

# Self-Calibrating Dynamic Projection Mapping System for Dynamic, Deformable Surfaces with Jitter Correction and Occlusion Handling

Muhammad Twaha Ibrahim\*  
University of California, Irvine

M. Gopi†  
University of California, Irvine

Aditi Majumder‡  
University of California, Irvine

## ABSTRACT

Dynamic projection mapping (DPM) is becoming increasingly popular, enabling viewers to visualize information on moving and deformable surfaces. Examples include large data visualization on the moving walls of tents deployed in austere remote locations during emergency management or defense operations. A DPM system typically comprises a RGB-D camera and a projector. In this paper, we present the first fully functional DPM system that auto-calibrates (without any physical props like planar checkerboard or rigid 3D objects) and creates a comprehensible display in the presence of large and fast movements by managing jitter and occlusion by passing objects.

Prior DPM systems need specific calibration props, manual inputs and in order to deliver sub-pixel calibration accuracy. Recalibration in the face of movement or change in system setup becomes a time consuming process where the calibration prop needs to be brought back. When rendering content using DPM, errors in calibration are exacerbated and the noise in the depth camera leads to jitter, making the projection unreadable or incomprehensible. Occlusion may disrupt operations completely by jumbling up even the unoccluded parts of the display.

In this paper we propose key hardware-agnostic methods for DPM calibration and rendering to make DPM systems easily deployable, stable and legible. First, we present a novel *projector-camera calibration* that does not need synchronization of the devices and leverages the moving surface itself, a counter-intuitive proposition. We project ArUCo markers on the moving surface and use corresponding detected features of these markers in the RGB and depth camera over multiple frames to accurately estimate the intrinsics and extrinsics of both the projector and the RGB-D camera. Second, we present a DPM *rendering* method that uses Kalman filtering models to reduce jitter and predict the surface shape in the presence of short term occlusions by other static objects. This results in the first DPM system, to the best of our knowledge, that can auto-calibrate in minutes and can render high resolution content like high-resolution text or images comprehensible even in the presence of fast movements, deformations and occlusions. We compare and evaluate the accuracy with prior methods and analyze the effect of surface movement on the calibration accuracy.

**Index Terms:** Computing Methodologies—Artificial Intelligence—Computer Vision—Image and Video Acquisition

## 1 INTRODUCTION

Dynamic Projection Mapping (DPM) is challenging due to the demands of continuous real-time reconstruction of the moving surface and low-latency adaptive projection. High-quality DPM depends on three critical components: (a) projector-camera calibration (i.e.

recovering intrinsic and extrinsic parameters of the devices) that achieves subpixel accuracy; (b) a method to capture a changing shape of the surface accurately in real time; and (c) an accurate rendering that adapts to the changing shape. Most prior work use custom solutions to achieve this.

[28, 46, 50] avoid shape capture by using high-speed coaxial projector-camera systems and [30, 31] use additional custom markers (e.g. special IR markers printed on the surface). Recently, more general DPM systems, that do not use any custom solutions, are becoming common [12, 13]. A consumer projector and an off-the-shelf RGB-D camera (e.g. Kinect), set up in a non-coaxial manner, is used to achieve DPM that adapts the projection on dynamic, and even deformable, surfaces. An RGB-D camera captures a coarse resolution registered depth image in addition to a high resolution RGB image at video rates. The captured depth of the moving deformable surface is leveraged by the projector to appropriately distort the projection to conform continuously to the dynamically changing surface shape.

Our goal in this paper is to achieve a *high quality* and *easily deployable* DPM using such general setups. We target applications where DPM is used to create data visualizations on moving tent walls in austere locations (e.g. military command and control, emergency management), manipulating the appearance of fabric [34], and illuminating screens carried by drones to display large-scale signs and advertisements [15]. To present stable, legible and high-resolution text and images in such situations requires highly accurate projector-camera calibration. Even after high-accuracy calibration, the DPM can deal with challenges of noisy depth data when using consumer devices. Unlike rigid objects where the shape is known a priori, DPM on deformable surfaces rely on the depth camera to reconstruct the changing shape of the surface every frame. However, the depth data is usually noisy and introduces tiny perturbations in the reconstructed shape, even if it is completely static. Since the projection adapts itself to this continuously changing noisy shape, it creates a jitter that can impede comprehension of projected content in addition to being simply annoying. Occlusions of the surface, for example, by a person walking in front of the projected display, is pretty common in any working situation. However, occlusions substantially change the reconstructed shape of the surface as the occluding object gets captured instead of the dynamic projection surface. This results in a significant distortion of the projected content even in the unoccluded regions, completely disrupting the viewer experience. Finally, in order to achieve ease of deployment in such expeditionary situations, the calibration process needs to be simple and fast. All current projector-camera calibrations use rigid physical props (e.g. precise checkerboard or 3D patterned objects) with involved manual intervention that takes time. Recalibration, if the system components move, needs the calibration prop to be brought back, and is not simple.

### 1.1 Main Contribution

In this paper, we present the first high quality DPM system that can be deployed easily and quickly using automated procedures. The contributions of this work that make such a system possible are as follows:

1. Contrary to common intuition that calibration processes should

\*e-mail: muhammti@uci.edu

†e-mail: gopi@ics.uci.edu

‡e-mail: majumder@ics.uci.edu

use static system set up, we propose a novel projector calibration method that leverages the moving surface to achieve an accurate and completely automated recovery of the intrinsic and extrinsic parameters of the projector and the camera. We project ArUCo markers on a moving surface and use a novel method that leverages the captured depth and color of the known projected features to achieve a completely automated, highly accurate, and fast (around a minute) projector-camera calibration without the use of any physical props. This does not need device synchronization and allows quick recalibration anytime the devices move.

2. Consumer depth cameras capture low resolution depth data that are often quite noisy. Not only does this create jitter during rendering, it also impacts the accuracy of the device calibration. We use a novel 3D plane fitting method that leverages the planar constraints imposed by the pinhole camera model to initialize the calibration procedure yielding intrinsic and extrinsic parameters of significantly higher accuracy.
3. Finally, we present a new content rendering method that handles both jitter and occlusion. For jitter, we employ Kalman filtering on the depth samples of the surface and smooth them over time to determine the surface shape. The Kalman filter is an efficient, recursive filter that uses a series of measurements over time to produce estimates and predictions of unknown variables (in our case, the true surface depth). This results in a comprehensible and perceptibly pleasing display. Leveraging the depth prediction by the Kalman filter, we devise an algorithm to handle occlusion of the surface by obstructing objects.

The above contributions result in the first high quality DPM system that can be deployed quickly; used for high-quality, jitter-free stable and legible data display in the presence of occlusions; and quick recalibration in the face of device movements – a system that can be deployed and used with confidence in expeditionary situations.

## 2 RELATED WORK

### 2.1 Device calibration

We discuss the large body of literature for projector-camera calibration, though not all are focused on DPM. Most prior works use a RGB camera. Some, like us, use a RGB-D camera (e.g. Kinect) and is considered as a single unit since the RGB and the Depth cameras are calibrated by the manufacturers to provide registered depth and color images. Further, there are different types of calibration methods. In some methods [1, 53], projector calibration is dependent on the results of camera calibration and therefore error in camera calibration is propagated to the projector parameters. However, other methods [29, 54] perform camera and projector calibration independently.

*Single Projector-RGB camera System:* Single projector-camera systems usually use structured light scanning (single or multi-pattern) of a known calibration object to establish pixel correspondences between the projector and camera followed by calibration [1, 7, 10, 11, 22, 29, 38, 44, 53, 54]. In [7, 10, 11, 22, 29], the calibration object, a checkerboard pattern printed on a planar surface, is first used to calibrate the camera using Zhang's method [55]. The projector is treated like an inverse camera [43] and is calibrated like a camera but using the projector-camera pixel correspondences generated from the structured light scan.

Methods that use only planar projection surface [1, 44, 54] need not rely on structured light scanning to generate pixel correspondences. Instead, they estimate a homography between the camera and projector using a rigid planar calibration board by pre-warping the projected image to align with printed fiducials on the calibration board [1, 44, 54]. Other methods [27, 51, 53] lift the restriction of

a planar calibration object, instead relying on objects of arbitrary geometry to perform calibration. This also allows these methods to project on non-planar surfaces.

*Single Projector, Single RGB-D camera System:* Resch et.al [38] use structured light scanning and a precise 3D mesh of an arbitrary object, obtained apriori by a laser scan, to iteratively refine the calibration parameters of the projector-camera system.

*Multi-Projector, Single RGB camera System:* Using measured priors (e.g. aspect ratio, size of radius) for specific non-planar surfaces (e.g. vertically extruded surfaces, swept surfaces or spherical surfaces) [39–42] use a single uncalibrated camera to find the device parameters in a multi-projector system. The priors help in constraining the optimization thereby removing the use of any special calibration pattern.

*Multi-Projector, Single RGB-D camera System:* Kurth et.al [23] use a depth camera to calibrate their multi-projector system used for dynamic projection mapping. They use a precise 3D mesh of an arbitrary complex object (e.g. a human bust) recovered via a laser scan to constrain the optimization process to achieve the parameters for the devices.

*Multi-Projector, Multi RGB-D camera System:* In a follow-up work [24] use multiple depth cameras, each connected to its own machine, to calibrate a dynamic multi-projector system in a distributed manner to support a large number of projectors. They identify various sources of error that cause visual artifacts in a DPM system and provide a *content-aware* method that seeks to hide these errors. During runtime, they use the depth camera connected to the main server to track the target object and perform DPM.

*Multi-Projector, Multi RGB-Camera Systems:* Explicit calibration of devices is not required on planar multi-projection systems and registration is achieved via homography based methods [3–5]. When handling multi-projector systems on non-planar surfaces, usually single projector-camera calibration methods are used to calibrate different device pairs separately and then connect them in a tree-like fashion to calibrate all the devices [35–37, 49]. However, errors creep in due to each device being calibrated separately and getting sub-pixel accuracy in registration becomes impossible. Further, anytime the devices are moved, a new calibration has to be performed bringing in the rigid 2D/3D calibration pattern.

*Comparison:* [48, 51] are the only works that achieve automated calibration (without the use of specific 2D or 3D props) of multiple projectors and RGB cameras projecting on a complex rigid 3D shape. Our work does the same for moving surfaces while using a RGB-D camera, enabling auto-calibration and quick recalibration.

Our work focuses on a single pipeline for automated calibration and rendering for DPM for quick and easy deployment of DPM systems. The sub-pixel accuracy achieved by our system is unmatched by any other prior system. Driven by the application of creating high quality visualizations on inside walls of tents in the presence of environmental elements (e.g. wind), we have provided several quantitative evaluations to show that the display created by the proposed DPM system indeed is a practical viable solution for highly comprehensible displays even in the presence of large and rapid movements and deformations. All methods including ours, with the exception of [24], present a content agnostic rendering.

### 2.2 Occlusion Handling

Methods for occlusion handling for projection mapping displays differ based on whether the object is static-rigid, dynamic-rigid or dynamic-deformable. For static-rigid and dynamic-rigid objects whose the shape is known apriori, multi-projector displays typically focus on identifying occluders for shadow removal by increasing the contribution of one or more projectors to compensate for an occluded projector in an overlapping region. However, for dynamic-deformable objects, the system must extract the surface shape every frame which can get affected by an occluder.

[18,19,47] predict an expected occluded projection and the actual captured occluded projector to identify shadows and remove them. However, these systems are not suitable for complex, deformable surfaces. [26,33] perform DPM using multiple projectors on a rigid object of known 3D shape. In order to remove shadows on the object caused by occlusions, they track occluding objects in the projection volume using a depth camera and adjust the blend weights of other projectors to compensate for the blocked projector. Pmomo [57] detects occluding objects in a single projector DPM system for rigid objects to cull projection on the occluder for a more realistic experience.

Flexpad [45] performs DPM on a deformable, non-stretchable surface that a user can interact with. They use optical surface material analysis on the raw IR image from the Kinect to distinguish skin pixels (e.g. the user’s hand) that may be occluding the surface. Narita et al. [32] perform DPM on a non-rigid object by printing a precise deformable dot cluster marker (DDCM) pattern on a surface using IR ink. These dot clusters are extracted, identified and tracked across frames using a high-speed camera ( $\sim 1000\text{fps}$ ) and compensated.

*Comparison:* Shadow removal is only pertinent to multiple projector systems. When compared with prior single projector systems that only detect specific occluders (e.g. skin) or need custom patterns, our method provides a general solution.

### 2.3 Jitter Correction

Jitter noise has been addressed in the domain of video captures. Tiny, high frequency motion caused by unstable camera support equipment or the user’s hand manifests as jitter which severely degrades the quality of videos. Therefore, most video stabilization techniques serve as a post-processing step to remove the jitter while preserving and smoothing the intentional motion of the camera. They do so by estimating the motion vectors in an image sequence, which can be achieved by pixel-based, block-matching or feature-based algorithms [9].

The Kalman filter has also been used to handle jitter. Jang et al. [16,17] use it to remove jitter noise from a sequence of images captured at various focuses for shape recovery, or shape-from-focus. Kwon et al. [25] compute motion vectors in an image sequence using phase correlation and smooth them over time using the Kalman filter. Yaman et al. [52] use the constant velocity Kalman filter for smoothing motion vector trajectories for image stabilization. More recently, some works have used deep learning to stabilize videos [6,56].

*Comparison:* The primary difference between prior works on jitter reduction for DPM and ours is its content-agnostic nature and real-time performance inspired by real-time video stabilization methods [20,21] using constrained constant velocity Kalman filter. Our method does not rely on the projected content to compute motion vectors. It smoothes frames as they come in and does not rely on future frames. Unlike Bermano et al. [2] that employ Kalman filter to smooth the relatively constrained movement of facial keypoints for DPM, we handle large and random movements of an unbridled fabric in the presence of different speeds of the wind. Furthermore, [2] use a coaxial projector-camera pair with a very specialized high-speed camera ( $\approx 1300\text{FPS}$ ) that result in very small movements in facial keypoints. Our method achieves this challenging goal using a consumer-grade camera operating at 30 FPS.

## 3 METHOD

Our setup consists of a projector and a RGB-D camera positioned to project on and capture a surface respectively that may be dynamic and deformable (Figure-1). We use an elastic tendo screen and use two fans on either side of the screen that can be set to three different speeds from slow to fast. We assume that the RGB-D

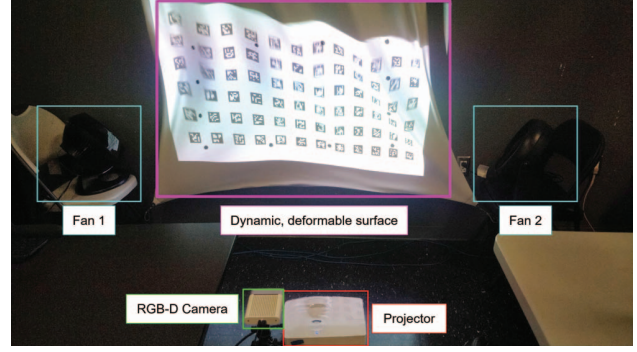


Figure 1: Our setup, comprising a RGB-D camera and a projector positioned towards a dynamic, deformable surface.

camera provides us with registered RGB image and depth mesh, in keeping with the norm for consumer RGB-D cameras.

### 3.1 Background

We model both the projector and camera with a pinhole camera model extended with radial and tangential distortion. We assume that the RGB-D camera’s center of projection (COP) is at the origin, its pose is aligned with the XYZ axis with the principal axis being along the positive Z axis. Let  $c \in \mathbf{R}^2$  represent the image of the 3D point  $d = [X \ Y \ Z]^T \in \mathbf{R}^3$  in the camera image plane. Then  $d$  and  $c$  are related by  $c = K_c \cdot \Lambda(\tilde{c})$  where  $\tilde{c} = [\tilde{c}_x \ \tilde{c}_y \ 1]^T = [\frac{x}{Z} \ \frac{y}{Z} \ 1]^T$  and  $K_c \in \mathbf{R}^{3 \times 3}$  is the camera intrinsic matrix and  $\Lambda(\cdot) \in \mathbf{R}^{3 \times 1}$  is the lens distortion function. The  $(3 \times 4)$  extrinsic parameter matrix is identity. Mathematically,  $K_c$  and  $\Lambda(\tilde{c})$  are given by:

$$K_c = \begin{bmatrix} f_x & 0 & n_x \\ 0 & f_y & n_y \\ 0 & 0 & 1 \end{bmatrix}; \quad (1)$$

$$\Lambda(\tilde{c}; D_c) = \begin{bmatrix} \tilde{c} \cdot \rho(\tilde{c}) + \tau(\tilde{c}) \\ 1 \end{bmatrix}; \quad (2)$$

$$\rho(\tilde{c}) = (1 + k_1 r^2 + k_2 r^4 + k_3 r^6) \quad (3)$$

$$\tau(\tilde{c}) = \begin{bmatrix} 2k_4 \tilde{c}_x \tilde{c}_y + k_5 (r^2 + 2\tilde{c}_x^2) \\ k_4 (r^2 + 2\tilde{c}_y^2) + 2k_5 \tilde{c}_x \tilde{c}_y \end{bmatrix}; \quad (4)$$

$$r^2 = \tilde{c}_x^2 + \tilde{c}_y^2; \quad (5)$$

where  $(f_x, f_y)$  is the focal length,  $(n_x, n_y)$  is the camera principle point,  $D_c = \{k_1, k_2, k_3, k_4, k_5\}$  are the distortion coefficients;  $(k_1, k_2, k_3)$  are radial distortion coefficients and  $(k_4, k_5)$  are the tangential distortion coefficients.

We assume the same pinhole camera model for the projector, and its rotation and translation with respect to the camera to be  $R_p \in \mathbf{R}^{3 \times 3}$  and  $T_p \in \mathbf{R}^{3 \times 1}$  respectively. Let  $p$  denote the projector pixel that illuminates the 3D point  $d$ . Then:

$$d' = [X' \ Y' \ Z']^T = R_p \cdot d + T_p \quad (6)$$

$$\tilde{p} = [\tilde{p}_x \ \tilde{p}_y \ 1]^T = \begin{bmatrix} X' \\ Y' \\ Z' \end{bmatrix} \quad (7)$$

$$p = K_p \cdot \Lambda(\tilde{p}; D_p) \quad (8)$$

where  $K_p, D_p$  are the projector intrinsic matrix and distortion coefficients respectively.

### 3.2 Image Acquisition

Calibration of the project-camera unit requires finding correspondences between 3D coordinate  $d_i$ , the 2D camera pixel  $c_j$  that images

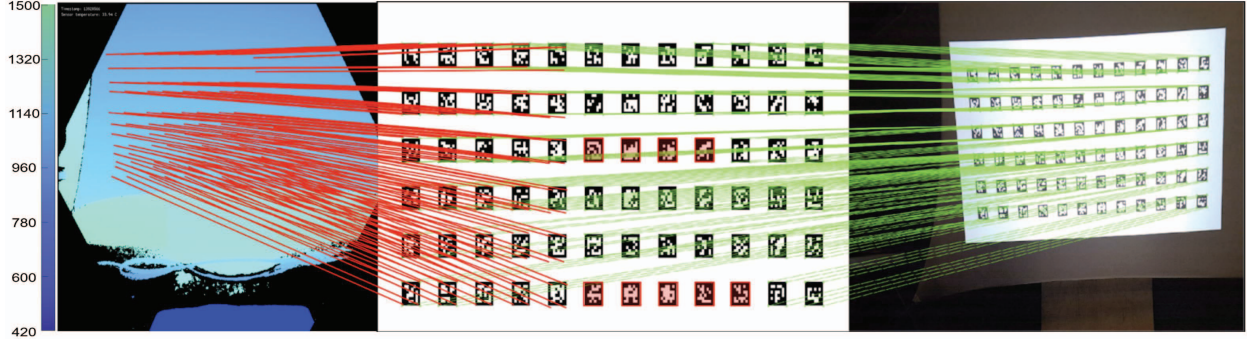


Figure 2: (Left) The depth map (in mm), (Middle) the projected image and (Right) the camera image. The corresponding pixels at the corners of the ArUCo markers between the projector, depth map and camera image are shown by the red and green lines respectively. Markers that were not detected in the camera image are highlighted in red.

it, and the corresponding 2D projector pixel  $p_i$  that illuminates it. Traditional systems usually use multi-frame structured light scan (SLS) methods to find the correspondences between  $c_i$  and  $p_i$  from which  $d_i$  is deciphered. However, in the proposed method, RGB-D camera directly provides us the correspondence between  $d_i$  and  $c_i$ . Also, we use a pattern akin to single-shot SLS patterns [10, 11] to link these correspondences to  $p_i$ .

Our pattern is based on ArUCo markers [8]. Each ArUCo marker comprises of a black and white square with a unique pattern and ID. When captured by a camera, an image containing ArUCo markers can be decoded to find the IDs and the four corners of the markers. We project a *sequence of grids of ArUCo markers* with known IDs and corners onto a dynamic, deformable surface. The RGB-D camera captures the projected sequence while the ArUCo marker detection algorithm detects the locations and IDs of the markers in that image and establishes correspondences between the detected  $c_i$ , its corresponding  $d_i$  and the known location of the detected points in the projector  $p_i$  (Figure-2).

Each image in the sequence is projected for several frames. Since the deformable projection surface moves during these frames, we can obtain a large number of  $c_i \rightarrow d_i$  correspondences even when using a single grid of ArUCo markers. Across different grids, we design the location of the corners and the IDs of the ArUCo markers to be different to increase the number of correspondences even further. Unique IDs across grids also prevents incorrect correspondences due to asynchronous projection and capture of the patterns if the camera and projector are not synchronized. We perform marker corner detection at sub-pixel accuracy in all the images captured by the RGB camera and use the depth from the depth camera at the registered location to establish the correspondence between  $c_i$ ,  $d_i$  and  $p_i$ .

### 3.3 Camera Calibration

Since our camera extrinsic matrix is identity, we need to only determine the camera intrinsic matrix and distortion coefficients using the correspondences. Let  $\{c_i^t, d_i^t\}$  denote the  $i$ -th correspondence at time  $t$ . Let  $K_c^t$  and  $D_c^t$  denote the intrinsic matrix and distortion coefficients at time  $t$ . We optimize for  $K_c, D_c$  by minimizing the reprojection error  $E_c^t$  given by:

$$\arg \min_{K_c^t, D_c^t} E_c^t = \frac{1}{2} \sum_i (|c_i^t - K_c^t \cdot \Lambda(\tilde{c}_i^t; D_c^t)|)^2 \quad (9)$$

$$\tilde{c}_i^t = \frac{d_i^t}{Z_i^t} = [X_i^t/Z_i^t \quad Y_i^t/Z_i^t \quad 1]^T \quad (10)$$

The camera parameters are optimized every frame using per-frame correspondences, initializing the optimization with the param-

eters from the previous frame. For the first frame  $t = 1$ , we assume the distortion coefficients to be zero, and compute the intrinsic matrix  $K_c^1$  directly from the 2D-3D correspondences. If the camera API provides an initial estimate of the intrinsics, we only use the intrinsic matrix and ignore the distortion coefficients. We assume the optimization for a frame  $t$  has converged when any one of the two conditions are met: (1) the number of iterations is greater than 1000, or (2) the absolute difference between the reprojection errors in two successive iterations is less than a threshold, i.e.  $|E^t - E^{t-1}| \leq \Delta_c$ .

### 3.4 Projector Calibration

For the projector, we need to estimate the projector intrinsics and extrinsics (i.e. rotation and translation with respect to the camera) both. Let  $\{p_i^t, d_i^t\}$  denote the  $i$ -th correspondence between a 2D projector coordinate  $p_i^t$  and 3D point  $d_i^t$  at time  $t$ . Let  $K_p^t, D_p^t$  denote the intrinsic matrix and distortion coefficients, and  $R_p^t, T_p^t$  denote the rotation and translation at time  $t$ . We optimize for  $K_p, D_p, R_p, T_p$  by minimizing the reprojection error  $E_p^t$  as

$$\arg \min_{K_p^t, D_p^t, R_p^t, T_p^t} E_p^t = \frac{1}{2} \sum_i (|p_i^t - K_p^t \cdot \Lambda(\tilde{p}_i^t; D_p^t)|)^2 \quad (11)$$

$$\tilde{p}_i^t = [\tilde{X}_i^t/\tilde{Z}_i^t \quad \tilde{Y}_i^t/\tilde{Z}_i^t \quad 1]^T \quad (12)$$

$$[\tilde{X}_i^t \quad \tilde{Y}_i^t \quad \tilde{Z}_i^t]^T = (R_p^t [X_i^t \quad Y_i^t \quad Z_i^t]^T + T_p^t) \quad (13)$$

We optimize the projector parameters using Equation-11 considering correspondences from each frame separately, moving on to the next frame when convergence for the current frame is reached. Optimizing likewise across multiple frames averages out random noise providing an accurate calibration despite noise in the depth camera.

However, Equation-11 is sensitive to initialization of the projector parameters. Poor initialization can result in the optimization getting stuck in local minima, resulting in a projection that will align with the surface only within a narrow depth range. For DPM systems where the surface may be moving in a large range, this can be problematic. Therefore, we devise a new method to obtain an initial estimate of the projector parameters to initialize the optimization in Equation-11 assuring robust convergence in optimization.

**Coarse Projector Parameter Estimation:** Pixels on any straight line in the projector coordinates correspond to a plane in 3D (Figure-3). Leveraging this property, we fit a plane through the 3D points corresponding to 2D pixel locations of ArUCo marker corners along every row, column and diagonal across all grids of markers. Intersecting all the planes along rows results in a line whose direction corresponds to the projector's X-axis. Intersection of all planes along columns results in a line along the projector's Y-axis. Taking

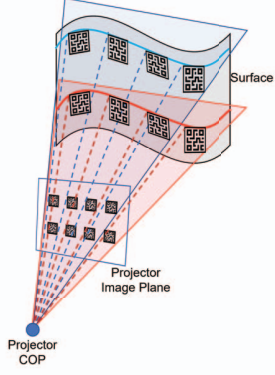


Figure 3: Illustration of plane fitting through 3D points of ArUCo marker corners on the surface that correspond to the same projector pixel rows. Intersecting all row planes gives the X-axis of the projector. Intersecting all column planes gives the Y-axis of the projector. The Z-axis is the cross product of the two. The projector COP is obtained by intersecting all row, column and diagonal planes.

the cross product of these two vectors gives the Z-axis. Thus, we determine the projector orientation  $R_p^0$  with respect to the camera. The projector center-of-projection (COP)  $T_p^0$  is estimated by finding the intersection of all the planes along the row, column and diagonals. Using this initial estimate of the projector orientation and COP, we compute  $K_p^0$ , assuming zero distortion.

**Projector Parameter Refinement:** We begin the optimization in Equation-11 with this more reasonable initialization. Note that if the 3D location of the projected points corresponding to markers along any row, column or diagonal are collinear, it will present a degenerate case to the plane fitting method. Therefore, this method will fail for static planar projection surface. In our setup (Figure-1) using a hanging fabric screen, such a scenario is unlikely.

### 3.5 Jitter Reduction

Following calibration, we use the recovered system parameters to adapt the projection to the dynamic, deformable surface using [12, 13]. At each frame, we find the 3D location of the equidistant black markers marking the desired rectangular area of projection and fit a B-spline to the detected region 3D. Using the recovered projector parameters, we then warp the projected image by projecting the B-spline surface into projector image plane. However, noise in the depth camera results in small perturbations in the fitted B-spline surface and the warped projection has jitters i.e. small, high frequency motion. This jitter is perceptually annoying and, more importantly, can render the display unreadable or incomprehensible. Therefore, we employ Kalman filtering on the 3D B-spline surface to account for the depth camera noise and reduce jitter.

The Kalman filter uses a series of measurements observed over time and produces estimates and predictions of the system state by taking into account the uncertainties of the system and the measurements. This results in a more accurate estimate of the system state than by considering a single measurement only. It consists of a predict-correct loop. Once initialized, it predicts the system state, denoted by  $x \in \mathbf{R}^{N_x \times 1}$  and uncertainties in the prediction at the next timestep. When a measurement, denoted by  $z \in \mathbf{R}^{N_z \times 1}$  is received, it updates (or corrects) the prediction of the current state to produce a new estimate. Mathematically, the predict step can be written as

$$\hat{x}_{t+1,t} = \mathbf{F}\hat{x}_{t,t} \quad (14)$$

$$\mathbf{P}_{t+1,t} = \mathbf{F}\mathbf{P}_{t,t}\mathbf{F}^T + \mathbf{Q} \quad (15)$$

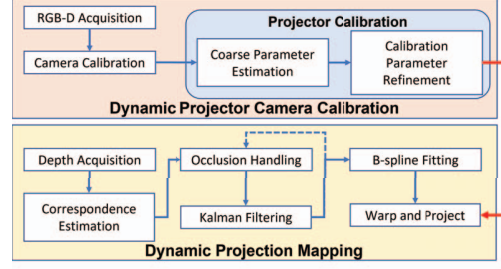


Figure 4: The flowchart of the proposed system.

where  $\hat{x}_{t+1,t}$  is the predicted state vector at time  $t + 1$ ,  $\hat{x}_{t,t}$  is the estimated state at time  $t$ .  $\mathbf{F} \in \mathbf{R}^{N_x \times N_x}$  is the state transition matrix that extrapolates the current state to the next timestep.  $\mathbf{P} \in \mathbf{R}^{N_x \times N_x}$  is a covariance matrix and represents the estimate uncertainty. A higher estimate of uncertainty will cause the Kalman filter to favor the measurement more than the current state in the estimation.  $\mathbf{Q} \in \mathbf{R}^{N_x \times N_x}$  is the process noise uncertainty.

In the update step, the Kalman filter computes the Kalman gain which weights the prediction and the current measurement to compute an estimate for the current system state. Mathematically,

$$\mathbf{K}_t = \mathbf{P}_{t,t-1}\mathbf{H}^T(\mathbf{H}\mathbf{P}_{t,t-1}\mathbf{H}^T + \mathbf{R}_t)^{-1} \quad (16)$$

$$x_{t,t} = x_{t,t-1} + \mathbf{K}_t(z_t - \mathbf{H}x_{t,t-1}) \quad (17)$$

$$\mathbf{P}_{t,t} = (\mathbf{I} - \mathbf{K}_t\mathbf{H})\mathbf{P}_{t,t-1}(\mathbf{I} - \mathbf{K}_t\mathbf{H})^{-1} + \mathbf{K}_t\mathbf{R}_t\mathbf{K}_t^T \quad (18)$$

where  $\mathbf{K} \in \mathbf{R}^{N_x \times N_x}$  is the Kalman gain,  $\mathbf{H} \in \mathbf{R}^{N_z \times N_x}$  is the observation matrix that maps the state vector to measurement vector space and  $\mathbf{R} \in \mathbf{R}^{N_z \times N_z}$  is the measurement uncertainty while  $\mathbf{I}$  is the identity matrix. The subscript  $t$  denotes the timestep.

In our case, the measurements  $z$  are the 3D points on the surface from the depth camera that is used to compute the B-spline surface. Our goal is to compute a more accurate estimate of these 3D points. Therefore, our system state vector  $x$  consist of all the 3D points used to compute the B-spline surface. We have implemented three different Kalman filter models based on the assumptions made by the prediction step: *the constant velocity*, *constant acceleration* and *constant jerk*. Depending on the model being used, the system state also comprises the velocity, acceleration or jerk of each 3D point. The *constant velocity* model assumes that the predicted location only relies on the velocity of the 3D point that does not change between successive frames. The *constant acceleration model* accounts for acceleration and velocity to predict the 3D location assuming that the predicted acceleration remains constant. The *constant jerk* model adds the jerk to the prediction, assuming that the predicted jerk remains constant. These three models therefore have different system states and state transition matrices. The constant velocity model has only the location and velocity of the 3D point in the state vector, while the constant acceleration and constant jerk models additionally include acceleration and jerk respectively. Irrespective of the model used, we only use the estimated location from the state vector at each timestep. The state transition matrix implements the equations of motion. At each frame, we acquire measurements of the surface from the depth camera. These measurements are fed into the Kalman filter which produces a more accurate estimate of each sampled 3D point on the surface which are then used to compute the B-spline and warp the content for projection.

### 3.6 Occlusion Handling

We leverage the Kalman filter prediction of the 3D points on the B-spline surface during jitter reduction to handle surface occlusion. We first use methods by [12, 13] to detect the 3D location of markers

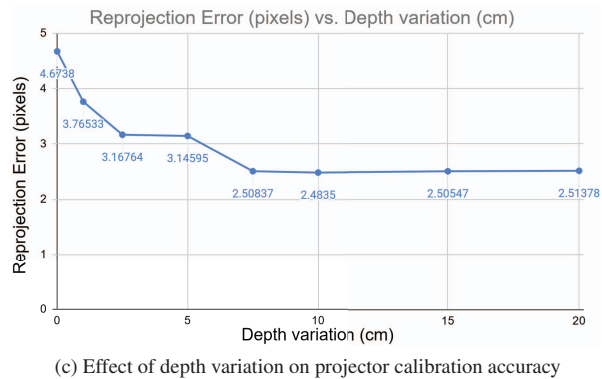
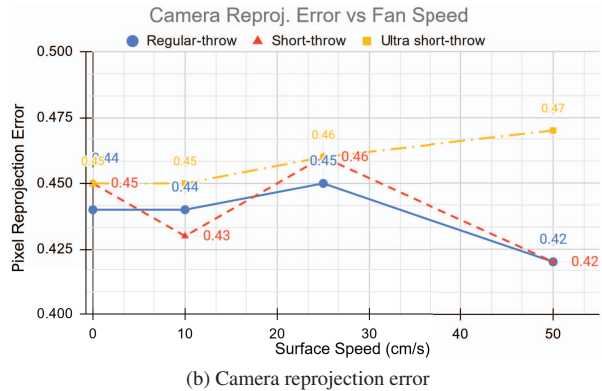
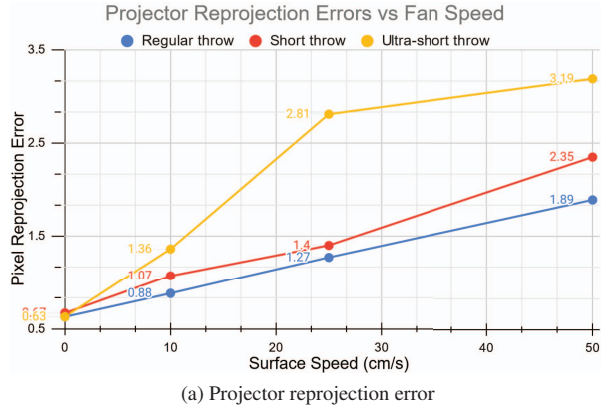
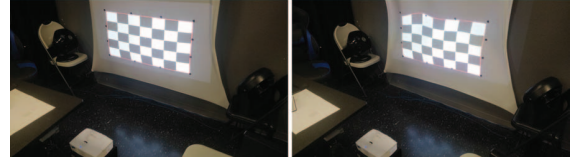
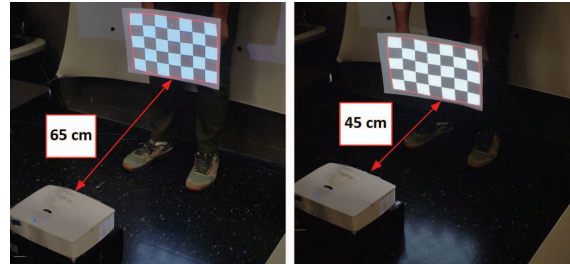


Figure 5: Effect of the surface speed on the calibration accuracy for (a) projector and (b) camera for different types of projectors. (c) shows the effect of depth variation on the projector calibration accuracy.

at the boundary of the projection region and all the 3D points within the desired boundary and compute the B-spline surface. In occlusion handling step we identify the region of the surface that is occluded by detecting the occluded points from the 3D points used for the B-spline surface. These occluded points are then replaced by the Kalman filter prediction. Assuming that an occluding object will always have depth that is less than that of the projection surface, we compute the minimum depth that was sampled in the B-spline region in the previous frame. In the current frame, we mask out any regions whose depth is less than the minimum depth (from the previous frame) by a certain threshold.



(a) Marker-based surface



(b) DPM outside calibration volume

Figure 6: DPM on (a) a marker-based and (b) a markerless deformable surface using our calibration. Note how the projection remains aligned with the markerless surface in (b) despite moving out of the calibration volume.

**Detecting Occluded Markers:** In order to determine which sampled points are occluded, we first start with the surface boundary. Our goal is to match each boundary point in the current frame with the detected boundary points in the previous frame. We do this by matching each boundary point in the previous frame to the closest 3D point in the current frame. If a boundary marker in previous frame is not matched, it is labeled as occluded and replaced by its Kalman filter prediction from the previous frame.

**Accounting for global movement:** Detecting occluding markers alone cannot take into account changes to the global position of the entire surface in the current frame. For example, the surface may move forward in unison. This global movement may not be correctly predicted by the Kalman filter and can cause the display to become distorted. Therefore, we additionally compute a rigid transform that transforms 3D points in the previous frame to the points in the current frame in a least-squares sense. For this, we use all the points – both unoccluded and also the occluded ones that are predicted by the Kalman filter. We then apply this rigid transform to points in the previous frame corresponding to occluded points in the current frame to get an estimate of the occluded 3D points in the current frame. Though not very accurate, computing a rigid transform is fast and efficient, does not result in extrapolation artifacts that nonlinear transforms may exhibit, and does not compromise the real-time performance of the display.

#### 4 IMPLEMENTATION AND RESULTS

Figure-4 shows the flowchart of our method. First, we calibrate the projector-camera pair. We project and detect ArUCo markers in the RGB-D capture to determine the 2D-3D correspondences for camera calibration. Next, we fit planes through the 3D points that correspond to the same projector row, column and diagonal to get a coarse estimate of the projector parameters. Finally, we refine the projector parameters to get an accurate calibration. With the calibration parameters, we perform DPM using Ibrahim et al.'s [12, 13] method, augmenting it with the proposed jitter control and occlusion handling methods.

We implemented the proposed system in C++. We used OpenCV's APIs for detecting ArUCo markers with sub-pixel accuracy and Levenberg-Marquadt optimization to implement our

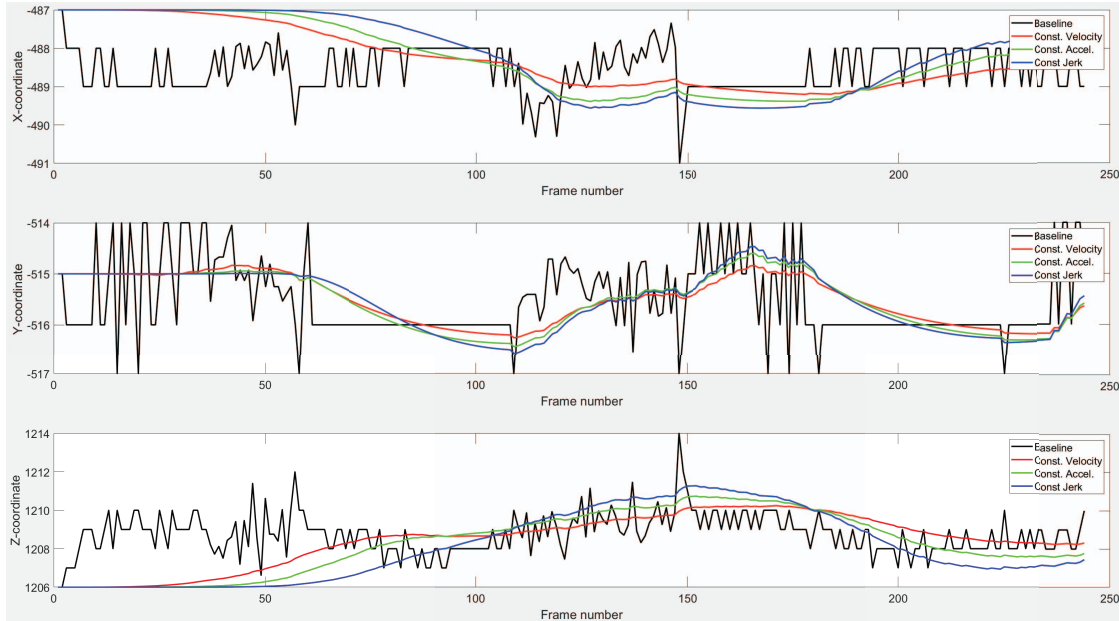


Figure 7: Kalman filter smoothing for a stationary 3D point over 250 frames. Note the amount of noise in the data from the depth camera (black). While all three models smooth it out, the constant velocity filtering (red) varies the least compared to the other two.

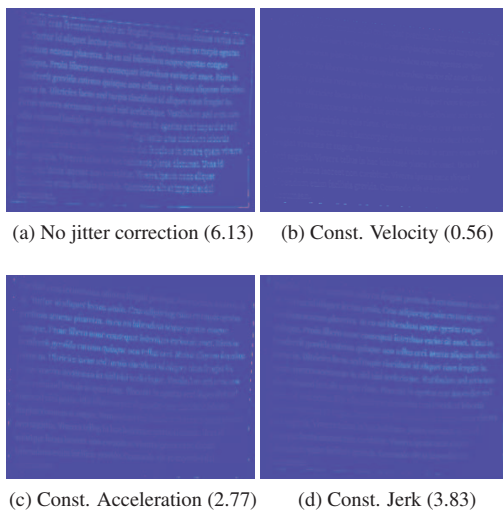


Figure 8: Maps of average absolute difference across a number of successive frames when projecting a static text image on a stationary projection surface. The average absolute differences are (a) 6.13, (b) 0.56, (c) 2.77 and (d) 3.83. Note how the difference is very low for constant velocity model shown in (b).

calibration routines. For the camera, we used an Azure Kinect. We tested our method on three different kinds of projectors - regular throw (Optoma ML750), short throw (Optoma EH200ST) and ultra-short throw (Optoma GT5600). The Azure Kinect itself comprises of two cameras: a time-of-flight IR/depth camera and an RGB camera. Although the depth camera resolution coarser than the RGB camera i.e.  $(640 \times 576)$  vs.  $(3840 \times 2160)$ , its API provides an interpolated point cloud rendered from the RGB camera view, similar to other consumer-grade depth cameras like the Intel Re-

alSense Depth Cameras [14]. In our calibration routine, we used 3D points from this interpolated point cloud. The projector resolution was  $(1920 \times 1080)$ . We projected a grid of ArUCo markers for 20 frames before switching to another grid layout.

#### 4.1 Projector-Camera Calibration Accuracy

We tested our proposed calibration method on a dynamic, deformable surface. The projector-camera system was placed around 864mm away from the screen. The width and height of the projection was 1270mm and 762mm respectively. This gives us a working volume of around 1270W x 760H x 865D cubic mm. We placed two fans on either side of the surface which would generate random waves and ripples across the surface and tested our calibration for various speeds of the fans (see Figure-1). This helped us to study the impact of the movement of the surface on the accuracy of our calibration technique and the quality of our display. In order to estimate the surface speed, we took the difference between two successive depth maps to measure the velocity and took the average value. We quantified calibration accuracy using reprojection error onto camera and projector images using the calibration parameters (Figure-5). Note that the camera reprojection error is less than 0.5 pixel and does not get impacted by the movement of the surface (see Figure-5b). A error of less than 0.5 pixel is considered good calibration as per best practices [29]. Even at the highest speed, our projector reprojection error is less than 4 pixels - less than 0.4% when considering the size of the projection image (see Figure-5a). The maximum 3D reconstruction error is 3.7mm which is less than 0.4% considering the size of the volume we are operating in. Considering the variation of the different parameters estimated by the calibration process (including projector and camera focal length, translation and rotation), we had an average standard deviation of less than 5%.

Since our method exploits surface movement for projector calibration, we studied the impact of depth variation in the 3D data on the calibration accuracy. We captured a calibration dataset with large movement in depth ( $\approx 30cm$ ). From this dataset, we selected a subset of 3D points by enforcing a limit on the amount of depth variation, performed projector calibration and measured the reprojection

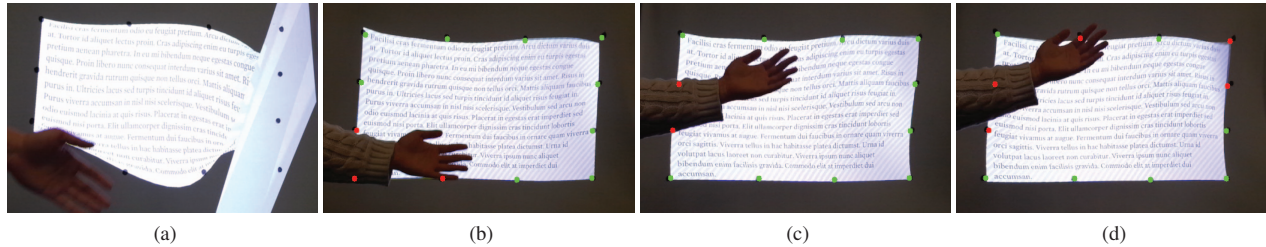


Figure 9: Occlusion of a moving surface by a hand moving across it. (a) shows the distorted projection without any occlusion handling. (b)-(d) Green points are detected points while red points are estimated points due to occlusion. Notice that sometimes, even unoccluded points are not detected by the camera (points on the right edge in (d)) but our algorithm is able to determine their position accurately.

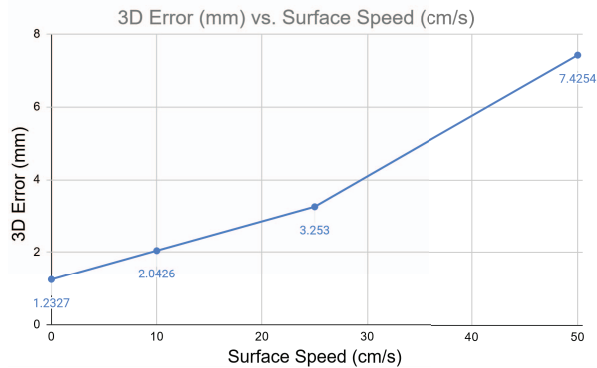


Figure 10: Average error between 3D points predicted using our occlusion handling method with their ground truth locations at various surface speeds.

error on this subset. Figure-5c shows the effect of depth variation on the projector reprojection error. Notice that for low depth variation, the reprojection error is high ( $> 3$  pixels) and lowers with increasing depth variation. This is because we get a better coarse parameter estimation with greater depth variation and more constraints during the optimization. However, the reprojection error remains largely unaffected for depth variation greater than 7cm. Therefore, a minimum depth variation of 7cm is recommended for the calibration.

Finally, we used the calibration parameters generated by our method to perform DPM on a dynamic, deformable surface. Note that DPM requires a very high level of calibration accuracy because the surface geometry is constantly changing. If the calibration does not accurately match the physical arrangement, then the projection will deviate from the marked region and show distortions. Figure-6a-6b shows the DPM display using calibration parameters generated by our method. Our calibration method is robust and able to generate calibration parameters that are valid even outside the calibration volume. Figure-6b shows a user moving a markerless display surface very close to the projector, well out of the calibration volume, which was approximately 85 cm away from the projector. Notice how the projection still remains well aligned within the surface using our calibration parameters. Note that methods like Moreno et al. [29] are not able to achieve this level of accuracy since their calibration parameters are valid for a small depth range. As the surface moves out of this range, the projection starts to misalign. In our setup, Moreno et al.'s method showed a 12% deviation from the marked region compared to at most 1% deviation using our method. This high calibration accuracy also enables high projection accuracy even when the surface goes significantly out of the calibration volume, whereas Moreno et al.'s calibration results in an additional 5% deviation.

As for the number of frames used for the calibration, we did not see any significant improvement to the calibration accuracy for more than 25 frames. In general, we capture around 15-25 frames for a successful calibration irrespective of the surface speed, which takes approximately 30 seconds to complete.

## 4.2 Jitter And Occlusion

The results with jitter and occlusion corrections are visibly less jittery compared to the display generated using the raw depth measurements. Figure-7 shows how the three filter types smooth out a jittery trajectory of a stationary 3D point. Figure-8 shows the average absolute difference maps over successive frames for each model when projecting static content on a stationary surface.

Figure-9 shows the occlusion of a rapidly moving surface by a hand. Notice how the projection gets severely distorted without any occlusion handling. However, our occlusion handling algorithm estimates the locations of the occluded points reasonably well to keep the projection coherent. Also notice that some points may not get detected by the camera even though they are unoccluded (see Figure-9d). These points are treated as occluded points and our algorithm estimates their positions.

To evaluate the accuracy of our occlusion handling method, we programmatically occluded some surface markers that were detected by our algorithm, estimated their locations using our occlusion handling method and then computed the distance from their true locations. We computed the average distance over multiple frames for several points. Figure-10 shows the error between the true locations and the estimated locations using our method for different surface speeds. Note that the error for a static surface is 1.2mm, which is not perceptible to viewers. With higher surface speeds, the error increases, reaching a maximum of 7.5mm when the surface movement resembles a strong gust of wind. Even at such high speeds, the mean error is less than a centimeter and viewers do not perceive it in the final display.

## 5 SUMMARY

In summary, we present the first work creating a stable, legible and easily deployable DPM system on dynamic and deformable surface using a completely automated projector-camera calibration and rendering with Kalmar filter based occlusion and jitter removal. In the future we plan to perform user studies to understand the extent of legibility achieved. We also plan to extend our work to multiple tiled projectors on dynamic deformable surfaces to create large seamless displays.

## 6 ACKNOWLEDGEMENT

This work was funded by a subcontract from US Air Force AFWERX SBIR funding from Summit Technology Laboratory (STL), Irvine, CA.



## REFERENCES

- [1] S. Audet and M. Okutomi. A user-friendly method to geometrically calibrate projector-camera systems. In *2009 IEEE computer society conference on computer vision and pattern recognition workshops*, pp. 47–54. IEEE, 2009.
- [2] A. H. Bermamo, M. Billeter, D. Iwai, and A. Grundhöfer. Makeup lamps: Live augmentation of human faces via projection. *Computer Graphics Forum*, 39(2):311–323, 2017.
- [3] E. Bhasker, R. Juang, and A. Majumder. Registration techniques for using imperfect and partially calibrated devices in planar multi-projector displays. *IEEE Transactions on Visualization and Computer Graphics*, 13(6):1368–1375, 2007.
- [4] E. S. Bhasker, P. Sinha, and A. Majumder. Asynchronous distributed calibration for scalable and reconfigurable multi-projector displays. *IEEE Transactions on Visualization and Computer Graphics*, 12(5):1101–1108, 2006.
- [5] H. Chen, R. Sukthankar, G. Wallace, and T. J. Cham. Calibrating scalable multi-projector displays using camera homography trees. *CVPR Technical Sketch*, 2001.
- [6] J. Choi and I. S. Kweon. Deep iterative frame interpolation for full-frame video stabilization. *ACM Transactions on Graphics (TOG)*, 39(1):1–9, 2020.
- [7] Z. Feng, D. Man, and Z. Song. A pattern and calibration method for single-pattern structured light system. *IEEE Transactions on Instrumentation and Measurement*, 69(6):3037–3048, 2019.
- [8] S. Garrido-Jurado, R. Muñoz-Salinas, F. J. Madrid-Cuevas, and M. J. Marín-Jiménez. Automatic generation and detection of highly reliable fiducial markers under occlusion. *Pattern Recognition*, 47(6):2280–2292, 2014.
- [9] W. Guilluy, L. Oudre, and A. Beghdadi. Video stabilization: Overview, challenges and perspectives. *Signal Processing: Image Communication*, 90:116015, 2021.
- [10] B. Huang, S. Ozdemir, Y. Tang, C. Liao, and H. Ling. A single-shot-perpose camera-projector calibration system for imperfect planar targets. In *2018 IEEE International Symposium on Mixed and Augmented Reality Adjunct (ISMAR-Adjunct)*, pp. 15–20. IEEE, 2018.
- [11] B. Huang, Y. Tang, S. Ozdemir, and H. Ling. A fast and flexible projector-camera calibration system. *IEEE Transactions on Automation Science and Engineering*, 18(3):1049–1063, 2020.
- [12] M. T. Ibrahim, A. Majumder, and M. Gopi. Dynamic projection mapping on deformable stretchable materials using boundary tracking. *Computers & Graphics*, 103:61–74, 2022.
- [13] M. T. Ibrahim, G. Meenakshisundaram, and A. Majumder. Dynamic projection mapping of deformable stretchable materials. In *26th ACM Symposium on Virtual Reality Software and Technology*, pp. 1–5, 2020.
- [14] Intel. Intel depth camera d435. <https://www.intelrealsense.com/depth-camera-d435/>. Accessed: 2022-09-11.
- [15] M. Iuchi, Y. Hirohashi, and H. Oku. Proposal for an aerial display using dynamic projection mapping on a distant flying screen. *2023 IEEE Conference Virtual Reality and 3D User Interfaces (VR)*, 2023.
- [16] H.-S. Jang, M. S. Muhammad, and T.-S. Choi. Removal of jitter noise in 3d shape recovery from image focus by using kalman filter. *Microscopy research and technique*, 81(2):207–213, 2018.
- [17] H.-S. Jang, M. S. Muhammad, and T.-S. Choi. Optimal depth estimation using modified kalman filter in the presence of non-gaussian jitter noise. *Microscopy Research and Technique*, 82(3):224–231, 2019.
- [18] C. Jaynes, S. Webb, R. Steele, M. Brown, and W. Seales. Dynamic shadow removal from front projection displays. In *Proceedings Visualization, 2001. VIS '01.*, pp. 175–555, 2001. doi: 10.1109/VISUAL.2001.964509
- [19] C. Jaynes, S. Webb, and R. M. Steele. Camera-based detection and removal of shadows from interactive multiprojector displays. *IEEE Transactions on Visualization and Computer Graphics*, 10(3):290–301, 2004.
- [20] C. Jia and B. L. Evans. Online motion smoothing for video stabilization via constrained multiple-model estimation. *EURASIP Journal on Image and Video Processing*, 2017(1):1–13, 2017.
- [21] N. Joshi, W. Kienzle, M. Toelle, M. Uyttendaele, and M. F. Cohen. Real-time hyperlapse creation via optimal frame selection. *ACM Transactions on Graphics (TOG)*, 34(4):1–9, 2015.
- [22] R. Juárez-Salazar and V. H. Díaz-Ramírez. Flexible camera-projector calibration using superposed color checkerboards. *Optics and Lasers in Engineering*, 120:59–65, 2019.
- [23] P. Kurth, V. Lange, C. Siegl, M. Stamminger, and F. Bauer. Auto-calibration for dynamic multi-projection mapping on arbitrary surfaces. *IEEE transactions on visualization and computer graphics*, 24(11):2886–2894, 2018.
- [24] P. Kurth, M. Leuschner, M. Stamminger, and F. Bauer. Content-aware brightness solving and error mitigation in large-scale multi-projection mapping. *IEEE Transactions on Visualization and Computer Graphics*, pp. 3607–3617, 2022.
- [25] O. Kwon, J. Shin, and J. Paik. Video stabilization using kalman filter and phase correlation matching. In *ICIAIAR*, pp. 141–148, 2005.
- [26] V. Lange, C. Siegl, M. Colaianni, M. Stamminger, and F. Bauer. Robust blending and occlusion compensation in dynamic multi-projection mapping. In *Proceedings of the European Association for Computer Graphics: Short Papers*, p. 1–4. Eurographics Association, 2017. doi: 10.2312/egsh.20171000
- [27] F. Li, H. Sekkati, J. Deglint, C. Scharfenberger, M. Lamm, D. Clausi, J. Zelek, and A. Wong. Simultaneous projector-camera self-calibration for three-dimensional reconstruction and projection mapping. *IEEE Transactions on Computational Imaging*, 3(1):74–83, 2017.
- [28] Y. Mikawa, T. Sueishi, Y. Watanabe, and M. Ishikawa. Variolight: Hybrid dynamic projection mapping using high-speed projector and optical axis controller. In *SIGGRAPH Asia 2018 Emerging Technologies*, pp. 1–2, 2018.
- [29] D. Moreno and G. Taubin. Simple, accurate, and robust projector-camera calibration. In *2012 Second International Conference on 3D Imaging, Modeling, Processing, Visualization & Transmission*, pp. 464–471. IEEE, 2012.
- [30] G. Narita, Y. Watanabe, and M. Ishikawa. Dynamic projection mapping onto a deformable object with occlusion based on high-speed tracking of dot marker array. In *Proceedings of the 21st ACM symposium on virtual reality software and technology*, pp. 149–152, 2015.
- [31] G. Narita, Y. Watanabe, and M. Ishikawa. Dynamic projection mapping onto deforming non-rigid surface using deformable dot cluster marker. *IEEE transactions on visualization and computer graphics*, 23(3):1235–1248, 2016.
- [32] G. Narita, Y. Watanabe, and M. Ishikawa. Dynamic projection mapping onto deforming non-rigid surface using deformable dot cluster marker. *IEEE Transactions on Visualization and Computer Graphics*, 23(3):1235–1248, 2017. doi: 10.1109/TVCG.2016.2592910
- [33] T. Nomoto, W. Li, H. Peng, and Y. Watanabe. Dynamic multi-projection mapping based on parallel intensity control. *IEEE Transactions on Visualization and Computer Graphics*, pp. 2125–2134, 2022.
- [34] P. Punpongsonan, D. Iwai, and K. Sato. Flexeen: Visually manipulating perceived fabric bending stiffness in spatial augmented reality. *IEEE Transactions on Visualization and Computer Graphics*, 2018.
- [35] R. Raskar, M. Brown, R. Yang, W. Chen, G. Welch, H. Towles, B. Scales, and H. Fuchs. Multi-projector displays using camera-based registration. *IEEE Proceedings Visualization*, pp. 161–522, 1999.
- [36] R. Raskar, M. Cutts, G. Welch, and W. Stuerzlinger. Efficient image generation for multiprojector and multisurface displays. In *Eurographics Workshop on Rendering Techniques*, pp. 139–144. Springer, 1998.
- [37] 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 displays. In *Proceedings of the 25th annual conference on Computer graphics and interactive techniques*, pp. 179–188, 1998.
- [38] 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 Transactions on Visualization and Computer Graphics*, 21(11):1211–1220, 2015.
- [39] B. Sajadi and A. Majumder. Markerless view independent geometric registration of multiple distorted projectors on vertically extruded surfaces using a single uncalibrated camera. *IEEE Transactions on Visualization and Computer Graphics*, 10, 2009.
- [40] B. Sajadi and A. Majumder. Auto-calibration of cylindrical multi-projector systems. In *2010 IEEE virtual reality conference (VR)*, pp.

- 155–162. IEEE, 2010.
- [41] B. Sajadi and A. Majumder. Automatic registration of multiple projectors on swept surfaces. In *Proceedings of the 17th ACM Symposium on Virtual Reality Software and Technology*, pp. 159–166, 2010.
  - [42] B. Sajadi and A. Majumder. Autocalibrating tiled projectors on piecewise smooth vertically extruded surfaces. *IEEE Transactions on Visualization and Computer Graphics*, 17(9):1209–1222, 2011.
  - [43] P. Sen, B. Chen, G. Garg, S. Marschner, M. Horowitz, M. Levoy, and H. Lensch. Dual photography. *ACM SIGGRAPH 2005 Papers*, 2005.
  - [44] M. Shahpaski, L. Ricardo Sapaico, G. Chevassus, and S. Susstrunk. Simultaneous geometric and radiometric calibration of a projector-camera pair. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4885–4893, 2017.
  - [45] J. Steimle, A. Jordt, and P. Maes. Flexpad: highly flexible bending interactions for projected handheld displays. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pp. 237–246, 2013.
  - [46] T. Sueishi, H. Oku, and M. Ishikawa. Robust high-speed tracking against illumination changes for dynamic projection mapping. In *2015 IEEE Virtual Reality (VR)*, pp. 97–104. IEEE, 2015.
  - [47] R. Sukthankar, T.-J. Cham, and G. Sukthankar. Dynamic shadow elimination for multi-projector displays. In *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001*, vol. 2, pp. II–II. IEEE, 2001.
  - [48] M. A. Tehrani, M. Gopi, and A. Majumder. Automated geometric registration for multi-projector displays on arbitrary 3d shapes using uncalibrated devices. *IEEE Transactions on Visualization and Computer Graphics*, 27(4):2265–2279, 2019.
  - [49] M. Waldner, C. Pirchheim, and D. Schmalstieg. Multi projector displays using a 3d compositing window manager. *Emerging display technologies*, pp. 1–4, 2008.
  - [50] L. Wang, H. Xu, Y. Hu, S. Tabata, and M. Ishikawa. Dynamic depth-of-field projection for 3d projection mapping. In *Extended abstracts of the 2019 CHI conference on human factors in computing systems*, pp. 1–4, 2019.
  - [51] S. Willi and A. Grundhöfer. Robust geometric self-calibration of generic multi-projector camera systems. In *2017 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*, pp. 42–51. IEEE, 2017.
  - [52] E. Yaman and S. Ertürk. Image stabilization by kalman filtering using a constant velocity camera model with adaptive process noise. *Proc. of ELECO*, pp. 152–155, 2001.
  - [53] S. Yamazaki, M. Mochimaru, and T. Kanade. Simultaneous self-calibration of a projector and a camera using structured light. In *CVPR 2011 WORKSHOPS*, pp. 60–67. IEEE, 2011.
  - [54] L. Yang, J.-M. Normand, and G. Moreau. Practical and precise projector-camera calibration. In *2016 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*, pp. 63–70. IEEE, 2016.
  - [55] Z. Zhang. A flexible new technique for camera calibration. *IEEE Transactions on pattern analysis and machine intelligence*, 22(11):1330–1334, 2000.
  - [56] M. Zhao and Q. Ling. Pwstabenet: Learning pixel-wise warping maps for video stabilization. *IEEE Transactions on Image Processing*, 29:3582–3595, 2020. doi: 10.1109/TIP.2019.2963380
  - [57] Y. Zhou, S. Xiao, N. Tang, Z. Wei, and X. Chen. Pmomo: Projection mapping on movable 3d object. In *Proceedings Of The 2016 CHI conference on human factors in computing systems*, pp. 781–790, 2016.