method

Our auto-calibration offers a distributed method that comprises of the following four steps:

1. *Connectivity graph construction* finds the connectivity between devices in the context of their overlapping field-of-view,
2. *Partial 3D reconstruction and camera calibration* step recovers the camera parameters and the shape S of the surface partially in regions that are seen by more than one camera,

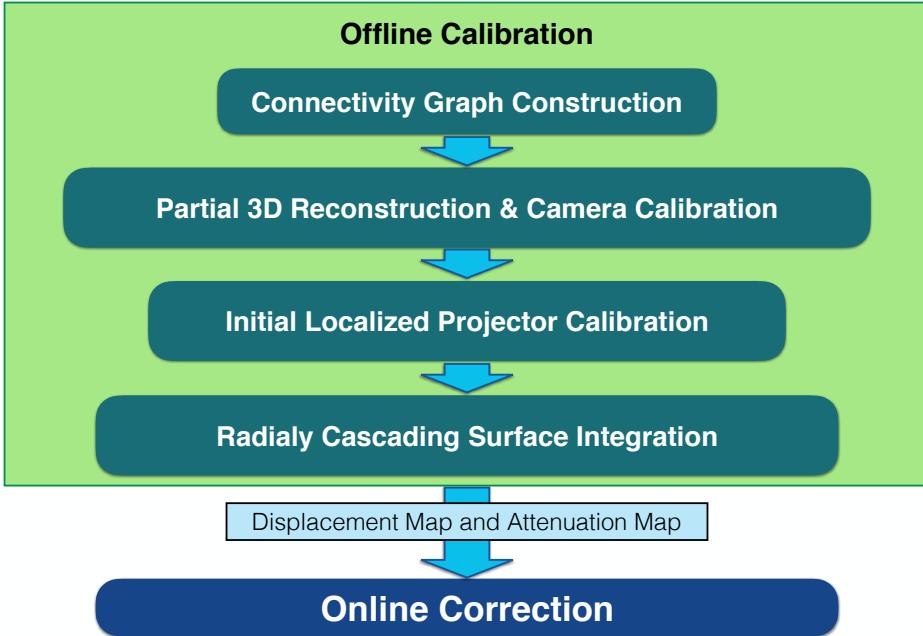


Figure 2.4: This figure shows the pipeline of our method. Our method has two components. First an offline calibration algorithm provides the displacement and attenuation map, then our online correction algorithm uses these maps to create a seamless imagery on the arbitrary shape.

3. *Initial localized projector calibration* achieves initial estimation of the linear and non-linear projector parameters using the partially-reconstructed display surface S and the camera parameters,
4. *Radially cascading surface integration* step refines the recovered projector parameters and integrate the 3D reconstruction of parts of S .

The above four steps yield complete reconstruction of S and an accurate calibration of all the devices.

2.3.1 Connectivity Graph Construction

Connectivity graph provides information about the overlap between the camera and projector field of views (FOV) and we recover this graph automatically. Many prior work [7, 85] that achieve projector calibration in multi-projector displays assume that the projectors and cameras form a regular grid. We avoid any such assumptions that are difficult to ensure in practice. However, we do assume that the connectivity graph is static and does not change during the calibration process.

The connectivity graph, A , is an undirected graph consisting of two sub-graphs: a *camera adjacency graph*, A_c , and a *projector adjacency graph*, A_p , with cross-edges between these two sub-graphs. Each node in A_c indicates a camera and an edge between u and v , $u, v \in j$, denotes that the FOV of C_u covers more than a fraction α , $0 < \alpha < 1$, of the FOV of C_v and vice versa. Typically, we set $\alpha = 0.2$ (a 20% overlap between cameras) to assure robust use of structure from motion (SfM) techniques, well known computer vision technique used for this purpose [71, 65]. Similarly, each projector is represented by a node in A_p and an edge between e and f , $e, f \in i$, denotes an overlap between P_e and P_f . Finally, we define *cross edges* between A_p and A_c . A node u in A_c is connected to a node e in A_p , $u \in j$ and $e \in i$, if C_u sees whole or part of P_e . Figure 2.3 shows an example of the connectivity graph of a system with six projectors and four cameras.

To construct A , each projector in the sorted order of their IP address projects white and the scene captured by all cameras after each projection.

Each camera C_j constructs the local adjacency around itself and the adjacency around each projector it observes using all the captured images (Figure 2.3 (b and c) shows the local adjacency graph for P_1 and C_1 constructed by images captured by C_1). Next,

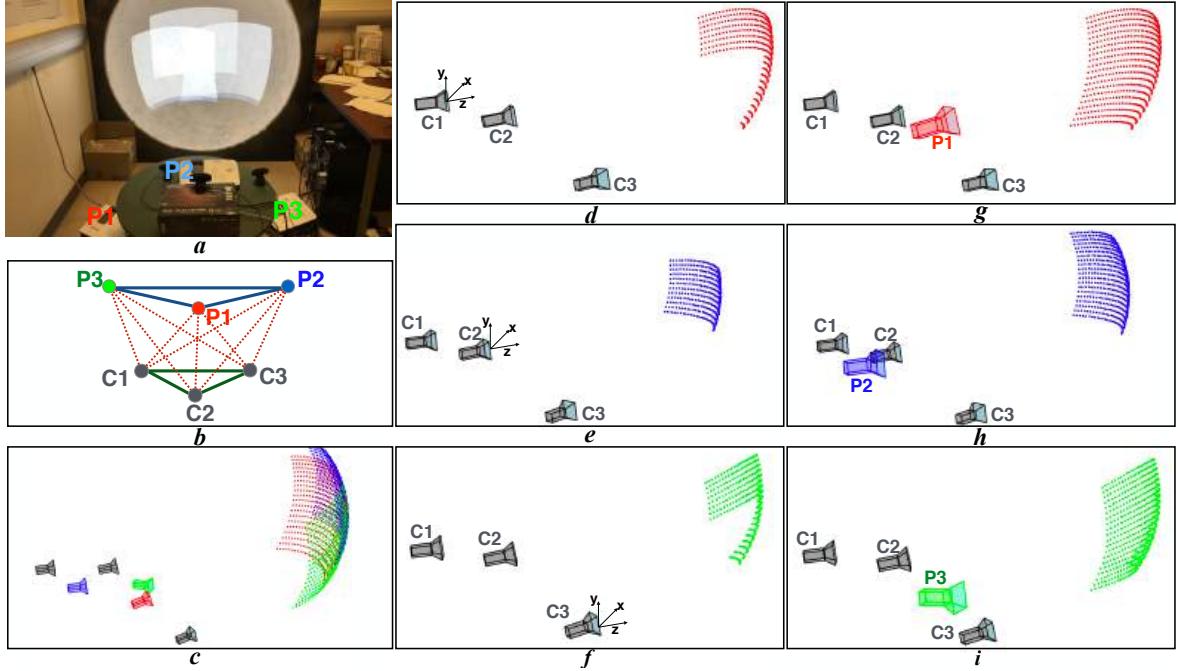


Figure 2.5: This figure shows steps of our algorithm. (a): Our setup with 3 projectors and 3 cameras on a dome. (b): Adjacency graph for projectors and cameras. (c,d and e): Camera calibration and partial 3D reconstruction for 3 projectors after Step 2. (f): Partially reconstructed display surface of P_1 , P_2 and P_3 in red, blue and green respectively after Step 2. Each of these 3D reconstructions are in different coordinate systems with different rotation, translation and scale factors. (g) Complete 3D reconstruction of the display surface of P_1 , P_2 and P_3 in red, blue and green respectively after step 3. (h): The 3D reconstruction of the display surface, and the calibrated projectors after Step 4 of radially cascading refinement.

the camera broadcasts the part of A it has constructed to all the projectors. Each projector consolidates the different parts of A it receives from all cameras to construct the complete adjacency information. Figure 2.3 (d) shows an example of the final connectivity graph between projectors and cameras.

2.3.2 Partial 3D Reconstruction and Camera Calibration

The goal of this step is to reconstruct the geometry of the part of S that is seen by two or more cameras and also recover the camera extrinsic properties. This is therefore

a SPMD (single program multiple data) process conducted on each projector P_i in parallel in a distributed manner.

The process proceeds in K iterations ($K \leq m$). In the k th iteration, $k < K$, a set of *non-overlapping* projectors, W_k , that forms a subgraph of A_p , project a set of binary blob patterns and all the cameras adjacent to W_k in A , capture the images. The exact value of K depends on the connectivity of sub-graph A_p . Densely connected graphs, indicating larger connectivity in A_p , results in higher values of K . In the worst case, $K = m$, where each projector projects the patterns sequentially.

The set of sets of non-overlapping projectors is computed using a greedy algorithm. All the projectors are tagged UNMARKED in the beginning of the algorithm. To choose the set of projectors W_k in iteration k , W_k is initialized to null set and the UNMARKED projector with the lowest ID that has no connectivity with W_k grabs the token, includes itself in W_k , changes its status to MARKED and then releases the token. This continues till all the UNMARKED projectors have a adjacent projector in W_k . In iteration k , all the projectors in the set W_k project binary blob patterns to use for 3D reconstruction. The above process continues through next iteration until all projectors are MARKED. These blobs are used to find correspondences across multiple devices.

The captured images of the patterns projected by each projector P_i give us a set of correspondences between P_i and its adjacent cameras which are then processed as described in Algorithm 1 to reconstruct the camera extrinsic parameters and regions of S seen by at least two cameras. The parts of S seen by only one camera will not be reconstructed in this step.

Using the mentioned steps in Algorithm 1, we have now reconstructed the part of S

Algorithm 1 Partial 3D Reconstruction

```

1: procedure PARTIAL 3D RECONSTRUCTION
2:    $G \leftarrow$  Set of cameras in  $A_c$  adjacent to  $P_i$  in  $A_p$  connected via the cross edges.
3:   Designate  $C^R \in G$  with the maximum connection with other cameras in  $G$  as reference camera for  $P_i$ .
4:    $D \leftarrow \{C^R\}$ 
5:    $G \leftarrow G - \{C^R\}$ 
6:   while  $G$  is not empty do
7:     Select  $C_i \in G$  with the maximum connectivity with  $D$ .
8:      $D \leftarrow D \cup \{C_i\}$ 
9:      $G \leftarrow G - \{C_i\}$ 
10:    Find  $C_u \in D$ ,  $u \neq i$ , that has overlap with  $C_i$ .
11:    Use correspondences between  $C_i$  and  $C_u$  in a structure from motion (SfM) method to (1) reconstruct the extrinsic parameters of  $C_i$  and  $C_u$ ; and (2) the 3D geometry of the surface seen by both  $C_i$  and  $C_u$ .
12:     $C_i^a \leftarrow$  set of all cameras in  $D$  that are adjacent to  $C_i$  in  $A_c$ .
13:    Perform a bundle adjustment optimization using all cameras in  $C_i^a$  refining their extrinsic parameters.
14:    Refine the 3D reconstruction of part of  $S$  seen in overlapping FOVs of cameras in  $C_i^a$  using all the correspondences from all pairs of cameras.

```

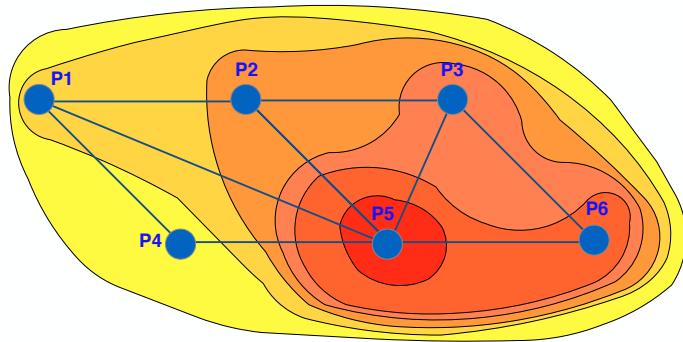


Figure 2.6: This figure shows the order that projectors added to set V from our previous example in Figure 2.3. The gradient of the color from red to yellow shows the order of the projectors - $P_5, P_6, P_3, P_2, P_1, P_4$.

illuminated by P_i that is seen by two or more cameras which are also calibrated during the process. We denote this part of the surface as S_{P_i} . However, this reconstructed region is in the local coordinate system defined by C^R . Figure 2.5(d,e and f) show partial 3D reconstruction of surface for three projectors. Notice that each of these figures is in local coordinate system based on the reference camera. For example, in Figure 2.5 '(d), (e) and (f), C_1 , C_2 and C_3 are the reference cameras respectively.

2.3.3 Initial Localized Projector Calibration

The result of the previous step leads to a set of correspondences between 2D projector coordinates of P_i , (x_i, y_i) , and the 3D coordinates of S_{P_i} , denoted by (X_i, Y_i, Z_i) . From these correspondences we solve a non-linear optimization for each P_i using maximum likelihood estimation via Levenberg Marquardt algorithm [63] to estimate K_i^p , E_i and α_i^k for $k = 1, 2 \dots 5$ and thus calibrate each projector P_i .

When achieving projector calibration, if all the points lie on a single plane, it will be a degenerate case for our method. Therefore, though we can handle partial planar region (as long as it does not cover the entire field of view of a projector), for a completely planar displays a large number of prior homography based methods can be used [85, 20, 18, 8, 15].

Following the projector calibration, for every surface point of S illuminated by P_i , we can find the correspondences in one or more calibrated cameras. Therefore, we can perform a multi-view 3D triangulation to find the 3D position of the center of each blob projected by P_i . To undo the non-linear distortion due to the projector, we use the recovered α_i^k , $k = 1, 2 \dots 5$, to remove the projector non-linearities before applying the multi-view 3D triangulation. Since this step is also performed locally by each P_i , we

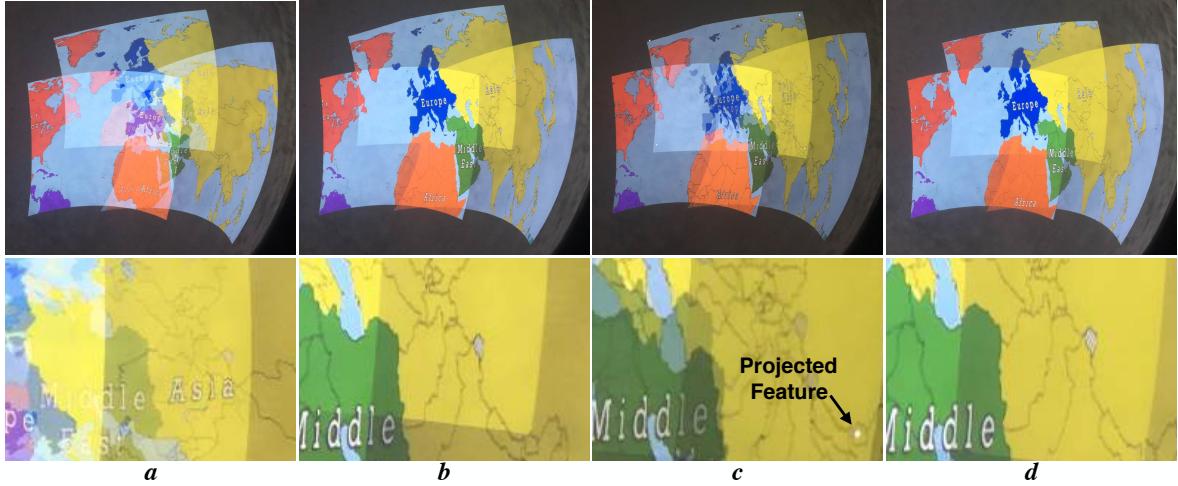


Figure 2.7: (a): Projected image with 3 projectors before correction. (b): Geometrically registered image with these 3 projectors. (c): Distorted image after moving a projector. (d): Corrected image after projector recalibration.

recover the surface geometry illuminated by each P_i in its own local coordinate system as seen in Figure 2.5(g,h and i). The 3D reconstruction of the surface illuminated by each projector is shown with different colors. However, these reconstructed regions are not connected to each other since each of them are still based on local coordinate system of each projector.

2.3.4 Radially Cascading Surface Integration

Finally, we integrate the reconstructed parts of S from all the different projectors with respect to one reference projector using a *radially cascading* mechanism inspired by [85] as described in Algorithm 2. We set the projector with maximum number of overlaps with other projectors as the reference projector and denote it as P_r . We denote the set of devices that are integrated with each other as V . Therefore, at this point $V = \{P_r\}$. Then at each step we choose a device (projector or camera) D_s as

described in Algorithm 2 and merge the display formed by the device in V and D_s . Figure 2.6 illustrates the order of the projectors added to the set V for an example graph A_p . In every step, the display formed by the devices in V is registered.

For integrating D_s with V we consider all the overlaps of D_s with other devices in V . Then we solve a non-linear optimization problem to find the scale factor γ and refine the extrinsic parameters of D_s to align reconstructed 3D points in the overlap areas of D_s and V . The objective function Err_s minimizes the sum of the distances of the all corresponding 3D points M_i and M'_i in the overlap of D_s and V (M_i and M'_i are 3D coordinates of point i in coordinate system P_s and V) is given by

$$Err_s = \min \sum_{i=1}^N |M_i - (\gamma R_s M'_i + T_s)|^2, \quad (2.6)$$

where R_s and T_s are the rotation and translation matrices of D_s , respectively and N is the number of corresponding points in overlap of D_s and V . We use Procrustes analysis described in [44, 11] to solve this problem.

The above step assures that D_s is connected to the display formed by all the devices in V . However, there is no guarantee that the overlap regions between D_s and all the other devices are represented consistently and similarly across all these devices. Therefore, we consider all the projectors and cameras that overlap with D_s and reconstruct the display surface across these devices to form a consistent display surface geometry. This assures that the overlap regions of multiple projectors in V are represented consistently. Figure 2.5(c) shows the integration of the display geometry reconstructed by all the projectors.

At the end of the Algorithm 2, V is the set of all projectors and we have identical 3D representation of display surface along with the estimation of parameters for all

Algorithm 2 Radialy Cascading Surface Integration

```
1: procedure RADIALY CASCADING SURFACE INTEGRATION
2:    $P_r \leftarrow$  The projector with the maximum degree in  $A_p$  (i.e. the projector with
   most overlaps with other projectors)
3:   The coordinate system of  $P_r$  is set as the global reference coordinate system.
4:    $V \leftarrow \{P_r\}$ 
5:    $H \leftarrow \{\text{All Devices}\} - V$ 
6:   while  $H$  is not empty. do
7:      $D_s \leftarrow$  The device in  $H$  with maximum number of edges to devices in  $V$ .
8:     Integrate 3D reconstructed points of  $D_s$  with  $V$ .
9:     Refine reconstructed geometry
10:     $V \leftarrow V \cup \{D_s\}$ 
11:     $H \leftarrow H - \{D_s\}$ 
```

the cameras and projectors assuring perfect and consistent registration. Note that since there is no metric scale during the calibration process, the final estimation of device parameters and 3D point coordinates are up to scale. However, much more importantly, the scale factor is consistent across all the m projectors and n cameras. For color correction, we apply simple edge blending described in [79]. The blending is encoded as a per pixel per channel attenuation map for each projector.

2.4 Online Image Correction

The one-time calibration step generates a per-pixel displacement map and a per-pixel attenuation map for each projector. It also defines the specific part I_i of the global image that should be cut out for each projector P_i . The displacement map defines how every pixel needs to be displaced to achieve geometric registration while the attenuation map defines how each pixel should be attenuated to achieve edge blending.

In the online image correction step, we cut out the appropriate image I_i for each projector P_i , warp I_i using the displacement map and blend them using the attenuation

map. These cut, warped and blended images when projected together from multiple projectors create a seamless image on the display surface. This correction is achieved in real-time using the GPU fragment shaders as in [88, 6, 89].

As we discussed before, we can achieve *view-independent* or *view-dependent* geometric registration of projected content. Since we reconstruct the shape of S , the image can be wallpapered on S resulting in a *view-independent* correction for multi-user applications. A simple example is indicating the height of the points in a relief map using different colors. Another example is pasting different map layers over the surface, e.g. satellite imagery, road map, or traffic map. These examples are demonstrated in Figure 2.8 and Figure 2.1. In *view-dependent* applications the image can be registered from any arbitrary viewpoint of a moving user. To achieve a view-dependent registration for single user applications, we first define the virtual camera (or viewpoint) with projection matrix M . Note that this can be different from the viewpoint of any of the calibrating cameras. Next we use the standard two-pass rendering approach in [76]. First, we render the 3D scene from this virtual camera using M . To generate the image I_i for each projector, at any pixel (x_i, y_i) we find the corresponding 3D coordinates of the surface geometry at (x_i, y_i) . Then we project this 3D point using M to find the corresponding pixel from the image rendered in the first pass. We use the color at this pixel to generate the image I_i .

2.5 Continuous Registration

Our system is capable of adapting to any change in the set up such as projector movement or change in the shape of the display surface. It is practical to assume that such changes occur one at a time instead of concurrently. In this section we explain

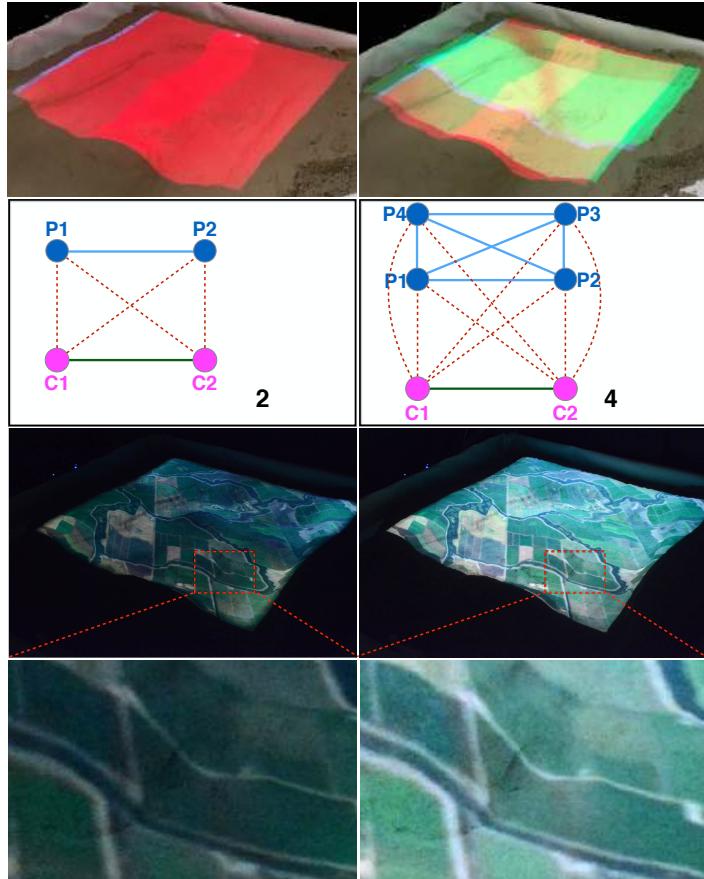


Figure 2.8: Top: Our setup with 2 projector (left) and 4 projector for super-imposed projection (right). Middle: Connectivity graph of the setups. Bottom: Result with two projectors (1x2 array) (left) and super imposed projection with 4 projectors (2x1 array on a 1x2 array) (right).

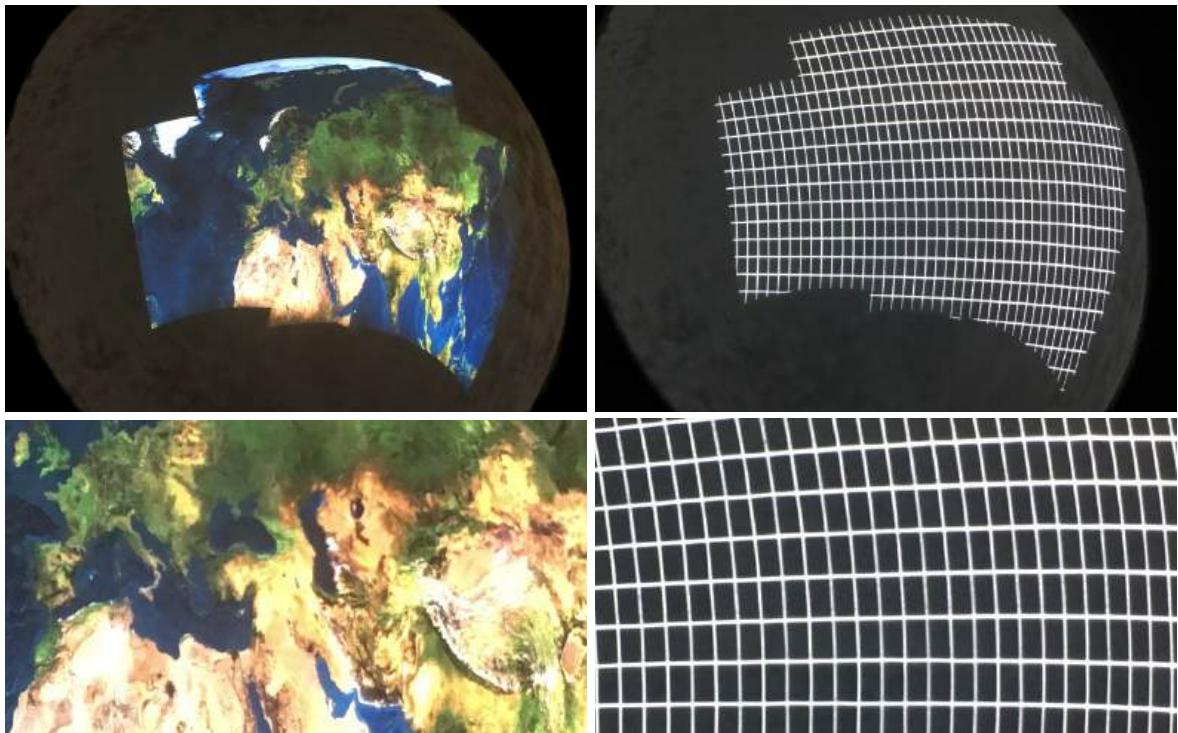


Figure 2.9: Final result of projecting on a dome with 3 projector and 3 cameras (top) and zoomed in image (bottom).

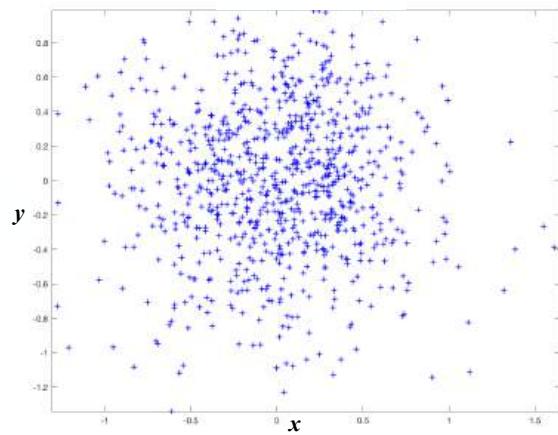


Figure 2.10: This figure shows the reprojection error (in pixel) of the 3D points on the projectors image plane in x and y direction in our simulator.

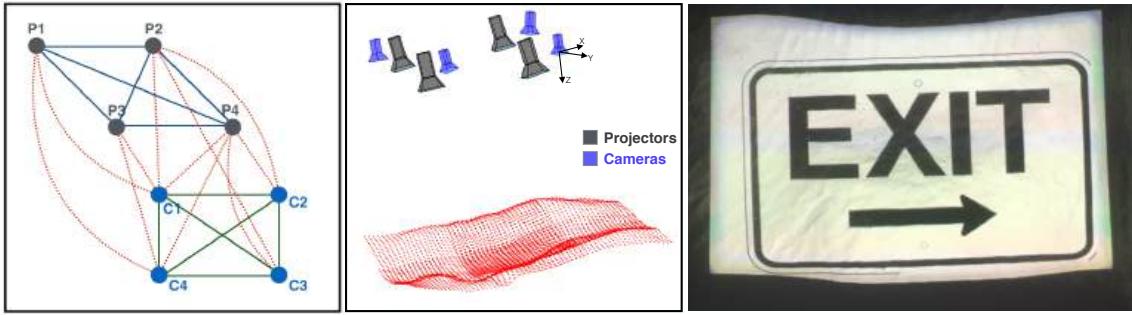


Figure 2.11: This figure shows our result projecting on a sand pit with 4 projectors and 4 cameras. Left: Connectivity graph for projectors and cameras. Middle: Recovered 3D geometry of the display surface and parameters of the projectors and cameras. Right: Final result.

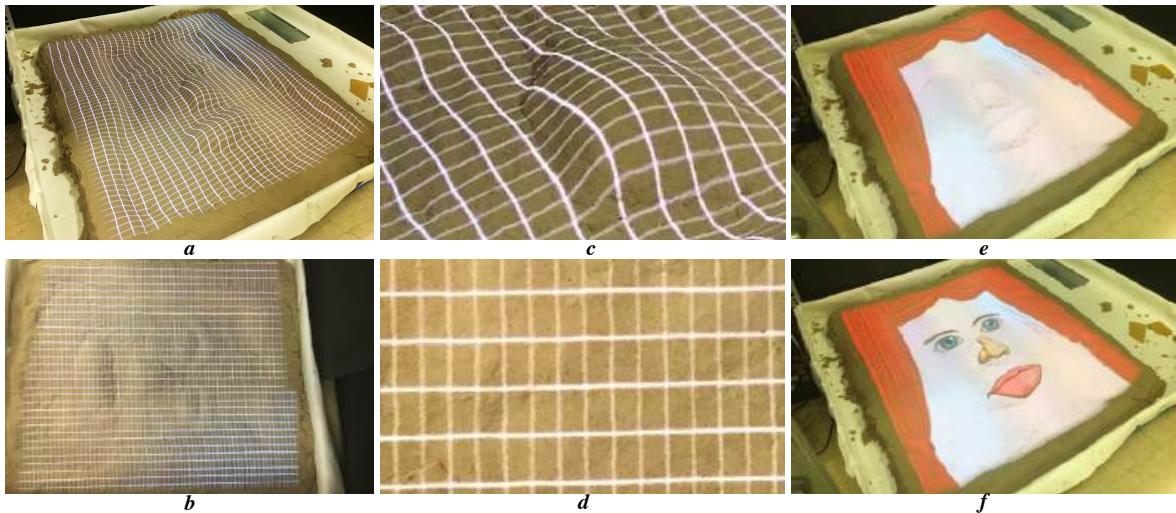


Figure 2.12: This figure shows our result projecting on a sand pit using same setup as Figure 2.1. (a and b): View-dependent projection from the incorrect view point(a) and correct view position(b). (c and d): Zoomed in image to show the accuracy of geometric registration. (e and f): Shows the result of changing the appearance of display geometry by projecting different contents.



Figure 2.13: Top: Showing the overlap of projectors for our setup with 10 projectors and 4 cameras on a cylindrical display surface. Bottom: Final result of projecting a graffiti on the cylindrical wall.

our method for recalibrating the system for the aforementioned changes in the set up.

2.5.1 Projector Movement

Unlike any prior work on multi-projector systems, since we calibrate the projector, we do not need to recalibrate the entire system when a projector moves. It is sufficient to just recalibrate the projector P_i that moved. When the projector movement happens, only its extrinsic parameter matrix E_i changes. Therefore, during the projector movement, we embed a few distinct features in the projected image (During this process the projectors can project a static image or a video). These features are seen by one or more cameras from which the corresponding 3D coordinates in S are deciphered. Using this data, we run the third step of our method (section 2.3.3) to find the new extrinsic parameters and recalibrate the projector. Assuming there is no zoom change during the projector movement, projecting only four features at the four corners of the moved projector results in an accurate and fast recalibration. Larger number of features will be required to achieve similar accuracy in the face of zoom changes which can compromise the speed of recalibration slightly. However, these can all be implemented in GPU to run in interactive rates. Further, features embedded in the content itself can be tracked using SIFT feature extraction to achieve the same recalibration without the projection of additional features.

2.5.2 Changing Display Surface Geometry

If the shape of parts of the surface changes, our system can trigger reconstruction of only the modified part of the surface. This process has two steps: (1) detection of the

changes in shape and (2) 3D reconstruction of the affected regions of the surface.

Detection: We assume that the cameras are monitoring the surface via continuous capture. Since all the device parameters, the surface geometry and the content to be projected are known, the image captured by each camera can be predicted. If the captured image differs beyond a noise dependent threshold from the expected captured image, a surface modification is detected. In other words, camera C_i detects a set of pixels U_i that do not have the expected values and thus reflect a change in the shape of the surface.

Reconstruction: Let $F_{j,i}(x_j, y_j) = (u_i, v_i)$ denote that pixel (x_j, y_j) of P_j corresponds to pixel (u_i, v_i) of camera C_i . From this correspondences, we compute the set of pixels U'_j from P_j that corresponds to the set of pixels U_i from C_i as

$$U'_j = \{(x_j, y_j) | \exists i, F_{i,j}(x_j, y_j) \in U_i\} \quad (2.7)$$

P_j then projects a sequence of structured light patterns in the area covered by U'_j to find a new set of correspondences between P_j and all the cameras. Having these new set of correspondences we can run the last step of our method (section 2.3.4) to compute the new 3D geometry of the surface.

2.6 Implementation and Results

We have implemented our method in four real systems. They come in different sizes (from large area immersive displays to table top objects), different shapes (from cylinders to domes to completely arbitrary surfaces made of sand), for different number



Figure 2.14: This figure shows our result on small objects. First row: Our setup with 6 projectors and 8 cameras in tiled configuration around the vase. The projectors illuminate the vase and other objects around it. Second row: Our final result from different view points. Third row: Showing a checker board pattern on the vase to demonstrate the accuracy of geometric registration(left) and the result of superimposed projection on the vase (right).

of devices, and for both superimposed and tiled projector configurations. This demonstrates the generality and scalability of our method.

The first system is a sand-pit of size 4'x3' lit by an array of two, four and six projectors in three different set-ups in both tiled and superimposed configuration seen by four cameras and driven by three machines i.e. $p=3$, $n=4$ and $m=2, 4$ and 6 . Figure 2.1 shows our result with six projectors and 3D reconstruction of the display surface. We show two different arbitrary geometries on the sandpit to emphasize the generality of our method. Figure 2.1 shows view-independent geometric registration and Figure 2.11 shows the view-dependent geometric registration using four projectors. Figure 2.12 shows our system with six projectors. The first column of Figure 2.12 shows the view-dependent geometric registration from two different view points and the grid shows the accuracy of our geometric registration. The Third column of Figure 2.12 shows how we can change the appearance of display surface as we know the 3D geometry of the surface. Figure 2.8 shows our result of superimposed projection to increase the brightness of the image. The sandpit is first lit by a 1x2 array of two projectors which are then superimposed by a 2x1 array of two more projectors to increase the brightness two-fold.

The second system is a hemisphere of diameter 5' lit by three projectors in a tiled configuration, seen by three cameras and run by three machines ($m=3$, $n=2$ and $p=3$). Figure 2.9 shows our result for this system. The projected grid shows the accuracy of our geometric registration in overlap of all projectors.

The third system is that of a table top vase lit by six projectors in a tiled configuration or in superimposed configuration seen by eight cameras driven by three machines. Figure 2.14(a) shows the system with six projectors in tiled configuration and part b

Table 2.2: System running time.

| | Spherical | Sandpit | Vase | Cylindrical |
|------------|------------------------|----------------------|----------------------|----------------------|
| Step 1 | 30sec | 40sec | 80sec | 100sec |
| Step 2 | 35sec | 45sec | 55sec | 70sec |
| Step 3 | 15sec | 25sec | 30sec | 45sec |
| Step 4 | 15sec | 20sec | 25sec | 35sec |
| Total Time | $\simeq 1.5\text{min}$ | $\simeq 2\text{min}$ | $\simeq 3\text{min}$ | $\simeq 4\text{min}$ |

and c shows final result of geometric registration around the vase. Figure 2.14(d) shows the result of superimposed projection on the vase.

The final system is a 90 degrees partial cylinder of 14' radius (22' wide and 6' high) lit by 5x2 array of projectors seen by four cameras and driven by five machines. Figure 2.13 shows the result for this system demonstrating the scalability of our method to a large number of projectors and cameras. Figure 2.7 shows the projected features during projector movement and the result of the geometric registration after recalibration of moving projector.

Our unoptimized Matlab implementation takes around 1.5 minutes on the sphere with three projectors and 2 minutes on the sandpit with six projectors. The time breakdown for each step is provided in Table 2.2. Re-calibration of the projectors takes approximately 10 seconds in our Matlab implementation.

We also evaluated our method in a simulator system that we developed of projectors and cameras. We used our method to calibrate all the devices in the simulator and find the 3D coordinates of the display surface. After calibration of the whole system, we projected back the reconstructed 3D coordinates of the points on each projector's image plane. Comparing the calculated coordinates with known coordinates of these points in the projector's image plane, we find that our average reprojection error is 0.44 pixel. Figure 2.10 shows the amount of the reprojection error for the 3D points on the

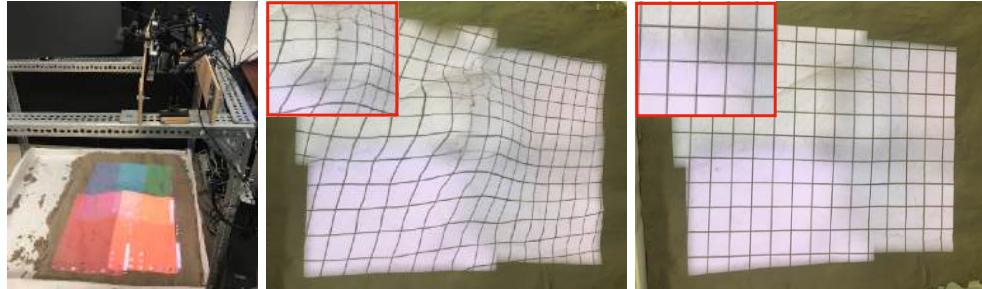


Figure 2.15: This figure shows the comparison of our method (right) with result of registration by estimating the device parameters by standard checkerboard calibration method and using this estimation for 3D reconstruction without cross-validation between devices (middle) on a display made of 4 projectors on a sandpit observed by 4 cameras (left).

Table 2.3: Comparison of the estimated projector intrinsic (in pixel) using checkboard and our system.

| | | F | Cx | Cy |
|------------|--------------|---------|--------|--------|
| Projector1 | Checkerboard | 1941.75 | 486.93 | 939.89 |
| | Ours | 1924.55 | 454.9 | 947.40 |
| Projector2 | Checkerboard | 2039.72 | 464.75 | 648.16 |
| | Ours | 1949.43 | 540.24 | 703.79 |
| Projector3 | Checkerboard | 1942.30 | 544.66 | 726.43 |
| | Ours | 1875.24 | 537.95 | 730.91 |
| Projector4 | Checkerboard | 1803.63 | 495.32 | 695.35 |
| | Ours | 1747.54 | 458.81 | 683.16 |

image plane of one of the projectors in x and y direction. We also compare the result of our projector calibration with projector calibration method using a checkerboard [64]. Table 2.3 show the comparison of estimated results. Figure 2.15 shows the result of registration by only estimating the device parameters by a checkerboard and using this estimation for 3D reconstruction without cross-validation between devices.

Chapter 3

Camera Intrinsic Calibration

In the previous chapter we explained how to accurately estimate intrinsic and extrinsic parameters of the projectors in addition to extrinsic parameters of the cameras. In this chapter we explain a practical method for fully automatic intrinsic calibration of cameras using directionally encoded light.

Accurate intrinsic camera calibration is one of the core requirements for a multitude of computer vision tasks. Although reliable methods exist to do so, most of them require substantial human interaction and a carefully configured setup to successfully compute accurate calibration parameters. Especially the accuracy of the estimated lens distortion parameters may suffer if the user is not sufficiently covering the complete area of the image sensor. This is particularly challenging for wide angle lenses where a large volume needs to be covered, or for tele-lenses where the calibration volume is at a distance.

Overcoming these issues is the primary goal of this work and therefore we propose a simple, fully automatic calibration method that estimates the intrinsic parameters of

a camera reliably without any human intervention. We present a practical calibration device that only requires a minimal working volume directly in front of the lens and the overall process can be carried out within few minutes with repeatedly reliable accuracy at the push of a button. Unlike most existing calibration methods we employ directionally encoded light which allows to calibrate additional intrinsic parameters, such as focus distance or aperture size. The proposed algorithm is not exclusively aimed for being used by scientists or engineers, but to simplify calibration for maintenance personnel, or amateurs who want to use a specific software application requiring a calibrated camera. From practical experience, achieving an acceptable camera calibration is a seriously difficult task for such non-experts.

In summary, our proposed method includes the following novel contributions:

- A reliable and fully automatic intrinsic camera calibration algorithm.
- A prototypical hardware setup which requires only a minimal working volume directly in front of the lens.
- An algorithm not requiring any system pre-calibration.
- The use of directionally encoded light to calibrate additional lens properties, such as focus distance and aperture size.
- An evaluation of the system based on various different lenses and cameras including a comparison with Zhang's method [115].

The remainder of the chapter will start with a short discussion of the related work (3.1), followed by a description of the proposed setup (3.2) and calibration method (3.3). In section 3.4 we will introduce our hardware prototype used to evaluate the system in section 3.5.

3.1 Background and Related Work

In the following we will give a short introduction about the basic camera model and the calibration method we are using, and will give an overview of the most relevant related work with respect to camera calibration as well as the proposed display configuration.

Pinhole Camera Model Our work is based on the widely used camera calibration introduced by Zhang et. al. [115] using a pinhole camera model. Assume (X, Y, Z) is the coordinate of a point in 3D space and (x, y) is the coordinate of the projection of this point on the 2D image plane of the camera sensor. Then the relation between these two coordinates is given by:

$$s \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = K \begin{bmatrix} R | t \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \quad (3.1)$$

Where s is an arbitrary scale factor. R and t are the rotation and translation matrices of the camera to the world coordinate system. K is the camera's intrinsic matrix and is defined as

$$K = \begin{bmatrix} f_x & \alpha & x_p \\ 0 & f_y & y_p \\ 0 & 0 & 1 \end{bmatrix} \quad (3.2)$$

Where f_x and f_y are the focal lengths of the camera in horizontal and vertical direction. (x_p, y_p) is the coordinate of the camera's principal point on the image plane. α is the skew between two image axes which, in our work, is assumed to be equal to zero which is a reasonable assumption for the vast majority of the existing modern cameras.

To also account for radial and tangential lens distortion, one of the most common models is used [13, 14, 23], where several coefficients k describe radial and tangential distortion depending on the distance to the system’s optical axis.

Multi-layer Displays Adding light-attenuating or -directing layers onto emissive displays has been widely explored for the generation of autostereoscopic displays for multi-view stereo [104] or near-eye displays [52]. These approaches are based on the principle of generating a spatially varying light field such that for varying observer positions a different content is presented. We apply such displays in the context of intrinsic calibration: directional light is generated by light-attenuating layers in front of a high-resolution screen, captured by the camera, and the information is used for calibration. The Bokode system [61] uses a similar optical design approach but within a different context.

Camera Calibration Methods There is a large body of research papers focusing on the task of accurately calibrating the intrinsic properties of cameras. They differ from each other by employing different optimization methods, camera models, calibration targets or user interfaces. Many of the existing methods use planar patterns for calibration and offer widely available toolboxes [12, 68] which require expert knowledge to reliably generate an accurate calibration. The method described in [115] is one of the most commonly used methods which uses multiple images of a plane captures from unknown orientations. Many of the existing works use this method as a reference to compare the accuracy of their results. Another approach was presented in [22] which uses parallel lines in the image to find vanishing points to calibrate the camera.

The performance of these methods strongly depends on the accuracy of the extraction of the correspondence points. [66] proposed a highly accurate feature detection method

for camera calibration using checker board patterns. [110] uses circular control marks to accurately find the location of features. [28] proposed a calibration algorithm for planar square, circle, and ring patterns. Here an iterative refinement approach to undistort and unproject the calibration pattern to the canonical plane is used to increase the accuracy of the localization.

Most existing works focus on the development of novel camera models and improved calibration accuracy, but the calibration process can be time consuming and partially even a manually guided process for identifying control points. There is a body of work focusing on automating these steps. Since most of these methods use planar markers, such as checkerboards, its robust detection is highly important: [10] proposed a method for automated checkerboard detection. In [29], [32], [4] and [84] self-identifying patterns for plane-based calibration are used which can also be partially occluded. The usage of random dot based markers instead of 2D tags or checkerboards is proposed in [70] which is based on the advantage that those can be detected more reliable in out of focus image regions. [51] presented an automatic technique for grid corner extraction. The paper of Ha et.al. [33] proposes a method which handles out of focused input images by displaying a series of structured light patterns on a small display close to the lens to generate display-plane-to-camera correspondences from a variety of different views. This method requires many different views to gather a sufficient number of correspondences and fails for systems with large sensor sizes and apertures.

All of these methods use images of differently oriented patterns, captured by the user. As a result, the calibration can vary for different image sets, and a bad collection of images can result in a poor quality. Especially for estimating accurate distortion coefficients it is crucial to have a set which covers all areas of the image plane. So an accurate calibration usually needs an expert user. [84] proposed an iterative approach

that uses the current calibration state to suggest the next most helpful pattern position to the user. They show that their system yields better calibration results than existing tools using a pre-captured set of images.

In [5], a single image calibration was proposed, but since it is focused on the calibration of endoscope cameras, it is not clear whether the claimed quality can be achieved for other camera and lens configurations as well. Another single image approach using lenticular arrays [96] was proposed recently. Although the method is able to estimate the focal length of a camera, it is not accurate enough to estimate distortion parameters and can only be applied using a centered principal point as well as with color cameras.



Figure 3.1: Sketches of the proposed setups. *Left:* Minimal configuration using two layers: the first layer is an emissive planar display panel and the second layer, a thin opaque sheet with a grid of holes, is placed slightly angled in front. *Center:* Using two tilted hole planes enables a better coverage for wide angle lenses and an additional calibration plane. *Right:* Three layered hole planes consisting of yellow, cyan and magenta filters allow to capture correspondences to three planes with one configuration.

3.2 Device Setup

The main goal of the proposed calibration method is to simplify and fully automate intrinsic camera calibration using a small device mounted directly in front of a lens. The proposed hardware can be assembled from low-cost off-the-shelf components and requires no pre-calibration. Unlike most existing calibration methods, we make use of directionally encoded light rays which allows us to calibrate additional intrinsic properties, such as aperture diameter and focal distance of the lens.

To generate such directional light rays we devise a multi-layer device in which one, the background layer, is an emissive high resolution display. The other layers are one or more thin planar sheets ($d < 0.2mm$), which are opaque for at least parts of the visible spectrum, but contain a grid of small holes ($r < 0.1mm$). With this configuration, light originating from the display can be directionally encoded while passing through the masking layers (cf. Fig. 3.1).

Such a setup leads to some unique effects which have to be considered during the calibration process: each of the holes in a masking layer approximates a pinhole projection, and thus the pixels of the background layer will be imaged in focus by the camera, irrespectively of the actual focus plane of the lens. This is beneficial, since it allows to place the calibration device very close to the lens during calibration. The holes themselves, however, will be subject to *defocus blur* depending on the size of the camera aperture. Furthermore, the pixels of the background layer may appear shifted on the camera sensor. In order to achieve accurate calibration results, these effects have to be accounted for as we show in the following.

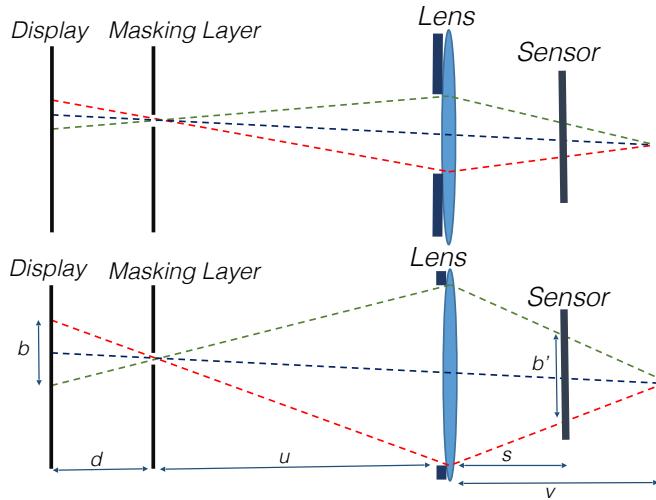


Figure 3.2: Defocus blur of a point source. The size and shape of the CoC depends on the lens' aperture size and shape. *Top*: A closed physical aperture creates a smaller image of the hole. *Bottom*: A large aperture size creates bigger images of the hole and a larger area of the display panel is visible through the hole.

3.2.1 Defocus Blur

If the camera was focused on the masking layer, then the image of each hole of the layer would amount to a single illuminated point on the camera sensor. However, since the device is meant to be mounted closely in front of the lens, which is likely focused at a farther distance, the projection of the holes will amount to circular disks on the sensor instead of points, known as circles of confusion (CoC) [39]. Size and shape of the disks depend on the one hand on the distance of the holes to the sensor and the focal plane and on the other hand on the shape and size of the camera aperture as illustrated in Fig. 3.2 and can be observed in Fig. 3.3.

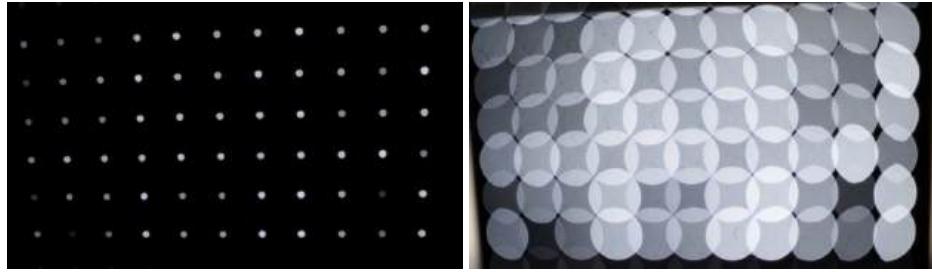


Figure 3.3: Captured images with closed (*Left*) and open aperture (*right*). The CoC of the holes is influenced by the aperture but also by its shape as well as the spatial location on the image plane.

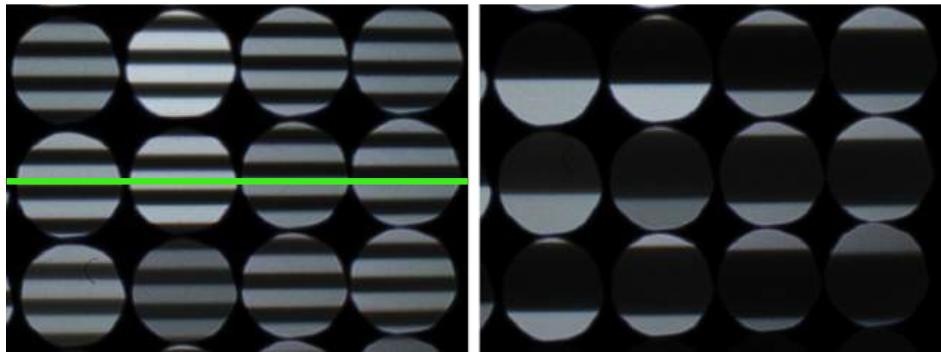


Figure 3.4: Captured images of a masking layer. Gray code patterns of two varying frequencies (*Left, Right*) are displayed on the background layer. Note how the pixel shift described in Sec. 3.2.2 leads to slightly shifted lines when observed through different holes, best seen in the left image at the green line.

3.2.2 Pixel Shift

As described above, the use of pinholes within the optical path leads to circular disks on the camera sensor. The content of these disks is a sharp image of a number of pixels of the background layer, since the holes are essentially pinhole projections, as can be seen in Fig. 3.4. Each pixel corresponds to a directional ray travelling through the optics of the camera. Rays which are not passing through the optical center of the lens will be distorted, which amounts in a translational shift on the sensor, which we call pixel shift. This phenomenon is illustrated in Fig. 3.5 and can be observed in Fig. 3.4, where the lines in neighbouring disks are slightly shifted. For accurate calibration, we

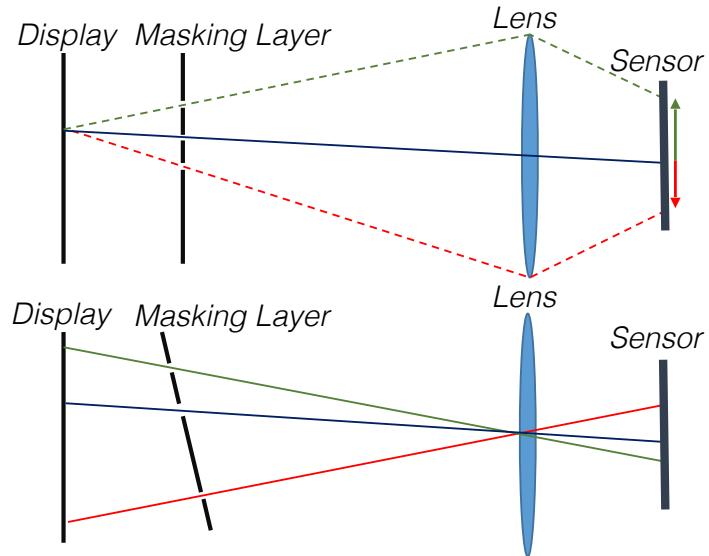


Figure 3.5: *Top*: Different light rays from a (defocused) pixel on the display will intersect with camera’s sensor at different position. Only the ray which passes through the optical center of the lens (blue ray) resides on the same line. *Bottom*: We only consider the rays passing the optical center of the lens for the calibration.

therefore rely solely on the rays that pass through the optical center of the lens and are hence distortion free. How to extract these will be described in Sec. 3.3.3.

3.3 Proposed Calibration Method

With the mentioned optical effects in mind, the proposed device can be used to estimate not only the standard intrinsic properties such as focal length and distortion parameters, but also the focal distance and aperture size. Having mounted the device in front of the camera, the correspondences between the holes of the masking layers and the camera pixels are computed and a mapping between camera and display pixels is generated. Since both background layer and masking layers are planar, we can use the computed correspondences to establish homographies between the planes and employ the method of [115] to calibrate the intrinsic matrix and distortion properties

of the camera. Other lens properties such as focal plane or aperture size can then be estimated by leveraging the directionally encoded light rays. Algorithm 3 shows an overview of our method. In the following sections we will explain each step in detail.

Algorithm 3 Intrinsic Calibration Method

- 1: Choose the smallest aperture for the camera (If possible)
 - 2: Show white for all pixels of the display layer
 - 3: Take an image of the hole layer
 - 4: Find correspondences of masking layer with camera's pixels
 - 5: Increase the aperture size such as to avoid CoC overlap (If possible)
 - 6: Show a sequence of gray code patterns on display layer and capture the images by the camera
 - 7: Generate the mapping between camera and display pixels by decoding the images of the gray codes.
 - 8: Calibrate the intrinsics of the camera and position and orientation of layers using the method of [115]
 - 9: Estimate the focal length and aperture size using the directionaly encoded light rays.
-

3.3.1 Correspondence to Masking Layer

To establish correspondence between the holes in the masking layers and their projection onto the imaging sensor we need to determine their projected location onto the sensor and identify their coordinate on the masking layer.

Hole pixel locations: To compute the projected location of the holes on the image sensor the background display is turned on completely, yielding homogeneously filled circles of confusion on the sensor. As shown in Fig. 3.3 the size of the aperture directly impacts the size of the CoC. The smaller the aperture the better but at the very least the disks should not overlap. If the aperture cannot be controlled or closed sufficiently, then the spacing of the holes on the masking layer has to be adjusted to prevent overlapping CoC.

Even with a small aperture, field curvature, optical aberration artifacts, the aperture shape as well as lens distortion might influence the shape of the CoC such that it deviates from a perfect circle (cf. Fig. 3.6). To account for that in the most generic way, the projected location of the holes is estimated by fitting a 2D Gaussian function to the image of each hole. Fig. 3.6 shows a sample of a fitted Gaussian function and the estimated hole center. Since the hole can be assumed to have infinitesimally small size, this method does not suffer from foreshortening effects known to pose challenges for location estimation using spatial fiducials such as circular feature points[43].

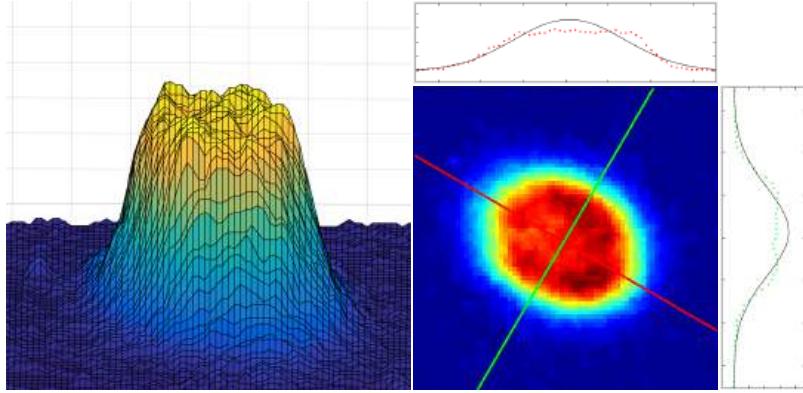


Figure 3.6: Color coded height field plot of a sample hole of one of the captured images of the blocking layers (*Left*) and the center found by fitting a 2D Gaussian function (*Right*).

Hole layer coordinates: The masking layers contain a regular grid of holes and we need to determine the local coordinates of all detected holes on the layers. Therefore, one of the holes is chosen as the origin of the coordinate system, and by detecting the grid we can find the coordinates of the other holes based on this origin. Other methods, such as automatically detected random patterns [111] would be possible as well, but the grid arrangement guarantees a regular pattern which is easier to process and more robust with respect to potentially overlapping circles.

If the lens contains strong distortions, a straightforward grid search is likely to fail.

However, as illustrated in the bottom row of Fig. 3.5 there is a linear projection between 2D coordinates of holes on the masking layer and the coordinates of corresponding pixels on the display panel. This describes a relationship between two planes independent of the camera and is free of any lens distortion. Fig. 3.7 shows the position of visible pixels through the holes on the display panel. Since there is only a relatively small angle between these planar layers and no non-linear distortion, only a small amount of keystoneing occurs, which can be neutralized using an affine transformation. As a result we can easily detect the grid once we know the correspondences between the holes and display pixels. How this is achieved will be described in the following Section.

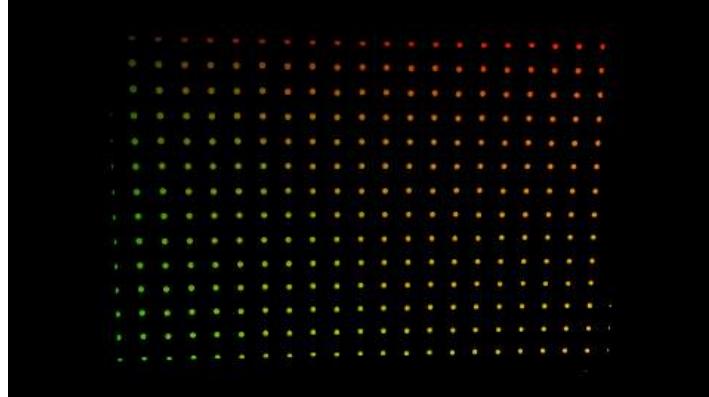


Figure 3.7: Color coded pixel coordinates sensed by the camera through the holes on the display. Note that the projection of the holes on the display is free of distortions and only contains a small amount of keystoneing.

3.3.2 Correspondence to Background Layer

Having established correspondence between the camera and masking layers, the ones to the background layer still have to be estimated. Despite the fact that the camera is out of focus, a sharp and magnified image of the display’s pixels is observed through each hole, since they are so small that they act like pinhole projections. This magnified image of the display helps finding accurate pixel correspondences to the camera. For

this purpose complementary black and white Gray code patterns [21] are displayed and captured ¹. Depending on the display's resolution, this process requires around 40-50 images to be captured to achieve reliable pixel correspondences. At this step, images are captured with larger aperture sizes, if possible, since this increases light throughput and allows to observe more display pixels through each hole. To avoid distortions resulting from a noticeable amount of light diffracting at the boundary of the hole, pixels close to the edge of a hole are masked and disabled for further processing. Next, by decoding the images of the Gray code patterns, pixel correspondences between the camera and the display are generated. To compute the plane homographies we select only those pixels on the background display whose rays pass through the optical center of the lens to avoid issues with pixel shift as described in Sec. 3.2.2. These can be easily determined, as they correspond to the pixels whose projection onto the image sensor coincide with the detected pixel locations of the holes. The other correspondences will be used later on to calibrate the focal plane of the lens.

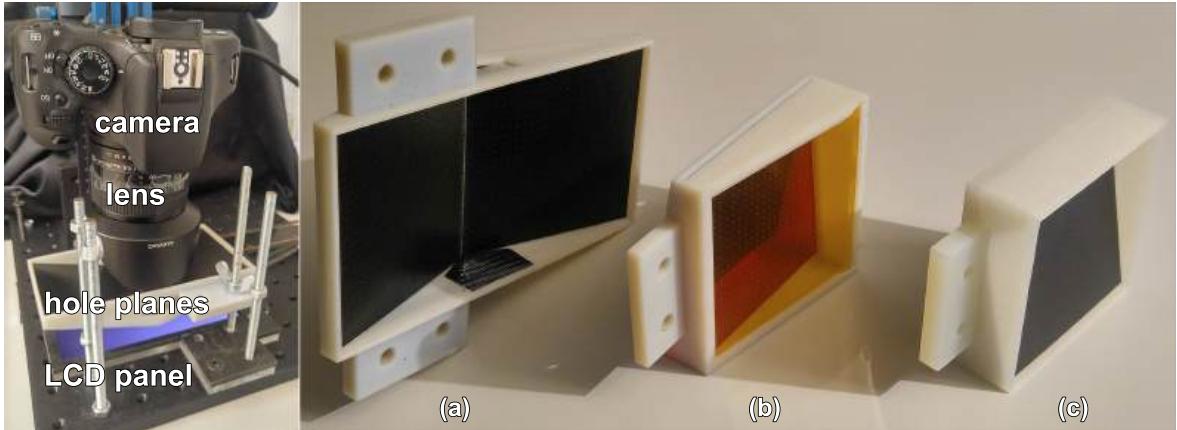


Figure 3.8: *Left:* Prototypical setup with two masking layers. The background layer is an emissive LCD panel and the other layers consist of thin opaque sheets perforated by a grid of holes. *Right:* Three different hole layer configurations: (a) v-shaped planes to get better coverage when calibrating wide angle lenses as shown on the left, (b) cyan, magenta and yellow layers to generate superimposed correspondences using color sensors, and (c) a minimal, single plane configuration.

¹The interested reader is referred to [95] for more information about structured light scanning

3.3.3 Camera and Lens Parameter Estimation

Having computed the correspondences from camera pixels to the masking layers as well as the background layer, the standard homography based calibration method proposed by Zhang et al. [115] can be used to calculate the intrinsic calibration matrix as well as distortion parameters. Furthermore, after having estimated this calibration, the distance to the focal plane as well as the current aperture size can be estimated by our system as well.

Intrinsic Matrix and Distortion Parameters The last Sections described how to gather a set of point correspondences between the display and the masking layers to the camera’s image plane. These correspondences are directly used for camera calibration using the method introduced by Zhang et al. [115]. Since parallel planes do not provide additional constraints for this calibration method and thus should be avoided, the masking layers are rotated by several degrees (cf. Fig. 3.1). First an initial guess of the camera parameters is estimated by solving the closed-form solution using the homographies between the planes and the camera. This result is further refined in a non-linear optimization step as described in [35]. Although this method is used for intrinsic calibration, the extrinsic information, i.e. rotation and translation between the camera and the individual planes is computed as well and can be used to estimate the lens aperture diameter as well as the focus distance as described in the following.

Aperture Size As discussed in Sec. 3.2.1 because of the defocus blur, the image of each hole creates a disk on the camera’s image plane. As a result, several pixels of the display panel are visible through each hole. The geometrical relationships are depicted in Fig. 3.2 and can be described as follows: lets denote the distance of a hole from the background layer as d and from the camera sensor as u and the camera’s aperture size

as a . Based on similarity of the triangles [61] the size of the visible part of the display panel (b) through the hole is

$$b = \frac{ad}{u}. \quad (3.3)$$

This relationship can be used to estimate the current lens's aperture setting. The effect of varying the camera aperture is illustrated in Fig. 3.3.

Mohan et al. [61] show that the size of the CoC for each hole is

$$b' = \frac{(v - s)a}{v}, \quad (3.4)$$

where s is the distance of the sensor from the lens, b' is the size of the CoC on the camera's image plane, and v is the distance of the image of the hole to the lens. Based on the thin lens model equation [37] we have

$$\frac{1}{f} = \frac{1}{u} + \frac{1}{v}. \quad (3.5)$$

Using above equation and removing v from 3.4 we have

$$b' = \frac{(fu - su + fs)a}{fu}. \quad (3.6)$$

The magnification scaling of the lens is $M = \frac{b'}{b}$. So using 3.3 and 3.6 we have

$$M = \frac{b'}{b} = \frac{fu - su + fs}{fd}. \quad (3.7)$$

The above equation can be used to estimate s , which describes the distance of the sensor from the lens, which also allows us to compute the distance of the focal plane from the camera.

In Equ. 3.3 it is shown that the size of the visible part of the display depends on the camera's aperture size and the distance of the hole from the lens and display panel. Having already calibrated the intrinsic and extrinsic parameters allows to calculate the distance of each hole from the lens and display panel. By having estimated the pixel correspondences between the camera's image plane and the display, the size of the visible part of the display panel through each hole can be calculated. With this information, Equ. 3.3 can be used to compute the lens' aperture size.

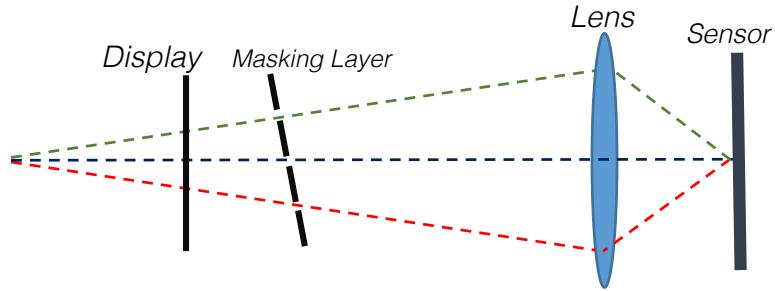


Figure 3.9: Extending the rays which intersect on the camera's sensor allows to estimate the focus distance.

Focal Plane Moreover, using our method we can find the distance of the focal plane from the camera. As we discussed in the last section, the magnification scaling in Equ. 3.7 can be used to find the distance of the sensor from the lens which can be used by Equ. 3.5 to find the distance of the focal plane. However, because of the error in calculating b (size of the visible part of display through each hole) and b' (size of CoC) the final result is not accurate enough. For this purpose we found an alternative solution which is more robust.

As mentioned already, the image of each hole creates a disk known as CoC. Assume that images of these holes have overlap with each other as it is shown in Fig. 3.3. In this case multiple rays from different holes are intersecting at the same position on the camera's image plane (Fig. 3.9). By extending these light rays we can find their

intersection point which is on the focal plane of the camera. For generating these light rays it is required to know the position of the holes and the pixels on the display. But it is not possible to directly look up the correspondences between the camera’s image plane and the display in the overlapping area, since this overlapping effect had to be avoided beforehand when displaying the Gray code patterns because they could not be decoded uniquely within these overlapping areas. To solve this problem first it is assumed that the circle of confusion for the holes do not overlap with each other. By displaying the Gray code patterns (cf. Sec. 3.3.2) the position of the visible pixels through each hole on the image plane is detected. Instead of using only the center correspondence per hole, all generated correspondences between the camera and the display for each individual hole are used to estimate a homography between display and camera pixels for each specific hole. This homography can now be used to virtually extend the image of each individual hole. As a result, virtual overlaps between the images of the holes can be created. Since rotation and translation of the planes are already known during the calibration step, all the required light rays for focal plane estimation can be generated (cf. Table 3.2).

3.4 Hardware Prototype

The algorithm was evaluated using three hardware prototypes based on the design presented in Section 3.2. The three prototypes are shown in Fig. 3.8). The emissive background layer was realized by a 6”, 2K (1440×2560 pixels) LCD IPS display controlled via HDMI². The size of each pixel is approximately $0.051mm$. The blocking layers were realized using black paper sheets which were perforated with a grid of small

²Topfoison TF60010A

Table 3.1: Comparison of the calibration results using the prototypes of the proposed method (cf. Fig. 3.8) and the algorithm proposed by Zhang: f_x and f_y are focal length of the camera in horizontal and vertical direction. (c_x, c_y) is the coordinate of the principal point of the camera, and $k_1 \dots k_4$ are the four estimated distortion parameters

| | | fx | cx | cy | k1 | k2 | k3 | k4 | |
|--------------------------|--------------|---------|---------|---------|---------|----------|----------|----------|----------|
| Canon EF-S 24mm f/2.8 | Prototype(a) | 4871.86 | 4875.97 | 2172.70 | 1396.46 | -0.16321 | 0.06961 | 0.00137 | -0.00114 |
| | Checkerboard | 4766.92 | 4748.65 | 2161.98 | 1419.37 | -0.12474 | 0.12403 | -0.00016 | 0.00060 |
| Canon EF 40mm f/2.8 | Prototype(c) | 7654.27 | 7654.69 | 2073.93 | 1393.93 | -0.05305 | -0.02518 | 0.00078 | 0.00106 |
| | Checkerboard | 7521.66 | 7498.10 | 2147.74 | 1429.39 | -0.03691 | -0.08091 | 0.00157 | 0.00112 |
| Samyang 14mm f/2.8 | Prototype(b) | 2821.82 | 2822.46 | 2147.65 | 1411.67 | -0.13445 | 0.04971 | -0.00038 | 0.00174 |
| | Checkerboard | 2839.23 | 2828.43 | 2144.77 | 1404.27 | -0.11572 | 0.03280 | -0.00043 | -0.00132 |
| LG V10 ~4.4mm f/1.8 | Prototype(b) | 4027.87 | 4026.39 | 2689.62 | 1480.87 | -0.00618 | 0.07026 | -0.00052 | 0.00092 |
| | Checkerboard | 3996.92 | 3977.37 | 2626.58 | 1502.02 | 0.02290 | -0.06092 | -0.00102 | -0.00218 |
| GoPro Hero2 Narrow | Prototype(b) | 1964.38 | 1980.31 | 975.03 | 583.19 | -0.38539 | 0.19380 | 0.00091 | -0.00232 |
| | Checkerboard | 1903.76 | 1924.59 | 945.47 | 588.38 | -0.3713 | 0.2103 | 0.00085 | -0.0017 |

holes using a laser cutter device³. To ensure that they are planar, they were integrated into customized 3D printed⁴ frames and placed 1 – 3cm above the display.

We also realized a color multiplexed setup in which three hole planes are stacked tilted on top of each other. Each plane consists out of one cyan, magenta or yellow color filter⁵, such that the camera captures red, green and blue colored dots from different layers. This has the advantage that it is able to generate superimposed planes with a fixed, unchanged setup.

3.5 Evaluation and Discussion

The accuracy of the proposed calibration was evaluated by comparing it to the method of [115]. 30 images of the checker board were captured with different orientations

³Epilog HELIX 24x18, 60W

⁴Stratasys Connex 350, material: digital ABS

⁵Kodak Wratten Color Filters #44A, #32, #12

and the toolbox implemented by Bouget [12] was used for calibration. The proposed method was tested with prototype (a) (cf. Fig. 3.8) using multiple lenses and cameras (cf. Table 3.1): A Canon 1100D with three different lenses ranging from 14mm up to 40mm, an action camera (GoPro Hero 2), and a smartphone camera (LG V10). The results of both methods were comparable in all setups. A 100mm DSLR lens (Canon EF 100mm f/2.8) was evaluated as well. For the latter, the resulting focal lengths were also comparable to the reference (f_x 22252.54 vs. 21802.86, f_y : 22262.68 vs. 21801.98), however, with such a combination the limits of the current prototype are reached since it is only able to generate a small number of correspondences because of the relatively wide hole spacing required due to the large CoC even with a minimum apertures size. To account for that, a larger back panel would be required to calibrate such lenses more reliably.

To estimate the accuracy of the aperture size calculation described in Sec. 3.3.3, the 24mm lens mounted on a Canon DSLR was used to compare the manually set aperture diameter to the estimated one for a range of f-stops. As shown in Fig. 3.10, all evaluated apertures diameters closely match the ground truth value. The same hardware was used to estimate the accuracy of the focal plane distance estimation. A comparison of the results with the measured ground truth values is given in Table 3.2). In the following we will explain how we obtained the ground truth data for the evaluation of the calculated aperture size and focal plane by our method.

Aperture Size First we calculate the focal length of the camera adequately accurate using a conventional checkerboard calibration or, if available, obtained the f-number from the camera and lens information. As a result we can use the following equation to find the aperture size of the camera: $\text{aperture size} = \text{focal length} / \text{f-number}$. The result of this equation is compared to the calculated aperture size derived from our

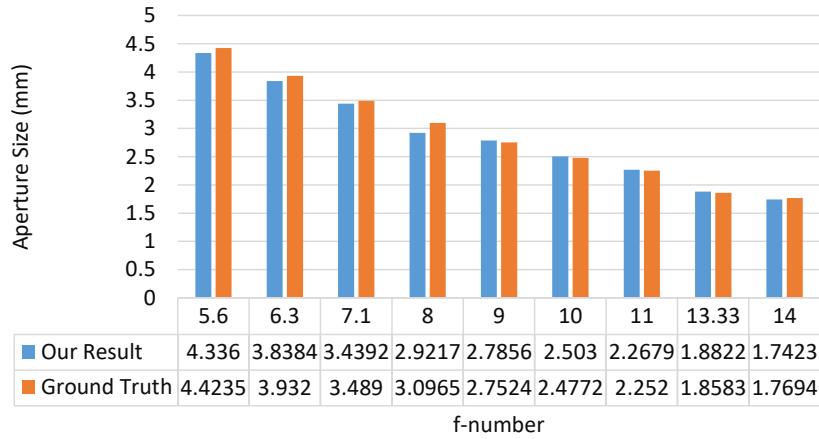


Figure 3.10: Estimated aperture sizes compared to ground truth values.

Table 3.2: Estimated distances of the camera's focal plane in comparison with the actual, i.e. measured, distance.

| | | | | | |
|----------------------|------|-------|-------|-------|-------|
| Proposed method (cm) | 31.8 | 41.41 | 50.76 | 57.02 | 65.55 |
| Ground truth (cm) | 30 | 40 | 50 | 60 | 70 |

method.

Focal Plane First we put a checker board at an arbitrary distance from the camera and adjust the camera to focus onto this checker board and take a picture of it. We also take several other images of the checker board from different orientations allowing us to use the conventional checker board calibration method to calibrate the camera and also find the position of the planes with respect to the camera. Then we can find the distance of the plane from the camera in the first image. Then, without changing the focus of the camera, we use our calibration method to estimate the distance of the focal plane.

Reprojection Errors The reprojection errors of our method were also compared to Zhang's method. For this purpose we used the result of our method for reprojecting the corners of a checkerboard in 15 different images. The average errors are shown in Table 3.3.

Table 3.3: Reprojection error comparison [pixels]

| | proposed method | checkerboard method |
|-------------|-------------------|---------------------|
| 24mm f/2.8 | [0.85979 0.74322] | [0.40055 0.46826] |
| 40mm f/2.8 | [0.51382 0.58675] | [0.36577 0.49497] |
| 14mm f/2.8 | [0.36540 0.34291] | [0.21734 0.25747] |
| 4.4mm f/1.8 | [2.06272 1.42654] | [1.81404 1.40415] |
| GoPro Hero | [0.35966 0.37716] | [0.20676 0.22861] |

3.6 Conclusion

In this chapter we proposed a novel, fully automatic method for intrinsic camera calibration using a small device mounted directly in front of a lens. This greatly simplifies calibration, which otherwise requires expert knowledge and tedious and lengthy manual calibration procedures. We demonstrate accurate and repeatable calibrations of a variety of different configurations ranging from wide angle lenses to tele optics, which are on par with expert checkerboard calibration. In addition to the intrinsics that can be calibrated with previous methods, we can also estimate the focal plane as well as the size of the aperture since we are employing directionally encoded light for calibration.

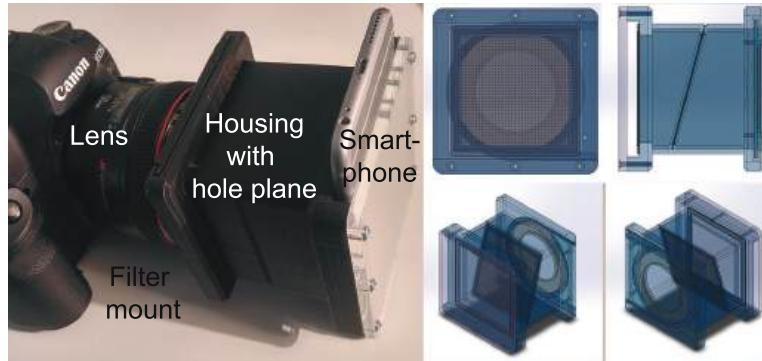


Figure 3.11: *Left:* 3D printed sample all-in-one prototype with lens mount and smartphone as display panel. *Right:* According CAD images.

Since the presented method does not require any specialized hardware except a high-resolution display which is a standard component of state-of-the-art smartphones, it can be realized by using the latter embedded into a frame holding the hole layers. We

3D printed such a sample calibration device with a smartphone as well as a lens mount using the filter thread as shown in Fig. 3.11. We envision that our proposed method could become a standard calibration tool.

In the future we are planning to investigate how further information can be extracted using this method such as lens aberration artifacts, the exact aperture shape, as well as spectral wavelength dependent issues.

Chapter 4

Photometric Calibration

In chapter 2 we discussed geometric calibration of multi-projector systems and explained our geometric calibration method to achieve high accuracy geometric registration on completely arbitrary geometry. However color seamlessness of such unconstrained shapes still pose a major challenge. This chapter presents the first method to achieve complete color seamlessness (both luminance and chrominance) for multi-projector displays on completely arbitrary shapes using a novel shape-aware 3D color gamut morphing method. Though earlier methods have introduced morphing of color parameters (like brightness or 3D gamut) to alleviate severe compression in gamut or dynamic range, they can only handle rectangular overlap regions between projectors, an impossibility in completely arbitrary shapes. Our shape-aware 3D morphing provides a solution that can scale to projection surfaces of any shape and size and overlaps of any shape achieving a perceptually seamless display.

4.1 Background and Related Work

A comprehensive color registration is critical when using off-the-shelf inexpensive projectors that often show severe spatial color variation, even when using projectors of same make and model. Color entails 2D chrominance and 1D luminance and 3D color gamut is a volumetric representation of the entire set of colors that can be reproduced by a device.

[117, 54, 56] categorize the color variation in three classes: intra-projector, inter-projector and overlap variation. Intra-projector variation entails both the hot-spot effect (created by radial fall-off of luminance from center to periphery of the projected image) and color blotches (created by variation in 2D chrominance gamut within a projector). Inter projector variation entails the difference in both chrominance and luminance across the different projectors. Finally overlap variation refers to the differently colored higher-brightness regions created by the overlap of multiple projectors.

However this categorization completely ignores the effect of the display surface shape on color variation problem, which can be rather significant. Variation in display shape result in significant variation in pixel density and angle of projection resulting in a new kind of luminance and chrominance variation. We call this variation *shape based color variation*.

Related Works: Early works [55, 72, 101, 105] used 3D gamut matching to address inter-projector color variation assuming uniform luminance within and across devices. Gamut matching finds the common intersection of multiple 3D gamuts and maps the color of all the different projectors to this common 3D gamut. Following gamut matching, the higher luminance overlap regions were addressed using edge blending [38, 81].

Edge blending simply feathers the brightness from different projectors in the overlap region in a complementary fashion so that the combined luminance in the overlap region matches that of the non-overlap region. [81] presents a software approach for edge blending that cannot reduce black offsets. Therefore, [38] proposes a hardware blending approach that can smoothly blend black offsets as well. For expensive color-uniform devices that show almost no intra projector variation, gamut matching and edge blending is sufficient to create a seamless multi-projector display.

As off-the-shelf projectors became common in spatially augmented reality solutions, spatial variation in color became a predominant feature. [57] presents an unified solution for only the luminance variations – intra-projector, inter-projector and overlaps. Therefore, a separate edge blending is not required. But, more importantly, it presents, for the first time, the concept of smoothing photometric responses instead of matching them. [57] shows that with off-the-shelf projectors, the luminance can vary greatly from pixel to pixel and can have a large range across the multi-projector display. Therefore, matching the luminance response results in a severe compression in the dynamic range of the multi-projector display rendering it almost useless. Therefore, a constraint luminance smoothing is proposed that uses human perception parameters to make the luminance variation unnoticeable while maximizing the dynamic range. This creates an high quality display that utilizes the luminance capability of each pixel optimally.

The inability of [57] is addressing any kind of spatial chrominance variation is alleviated in [88] that provides an unified solution for both inter and intra projector chrominance and luminance variation that became quite prevalent in off-the-shelf low-cost projectors. [88] presents a gamut morphing solution that smoothly morphs the 3D gamut of the mutli-projector display from one pixel to another in a perceptually seamless manner thereby compromising the 3D color capability of at each pixel minimally.

However, [57] and [88] both still assumes roughly similar density pixels and therefore is not shape-aware. [88] in addition assumes rectangular overlap regions further constraining the projector arrangements.

[98] presents the first shape-aware solution for color variation in multi-projector displays. [98] simulates the diffuse direct light transform and presents a method to smoothly blend luminance across projectors. Therefore, this work only addresses the luminance variation across the surfaces ignoring the significant variation of the chrominance across the projectors.

4.1.1 Main Contribution

In this chapter we present the first fully automatic camera based color registration technique that addresses all kinds of variation in both luminance and chrominance in a shape-aware manner. We present a shape-aware gamut morphing algorithm which achieves a smooth morphing of the 3D color gamut on an arbitrarily shaped projection surface illuminated by multiple projectors. Table 4.1 shows the comparison of our method with the aforementioned previous works. Our main contributions are as follows

1. Our smooth 3D color gamut morphing respects the variation in 3D shape of the display surface and assures a perceptually unnoticeable chrominance and luminance transition across the display surface.
2. Our method provides a comprehensive solution that addresses all kinds of luminance and chrominance variation including intra-projector, inter-projector and overlap variations, while compensating for the effect of shape of arbitrarily shaped Lambertian displays.

Table 4.1: This table shows the comparison of our method with existing works.

| Method | Intra Lum | Inter Lum | Chr | Overlap Lum | Chr | Shape- aware |
|----------------|--------------|--------------|-----|----------------|-----|-----------------|
| [101, 72, 105] | | X | X | | | |
| [55] | | X | | | | |
| [38, 81] | | | | X | X | |
| [57] | X | X | | X | | |
| [98] | X | X | | X | | X |
| [88] | X | X | X | X | X | |
| Ours | X | X | X | X | X | X |

3. We show that applying this 3D gamut morphing in the spherical coordinates allows efficiency enhancement via distributed calculation over the multiple computers connected to the projectors. Therefore, our method can scale easily to larger number of projectors.

4.2 Notations

In this section, we first describe the geometric and color notations used in this chapter.

Geometric Notations: We consider a multi-projector display made of M overlapping projectors denoted by P_i , $1 \leq i \leq M$, projecting on an arbitrary shaped display surface S . We denote each 2D pixel coordinate of projector i by (p_i, q_i) , the corresponding 3D coordinate of each pixel on S by (U, V, W) and the corresponding camera pixel by (u, v) . We assume the 3D coordinate of all projector pixels are estimated apriori. For this purpose we use geometric registration method for multi-projector displays as described in chapter 2. Therefore, the 3D coordinate of each pixel is given by

$$(U, V, W) = F_i(p_i, q_i) \quad (4.1)$$

where F_i is the 2D to 3D transformation function of projector i . Furthermore, each pixel (u, v) in camera space is related to the pixel (p_i, q_i) of projector i by the function G , i.e. $(u, v) = G_i(p_i, q_i)$.

Color Notations: Our 3D gamut morphing happens in the CIE XYZ space where color is defined by three values (X, Y, Z) . Note than Y is the luminance. We denote the brightness of a color, I , by its total tristimulus value,

$$I = X + Y + Z \quad (4.2)$$

The chrominance of a color is then specified by its chromaticity coordinates (x, y) given by

$$x = \frac{X}{X + Y + Z}, \quad y = \frac{Y}{X + Y + Z} \quad (4.3)$$

At any pixel where K projectors overlap, each projecting a different color $C_i = (X_i, Y_i, Z_i)$, the XYZ coordinate of the resulting color is given by $C_s = (X_s, Y_s, Z_s)$,

$$(X_s, Y_s, Z_s) = \left(\sum_{i=1}^K X_i, \sum_{i=1}^K Y_i, \sum_{i=1}^K Z_i \right) \quad (4.4)$$

The brightness I_s of C_s is equal to the sum of the brightness of all C_i 's ($I_s = \sum_{i=1}^K I_i$).

The chromaticity coordinate (x_s, y_s) of C_s is given by the weighted sum of the chromaticity coordinates of C_i 's where the weight w_i of C_i is given by the proportion of I_i in I_s . Therefore,

$$(x_s, y_s) = \sum_{i=1}^K \frac{I_i}{I_s} (x_i, y_i) \quad (4.5)$$

Color Gamut: Chromaticity chart is a projection of the 3D XYZ space on a plane with normal $(1, 1, 1)$ that phases out the brightness information retaining only chrominance

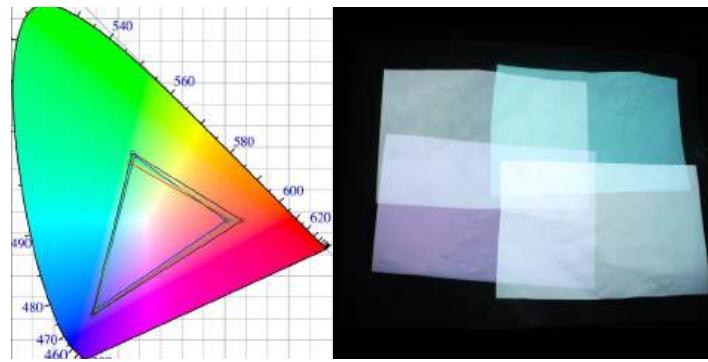


Figure 4.1: This figure shows the color gamut of 4 different projectors (left) and projecting white color with these projectors (right). Notice a small variation in color gamut creates a noticeable color variation in user's perception.

information. Assuming three primary systems, each projector has three channels. Each channel is denoted by c , $c \in \{r, g, b\}$. Let the three channel projector input be denoted by (n_r, n_g, n_b) , $0 \leq n_c \leq 1$. Let (X_l, Y_l, Z_l) be the XYZ value of the color created by the input of 1.0 for one channel $c = l$ and 0.0 for the other two channels $c \neq l$. Therefore, as n_l goes from 0 to 1, the XYZ value of the color created moves along the vector connecting origin to (X_l, Y_l, Z_l) . The three vectors (X_r, Y_r, Z_r) , (X_g, Y_g, Z_g) , and (X_b, Y_b, Z_b) form a parallelepiped in XYZ space which includes all the colors that can be reproduced by the projector. This is called the 3D color gamut of the device. Therefore, the XYZ value of the color of a pixel with input values (n_r, n_g, n_b) is given by

$$\sum_{c \in \{r, g, b\}} n_c(X_c, Y_c, Z_c) \quad (4.6)$$

The brightness of each channel is given by $I_c = X_c + Y_c + Z_c$. Further, the brightness I_w of the input white given by $(1, 1, 1)$ is therefore given by the addition of the brightness of the three channels, i.e. $I_w = I_r + I_g + I_b$.

The constant chrominance of each channel, denoted by $(x_c, y_c) = (\frac{X_c}{I_c}, \frac{Y_c}{I_c})$, $c \in \{r, g, b\}$ forms a triangle in the chromaticity chart that denotes the 2D gamut of the projector.

Figure X is an illustration showing the 3D XYZ color space, example 3D gamut of two different projectors, the projection of the 3D XYZ color space on the plane with normal $(1, 1, 1)$ to create the chromaticity chart, and the projection of the two 3D color gamuts on the same plane yield two different 2D color gamuts. Note that the chromaticity coordinate of the white input (x_w, y_w) is given by

$$(x_w, y_w) = \sum_{c \in \{r, g, b\}} \frac{I_c}{I_w} (x_c, y_c) \quad (4.7)$$

Explaining Spatial Color Variation: Now we will explain the different kinds of color variation using the aforementioned color notation.

Intra-Projector: The channel chrominance is usually constant within a single projector. However, the brightness of each channel I_c varies spatially. It is highest at the center or bottom center (to allow for offset projection when the projector is sitting on a table or mounted on a ceiling) and falls off radially away from the center. This is called the vignetting effect. Therefore, intra-projector variation is primarily a variation in brightness. We can describe this variation as

$$I_c(p, q) = V(p, q) I_c, c \in \{r, g, b\} \quad (4.8)$$

where $I_c(p, q)$ denotes the channel brightness at pixel (p, q) and $V(p, q)$, $0 \leq V(p, q) \leq 1$ denotes the vignetting effect. Note that, since the brightness variation is channel independent and changes equally for all channels the chromacity of white is constant across the projector.

Inter-Projector: The inter-projector variation is due to the difference in intra-projector brightness variation and the difference in the 2D chrominance gamut of each projector.

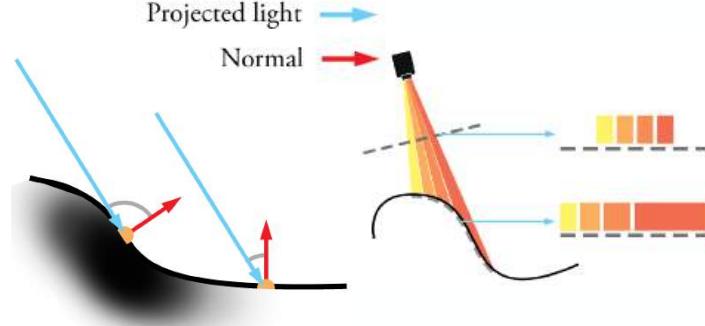


Figure 4.2: Left: This figure shows the effect of incoming light direction on brightness. Right: showing the effect of surface geometry of projector's orientation on size and density of pixels.

Figure 4.1 shows the real data of 2D color gamut of 4 different projectors. Notice that a small spatial variation of 2D color gamut of these projectors creates a noticeable color variation, especially for whites.

Overlap Variation: The overlap variation is caused by the addition of brightness in the overlap region. In addition, combination of the different chrominance from different projectors in different proportion of their brightnesses lead to different chrominance in the overlap region than in the non-overlap region as shown in Figure 4.1 as well.

Shape based Variation: Next, we discuss the color variation due to the variation in shape of the projection surface. Let us consider a point Q on the Lambertian display surface. The brightness I_c^Q viewed at Q depends on the direction of the illumination L_Q and the normal N_Q to the surface at Q and is given by

$$I_c^Q = (N_Q \cdot L_Q) I_c(p, q) \quad (4.9)$$

where $(N_Q \cdot L_Q)$ is the inner product of N_Q and L_Q . Figure 4.2 shows the effect of display shape on the brightness of display. Moreover, the density and size of projected pixels on point Q depends on distance of the projector to display surface and direction

of the the illumination at each point. Figure 4.2 shows the effect of direction of the illumination on size of each pixel. The projected pixel at point Q has smallest size when N_Q and L_Q are parallel. Increasing the angle between N_Q and L_Q will increase the size of the projected pixel and hence, decreasing the density of pixels arround Q .

4.3 Shape-Aware 3D Gamut Morphing

Difference in color gamut of projectors creates noticeable color variation across the display. Most of the existing works perform gamut matching on the devices. However, this will reduce the producible color space along the entire display. Our goal is to smoothly morph the color gamut across the display surface. Though we want to change the color gamut, but we can only change the brightness of each projector. Therefore, no changes of color can happen in non-overlap regions. However, changing the proportion of brightness from contributing projectors can help in changing the color in the overlap regions.

The crux of the method is to first morph 2D chrominance gamut in overlap regions using Laplacian based smoothing considering the shape of the display surface. Though this achieves chrominance morphing, brightness discontinuities still exist. Therefore, next, we apply a perceptual constrained brightness smoothing in a way that does not undo the chrominance smoothing. Together, these two steps achieve the shape aware 3D gamut morphing. Figure 4.3 shows the 3D gamut morphing in the overlap area of the projectors. Finally the aforementioned steps achieve optimized quality only with white point balancing. Therefore, before color gamut morphing step, first we perform a white point balancing to achieve same desired white point across the display surface.

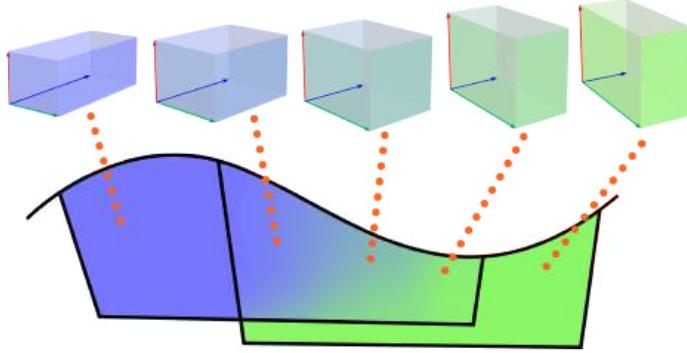


Figure 4.3: This figure show the variation of 3D color gamut of two projectors. We smoothly morph the color gamut in the overlap of projectors. The color of each 3D color gamut in this figure indicates the color of white.

4.3.1 Per Projector White Point Balancing

Since the human’s perception is more sensitive to white, the user can notice the variation of white over the surface. Hence, we want to have a constant white color over the entire display surface. For this purpose we used the proposed method in [88] to find a per channel scale factor ψ_c^i , $c \in \{r, g, b\}$ and $1 \leq i \leq M$, for each projector to achieve same chrominance of white color for all projectors. This is called white point balancing. Note that white point balacning only changes the chrominance of white by scaling the brightness of each channel of the projector and still different projectors have different color gamuts. Figure 4.4 shows the result of white point balancing in a system with four projectors.



Figure 4.4: This figure shows the projection of white color using four projectors on a relief map before (left) and after (right) white point balancing.

4.3.2 Chrominance Gamut Morphing

As we discussed earlier the goal of this step is to smoothly morph the 2D chrominance gamut in the overlap of projectors from one projector to another one. Assume Q is a point on the display surface and is covered by K projectors. After geometric registration all the projectors project same RGB value on their corresponding pixel projecting on Q . Assume $(x_c^i, y_c^i), c \in r, g, b$ denoted the three vertices of 2D chrominance color gamut of projector i . As we discussed before the chrominance color gamut at point Q lighted by K projectors can be calculated as

$$(x_c, y_c) = \sum_{i=1}^K \alpha_i^Q (x_c^i, y_c^i) \quad (4.10)$$

where α_i is the proportion of brightness of projector i to sum of brightness of all projectors at point Q . Therefore, we have

$$\alpha_i^Q = \frac{I_c^i(p_i, q_i)}{\sum_{j=1}^K I_c^j(p_j, q_j)} \quad (4.11)$$

From Equation 4.11 we have $\sum_{i=1}^K \alpha_i^Q = 1$. Hence the 2D color gamut of the display at point Q is a linear combination of the color gamuts of overlapping projectors at that point. Using equation 4.10 and 4.11 we can smoothly morph the color gamut between the projectors by controlling the brightness of each channel of the projectors at point Q . For this purpose we have to find a per pixel scale factor $\gamma^i(p_i, q_i)$ to scale the brightness of the each channel of pixel of the projector i . So we have

$$\alpha_i^Q = \frac{\gamma^i(p_i, q_i) I_i(p_i, q_i)}{\sum_{j=1}^M \gamma^j(p_j, q_j) I_j(p_j, q_j)} \quad (4.12)$$

In order to have a seamless color morphing we should smoothly change α_i^Q in the overlap of the projectors. Note that, The density of pixels varies across the display surface based on relative orientation of the projector with each small patch of the display surface (Figure 4.2). As a result a uniform variation of color in projectors space result in an ununiform color variation on the display surface. Hence, we need to consider the 3D geometry of the display surface to achieve an unnoticeable gamut morphing across the surface of arbitrary shaped display.

In the following we explain our algorithm to achieve smooth chrominance color gamut morphing across the projectors. Our 2D chrominance gamut morphing has three steps: (1) Graph based spatial sampling, (2) Spherical space transformation and (3) Finding per pixel scale factor for each projector.

4.3.2.1 Graph Based Spatial Sampling

Our goal is to find the attenuation factor $\gamma^i(p_i, q_i)$ for all pixels of the projectors. To achieve this, first, we calculate the attenuation factor for a set of sample points and

finally we estimate it for rest of the pixel by mesh interpolation.

Selecting sample points: After geometric calibration of the system we have an accurate 3D geometry of the display surface. We uniformly select sample points from the projection area of each projector on the 3D geometry of the display surface. We select denser sample points on the boundary of each projector. This assures no visible variation of color right at the periphery of each projector.

Creating the graph: Then, we apply constraint Delaunay triangulation on these sample points to create a connectivity graph connecting each sample points to its neighbor points. Constraint Delaunay triangulation forces certain required edges into the triangulation. In our case we force each sample point on the periphery of projectors to be connected to its closest sample point on periphery of the same projector. As a result there is no edges in the graph which is passing through the boundary of the projectors or in other words there is no edge between a sample point inside the projection area of one projector to another sample point out side of the projection area of same projector.

Assigning weights: As we explained in equation 4.10 the 2D color gamut of each point is a linear combination of color gamut of overlapping projectors on that point. This linear combination is achieved by a set of scalar coefficients $\alpha_i, 1 \leq i \leq M$. Thus, we assign a M dimensional vector $V = (\alpha_1, \alpha_2, \dots, \alpha_M)$ to each node of the graph. At this point the value of $\alpha_i, 1 \leq i \leq M$, for some nodes is known and for some of them is not. First, if a point is lighted by only one projector i then $\alpha_i = 1$ and $\alpha_j = 0, j \neq i$. Second if a point is lighted by a set of projectors denoted by H then for this point we have

$$\begin{cases} \alpha_i = 0, i \notin H \\ \alpha_i = \text{unknown}, i \in H \end{cases} \quad (4.13)$$

Our goal is to estimate unknown parameters of vector V for all nodes of the graph in a way that α_i changes smoothly across the graph based on the 3D geometry of the display surface. Therefore, we assign a weight to each edge of the graph which connects two sample points. The weight of each edge is equal to Euclidean distance of 3D coordinate of the two sample points on the geometry of the display surface. So the wieght of the edge between two connected sample point r and t is given by

$$w_{rt} = Distance([U_r, V_r, W_r], [U_t, V_t, W_t]) \quad (4.14)$$

where $[U_r, V_r, W_r]$ and $[U_t, V_t, W_t]$ are the 3D coordinate of two points r and t . The goal of using a weight for each edge of the graph is to control the difference of value of α_i for node t and r assuring a smooth transition of α_i on the surface based on geometry of the arbitrary shaped display.

At this point we can solve a weighted Laplacian equation with Dirichlet boundary conditions over this graph for each α_i to smoothly change α_i over the graph. Our Dirichlet boundary condition is the known value of α_i for a subset of nodes. But this does not guarantee the following constraint for elements of vector V .

$$\sum_{i=1}^M \alpha_i = 1 \quad (4.15)$$

The above constraint makes values of α_i , $1 \leq i \leq M$, dependent to each other. To solve this issue, we have to add above constraint to our Laplacian equation and estimate values α_i s together. This has serveral drawbacks. First, as we mentioned the dependency between values of α_i s prevent us from distributing the task of calculation of α_i s over the computers connected to projectors. Furthermore, the time complexity increases by

the number of projectors. Finally, by adding or removing a projector to the system we have to estimate value of all α_i s, $1 \leq i \leq M$, again even if projector i does not have any overlap with the added or removed projector. To overcome this problem we perform a spherical space transformation as explained in the following section.

4.3.2.2 Spherical Space Transformation

As we mentioned elements of vector V can not be calculated independently because of the constraint 4.15. We can relax this constraint by transforming the Laplacian based smoothing from Cartesian coordinate system to hyper spherical coordinate system as explained bellow.

For this purpose, we replace vector V by its square root as vector $V' = (\sqrt{\alpha_1}, \sqrt{\alpha_2}, \dots, \sqrt{\alpha_M})$.

Now the constraint in Equation 4.15 is equivalent to

$$\text{Length}(V') = \sqrt{\alpha_1 + \dots + \alpha_M} = 1 \quad (4.16)$$

Now our new constraint is to have unit size vector V' for each node. To satisfy this constraint we transform vector V' from Cartesian coordinate system to M dimensional Spherical coordinate system which the coordinates consist of a radial coordinate r and

$M - 1$ angular coordinates $\phi_1, \phi_2, \dots, \phi_{M-1}$. so we have

$$\begin{aligned}\sqrt{\alpha_1} &= r \cos \phi_1 \\ \sqrt{\alpha_2} &= r \sin \phi_1 \cos \phi_2 \\ &\dots \\ \sqrt{\alpha_{M-1}} &= r \sin \phi_1 \sin \phi_2 \dots \cos \phi_2 \\ \sqrt{\alpha_M} &= r \sin \phi_1 \sin \phi_2 \dots \sin \phi_2\end{aligned}\tag{4.17}$$

Based on the above equation we have $r = \sqrt{\alpha_1 + \dots + \alpha_M}$. Hence, we can simply assume $r = 1$ for all nodes. Now each node has a $M - 1$ dimensional vector $(\phi_1, \phi_2, \dots, \phi_{M-1})$. Here again the value of ϕ_i is known for only a subset of nodes, and our goal is to smoothly morph the value of each ϕ_i over the graph. But since there is no constraint between values of ϕ_i s, each angle ϕ_i can be morphed independently across the display surface.

For morphing ϕ_i we use a weighted Laplacian equation for each ϕ_i to smoothly morph the value of ϕ_i across the graph. These Laplacian equations can be solved distributed over the individual computers connected to each projector for each ϕ_i . Finally, we can transform the result from Spherical space to Cartesian space to find the value of each α_i .

4.3.2.3 Finding per pixel scale factor

In the previous step we calculated the value of vector $V = (\alpha_1, \alpha_2, \dots, \alpha_M)$ for all the sample points. As we explained in equation 4.10 the value of α_i is equal to ratio of brightness of projector i to sum of brightness of all projectors at each point for each

color channel. However scaling each color channel individually will change the color of white across the projector. Therefore, as discussed in [88] we only find one common scale factor for all channels by scaling the brightness of white. This still assures a smooth morphing of color gamuts across the display surface. So we need to find per pixel factor γ_i to scale brightness of each projector pixel to achieve desire α_i at each point. Hence, we have

$$\alpha_i = \frac{\gamma_i(p_i, q_i) I_{white}^i(p_i, q_i)}{\sum_{j=1}^M \gamma_j(p_j, q_j) I_{white}^j(p_j, q_j)} \quad (4.18)$$

Above system of equations is an underconstrained system to compute the value of all γ_i , $1 \leq i \leq M$. For this purpose we fix the value $\gamma_i(p_i, q_i)$ for one of the projectors to be $\gamma_i(p_i, q_i) = 1$ and then we find the value of $\gamma_j(p_j, q_j)$, $j \neq i$, for other projectors. Then if the value of one of γ_j exceed 1, we fixed the value for another project and repeat the calculations.

Now we have the scale factor for all the samples. Then we use interpolation to find the scale function $\gamma_i(p_i, q_i)$ for all the pixels of each projector. Finally by scaling the brightness of each pixel of projectors we achieve a smooth chrominance gamut morphing across the display surface. Figure 4.6 shows the result of chrominance gamut morphing.

4.3.3 Brightness Smoothing

After above steps the 2D color gamut changes smoothly over the display surface. However, still there is a noticeable brightness variation, specially in overlap areas (Figure 4.6). In this step we apply a perceptual constrained brightness smoothing presented in [57]. [57] assumes a planar display surface and shows that the constraint for having

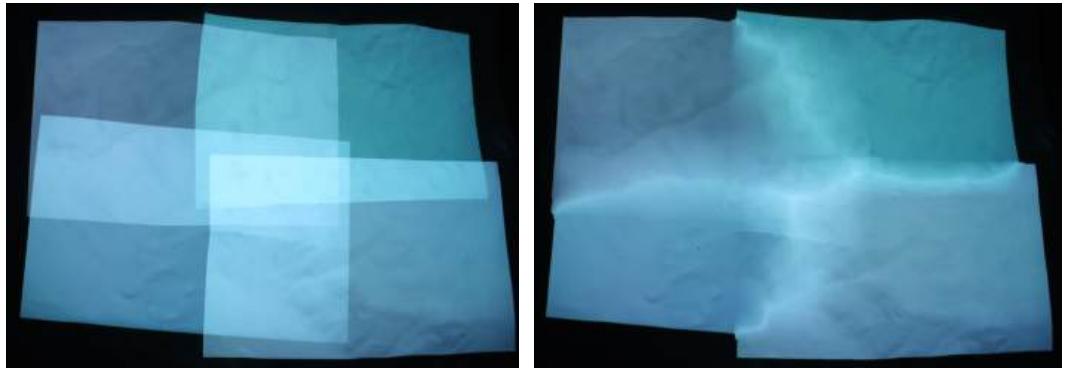


Figure 4.5: This figure shows projection of white color using four projector before (left) and after (right) 2D chrominance gamut morphing. As you can see despite the fact that the chrominance color is smoothly morphed across the display surface, still there is visible brightness variation in overlap areas.

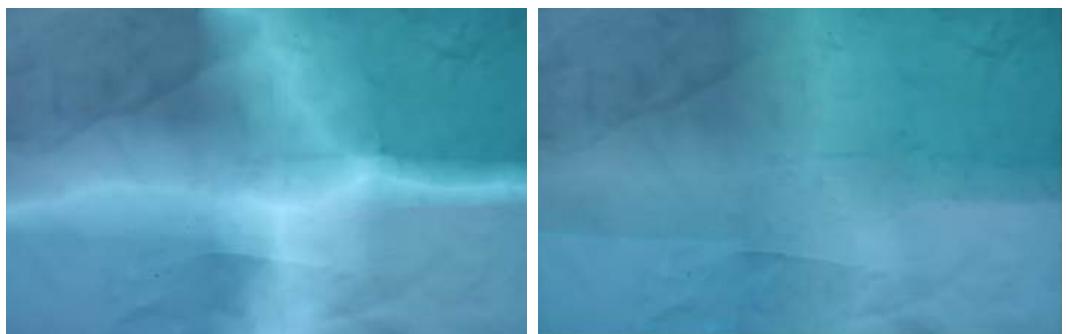


Figure 4.6: This figure shows our result before (left) and after (right) Brightness smoothing and Bezier curve fitting.

an imperceptible brightness to humans is given by

$$\frac{|I'(s, t) - I'(s', t')|}{\sqrt{(s - s')^2 + (t - t')^2}} \leq \frac{1}{\lambda} I'(s, t) \quad (4.19)$$

where (s, t) is the coordinate of a pixel on the planar display's coordinate system and (s', t') is the coordinate of any adjacent pixel to (s, t) . I' indicates the brightness of each point after perceptual brightness smoothing and λ is the smoothing parameter. Since we assume any arbitrary shape surfaces and we desire to consider the shape of the display surface in our brightness smoothing we will have

$$\frac{|I'(U, V, W) - I'(U', V', W')|}{\sqrt{(U - U')^2 + (V - V')^2 + (W - W')^2}} \leq \frac{1}{\lambda} I'(U, V, W) \quad (4.20)$$

where (U, V, W) and (U', V', W') are the 3D coordinate of two adjacent pixels on the display surface.

This will assures a smooth change of brightness across the display surface. But it still does not have C1 continuity. Therefore, our brightness smoothing is followed by a Bezeir surface fitting to achieve C_1 continuity. This gives us an attenuation map μ for the entire display surface. Then we use the coordinate transformation function between the display coordinate and projectors to transformn μ to attenuation map $\mu_i(p_i, q_i)$ for each projector's pixel. Note that since all the projectors have same attenuation factor at each point on the overlapping area, as a result, the brightness of projectors scaled equally at each point and hence, it does not affect the result of gamut morphing step.

Figure 4.6 shows the result of brightness smoothing.



Figure 4.7: This figure shows the result of our photometric registration method in a system made of four overlapping projectors projecting on a arbitrary shaped relief map.

4.4 Online Image Correction

Finally for having a seamless imagery we have to change the input image of each projector based on attenuation maps generated from previous steps. Each projector has two per pixel attenuation map $\gamma_i(p_i, q_i)$ generated by chrominance gamut morphing step and $\mu_i(p_i, q_i)$ generated by perceptual brightness smoothing, besides a per channel scale factor ψ_c^i from the white balancing step. As a result we need to scale the value of each input pixel using these attenuation maps. Assume $A_c^i(p_i, q_i)$ is the value of channel c , $c \in \{r, g, b\}$, of pixel (p_i, q_i) for projector i after geometric registration step. Therefore the value of this pixel after photometric registration denoted by $B_c^i(p_i, q_i)$ is given by

$$B_c^i(p_i, q_i) = h_i^{-1}(\psi_c^i \mu_i(p_i, q_i) \gamma_i(p_i, q_i) A_c^i(p_i, q_i)) \quad (4.21)$$

where h_i^{-1} is the inverse transfer function of the projector i . Since, the projector is not a linear device it is necessary to recover its transfer function. For this purpose we project different shades of color for each projector and capture it with the camera. Then we use proposed method in [57] to recover the transfer function of each projector.



Figure 4.8: This figure shows a system made of three projectors projecting on a dome before photometric correction (left), result of our method (middle) and result of RGB blending (right).

4.5 Implementation and Results

As we described earlier our algorithm consist of two steps. The offline photometric calibration step which calculates the attenuation maps is implemented in Matlab. The online image correction which is implement in GPU using OpenGL and can correct the image in real time. For reconstructing the color gamut of projectors we used a sRGB camera (Canon EOS Rebel Xsi). Since most projectors have a color gamut which is within sRGB gamut, we can accurately reconstruct the color gamut of projectors.

We have implemented and tested our algorithm on two different systems. First system consist of four tiled projectors, projecting on a relief map. Figure 4.7 shows the result of color registration in this system. We also tested our system using three projectors, projecting on a relief map. Figure 4.8 shows our result and the difference between the result of our gamut morphing method with result of RGB blending. Note that using RGB blending still the color variation is noticeable across the display surface.

Chapter 5

User Interaction

Lightweight inexpensive handheld projectors are now available widely (e.g. Sony MP-CL1A of weight 210g and size $7.7 \times 14.9 \times 1.3\text{cm}$, Optoma PK201 of weight 159g and size $6 \times 11.6 \times 1.7\text{cm}$). This opens up the possibility of multiple different kinds of interactions using these projectors as handheld I/O devices. In this chapter, we present an interaction paradigm called *Handheld Interactive Projector (HIP)* that allows one or more handheld projectors to interact with a multi-projector display on an arbitrary 3D geometry when being observed by one or more 2D feedback cameras without the use of any physical markers, fiducials or IMUs. We assume that the multi-projector display is a calibrated system where the 3D geometry of the projection surface and the device parameters (e.g. camera and projector parameters) are already recovered. The uncalibrated handheld projector can interact with both the underlying arbitrary 3D surface and the overlaid digital content either simultaneously or separately.

HIP is enabled by dynamic projector calibration that tracks the 3D location and orientation of the projector using one or more feedback cameras observing 2D features

introduced or available in the content. This is different than almost all prior works that track the projector using 3D fiducials, or IR tags, or a combination of 3D or 2D sensors on the projector hardware and has the following shortcomings.

a) The physical fiducials on the hardware can be occluded by the user making it hard for the tracking to be operational under all conditions. b) Accuracy of 3D sensors are usually much lower than those of 2D sensors. Therefore, the lower tracking accuracy does not allow the content from the handheld projectors to be seamlessly registered with that of overlaid content. Therefore applications that need accurate detection of interaction between contents, as is demonstrated in this work, cannot be achieved. For example, detecting collision between two objects projected from two handheld projectors or between objects from the handheld projector and overlaying content from other static projectors needs accurate registration of the content with the surface which needs highly accurate 3D tracking. Therefore, by using 2D fiducials we achieve a highly accurate 3D tracking that allows subpixel level registration with the digital content from multiple handheld projectors with the underlying surface geometry and the overlaid content from one or more other static projectors providing a truly seamless integration of the real (the 3D surface) and virtual (the overlaid digital content), the static (the 3D surface and static multi-projector display) and dynamic (the moving handheld projectors). Practically also, using only 2D sensors minimizes the number and kind of sensors making the HIP paradigm more attractive for quick deployment in non-permanent systems. Further, since shadows are complementary to illumination, the same HIP paradigm can be used to achieve the same capabilities via shadows created by blocking the light from the handheld projectors.

Table 5.1: Categorized of handheld projector based interaction paradigms.

| Handheld Projector Interaction | Planar | Non-planar |
|--------------------------------|--------------------------|------------|
| No Overlaid Digital Content | [9, 16, 27, 78, 80, 108] | [62] |
| Overlaid Digital Content | [17, 106, 107], HIP | HIP |

5.1 Related Work

Prior work on using handheld projectors for interaction usually focus on either interaction with a non-planar surface or interaction with the overlaid digital content on a planar surface. Each of the above can be considered a special case of the HIP paradigm providing a general framework to tie prior works together. Multiple prior works propose interacting with single or multi-projector systems using hand or body gestures observed by feedback cameras [34, 60, 85] Since these do not use any handheld devices, they are not relevant to HIP. A survey of such methods is available in [109]. There is also a large body of literature on registration of multi-projector displays [79, 87, 90, 94, 112] which is relevant as a scalable technology for illuminating arbitrarily shaped 3D surfaces, but not central to the HIP paradigm. Therefore, we do not include a detailed account of these works here and focus only on handheld projector based interaction paradigms. We categorize prior interaction paradigms using handheld projectors in four classes based on whether the 3D surface is non-planar or planar and whether it is already overlaid by digital content from one or more other projectors in a calibrated system (Table 5.1).

Planar Surface with No Overlaid Digital Content: These systems introduced a “flashlight” interaction metaphor where illuminating a planar surface with a dynamic handheld projector reveals a “hidden” content (e.g. part of a large mural) [16, 17]

Table 5.2: Comparison of the HIP paradigm with prior work

| Method | Geometry | Spatial Extent | Accuracy | Sensors Used | Markers |
|--------------------|---------------------------|----------------|----------|---|-------------|
| iLamp [78, 80] | Planar | 6DOF | NR | 2D/3D - Camera, Tilt sensor, IR tags | Physical/3D |
| Cao [16, 17] | user defined multi-planar | 6DOF | NR | 2D - Camera | Physical/3D |
| SurfacePhone [108] | Planar | 6DOF | NR | 3D - Camera, Motion sensor | |
| Blasko [9] | Planar | 6DOF | NR | 3D - Position and orientation sensor | |
| Harrison [34] | Planar | None | | 3D - Kinect | |
| SidebySide [107] | Planar | None | | 3D - IR Camera, IMU | Digital/2D |
| MotionBeam [27] | Planar | 6DOF | NR | 3D - Accelerometer, Gyroscope, Magnetometer, infrared receiver Ultrasonic distance sensor, | Physical/3D |
| RoomProjector [62] | Arbitrary | 6DOF | 31mm, 9 | 3D - Kinects, IMU, Accelerometer | Physical/3D |
| SlamProjector [62] | Arbitrary | 6DOF | 34mm, 2 | 3D - Kinects | |
| HIP paradigm | Arbitrary | 6DOF | 2mm, 0.3 | 2D only - Camera | Digital/2D |

or augments the underlying surface with relevant information [78, 80]. Therefore, all these works use a notion of *spatially aware* projectors by tracking the 3D location and orientation of the projectors. This tracking of the dynamic projector is achieved in [16, 17] by using physical fiducials on the handheld projectors which are then observed by one or more surround cameras. Unfortunately, occlusion reduces the exibility of the system since the markers need to be visible continuously to track the handheld projectors. [78, 80] uses a tilt sensor and IR tags in addition to the fiducial markers. [27] presents a “MotionBeam” interaction metaphor that uses handheld projectors for character animations on planar surfaces. It uses a 9DOF sensor, a distance sensor, and IR tags to find the orientation and location of the handheld projector. [108] uses the same techniques for spatially aware projectors but tracks a camera attached to the projector whose location with respect to the projector is known and fixed. In comparison, we use only 2D cameras to track the 3D projector location and position without the use of any tags, sensors or fiducials on the projection hardware, but simply using 2D image features introduced or available in the content projected from the handheld projectors.

Non-Planar Surfaces with No Overlaid Digital Content: All the above men-

tioned works are not *geometry aware*, i.e. they do not know the underlying 3D surface on which the handheld projector is projecting and therefore cannot account for the distortion caused by non-planar surfaces. [62] presents a system which is geometrically aware in addition to being spatially aware. One or more Kinects are used to reconstruct the 3D surface apriori to achieve geometric awareness. The first prototype in [62], a RoomProjector, uses the cameras on Kinect to monitor physical markers on the projector to detect the device position and an IMU on the projector is used to generate device orientation. This results in upto 9 degrees and 31mm error in orientation and position recovery primarily due to low accuracy in the measurements from the IMU. The orientation accuracy is improved to 2 degrees in the second prototype, the Slam-Projector, by augmenting the projector with a Kinect. But, the positional accuracy is compromised due to error accumulation in the SLAM method. In contrast we achieve a 0.3 degree and 2mm orientation and position recovery accuracy as we dynamically calibrate the projector from 2D markers in its projection and find its 3D position and orientation.

Planar Surface with Overlaid Digital Content: [107] have explored overlaying digital content from multiple handheld projectors on a planar surface that can interact with each other. A camera and hybrid infrared/visible projectors are use to track the images from multiple devices relative to each other using invisible infrared patterns. This work is closest to our work in terms of using projected images for tracking purposes, but this work focuses on 2D tracking of the projected images instead of 3D tracking of the projection hardware. [17, 46] shows interaction between projected contents from two users for collaborative access and sharing of data (e.g. sharing pics via overlapping of their display by projectors, moving objects from paper displayed by one

user to another, or beaming files from one users directory to another). [106] introduced the focus+context concept by integrating a handheld projector that provides ‘focus’ related information (e.g. high resolution view for a picture, or extra information on an object) with a stationary display that shows the context (e.g. the entire image or assembly of objects). Both these works use markers on the projectors that are tracked with a camera.

A detailed comparison of the HIP paradigm with prior work is summarized in Table 5.2. The first column of the table shows the kind of surface geometry handled by the system. The next column shows spatial-awareness of the system which refers to the ability of the system to track the 3D location and orientation of the projector. Finally, the last two columns list the different kinds of sensors each system use 2D or 3D or a mix of both and whether they use markers on the projector (Physical/3D) or digital markers embeded in 2D content (Digital/2D) for tracking. Note that unlike any prior work, HIP depends only on 2D sensors and 2D fiducials in the projected image to achieve the 3D tracking.

5.2 Main Contributions

We present HIP, a new interaction paradigm that allows interaction of handheld projection with an arbitrary surface overlaid with digital content from multi-projector displays. We process the 2D projected content from the handheld projectors (with introduced or available features) observed by 2D cameras to achieve 3D tracking of the handheld device. The accuracy of our calibration is superior to any other interaction paradigm demonstrated before allowing a seamless registration of the interactive content with the underlying geometry and the static overlaid content. We demonstrate

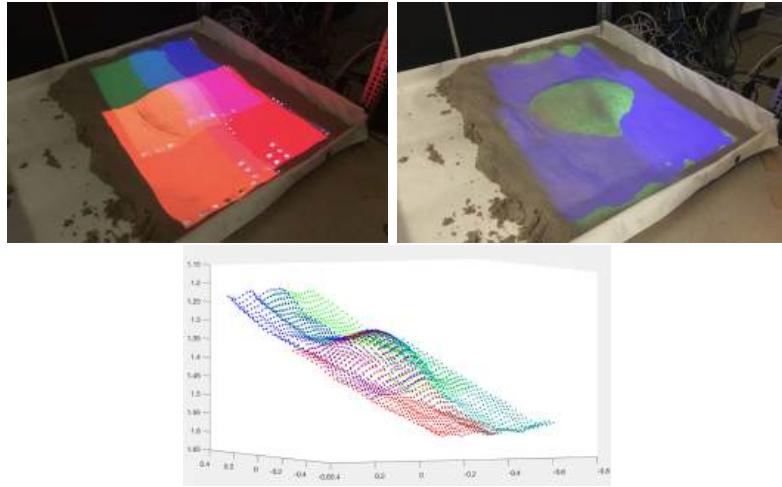


Figure 5.1: This figure shows (a) four overlapping projectors projecting on the sandbox (b) result of geometric registration of the multiple stationary projectors by automatically projecting environment texture (grass on high elevations and water on low elevations) and (c) result of 3D reconstruction of the display surface.

example of such interactions from one or more handheld projector on a sand-pit a 3D surface whose shape can be remodeled easily illuminated by a 4-projector display.

5.3 Projector Tracking

The mainstay of the HIP interaction paradigm is the projector tracking algorithm. The system comprises of an arbitrary 3D surface with digital content overlaid from n tiled or superimposed projectors, $n \geq 1$, observed by m cameras. The system set up involves using structured light techniques to achieve geometric registration and recovery of the 3D geometry of the surface which has been explained in chapter 2. Figure 5.1 illustrates this system set-up for a sandpit lit by four projectors. The information that we use for tracking from this initialization step are (a) the 3D geometry of the surface; and (b) the correspondences between the camera pixels and display surface.

5.3.1 Dynamic Projector Calibration

We dynamically calibrate the projector to find its extrinsic parameters (i.e. 3D position and orientation) using one or more 2D cameras that watch the projection surface instead of the projector. Note that we use the same cameras that are used for camera based registration in projection-based displays to observe the 3D projection surface. Hence, we do not need any additional cameras or any other sensors. We detect features in the images taken by the cameras to calibrate the handheld projector. These features can be inherent to the digital content that can be detected via standard techniques like SIFT or some salient features (e.g. a few colored dots) that we introduce in the image space of the handheld projector. We require a minimum of four features and therefore most digital contents can be easily used for this purpose. Even if we introduce the features in the 2D image, we place them in the periphery of the projector image space. Since the features are small in size, they are not obtrusive visually.

Using the widely used pin-hole camera model for the projector, the 2D location (x, y) of a feature in the projectors image plane can be related to the 3D position (X, Y, Z) of the feature on the display surface by

$$\alpha \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = K[R|RT] \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \quad (5.1)$$

where α is the scale factor and $[R|RT]$ is the projector's extrinsic matrix including the 3×3 rotation R and 3×1 translation T . K is the projector's intrinsic matrix given



Figure 5.2: This figure shows the result of extraction of the projected ball by handheld projector before (left) and after (right) removing environment background.

by

$$K = \begin{bmatrix} f & 0 & u_x \\ 0 & f & u_y \\ 0 & 0 & 1 \end{bmatrix} \quad (5.2)$$

where f is the focal length of projector and (u_x, u_y) is the projectors principal point.

Note that though for most cameras today, $u_x = u_y = 0$, it is not true for projectors primarily due to the offset projection capability that is critical for projection on regions above a tabletop, *and not below*, when placed on such a surface.

Finding Intrinsic Parameters: In order to find the entries of K , we first place the handheld projector at a stationary position facing the projection surface and project a sequence of binary blobs patterns. From the features detected by the observing cameras and the camera to 3D surface correspondences calculated during system initialization, we find the 2D to 3D surface correspondences of the features projected by the handheld projector. These correspondences can then be used to compute the intrinsic parameters in K using standard linear techniques for camera calibration [35]. However, unlike cameras, most projectors show non-linear distortion (i.e. curving of straight lines). In those cases, we use a non-linear optimization (as Levenberg Marquardt algorithm [63]) to find K and the radial and tangential distortion parameters.

Following this one-time intrinsic parameter calibration, the user can now start moving the handheld projector. The extrinsic parameters of the handheld projector are computed dynamically to achieve complete projector tracking. After system initialization, the feedback cameras observing the static multi-projector display can predict the image they are supposed to see if no handheld projector is present in the system. Subtracting this predicted image from what is actually observed extracts the image projected by the handheld projector. The features in this image are detected and their corresponding locations in the 3D display surface are deciphered. Since the location of the same features can be found on the image plane of the handheld projector, we now have 2D feature to 3D display surface correspondence for each feature in the handheld projector. The extrinsic matrix parameters can be extracted from these correspondences using linear optimizations [35]. Figure 5.2 shows the extraction of the soccer ball image projected by moving projector. Since the number of image features required are small, different colors in the texture of the soccer ball can be used in order to find enough unique features.

Availability of features is important for HIP paradigm to work. Since only a few features are required (with a minimum of four features), it is usually easy to find features in the handheld projection. In the worst case, some features can be introduced innocuously at the corners of the projector. Note that a larger variation in the location of the features yeilds more accurate results. Again, since only a few features are adequate, it is not very hard to have spatially distributed features.

5.3.2 Shadow Based Interaction

Since shadows are complementary to projected light, we have developed a modality for the user to interact with the content using the shadow of an object (e.g.their own hand) blocking the light from the handheld projector. For example, an user can push or stop the soccer ball using the shadow of their hand.

In section 2.1 we extract the projected content of the handheld projector by subtracting the predicted image from the observed image. For detecting shadows, we have to do just the opposite. Subtracting the observed image from the predicted image will reveal the darker shadow region. Then, we can again use standard feature detection techniques to find features in the shadow. In particualr, for hand, we used the method introduced in [86] to detect the hand and its finger tips.

5.4 Interaction Modalities

We demonstrate different interaction modalities using handheld projectors in the system made of a 4' by 3' sand pit whose geometry can be changed easily. We use four tiled projectors and four feedback cameras to create a multi-projector display that would project a 3D terrain environment on which one of more users can interact with the geometry or the color of the overlaid content using handheld projectors. An example terrain is shown in Figure 5.1 where grass covers the higher elevations while water resides in the lower elevations. This is our first prototype called the “Terrain 554 Prototype” (Figure 5.3). Figure 5.4 shows the second prototype that demonstrates interaction with dynamic overlaying content called “Live Content Prototype”. The third prototype is a “flashlight” system (Figure 5.5). We demonstrate several inter-

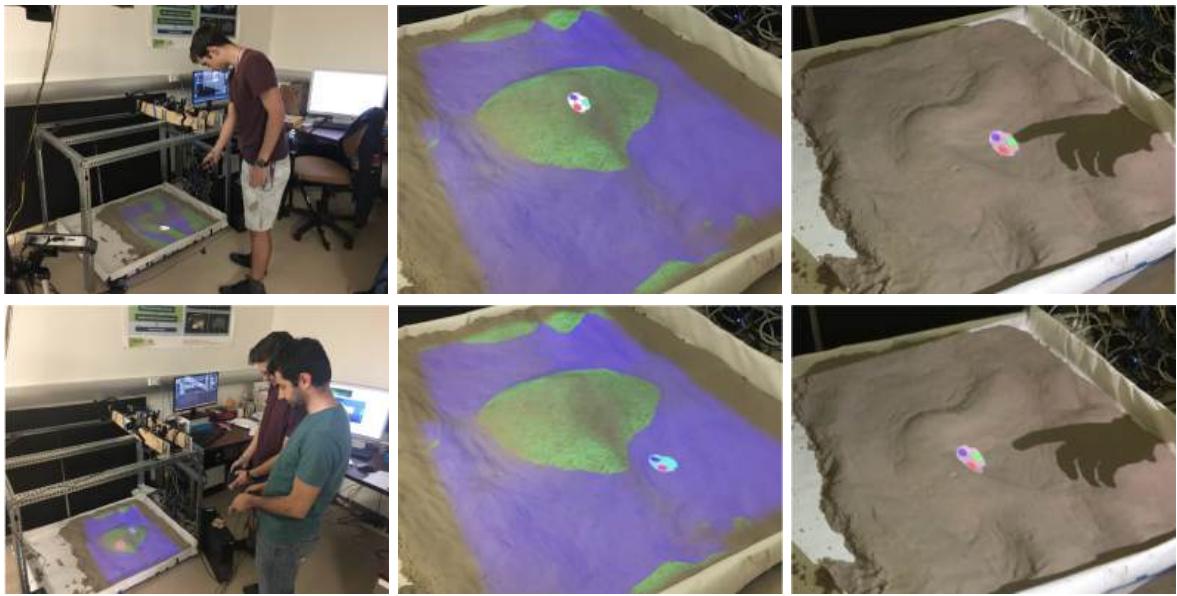


Figure 5.3: This figure shows (a) single or multiple user interacting with the system by moving the ball using handheld projector (first column),(b) Iteraction of the projected content with geometry of the display surface (second column) and (c) interaction of the user with the projected content using hand shadow (third column). First row shows the initial position of the ball and second row shows the position of the ball after interaction with shadow and geometry of the surface. Note that in these figures the projector does not move and only the projected content moves.

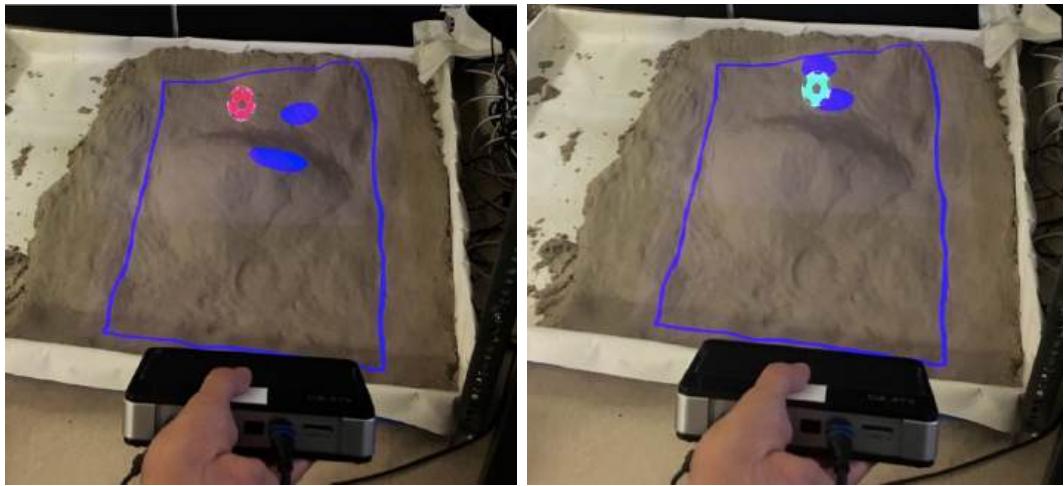


Figure 5.4: This figure shows the interaction of the ball with dynamic overlaying content in our Live Content Prototype. The color of the ball is changed by hitting other moving projected contents.

action modalities triggered by projector motion, underlying geometry characteristics, overlaying static or dynamic content, and with other handheld moving projectors.

Our first demonstration shows the user projecting a ball on the Terrain Prototype that interacts with the underlying geometry and changes speed based on the gradient. The user can move the ball, which triggers a projector motion based interaction, and tries to keep it from running into the water. We demonstrate this on two different terrains to show the interactions more clearly in higher gradient reliefs. We also demonstrate multi-user interaction where two users are trying to move their balls as required, but if the balls touch each other they bounce off (Figure 5.3 and video). In this same prototype, we demonstrate a shadow based interaction where user moves the ball using the shadow of the tips of his fingers (Figure 5.3 and video).

Next, we show the Live Content Prototype where multiple objects projected by static projectors move around the surface area and interact with the ball projected from the handheld projector. The users goal is to move the projector and keep the ball from colliding with the balls projected by the static projectors. A collision is indicated by change of color of the ball from the handheld projector. This demonstrates interaction with overlaid digital content (Figure 5.4 and video).

To demonstrate compliancy with interaction metaphors presented in earlier works we demonstrate a flashlight metaphor where moving on terrain provides data about the terrain (e.g. visualization of different animals living in different elevations of the terrain). We demonstrate multi-user interaction here as well where we show two users overlaying two different data, one animals and the other objects. The users can explore different regions separately or can overlay both information on the same region simultaneously (Figure 5.5 and video).

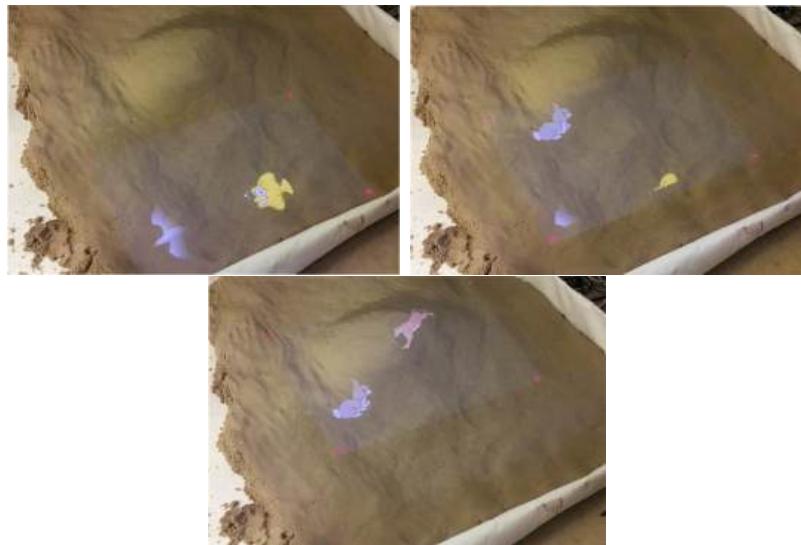


Figure 5.5: This figure shows the result of our flash light system. The user can see different parts of the content by moving the handheld projector. For instance in this example we created a prototype for small kids that they can find the position of different animals living in different elevations of the terrain

Implementation Details: We used four Qumi projectors and four logitech webcam c930 for 3D reconstruction of the display surface and projecting the different environments on the sandpit. We also used a Vivitek QUMI Q2 LED based projector (with size $3 \times 16 \times 10\text{cm}$ and weight 490g) as the handheld projector. Though the Sony Laser projectors are lot smaller and more apt to be called handheld, we used the LED projectors for their much higher brightness which allows our prototypes to operate in bright ambient light. We used OpenCV and C++ for detection of shadows and calibrating the devices and OpenGL shader for rendering the projected content by handheld projector. For each frame we calculate the location and orientation of the moving projector. Then using the 3D model of the display surface and all the intrinsic and extrinsic parameters of the projector we project back the desired content on the display surface on projectors image plane and use this image as the input of the handheld projector.

5.5 Conclusion

In summary, we present the HIP interaction paradigm that allows us to interact with 3D surfaces and overlaid digital content separately or simultaneously using one or more handheld projectors. The core of our method resides on a very accurate dynamic projector calibration technique that finds the 3D position and orientation of the projector in real-time using 2D fiducials embedded in the digital content from the handheld projectors. We anticipate such interaction modalities can be hugely useful in many mixed reality systems for designing multi-user games and visualizations for edutainments.

Chapter 6

Conclusion and Future Work

In summary, in this thesis, I presented an uniform and consistent calibration method for geometric registration and calibration of spatially augmented reality systems. Our approach is fully automatic without using any physical markers or known 3D geometry. The stereo requirement for cameras has been removed since the projectors are used as dual of cameras. I showed that projectors and cameras can be casually aligned and the system achieves high geometric registration accuracy by using large number devices. Further, the dynamic calibration of devices enables various type of flexible interaction paradigms including interaction with overlaid digital content, surface geometry and shadows of users. I showed that the user can move the projectors or surface geometry and the system will correct the mis-registration by recalibrating the projectors. The current system is implemented in CPU and as part of the future work the method can be adopted to run on GPU to achieve real-time correction.

Moreover, I presented a fully automatic calibration method based on directionaly encoded light rays by using a small prototype device placed in front of the lens of the

camera. Besides the accurate recovery of all intrinsic parameters of the camera, our camera calibration algorithm is the first method which can estimate the aperture size and focal plane of the camera.

For color registration, I developed a fully automatic shape-aware 3D gamut morphing method to smoothly morph the 3D color gamut across the display surface. I presented the first method which considers the effect of 3D geometry of display surface on color variation and addressed all kinds of color variation across multi-projectors displays. Our spherical space transformation allow us to distribute the calculations over multiple computers. However, we assume Lambertian surfaces and exploring the effect of specular reflection on color registration is SAR systems is remained for future works.

Bibliography

- [1] D. Aliaga. Digital inspection: An interactive stage for viewing surface details. *Proc. ACM Symp. on I3D*, 2008.
- [2] D. Aliaga and Y. Xu. Photogeometric structured light: A self-calibrating and multi-viewpoint framework for accurate 3d modeling. *Proc. of IEEE CVPR*, 2008.
- [3] D. Aliaga, Y. H. Yeung, A. J. Law, B. Sajadi, and A. Majumder. Fast high-resolution appearance editing using superimposed projections. *ACM Transactions on Graphics*, 2012.
- [4] B. Atcheson, F. Heide, and W. Heidrich. Caltag: High precision fiducial markers for camera calibration. In R. Koch, A. Kolb, and C. Rezk-Salama, editors, *VMV*, pages 41–48. Eurographics Association, 2010.
- [5] J. P. Barreto, J. Roquette, P. Sturm, and F. Fonseca. Automatic Camera Calibration Applied to Medical Endoscopy. *Proceedings of the British Machine Vision Conference 2009*, (20091000069650):52.1–52.10, 2009.
- [6] E. Bhasker, R. Juang, and A. Majumder. Registration techniques for using imperfect and partially calibrated devices in planar multi-projector displays. *IEEE TVCG*, 2007.
- [7] E. Bhasker, P. Sinha, and A. Majumder. Asynchronous distributed calibration for scalable reconfigurable multi-projector displays. *IEEE Transactions on Visualization and Computer Graphics (Visualization)*, 2006.
- [8] O. Bimber, D. Iwai, G. Wetzstein, and A. Grundhöfer. The visual computing of projector-camera systems. In *ACM SIGGRAPH 2008 Classes*, SIGGRAPH ’08, New York, NY, USA, 2008. ACM.
- [9] G. Blasko, S. Feiner, and F. Coriand. Exploring interaction with a simulated wrist-worn projection display. In *Proceedings of the Ninth IEEE International Symposium on Wearable Computers*, ISWC ’05, Washington, DC, USA, 2005. IEEE Computer Society.

- [10] Y. Bok, H. Ha, and I. S. Kweon. Automated checkerboard detection and indexing using circular boundaries. *Pattern Recogn. Lett.*, 71(C):66–72, Feb. 2016.
- [11] F. L. Bookstein. Morphometric tools for landmark data. *Biometrical Journal*, 35(4):512–512, 1993.
- [12] J. Y. Bouguet. Camera calibration toolbox for Matlab, 2008.
- [13] D. Brown. Close-range camera calibration. *Photogrammetric Engineering*, 37(8):855–866, 1971.
- [14] D. C. Brown. Decentering distortion and the definitive calibration of metric cameras. *Annual Meeting of the Americann Society of Photogrammtric Engineering*, 1965.
- [15] M. Brown, A. Majumder, and R. Yang. Camera-based calibration techniques for seamless multiprojector displays. *IEEE Transactions on Visualization and Computer Graphics*, Mar. 2005.
- [16] X. Cao and R. Balakrishnan. Interacting with dynamically defined information spaces using a handheld projector and a pen. *ACM Symposium on User Interface Software and Technology*, 2006.
- [17] X. Cao, C. Forlines, and R. Balakrishnan. Multi-user interaction using handheld projectors. *ACM Symposium on User Interface Software and Technology*, 2007.
- [18] H. Chen, R. Sukhthankar, G. Wallace, and T. jen Cham. Calibrating scalable multi-projector displays using camera homography trees. In *In Computer Vision and Pattern Recognition*, pages 9–14, 2001.
- [19] H. Chen, R. Sukthankar, G. Wallace, and K. Li. Scalable alignment of large-format multi-projector displays using camera homography trees. *Proc. of IEEE Vis*, 2002.
- [20] H. Chen, R. Sukthankar, G. Wallace, and K. Li. Scalable alignment of large-format multi-projector displays using camera homography trees. In *Proceedings of the Conference on Visualization '02*, VIS '02, Washington, DC, USA, 2002. IEEE Computer Society.
- [21] O. Choi, H. Lim, and S. C. Ahn. Robust binarization of gray-coded pattern images for smart projectors. In *2016 International Conference on Electronics, Information, and Communications (ICEIC)*, pages 1–4, Jan 2016.
- [22] R. Cipolla, T. Drummond, and D. Robertson. Camera calibration from vanishing points in images of architectural scenes, 1999.

- [23] A. E. Conrady. Decentered lens-systems. *Monthly Notices of the Royal Astronomical Society*, Vol. 79, p.384-390, 1919.
- [24] D. Cotting, M. Naes, M. Gross, and H. Fuchs. Embedding imperceptible patterns into projected images for simultaneous acquisition and display. *International Symposium on Mixed and Augmented Reality*, 2004.
- [25] D. Cotting, R. Ziegler, M. Gross, and H. Fuchs. Adaptive instant displays: Continuously calibrated projections using per-pixel light control. *Proc. of Eurographics*, pages 705–714, 2005.
- [26] S. Cronk, C. Fraser, and H. Hanley. Automated metric calibration of colour digital cameras. *The Photogrammetric Record*, pages 355–372, 2006.
- [27] K. D. D. Willis, I. Poupyrev, and T. Shiratori. Motionbeam: A metaphor for character interaction with handheld projectors. *Conference on Human Factors in Computing Systems - Proceedings*, 2011.
- [28] A. Datta, J. Kim, and T. Kanade. Accurate camera calibration using iterative refinement of control points. In *Workshop on Visual Surveillance (VS), 2009 (held in conjunction with ICCV)*, October 2009.
- [29] M. Fiala and C. Shu. Self-identifying patterns for plane-based camera calibration. *Machine Vision and Applications*, 19(4):209–216, 2008.
- [30] J. D. Francis Li, Hicham Sekkati. Simultaneous projector-camera self-calibration for three-dimensional reconstruction and projection mapping. *IEEE Transactions on Computational Imaging*, 2017.
- [31] S. Garrido-Jurado, R. Muoz-Salinas, F. Madrid-Cuevas, and M. Marín-Jiménez. Simultaneous reconstruction and calibration for multi-view structured light scanning. *Journal of Visual Communication and Image Representation*, 2016.
- [32] S. Garrido-Jurado, R. M. noz Salinas, F. Madrid-Cuevas, and M. Marín-Jiménez. Automatic generation and detection of highly reliable fiducial markers under occlusion. *Pattern Recognition*, 47(6):2280 – 2292, 2014.
- [33] H. Ha, Y. Bok, K. Joo, J. Jung, and I. S. Kweon. Accurate Camera Calibration Robust to Defocus Using a Smartphone. *2015 IEEE International Conference on Computer Vision (ICCV)*, pages 828–836, 2015.
- [34] C. Harrison, H. Benko, and A. Wilson. Omnitouch: Wearable multitouch interaction everywhere. In *Proc. ACM UIST*, 2011.
- [35] R. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2000.

- [36] M. Harville, B. Culbertson, I. Sobel, D. Gelb, A. Fitzhugh, and D. Tanguay. Practical methods for geometric and photometric correction of tiled projector displays on curved surfaces. *IEEE PROCAMS*, 2006.
- [37] E. Hecht and A. Zajac. *Optics*. Addison-Wesley world student series. Addison-Wesley, 1987.
- [38] M. Hereld, I. Judson, and R. Stevens. Introduction to building projection-based tiled display systems. *IEEE Computer Graphics and Applications*, 2000.
- [39] R. Jacobson, S. Ray, G. Attridge, and N. Axford. *Manual of Photography*. Taylor & Francis, 2000.
- [40] T. Johnson and H. Fuchs. Real-time projector tracking on complex geometry using ordinary imagery. *IEEE CVPR Workshop on Projector Camera Systems (PROCAMS)*, 2007.
- [41] T. Johnson, F. Gyarfas, R. Skarbez, H. Towles, and H. Fuchs. A personal surround environment: Projective display with correction for display surface geometry and extreme lens distortion. *IEEE Virtual Reality*, 2007.
- [42] T. Johnson, G. Welch, H. Fuchs, E. L. Force, and H. Towles. A distributed cooperative framework for continuous multi-projector pose estimation. *IEEE Virtual Reality*, 2009.
- [43] K. Kanatani, Y. Sugaya, and Y. Kanazawa. *Ellipse Fitting for Computer Vision: Implementation and Applications*. Synthesis Lectures on Computer Vision. Morgan & Claypool Publishers, 2016.
- [44] D. G. Kendall. A survey of the statistical theory of shape. *Statistical Science*, 4(2):87–99, 1989.
- [45] A. Law, D. Aliaga, B. Sajadi, A. Majumder, and Z. Pizlo. Perceptually-based appearance modification for compliant appearance editing. *Computer Graphics Forum*, 2011.
- [46] M. Lazarov, H. Pirsiavash, B. Sajadi, U. Mukherjee, and A. Majumder. Data handling displays. *IEEE ACM Workshop on Projector Camera Systems*, 2009.
- [47] J. C. Lee, P. H. Dietz, D. Maynes-Aminzade, R. Raskar, and S. E. Hudson. Automatic projector calibration with embedded light sensors. In *Proceedings of the 17th Annual ACM Symposium on User Interface Software and Technology*, pages 123–126, New York, NY, USA, 2004. ACM.
- [48] Z. Li, Y. Shi, C. Wang, and Y. Wangi. Accurate calibration method for a structured light system. *Optical Engineering*, 2008.

- [49] J. Liao and L. Cai. A calibration method for uncoupling projector and camera of a structured light system. *International Conference on Advanced Intelligent Mechatronics*, 2008.
- [50] G. M. Liming Yang, ean-Marie Normand. Practical and precise projector-camera calibration. *IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*, 2016.
- [51] V. A. V. Lixia Yang, Chao Tian and A. R. Reibman. An automatic grid corner extraction technique for camera calibration. *2012 19th IEEE International Conference on Image Processing*, pages 349 – 352, 2012.
- [52] A. Maimone, D. Lanman, K. Rathinavel, K. Keller, D. Luebke, and H. Fuchs. Pinlight displays: Wide field of view augmented reality eyeglasses using defocused point light sources. *ACM Trans. Graph.*, 33(4):89:1–89:11, July 2014.
- [53] A. Majumder. Camera based evaluation of photometric compensation methods on multi-projector displays. *Proceedings of IEEE International Conference on Image Processing (ICIP)*, 2004.
- [54] A. Majumder and M. Gopi. Modeling color properties of tiled displays. *Computer Graphics Forum*, June 2005.
- [55] A. Majumder, Z. He, H. Towles, and G. Welch. Achieving color uniformity across multi-projector displays. *Proceedings of IEEE Vis*, 2000.
- [56] A. Majumder and R. Stevens. Color nonuniformity in projection-based displays: Analysis and solutions. *IEEE Transactions on Vis and Computer Graphics*, 10(2), March–April 2003.
- [57] A. Majumder and R. Stevens. Perceptual photometric seamlessness in tiled projection-based displays. *ACM TOG*, 2005.
- [58] T. P. Martin Bujnak, Zuzana Kukelova. 3d reconstruction from image collections with a single known focal length. 2009.
- [59] W. B. S. Michael S. Brown. A practical and flexible tiled display system. *10th Pacific Conference on Computer Graphics and Applications, 2002. Proceedings.*, 2002.
- [60] L. C. Mistry Pranav, Pattie Maes. Wuw - wear ur world - a wearable gestural interface. *ACM CHI extended abstracts*, 2009.
- [61] A. Mohan, G. Woo, S. Hiura, Q. Smithwick, and R. Raskar. Bokode: Imperceptible visual tags for camera based interaction from a distance. *ACM Trans. Graph.*, 28(3):98:1–98:8, July 2009.

- [62] D. Molyneaux, S. Izadi, D. Kim, O. Hilliges, S. Hodges, X. Cao, A. Butler, and H. Gellersen. Interactive environment-aware handheld projectors for pervasive computing spaces. Lecture Notes in Computer Science. Springer, 2012.
- [63] J. J. Moré. *The Levenberg-Marquardt algorithm: Implementation and theory*, pages 105–116. Springer Berlin Heidelberg, Berlin, Heidelberg, 1978.
- [64] D. Moreno and G. Taubin. Simple, accurate, and robust projector-camera calibration. *3D Data Processing, Visualization and Transmission*, 2012.
- [65] P. Moulon, P. Monasse, and R. Marlet. Adaptive structure from motion with a contrario model estimation. In *Proceedings of the 11th Asian Conference on Computer Vision - Volume Part IV*, ACCV'12. Springer-Verlag, 2013.
- [66] M. Mühlich and T. Aach. High accuracy feature detection for camera calibration: A multi-steerable approach. In F. A. Hamprecht, C. Schnörr, and B. Jähne, editors, *Pattern Recognition: 29th DAGM Symposium, Heidelberg, Germany, September 12-14, 2007. Proceedings*, pages 284–293, Berlin, Heidelberg, 2007. Springer Berlin Heidelberg.
- [67] R. K. Oliver Fleischmann. Fast projector-camera calibration for interactive projection mapping. *23rd International Conference on Pattern Recognition (ICPR)*, 2016.
- [68] OpenCV. Open source computer vision library. <https://github.com/itseez/opencv>, 2015.
- [69] J.-N. Ouellet, F. Rochette, and P. Hebert. Geometric calibration of a structured light system using circular control points. In *3D Data Processing, Visualization and Transmission*, 2008.
- [70] Y. Oyamada, P. Fallavollita, and N. Navab. Single camera calibration using partially visible calibration objects based on random dots marker tracking algorithm. In *ISMAR*, Atlanta, USA, November 2012.
- [71] O. Özyesil, V. Voroninski, R. Basri, and A. Singer. A survey on structure from motion. *CoRR*, 2017.
- [72] B. Pailthorpe, N. Bordes, W. Bleha, S. Reinsch, and J. Moreland. High-resolution display with uniform illumination. *Proceedings Asia Display IDW*, pages 1295–1298, 2001.
- [73] S.-Y. Park and G. G. Park. Active calibration of camera-projector systems based on planar homography. In *ICPR*, pages 320–323. IEEE Computer Society, 2010.

- [74] A. Raij, G. Gill, A. Majumder, H. Towles, and H. Fuchs. Pixelflex 2: A comprehensive automatic casually aligned multi-projector display. *IEEE PROCMAS*, 2003.
- [75] A. Raij and M. Polleyfeys. Auto-calibration of multi-projector display walls. *Proc. of ICPR*, 2004.
- [76] R. Raskar. Immersive planar displays using roughly aligned projectors. In *Proc. of IEEE VR*, 2000.
- [77] R. Raskar, J. V. Baar, T. Willwacher, and S. Rao. Quadric transfer function for immersive curved screen displays. *Eurographics*, 2004.
- [78] R. Raskar, P. Beardsley, J. van Baar, Y. Wang, P. Dietz, J. Lee, D. Leigh, and T. Willwacher. Rfig lamps: Interacting with a self-describing world via photosensing wireless tags and projectors. *ACM Trans. Graph.*, Aug. 2004.
- [79] R. Raskar, M. Brown, R. Yang, W. Chen, H. Towles, B. Seales, and H. Fuchs. Multi projector displays using camera based registration. *Proc. of IEEE Vis*, 1999.
- [80] R. Raskar, J. van Baar, P. Beardsley, T. Willwacher, S. Rao, and C. Forlines. ilamps: Geometrically aware and self-configuring projectors. *ACM Transaction on Graphics (SIGGRAPH)*, 2003.
- [81] 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 immersive display. In *Proceedings of ACM Siggraph*, pages 168–176, 1998.
- [82] C. Resch, H. Naik, P. Keitler, S. Benkhardt, and G. Klinker. On-site semi-automatic calibration and registration of a projector-camera system using arbitrary objects with known geometry. *IEEE Trans. Vis. Comput. Graph.*, 2015.
- [83] A. Z. Ricardo R. Garcia. Geometric calibration for a multi-camera-projector system. *IEEE Workshop on Applications of Computer Vision (WACV)*, 2013.
- [84] A. Richardson, J. Strom, and E. Olson. AprilCal: Assisted and repeatable camera calibration. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, November 2013.
- [85] P. Roman, M. Lazarov, and A. Majumder. A scalable distributed paradigm for multi-user interaction with tiled rear projection display walls. *IEEE Transactions on Visualization and Computer Graphics*, 2010.
- [86] P. K. K. Ruchi Manish Gurav. Real time finger tracking and contour detection for gesture recognition using opencv. *International Conference on Industrial Instrumentation and Control (ICIC)*, 2015.

- [87] B. Sajad and A. Majumder. Automatic registration of multiple projectors on swept surfaces. *ACM Virtual Reality and Software Technology*, 2010.
- [88] B. Sajadi, M. Lazarov, A. Majumder, and M. Gopi. Color seamlessness in multi-projector displays using constrained gamut morphing. *IEEE Transactions on Visualization and Computer Graphics (TVCG)*, 2009.
- [89] B. Sajadi and A. Majumder. Markerless view-independent registration of multiple distorted projectors on vertically extruded surface using a single uncalibrated camera. *IEEE Transactions on Visualization and Computer Graphics (TVCG)*, 2009.
- [90] B. Sajadi and A. Majumder. Auto-calibration of cylindrical multi-projector systems. *IEEE Virtual Reality*, 2010.
- [91] B. Sajadi and A. Majumder. Scalable multi-view registration for multi-projector displays on vertically extruded surfaces. *Computer Graphics Forum*, 2010.
- [92] B. Sajadi and A. Majumder. Auto-calibrating projectors for tiled displays on piecewise smooth vertically extruded surfaces. *IEEE Transactions on Visualization and Computer Graphics*, 2011.
- [93] B. Sajadi and A. Majumder. Automatic registration of multi-projector domes using a single uncalibrated camera. *Computer Graphics Forum*, 2011.
- [94] B. Sajadi, M. A. Tehrani, M. Rehimzadeh, and A. Majumder. High-resolution lighting of 3d relief maps using a network of projectors and cameras. *3D-TV conference on Immersive and Interactive 3D Media Experience Over Networks*, 2015.
- [95] J. Salvi, S. Fernandez, T. Pribanic, and X. Llado. A state of the art in structured light patterns for surface profilometry. *Pattern Recogn.*, 43(8):2666–2680, Aug. 2010.
- [96] I. Schillebeeckx, S. Louis, R. Pless, and S. Louis. Single Image Camera Calibration with Lenticular Arrays for Augmented Reality. 2016.
- [97] T. K. Shuntaro Yamazaki, Masaaki Mochimaru. Simultaneous self-calibration of a projector and a camera using structured light. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2011.
- [98] C. Siegl, M. Colaianni, L. Thies, J. Thies, M. Zollhöfer, S. Izadi, M. Stamminger, and F. Bauer. Real-time pixel luminance optimization for dynamic multi-projection mapping. *ACM Trans. Graph.*, Oct. 2015.

- [99] N. Snavely, S. M. Seitz, and R. Szeliski. Photo tourism: Exploring photo collections in 3d. *ACM Transactions on Graphics (SIGGRAPH)*, 2006.
- [100] P. S. H. Song Zhang. Novel method for structured light system calibration. *Optical Engineering*, 2006.
- [101] M. C. Stone. Color balancing experimental projection displays. *9th IS&T/SID Color Imaging Conference*, 2001a.
- [102] W. Sun, I. Sobel, B. Culbertson, D. Gelb, and I. Robinson. Calibrating multi-projector cylindrically curved displays for "wallpaper" projection. *IEEE/ACM Workshop on PROCAMS*, 2008.
- [103] J. G. T. T. Li, H. Y. Zhang. Geometric calibration of a camera-projector 3d imaging system. *25th International Conference of Image and Vision Computing New Zealand (IVCNZ)*, 2010.
- [104] H. Urey, K. V. Chellappan, E. Erden, and P. Surman. State of the art in stereoscopic and autostereoscopic displays. *Proceedings of the IEEE*, 99(4):540–555, April 2011.
- [105] G. Wallace, H. Chen, and K. Li. Color gamut matching for tiled display walls. *Immersive Projection Technology Workshop*, 2003.
- [106] M. Weigel, S. Greenberg, and A. Tang. From focus to context and back: Combining mobile projectors and stationary displays. In: *Proceedings of GRAND 2013*, 2012.
- [107] K. D. Willis, I. Poupyrev, S. E. Hudson, and M. Mahler. Sidebyside: Ad-hoc multi-user interaction with handheld projectors. In *Proceedings of the 24th Annual ACM Symposium on User Interface Software and Technology*, UIST '11, 2011.
- [108] C. Winkler, M. Löchtefeld, D. Dobbelstein, A. Krüger, and E. Rukzio. Surfacephone: A mobile projection device for single- and multiuser everywhere tabletop interaction. In *Proceedings of the 32Nd Annual ACM Conference on Human Factors in Computing Systems*, New York, NY, USA, 2014.
- [109] K. Wolf, M. Funk, P. Knierim, and M. Löchtefeld. Survey of interactive displays through mobile projections. *Int. J. Mob. Hum. Comput. Interact.*, 8(4):29–41, Oct. 2016.
- [110] Y. Xiao and B. Fisher. *Accurate Feature Extraction and Control Point Correction for Camera Calibration with a Mono-Plane Target*. 2010.

- [111] L. Yang, J. M. Normand, and G. Moreau. Local geometric consensus: A general purpose point pattern-based tracking algorithm. *IEEE Transactions on Visualization and Computer Graphics*, 21(11):1299–1308, Nov 2015.
- [112] R. Yang, D. Gotz, J. Hensley, H. Towles, and M. S. Brown. Pixelflex: A reconfigurable multi-projector display system. *Proc. of IEEE Vis*, 2001.
- [113] R. Yang and G. Welch. Automatic projector display surface estimation using every-day imagery. *9th International Conference in Central Europe on Computer Graphics, Visualization and Computer Vision*, 2001.
- [114] X. Zhang and L. Zhu. Projector calibration from the camera image point of view. *Optical Engineering*, 2009.
- [115] Z. Zhang. Flexible camera calibration by viewing a plane from unknown orientations. *International Conference on Computer Vision*, 1999.
- [116] J. Zhou, L. Wang, A. Akbarzadeh, , and R. Yang. Multi-projector display with continuous self-calibration. *IEEE/ACM Workshop on Projector-Camera Systems (PROCAMS)*, 2008.
- [117] C. Zoido, J. Maroto, G. Romero, and J. Felez. Optimized methods for multi-projector display correction. *International Journal on Interactive Design and Manufacturing (IJIDeM)*, 7(1), Feb 2013.
- [118] S. Zollmann, T. Langlotz, and O. Bimber. Passive-active geometric calibration for view-dependent projections onto arbitrary surfaces. *In Workshop on Virtual and Augmented Reality of the GI-Fachgruppe AR/VR*, 2006.