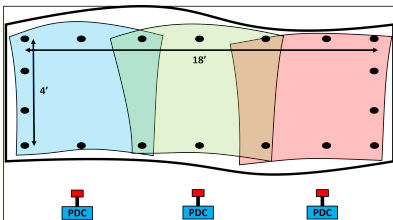# Real-time Seamless Multi-Projector Displays on Deformable Surfaces
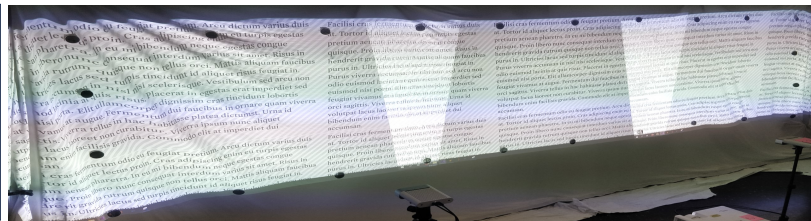
Muhammad Twaha Ibrahim*

University of California, Irvine

M. Gopi†

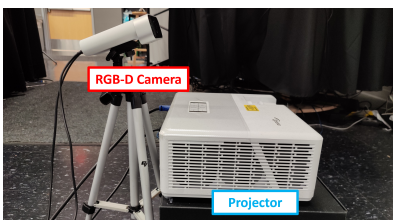University of California, Irvine

Aditi Majumder‡
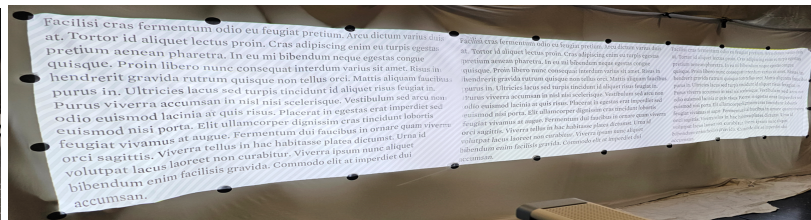
University of California, Irvine

(a) Display Setup

(b) Uncalibrated display

(c) PDC unit

(d) Calibrated display

Figure 1: **(a)** Our setup, comprising 3 PDCs with overlapping projection covering a 18' × 4' (5.5m×1.2m) deformable surface. **(b)** The uncalibrated display without any warp and blend. **(c)** A single PDC unit. **(d)** The final seamless display.

ABSTRACT

Prior works on multi-projector displays have focused primarily on static rigid objects, some focusing on dynamic rigid objects. However, works on projection based displays on deformable dynamic objects have focused only on small scale single projector displays. Tracking a deformable dynamic surface and updating projections precisely in real time on it is a significantly challenging task, even for a single projector system.

In this paper, we present the first end-to-end solution for achieving a real-time, seamless display on deformable surfaces using multiple unsynchronized projectors without requiring any prior knowledge of the surface or device parameters. The system first accurately calibrates multiple RGB-D cameras and projectors using the deformable display surface itself, and then using those calibrated devices, tracks the continuous changes in the surface shape. Based on the deformation and projector calibration, the system warps and blends the image content in real-time to create a seamless display on a surface that continuously changes shape. Using multiple projectors and RGB-D cameras, we provide the much desired aspect of scale to the displays on deformable surfaces.

Most prior dynamic multi-projector systems assume rigid objects and depend critically on the constancy of surface normals and non-existence of local shape deformations. These assumptions break in deformable surfaces making prior techniques inapplicable. Point-based correspondences become inadequate for calibration, exacerbated with no synchronization between the projectors. A few

works address non-rigid objects with several restrictions like targeting semi-deformable surfaces (e.g. human face), or using single coaxial (optically aligned) projector-camera pairs, or temporally synchronized cameras.

We break loose from such restrictions and handle multiple projector systems for dynamic deformable fabric-like objects using temporally unsynchronized devices. We devise novel methods using ray and plane-based constraints imposed by the pinhole camera model to address these issues and design new blending methods dependent on 3D distances suitable for deformable surfaces. Finally, unlike all prior work with rigid dynamic surfaces that use a single RGB-D camera, we devise a method that involve all RGB-D cameras for tracking since the surface is not seen completely by a single camera. These methods enable a seamless display at scale in the presence of continuous movements and deformations.

This work has tremendous applications on mobile and expeditionary systems where environmentals (e.g. wind, vibrations, suction) cannot be avoided. One can create large displays on tent walls in remote, austere military or emergency operations in minutes to support large scale command and control, mission rehearsal or training operations. It can be used to create displays on mobile and inflatable objects for tradeshows/events and touring edutainment applications.

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

1 INTRODUCTION

Multi-projector displays are an easy way to create seamless displays at scale without undertaking massive engineering feats of tiling LCD/LED panels together. Though multi-projector displays on rigid objects (static or dynamic) have been explored extensively, expeditionary or mobile systems often demand such displays on surfaces that continuously move and deform. Examples include emergency

*e-mail: muhammti@uci.edu

†e-mail: gopi@ics.uci.edu

‡e-mail: majumder@ics.uci.edu

management or military tents in austere locations where large data demand large visualizations for command and control, decision making, mission rehearsal and even training during long idle periods. The easiest way to setup such displays is to use projectors on the tent walls. However, such fabric based surfaces suffer continuous movement and deformations from environmentals like wind and vibrations. Similar situation is faced in commercial domain when mobile and inflatable displays need to be setup on tradeshow floors in the presence of ventilation systems blowing air or vibrations nearby from other demonstrations. Mobile educational systems (e.g. mobile planetariums) that travel to serve geographically under-served population can also benefit from displays on moving deforming surfaces.

Except for [1, 55], all *multi-projector* systems explored so far focus on rigid objects, either static [57] or dynamic [35, 54]. Any rigid object offers constant surface normals and no local deformation of the surface. Assuming prior knowledge of the 3D surface along with these invariants, [53] performed precise device calibration on a static rigid object for a dynamic projection mapping setup. Note that the precision required is rather exacting – if the physical set up and its estimation do not match, misregistrations result. To achieve the precision in the presence of movement, the rigid object is usually tracked by a single RGB-D camera. Note again, high accuracy tracking is essential to create a seamless display in which the projected image "sticks" to the rigid object as it moves. Finally, the blending between two overlapping projectors on the rigid object is achieved in real-time leveraging the fact that surface normals do not change with movement. [55] uses multiple projectors to project on a human face by fitting a parametric face model to the depth camera output. The changing surface normals of the tracked face are used to select the projector rays with the highest illumination quality only. This effectively narrows the blend width in the overlapping region to reduce the visibility of misregistration artifacts.

Ahmed et al. [1] is the only multi-projector system that handles deformable surfaces like fabrics. They use two projectors and two temporally synchronized IR cameras, each pro-cam pair being individually coaxial to achieve a super-imposed display. They calibrate each coaxial pair separately, establishing pixel-to-pixel correspondences between the camera and its corresponding coaxial projector. During projection, each IR camera separately tracks fiducials on an IR grid painted on the deformable surface and using calibration parameters of its corresponding projector, warps the display content for projection, resulting in a superimposed display that does not require realtime blending.

However, when the projection surface is deformable (like a moving fabric), the devices are not temporally synchronized or optically aligned, all the previously mentioned conditions that are leveraged to deliver a seamless display break. The surface normals change continuously and local deformations of the surface are significant. This renders most of the techniques involved (e.g. calibration, blending) inapplicable. In order to scale, single RGB-D camera based tracking is inadequate since the entire deformable surface is not seen by a single RGB-D camera.

1.1 Main Contributions

In this work, we present a novel calibration, blending and tracking methods that enable seamless multi-projector displays at scale on deformable dynamic surfaces. Following are our main contributions.
1. *Calibration:* In the absence of point based correspondences in a moving deformable surface, we use ray or plane based constraints imposed by the pinhole camera model on the moving surface to achieve highly precise device calibration.
2. *Multi-camera Surface Tracking:* Our method takes partial information from each RGB-D camera to generate highly accurate tracking of every location of the deformable projection surface.
3. *Multi-projector Blending:* In the absence of constant surface

normals, we present a new distance based blending method that can work with deformable dynamic surface.
4. *Complete System Pipeline:* Using the above information we warp and blend the content from mutliple projectors that adapts in real time to the changing deformations and movements in the projection surface creating a stable seamless display. To the best of our knowledge, this is the first end-to-end pipeline that allows multiple, temporally unsynchronized projectors and cameras to adapt to the changes in a moving deformable projection surface to create a registered, stable and seamless display.

Our work is closest to [1], but is significantly different from their work in the following respects. **(i)** We calibrate all our devices and align them in world space instead of using coaxial pro-cam pairs that are not calibrated with respect to each other. This allows us to **(ii)** place our cameras and projectors so each device need not view/illuminate the entire surface, enabling large-scale displays unlike [1]. Additionally, **(iii)** unlike [1], we reconstruct the entire 3D surface geometry, **(iv)** without requiring temporal synchronization of the cameras. Finally, **(v)** our display requires real-time blending based on the reconstructed surface shape, unlike [1] who implement a two-projector superimposed setup and hence, do not require it.

2 RELATED WORK

Although there has been a significant amount of work in single projector systems, the primary focus of this section is on multi-projector systems that are most relevant to us. We consider the prior work in three categories here, *Rigid Static Surfaces*, *Rigid Dynamic Surfaces* and *Deformable (Dynamic) Surfaces*.

2.1 Rigid Static Surfaces

When multi-projector displays are used on planar surfaces, the device calibration is avoided by using homography based methods for registration [5, 6, 9, 45]. [46–49] shows that registration can be achieved without precise device calibration if a prior on the category of shape (e.g. vertically extruded, swept or spherical surfaces) and measurements (e.g. aspect ratio of a cylindrical surface, radius of a spherical surface) of the display surface is known.

When an arbitrary geometry is used, some works use projectors and cameras separately. [39–41, 59] use a single camera to calibrate different devices separately and combine the results in a tree-like fashion. However, errors across device pairs make it nearly impossible to achieve sub-pixel accuracy in registration. Others calibrate a pro-cam unit (i.e. a unit made of a projector and a RGB camera) together and use these pro-cam units as building blocks for creating a display that has the same number of projectors and cameras. Single pro-cam systems typically employ structured light scanning of a known calibration object to establish pixel correspondences between the projector and camera followed by calibration [3, 10, 13, 14, 20, 32, 44, 51, 63, 64]. The calibration object can be a checkerboard pattern [10, 13, 14, 20, 32] or fiducials (e.g. QR codes) printed on a planar board [3, 51, 64], or even objects of arbitrary geometry to perform calibration, the latter of which allows projection mapping on non-planar surfaces. Such methods capture the static calibration object in several different poses, model the projector like an inverse camera [50] and use the camera calibration method by Zhang [66] to achieve full calibration. [57, 62] 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.

In terms of blending, systems with rigid static surface need to compute a blend mask only once. As noted by [52], the simplest approach to intensity blending in multi-projector displays is to illuminate each surface point from only one projector. However, this approach can make misregistrations due to calibration errors as well as color and intensity differences between projectors clearly visible. [39, 42, 65] implement blending across multiple projectors by

feathering the blend mask in the overlapping regions. However, the blend weights are computed in 2D projector-image space and does not consider the 3D geometry of the surface. Therefore, these methods are restricted to surfaces with simpler geometry (e.g. planes). There exist more advanced methods that approach blending as a part of the overall color non-uniformity of the multi-projector display. They compute blend maps that make color variations imperceptible while maximizing display quality, for planar [28] or arbitrary surface geometry [58].

2.2 Rigid Dynamic Surfaces

More recently, with the advent of RGB-D cameras, a few works explore the advantage of having an additional depth camera, albeit noisy and low-latency, in handling dynamic rigid objects. Use of PDC units (i.e. a unit made of a Projector, a Depth camera and a RGB Camera) became common in such systems. A large body of work focus on a single PDC unit and completely avoid calibration or shape recovery using a coaxial setup with highly specialized high-speed RGB-D devices [21, 29, 56, 60] that can additionally alleviate both the noise and latency issues. [2, 12, 43, 44, 67] calibrate a single PDC unit with prior knowledge of a precise 3D mesh of an object that serves as the calibration object. [23, 25, 26, 35, 53, 54] use multiple projectors to illuminate a moving rigid object. [23, 25, 26, 53, 54] use a single depth camera to tie all the projectors together via an object with known 3D shape. [35] uses several motion capture cameras to track markers on the target object. These works focus on illuminating small 3D objects (e.g. a bust) that can be contained within the field of view of a single RGB-D camera.

Kurth et al. [24] then advance to systems with multiple RGB-D cameras to allow 360 degree surround illumination of rigid moving objects. Multiple RGB-D cameras are required for covering the surround field of view created by all the projectors. This work proposes a calibration method that supports multiple RGB-D cameras and projectors, however still uses a single RGB-D camera for tracking. Since the rigid object does not change shape, as it moves the part visible to the RGB-D camera can be matched to the known rigid geometry via iterative closest points (ICP) based techniques to find the pose and orientation.

In a dynamic system, the overlapping regions of the projected imagery from multiple projectors change continuously due to the movement of the projection surface. Therefore, blend masks for each projector need to be recomputed in real-time. [27] proposes a method to compute blend masks for dynamic rigid objects using the dot product between the surface normal at a point on the object and the unit-rays from each projector. However, they do not take the 3D distance into account. [35] introduces a blending method that considers the surface geometry, distance to the surface as well as the pixel size on the surface to compute blend weights across projectors. By constructing a light transport matrix, [54, 55] solve a global optimization problem using a GPU-based solver to achieve real-time *content dependent* projector blending on a dynamic, rigid object. However, this solution does not scale as the light transport matrix becomes too large with more projectors. [24] proposes a high-performance distributed solver to achieve real-time performance in small systems. Although blending techniques that respect surface discontinuities have been proposed [4], they are not real-time and used for a different purpose of illuminating an animatronic head with limited local motions to render different expressions and manipulating the projection images on it to enhance its expressiveness.

2.3 Deformable Dynamic Surfaces

The rigid 3D object of known shape provides the critical anchor for all the devices in a system that illuminates rigid dynamic objects. The biggest challenge when moving to deformable objects is the constantly changing shape of the object that removes any way to anchor the system. The changing shape of the surface has to be

tracked continuously and the projection has to adapt to these changes in real-time. This challenge becomes multi-fold when considering geometrically stitching images projected from multiple PDCs to achieve a seamless display, especially when the PDCs are not synchronized. Therefore, most prior work skirts around the problem of device calibration and surface recovery by using coaxial pro-cam units [7, 8, 22, 38]. Some works embed retroreflective markers [11, 31] or markers painted with IR ink [36, 37] on the deformable surface to track it and recover its shape. [33, 34] annotate the display surface with a precisely printed dot cluster marker grid using IR ink.

Some highly specialized systems like the MIDAS system [30] use a specialized, high-speed projector, three high speed cameras and three near-IR light sources of different bands, all setup in a co-axial arrangement using mirrors so that they have the same center of projection. Each of them is equipped with color filters to sense each of the three NIR light sources. This setup enables them to use photometric stereo to compute the object normals at every camera pixel. With this information, they render their projection image to change the appearance of the object to a different material. Due to the use of coaxial high-speed setup, this system can achieve dynamic projection mapping without much perceptible lag.

Ahmed et al. [1] implement a two-projector superimposed display on a deformable surface. Each projector is coaxially aligned with an IR camera. Although they calibrate each coaxial pair separately, they do not calibrate all the devices together and hence, do not reconstruct the 3D surface geometry. Instead, each camera tracks an IR grid painted on the surface and the display content is warped separately for each projector.

Ibrahim et al. [15, 16] is the only method that calibrates a consumer-grade *non-coaxial* single PDC unit on a dynamic, deformable surface without using any embedded markers. They project a grid of ArUCo markers to establish projector-camera pixel correspondences on a moving surface before calibrating the PDC unit. However, their method cannot be used to calibrate multiple PDCs.

2.4 Comparison

Multi-PDC systems have not been explored in the context of deformable dynamic objects. Therefore, blending techniques are non-existent for this scenario. The normal-based blending applied for rigid dynamic system are not applicable since surface normals change continuously in a deformable dynamic object. Instead, we propose a novel blending technique dependent on distances between the devices, their view frusta and the changing projection surface. Finally, since no single RGB-D camera sees the entire projection surface, we devise a new tracking method that combines the information from multiple unsynchronized RGB-D cameras to provide acceptable tracking updates. However, for calibration, our method starts from the solutions proposed for single PDC units by [15, 16] and design more comprehensive methods based on ray or plane-based constraints imposed by the pin-hole camera model to achieve accurate device calibration for multiple PDC units.

3 OUR SYSTEM

The system setup consists of *N* PDCs projecting on a large, continuous and smooth deformable surface in a tiled manner such that the projections overlap (see Figure 1 (a)-(b)). The boundary of the display region is marked on the surface by black dots placed equidistant along each edge, resulting in a final display. Each projector has a corresponding RGB-D camera that observes part of the surface, such that the camera fields of view also overlap. We assume that the color camera and the depth camera in each RGB-D camera are temporally synchronized and geometrically registered, which is common for almost all consumer RGB-D cameras. Further, we do not require any knowledge of the projector or camera intrinsics or extrinsics.

Our method begins by each projector projecting ArUCo patterns on the moving deformable surface while all RGB-D cameras capture
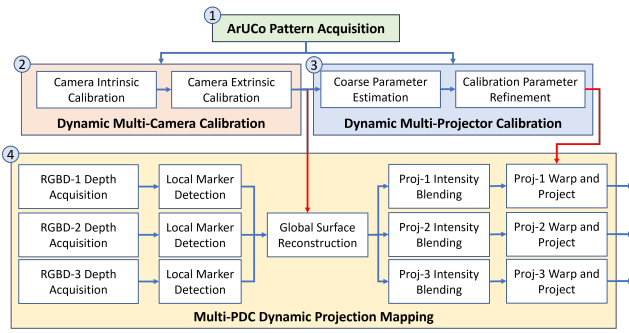
Figure 2: The flowchart of the proposed system.

those patterns and decode them to generate pixel correspondences. Then, we proceed to camera calibration, followed by projector calibration. Using the calibration parameters, we track the surface shape and perform intensity blending in real-time to adapt to the changing surface shape and create a stable seamless multi-projector display on a dynamic, deformable surface.

The goal of a single-projector projection mapping system is to determine a texture-mapping function $\Omega(\cdot)$ that warps the source image I_{src} to a target image I_{tgt} such that I_{tgt} conforms to the surface shape when it gets projected out [17]. The texture mapping function is defined as: $\Omega(p) = s$, which maps a projector pixel $p \in \mathbf{R}^2$ in I_{tgt} to a texture coordinate $s \in \mathbf{R}^2$ in I_{src}, i.e. $I_{tgt}(p) = I_{src}(\Omega(p))$. In our multi-projector system, each projector j has its own texture mapping function, denoted by $\Omega^j(p) = s$ which must be computed based on the current surface shape. Therefore, in order to compute the correct $\Omega^j(p)$ and render a *registered* multi-projector display, we need the following information: the texture coordinate $s = (u, v)$, the 3D point d on the surface that it maps to and the projector pixel p^j in projector j that will illuminate the surface at that point. Since there are multiple cameras, each observing a part of the surface, we require accurate camera calibration in order to merge their depth in a unified world space. In order to map the 3D point d to its correct projector pixel p^j in projector j, we require accurate projector calibration.

Finally, to render a *seamless* display, we must blend the projectors in their overlapping regions. This is achieved by computing a per-projector intensity mask, denoted by $\alpha_j(p), 0 \leq \alpha^j(\cdot) \leq 1$. Therefore, for the i-th point on the surface illuminated by projector j, we require the following tuple of correspondences: $f_i^j = (s_i, d_i, p_i^j, \alpha_i^j)$, where $s_i = \Omega(p_j^i)$, and d_i is the 3D point on which pixel p_j^i gets projected on. The final image that is rendered by projector j in a multi-projector setup is $I_{tgt}^j(p) = \alpha^j(p) I_{src}(\Omega^j(p))$. Note that for systems handling deformable dynamic surfaces that continuously change in shape, $\alpha^j(\cdot)$ and $\Omega^j(\cdot)$ are additionally functions of time. However, we omit that for the sake of brevity.

Figure 2 gives an overview of the complete pipeline of our system. First, we perform calibration, which recovers accurate projector and camera calibration parameters. Next, we use the cameras to capture and track the surface shape. With the camera calibration parameters, we align the surface depth from each camera to the world space and reconstruct the surface. Using the projector parameters, we compute the blend weights $\alpha^j(\cdot)$ for each sampled point on the surface and reproject it to the projector to determine its corresponding projector pixel. This enables us to compute the texture mapping functions $\Omega^j(\cdot)$. Finally, we render the display by passing all this information to a shader. In subsequent sections, we describe in detail our calibration, surface tracking and intensity blending methods that come together in the above pipeline to create a seamless, real-time display on a dynamic, deformable surface.

4 CALIBRATION

The first step in achieving a registered multi-projector display on a dynamic, deformable surface is to calibrate every projector and camera in the system. As in [15, 16], we model the projectors and cameras with a pinhole camera model extended with radial and tangential distortion, where a 2d point p (of a projector or camera) corresponds to a 3D point d on the surface. The calibration process needs to determine the intrinsics that include the intrinsic matrix $K \in \mathbf{R}^{3 \times 3}$ and distortion coefficients $D \in \mathbf{R}^{5 \times 1}$, and the extrinsics that include the rotation $R \in \mathbf{R}^{3 \times 3}$ and translation $T \in \mathbf{R}^{3 \times 1}$. Thus, for each device, we have nine intrinsic parameters (two for focal length, two for principal point and five distortion coefficients) and six extrinsic parameters (three each for rotation and translation) resulting in a total of fifteen parameters per device. We denote the projection function M^i that takes a 3D point d and maps it to a pixel p using the calibration parameters of device i as $p = M^i(d; \{K^i, D^i, R^i, T^i\})$.

In order to calibrate a multi-PDC system, we require correspondences between the cameras and projectors. In static projection mapping systems, this is achieved by structured light scanning, where each projector projects known patterns, while all the cameras capture them. The patterns are decoded to establish pixel correspondences between the projectors and cameras, and are triangulated using multi-view geometry.

However, in a moving deformable surface that changes shape continuously, this approach is inapplicable, especially when the devices are not synchronized. There is no guarantee that the surface shape remained the same when computing camera pixels corresponding to the same projector pixel across multiple cameras. In fact, in all likelihood, the correspondences belong to two different surface shapes. This lack of synchronization implies that the surface cannot be triangulated even if they are correctly decoded. Therefore, the system cannot be accurately calibrated to match the physical setup, a critical requirement for creating a registered display.

In the absence of inter-camera correspondences, we devise a method that leverages the depth from each camera via ray and plane based constraints offered by the pinhole model to achieve accurate calibration despite the surface motion. For this we need to acquire images of some projected patterns. We project a grid of ArUCo patterns, as proposed by [15, 16]. Note that the movement of the deformable surface is important for this step. Each projector projects several sequences of ArUCo grids one at a time for a few camera frames, while all cameras capture it. The captured images are then decoded to determine the camera pixels of the ArUCo pattern corners, while the corresponding 3D point is obtained from the depth map. Thus, for the k-th ArUCo pattern m_k^i detected in camera i, we have a 3D point d_k^i, the camera pixel c_k^i and the projector pixel p_k^i.

4.1 Camera Intrinsic Calibration

The first step in achieving full calibration is to estimate the camera intrinsics. For each camera, we fix the extrinsics to be at the origin, with the camera looking down the positive Z-axis. Thus, for each camera i, we optimize its intrinsic parameters denoted by $\{K_c^i, D_c^i\}$ using the correspondences (c_k^i, d_k^i) from all markers m_k^i detected in that camera. We minimize the reprojection error E_c^i, as:

$$\arg \min_{K_c^i, D_c^i} E_c^i = \frac{1}{2} \sum_k (|c_k^i - M_c^i(d_k^i; \{K_c^i, D_c^i, \mathbf{0}, \mathbf{0}\}))^2. \quad (1)$$

We initialize the optimization assuming the distortion coefficients to be zero, and compute the intrinsic matrix K_c^i directly from the 2D-3D correspondences. Each camera's intrinsics are optimized separately. If the camera API provides the camera intrinsics, we use them to initialize the optimization instead. We continue optimizing the parameters until convergence.
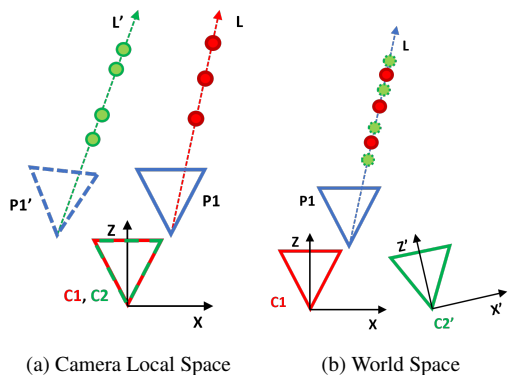
(a) Camera Local Space   (b) World Space

Figure 3: Illustration of camera extrinsic calibration. Cameras C1 and C2 both capture the world from their local origin and view the same projector at different locations (P1 and P1'). **(a)** They capture 3D points corresponding to the same projector pixel in their local spaces, giving points along ray L (red) and L' (green). However, in world space, points along ray L' must align with points along ray L. **(b)** Our camera extrinsic calibration finds a rigid transform that aligns points on ray L' along the ray L to determine the camera pose.

4.2 Camera Extrinsic Calibration

The largest impact of the absence of inter camera correspondences in an unsynchronized system is estimating the camera extrinsic parameters. Traditional multi-view geometry methods that assume a static surface become inapplicable. We present a new method to estimate the camera extrinsics that uses 3D depth captured by each camera instead of the 2D pixel correspondences.

Consider only the set of pattern corners that lie on the same row of pixels in the projector image when it is displaying a grid of ArUCo patterns on the moving deformable surface. This row of 2D pixels forms a plane in 3D. Each camera captures this ArUCo grid and records the 3D points corresponding to the pixel row of pattern corners. If we fit a plane through each set of 3D points acquired by each camera, we will get two different plane parameters, even though they are the same 3D plane in world space. Similarly, consider only 3D points captured by two cameras that correspond to the same projector pixel i.e. ArUCo marker corner. If we fit a line through each of these sets of 3D points, we will get two different line parameters, even though they correspond to the same 3D ray in world space.

Therefore, we need to transform the point clouds such that 3D points from different cameras corresponding to the same projector row/column pixels lie on the same 3D plane, and 3D points from different cameras corresponding to the same projector pixel lie on the same 3D ray. This transform implicitly gives us the camera extrinsics. Figure 3 illustrates this concept using points along a ray.

Thus, we perform an optimization that computes a rigid transform by minimizing the error between corresponding 3D lines and 3D planes in the point clouds captured by multiple cameras. We minimize the error, $E_{ray}^i, E_{row}^i, E_{col}^i$, between corresponding lines, row planes and column planes respectively in the 3D point cloud captured by the reference camera and the point cloud captured by the i-th camera transformed by the camera pose R_c^i, T_c^i given by:

$$\arg \min_{R_c^i, T_c^i} E_t^i = E_{ray}^i(d^i, d^{ref}; \{R_c^i, T_c^i\}) \\ + E_{row}^i(d^i, d^{ref}; \{R_c^i, T_c^i\}) + E_{col}^i(d^i, d^{ref}; \{R_c^i, T_c^i\}) \tag{2}$$

The error is computed as the distance between the points in the transformed point cloud and their projection on the corresponding ray/plane in the reference point cloud. Once we have each camera's
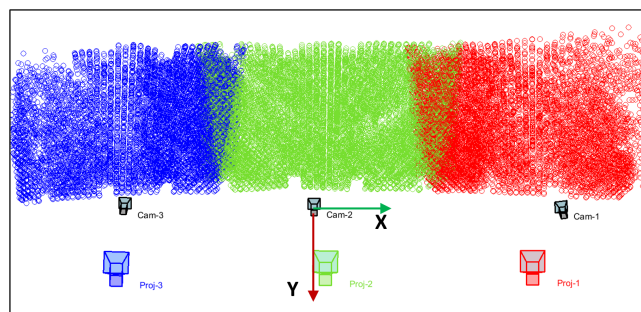

Figure 4: The calibration results from our method. The cameras are shown in black, while the three projectors and their corresponding depth data are colored in red, green and blue.

extrinsics, we transform the point cloud from each camera to the world space.

Note that standard 3D point cloud alignment algorithms like ICP (Iterative Closest Point) cannot be used here. Due to the dynamic nature of the surface, the 3D points corresponding to the same pattern corner lie on a straight line. One camera may have captured more 3D points of that pattern than another. Since ICP computes point matches every iteration, it may converge to the incorrect solution because of incorrect matches. Therefore, while ICP serves as a good initialization to the optimization, it does not result in a precise alignment of the point clouds.

4.3 Coarse Projector Calibration

At this point, the cameras in the system are fully calibrated. Next, we perform projector calibration. We use the method by [15] that uses the point cloud to perform a coarse projector calibration. This is used as an initialization to an optimization that refines the projector parameters.

To calibrate a particular projector, we use all the 3D points corresponding to it that have been captured by all cameras and transform them to world space using the camera extrinsics. Next, we again use the plane based constraint imposed by the pinhole camera model to fit a plane through each set of 3D points that correspond to a row of pixels in the projector image. This gives us a set of row planes. As dictated by the pin hole camera model, we intersect all the row planes to form a 3D line that contains the projector location in world space and find the direction of this 3D line to estimate the projector's X-axis in world space. We repeat this procedure for the set of 3D points that correspond to the same column pixels in the projector image, fitting planes and intersecting them to get the projector's Y-axis direction. Taking a cross product of the X and Y-axes gives the projector's Z-axis. Finally, we intersect the 3D lines corresponding to the X- and Y-axes to get an estimate of the projector's center of projection (aka location).

We also estimate the intrinsic matrix of the projector using the pose estimate and the 2D-3D correspondences of that projector. This coarse projector calibration serves as an initialization to the optimization that refines each projector's parameters.

4.4 Fine Projector Calibration

Let $(p_k^j, \tilde{d}_k^{\ j})$ denote the k-th correspondence between a 2D projector coordinate p_k^j and its transformed 3D point $\tilde{d}_k^{\ j}$ for the j-th projector. Let $\{K_p^j, D_p^j, R_p^j, T_p^j\}$ denote projector parameters. We optimize

these parameters and minimize the reprojection error E_p^j:

$$\arg \min_{K_p^j, D_p^j, R_p^j, T_p^j} E_P^j = \frac{1}{2} \sum_k (|p_k^j - M_P^j(\tilde{d}_k^{\ j}; \{K_p^j, D_p^j, R_p^j, T_p^j\}))^2 \quad (3)$$

We initialize the projector parameters from the coarse projector calibration in the previous step, assuming distortion coefficients are zero. The optimization is then performed until convergence. Figure 4 shows the final, fully calibrated system.

5 MULTI-CAMERA SURFACE TRACKING

Surface tracking is achieved by a realtime 3D reconstruction of the continuously changing surface shape. The goal of the surface reconstruction step is to determine 3D points on the surface that correspond to the uniformly sampled texture grid, where each texture coordinate is denoted by $s_k = (u_k, v_k)$. Each black marker on the surface corresponds to a known texture coordinate. However, because they are all identical (to keep the system simple for deployment for lay users), we must first determine the (u,v) coordinate of each marker in every camera frame. We employ the method proposed by [17, 18] to track the surface, but adapt it to handle multiple depth cameras, each observing only part of the surface.

We start by detecting the border markers in the IR frame of each camera and determine the corresponding 3D point from the depth map. Since each 3D point is in the local space of the camera that detected it, we use the corresponding extrinsics to transform each 3D point to world space. However, because the camera FOVs overlap, some markers are detected by multiple cameras. In order to reconstruct the surface accurately, we must identify these duplicate detections and merge them.

Note that while all the 3D points are in world space, they are randomly ordered. To remove this random ordering, we compute the average 3D point of all the points in world space and then sort them all in a clockwise fashion around the average 3D point. This arranges each 3D point with its adjacent neighbor around the display boundary. Next, we proceed to merging markers detected by multiple cameras. For each 3D point, we compare the distances with its left and right neighbors. If the distance between a pair of points is less than a threshold ($\leq 50mm$), we mark those points as duplicates. Each set of duplicate detections is assigned a unique ID and after all duplicates have been detected, the 3D points in each set are merged by averaging them.

Now that we have one 3D point for each marker in world space, we identify the four corners of the display region by computing the angle each marker makes with its two adjacent neighbors. Four markers with angles close to $90°$ are determined as the display corners. From these four corners, we determine the top-left corner and assign its texture coordinate $(u,v) = (0,0)$. Since all the markers are sorted in a clockwise fashion, we can assign each marker its corresponding (u,v) starting from the top-left corner, going clockwise around the display boundary.

Next, we compute correspondences inside the display region. Instead of interpolating normals like [17, 18], we use bilinear interpolation to estimate the internal correspondences. These interpolated 3D points are not guaranteed to be on the surface and using them for rendering will result in visible misregistrations. Assuming a calibrated multi-PDC rig, we reproject the internal 3D points back into every camera and re-sample the depth at that pixel location if it is valid (see Figure 5). We transform these re-sampled 3D points back into world space and average any points that were seen by multiple cameras.

Thus, by tracking the markers from each camera, resampling the surface shape and combining all 3D points in world space, we are able to achieve multi-camera surface tracking. At this point, the entire texture grid has a corresponding 3D point. Finally, we use the projector calibration parameters to determine the corresponding 2D pixels in every projector and compute the function $\Omega^j(\cdot)$.
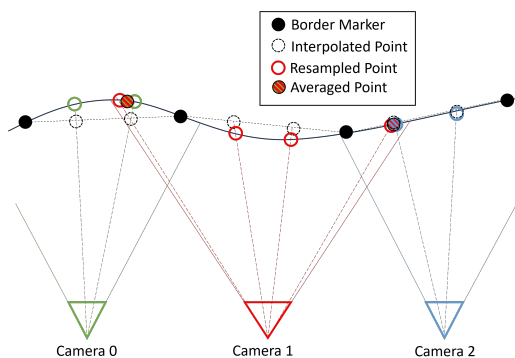

Figure 5: Surface sampling by the multi-camera rig. The filled black markers represent the surface border. Points are estimated by linear interpolation between the markers (empty black circles). These points are reprojected into the cameras, which measure the 3D surface at that pixel (colored circles). Points seen by multiple cameras are averaged (striped circles).
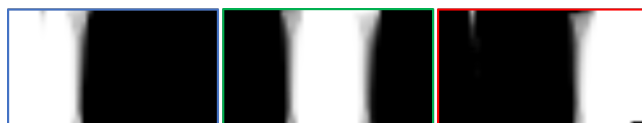

Figure 6: The projector blend masks in (u,v) space. Note that each mask represents the entire display region.

Note that calibration errors could introduce misregistration artifacts when we transform the points back into world space. However, misregistrations will only be visible in the projector overlapping regions. Since the overlapping regions are seen by multiple cameras and we average the 3D points seen by each camera, this mitigates the misregistration artifacts caused by imperfect calibration.

6 PROJECTOR INTENSITY BLENDING

To blend the contributions from different projectors in the overlapping region, we need to determine the contribution by each projector to all 3D points. The challenge in a system with dynamic deformable projection surface stems from the fact that these regions are continuously changing. Therefore, the projector blend masks need to be computed every frame efficiently to maintain a real-time FPS (frames per second) of the display.

Prior works on multi-projector dynamic systems only handle rigid objects where the projector light completely floods the target object. In such a rigid object, the surface normals at each 3D point are known apriori. This allows such methods to use the angle of the surface normal with respect to the projector locations (by computing the dot product) to determine the blend weights for each projector. Thus, if the surface normal is at a grazing angle with respect to a projector, the blend weight is close to zero. During runtime, the normals are transformed based on the current object pose to determine the blend weights for each projector contributing to a 3D point. The 3D distance of contributing projectors is also included in the blend weights to account for distance attenuation. Such methods are well suited to handle sharp edges and occlusions of rigid object with respect to projectors.

On the other hand, projector intensity blending on deformable surfaces cannot assume static surface normals and hence, have to be recomputed every frame. However, surface normals are not well suited for blending in our case. Aside from the added latency to compute them every frame, surface normal based blending weights do not feather off the masks to zero at the edges of a projector

Figure 7: The final multi-projector display on a moving, deformable surface.

since the projector rays do not hit the surface at grazing angles, resulting in visible brightness seams. Additionally, noise in the depth camera can impact surface normal computation, resulting in artifacts in the blend masks. The mandate is for the projection intensity mask to fall off gradually to zero the closer it gets to the projection edge in the overlapping region. At the same time, we want to account for the distance of a 3D point to the projectors as well, with the closer projector having a higher contribution. Therefore, we compute distance-based blending weights using the correspondences generated in Section 5.

First, for each projector, we pre-compute the 3D planes forming its view frustum using its calibration parameters. Next, for each 3D point d_k on the surface, we check whether its reprojection in projector j is valid. If not, its blend weight is automatically set to zero. Otherwise, we compute **(a)** the 3D distance between the 3D point and its closest view frustum planes for that projector, and **(b)** the 3D distance of the corresponding 3D point to that projector. Finally, the blend weight at the k-th 3D point in projector j, denoted by α_k^j, is computed as:

$$\alpha_k^j = \frac{w_k^j L(d_k)}{\sum_q^N w_k^q L(d_k)}, \quad L(d) = \frac{G(d)^2}{H(d)^2}, \tag{4}$$

where d_k is a 3D point on the reconstructed surface, N is the number of projectors, $G(d)$ computes the 3D distance between the 3D point d_k and its closest view frustum plane in projector j, and $H(d)$ computes its 3D distance to the projector location. w_k^j is 1 if the reprojected 3D point is valid in projector j, and 0 otherwise. In other words, if a 3D point in the overlapping region is closer to a projector, that projector has a higher contribution to that point. At the same time, if that 3D point is close to the projection edge, its contribution is reduced.

The result is a blend mask for each projector where the pixel intensities in the non-overlapping regions are one, but gradually fall off to zero in the overlapping regions as they approach the projector edges, resulting in a smooth display. Figure 6 shows the blend masks for the three projectors in (u, v) space. Note that each mask represents the entire display region. Therefore, while the falloffs look small in the image, they span 2' (0.7m) on the 3D surface.

7 RESULTS

Figure 1 and Figure 7 show the full seamless multi-projector display on a deformable fabric achieved by our method. We used three PDCs

Table 1: Reprojection error (in pixels) for the 3-PDC system

Camera Reproj. Error	Projector Reproj. Error
0.113	1.156
0.126	1.234
0.124	1.249

that illuminated a display surface of size 18'×4' (5.5m×1.2m). The display surface comprises a loosely hanging white fabric. Ten boundary markers annotate the top and bottom edges across, while four markers annotate the left and right edges down the fabric. The PDCs are placed approximately 4' (1.2m) away from the display surface and 6.5' (2m) apart from each other to have a reasonable overlap while still covering the entire display area. Two powerful fans placed on either side of the surface are used to simulate surface motion of varying intensities. Figure 8 shows the seamless registration and blending of overlapping projectors.

We used three Azure Kinect RGB-D cameras. The Azure Kinect is a time-of-flight IR/depth camera that provides registered RGB and depth images. It provides multiple operating resolutions. In our work, we used the Azure Kinect in the wide field-of-view setting, capturing 1280×720 RGB-D images at 30fps. This enables us to cover a large display. At this setting, the depth camera resolution is 512×512, significantly smaller than the RGB camera resolution. However, the camera API provides methods that transform and interpolate the depth map to the RGB camera space, similar to other consumer-grade depth cameras like the Intel RealSense Depth Cameras [19]. In our work, we use 3D points from this interpolated depth map. For the projectors, we used three Optoma ZH406STx short-throw projectors, operating at 1920×1080 resolution. At a time, only one projector projects an ArUco marker grid while all cameras capture in parallel. We capture approximately 25 frames per projector. Although the Kinect API provides camera intrinsic and distortion parameters, we have found them to be inaccurate for our application. Therefore, we always refine camera intrinsic and distortion parameters. Note that the cameras are not temporally synchronized, either during calibration or during real-time display. We tested the robustness of our system by placing the PDCs in different positions and orientations with respect to the surface, such as tilted upwards or sideways. So long as each border marker is visible to at least one camera and the display area is completely illuminated by the combined projections, our system operates smoothly.

The proposed system was implemented in MATLAB and C++. Our machine had an Intel Core i7 CPU, with 32GB RAM with a nVidia Quadro P4000 GPU. We used OpenCV APIs for detecting ArUco markers with sub-pixel accuracy. We used Levenberg-Marquadt optimization to implement our calibration routines. All calibration routines were implemented in C++, except the camera extrinsics calibration, which was in MATLAB. Calibration routines run until convergence, which happens when any one of the following two conditions are met: (1) the number of iterations exceeds a user-defined limit (typically 1000), or (2) the absolute difference between the errors in two successive iterations is less than a threshold $\Delta = 10^{-4}$. It takes approximately 20 minutes to fully calibrate a 3-PDC system, of which ArUco acquisition takes 40 secions, camera extrinsics calibration takes 11 minutes and projector calibration takes 7 minutes. Note that our code is not optimized and calibration is done only once for the system. The calibration parameters are then used to achieve the real-time seamless projection on the deformable surface. We used OpenGL for rendering.

7.1 Evaluation

We evaluated four main aspects of our system: **(i)** calibration accuracy, **(ii)** run-time efficiency, **(iii)** accuracy of the 3D surface reconstruction and **(iv)** display registration quality.

Figure 8: Results of our calibration, rendering and display method. *Left column*: No calibration. *Middle column*: Calibrated result without blending. Note the accuracy of our registration. *Right column*: Our final blended result.
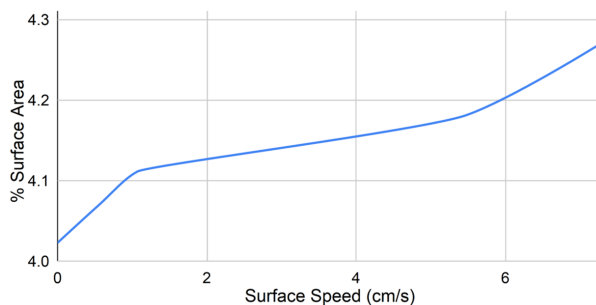

Figure 9: Plot of the misregistered surface area (%) vs. surface speed (cm/s), showing the effect of surface motion on the 3D surface reconstruction.

Table 2: Display run-time breakdown for the 3-PDC system to render a single frame.

Step	Time (ms)
Border detection	3.78
Surface Reconstruction	3.37
Blending	3.56
Warping	10.65
Total	**21.36**

Calibration Accuracy: Table 1 shows the reprojection error achieved using our calibration method for each camera and projector in our system. The reprojection error for all cameras is less than 0.2 pixels, whereas for projectors, it averages around 1.5 pixels. The reason for the relatively higher reprojection error for the projectors is due to noise in the depth capture by the Kinects.

With the exception of [15], existing calibration methods require the surface to be static, rigid and/or of known shape, none of which apply to our case. The calibration parameters provided by such methods are accurate only for a specific surface geometry. If the surface then moves, we will see misregistrations immediately. In contrast, our method can exploit the surface movement to provide a more accurate calibration.

Runtime Efficiency: Table 2 shows the runtime breakdown of our system to render a single frame for the display with 3 PDCs after each RGB-D camera has provided a frame for processing. Note that the total time taken by our system to render a single frame for the deformable surface is 22ms, which is less than the camera frame rate (33ms/frame). Therefore, our system is limited by the camera frame rate and theoretically, we can support cameras where each frame is rendered in 22ms or higher (i.e. maximum of 45 fps).

Surface Reconstruction Accuracy: To evaluate the surface reconstruction accuracy, we computed the surface area of the reconstructed surface that where the depth differed from the surface reconstructed from the dense depth maps by more than 1cm. We captured data at various surface speeds, ranging from stationary to motion mimicking strong gusts of wind. Figure 9 shows these areas of the reconstructed surface as a percentage of the total surface area. Notice that with increasing surface speed, the error increases only slightly, reaching a maximum of 4.25%. This amounts to an area of 3 sq. ft. distributed across the total 72 sq. ft. display ($0.25m^2$ of $7m^2$) and is imperceptible. Notice also that there is still some error even when the surface is stationary. This is not due to calibration errors (which would result in visible misregistrations of the display) but the small wrinkles in the fabric which our reconstruction method interpolates over.

Display Registration Quality: We captured a video of the overlapping regions of the final display when projecting text at various surface speeds and compared it to the original image using the Struc-
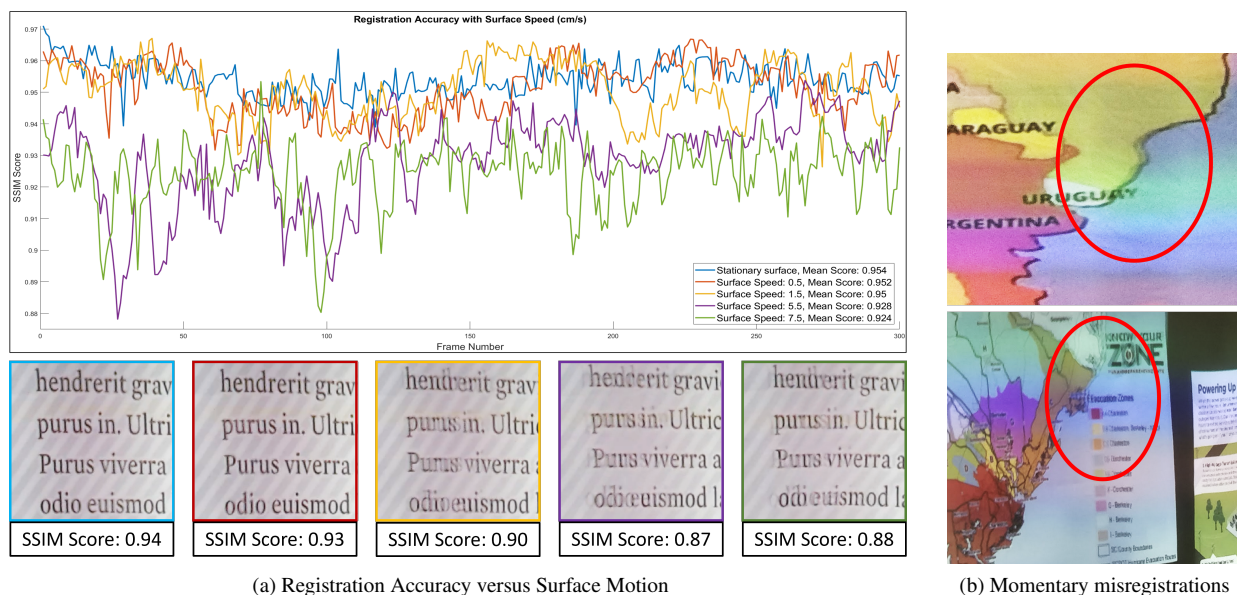
(a) Registration Accuracy versus Surface Motion

(b) Momentary misregistrations

Figure 10: Effect of surface motion on registration accuracy. **(a)** The graph shows the SSIM score of a video of the overlapping projection compared to the original text image. The images underneath show the projected frames with the lowest SSIM scores at the corresponding surface speeds. **(b)** shows momentary misregistrations for other types of display content.

tural Similarity Index Measure (SSIM) [61]. The SSIM score ranges from 0 to 1, with lower values corresponding to poorer quality, while a score of 1 indicates an exact match. Figure 10 shows the variation in the SSIM score over 300 video frames. For smaller surface movements, the SSIM score remains generally high, greater than 0.94. However, as expected, with increasing surface speed, the SSIM score decreases, going below 0.9 for a few frames which is when misregistrations become obvious. This is confirmed by a visual analysis of the overlapping projections as well. The images of the text in Figure 10a show the display registration in the video frame with lowest SSIM score for each surface speed across the 300 frames. It is important to note that these misregistrations are only momentary as the display continues to adapt to the changing surface geometry.

7.2 Discussion and Limitations

Latency: Although our calibration and surface reconstruction is quite accurate even in strong winds, we still see misregistrations in the final display at high surface speeds (Figure 10). This is mainly due to the lag caused by our system. It takes 22ms to render a single frame for a 3-PDC system and this lag is enough to see minor misregistrations in the overlapping regions for a brief moment. One possible hardware solution is to use higher speed RGB-D cameras e.g. 45fps to mitigate these misregistrations. A possible algorithmic solution is to use motion prediction. While prior work has used motion prediction for rigid bodies in a multi-projector DPM setup [54], these techniques are not applicable in our case where the surface is deformable and the movements are random and complex.

Depth Camera Noise: A problem with using commercial depth cameras such as the Azure Kinect is noise in the captured depth map. Using noisy depth for calibrating the system can result in calibration errors. However, our novel calibration method uses ray and plane -based constraints to mitigate those effects. During display runtime however, noise in the depth map manifests as a jittery display. [15] is the first work that addresses this jitter in the context of DPM by using the Kalman filter to smooth the points. However, the Kalman filter introduces some lag which can manifest as misregistrations in the overlapping regions. Therefore, in the future, we would like to explore other more accurate noise removal methods.

Border Markers: We place border markers on the surface to serve as features demarcating the display boundary. This is necessary for deformable surfaces as the projection areas change based on the surface movement. While [17] explain a marker-less method of performing DPM, they rely on a clear depth discontinuity between the surface and the background. This is not guaranteed in our case and we would like to explore methods that do not require any boundary features to perform DPM on deformable surfaces in the future.

Display Readability at High Surface Speeds: Our goal was to create a projection display that conforms to the surface shape as it is perceptibly pleasing for viewers. However, with high surface speeds, the display readability may be compromised due to the rapid surface motion. In the future, we would like to study this effect of rapid surface motion on display readability by conducting user studies and exploring algorithms that can stabilize the projection at high surface speeds.

Display Color Quality: As shown in Figure 1(d), Figure 7 and Figure 8, our distance-based blending method is able to blend the overlapping regions to create a seamless display in real-time. Though normal based blending does not work in this scenario, we would like to explore adapting other color calibration methods such as [24, 58] for deformable dynamic surfaces in the future.

8 SUMMARY

In summary, we present the first work to achieve large seamless displays on dynamic and deformable surfaces using multiple projectors and RGB-D cameras. We present new methods that accurately calibrate the devices, track the surface and blend the projectors in realtime. In the future, we plan to improve our system by addressing existing limitations.

ACKNOWLEDGMENTS

REFERENCES

[1] B. Ahmed and K. H. Lee. Projection mapping onto deformable nonrigid surfaces using adaptive selection of fiducials. *Journal of Electronic Imaging*, 28(6):063008–063008, 2019.

[2] H. Asayama, D. Iwai, and K. Sato. Fabricating diminishable visual markers for geometric registration in projection mapping. *In IEEE Transactions on Visualization and Computer Graphics*, 24:1091–1102, 2018.

[3] 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.

[4] A. Bermano, P. Brüschweiler, A. Grundhöfer, D. Iwai, B. Bickel, and M. Gross. Augmenting physical avatars using projector-based illumination. *ACM Trans. Graph.*, 32(6), nov 2013. doi: 10.1145/2508363.2508416

[5] E. Bhasker, R. Juang, and A. Majumder. Registration techniques for using imperfect and par tially calibrated devices in planar multi-projector displays. *IEEE Transactions on Visualization and Computer Graphics*, 13(6):1368–1375, 2007.

[6] 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.

[7] W. Bi and B. Xiao. Perceptual constancy of mechanical properties of cloth under variation of external forces. In *Proceedings of the ACM Symposium on Applied Perception*, pp. 19–23, 2016.

[8] K. L. Bouman, B. Xiao, P. Battaglia, and W. T. Freeman. Estimating the material properties of fabric from video. In *Proc. IEEE Int. Conf. Comput. Vis.*, p. 1984–1991, 2013.

[9] H. Chen, R. Sukthankar, G. Wallace, and T. J. Cham. Calibrating scalable multi-projector displays using camera homography trees. *CVPR Technical Sketch*, 2001.

[10] 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.

[11] Y. Fujimoto, R. T. Smith, T. Taketomi, G. Yamamoto, J. Miyazaki, H. Kato, and B. H. Thomas. Geometrically-correct projection-based texture mapping onto a deformable object. *In IEEE Transactions on Visualization and Computer Graphics*, 20:540–549, 2014.

[12] N. Hashimoto, R. Koizumi, and D. Kobayashi. Dynamic projection mapping with a single ir camera. *In International Journal of Computer Games Technology*, 2017.

[13] B. Huang, S. Ozdemir, Y. Tang, C. Liao, and H. Ling. A single-shot-per-pose 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.

[14] 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.

[15] M. T. Ibrahim, M. Gopi, and A. Majumder. Auto-calibrating dynamic projection mapping system with jitter and occlusion management. In *2023 22nd IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*, 2023.

[16] M. T. Ibrahim, M. Gopi, and A. Majumder. Projector-camera calibration on dynamic, deformable surfaces. In *2023 IEEE Conference on Virtual Reality and 3D User Interfaces Abstracts and Workshops (VRW)*, pp. 905–906, 2023. doi: 10.1109/VRW58643.2023.00295

[17] M. T. Ibrahim, A. Majumder, and M. Gopi. Dynamic projection mapping on deformable stretchable materials using boundary tracking. *Computers & Graphics*, 103:61–74, 2022.

[18] 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.

[19] Intel. Intel depth camera d435. `https://www.intelrealsense.com/depth-camera-d435/`. Accessed: 2022-09-11.

[20] R. Juarez-Salazar and V. H. Diaz-Ramirez. Flexible camera-projector calibration using superposed color checkerboards. *Optics and Lasers in Engineering*, 120:59–65, 2019.

[21] S. Kagami and K. Hashimoto. Animated stickies: Fast video projection mapping onto a markerless plane through a direct closed-loop alignment. *In IEEE Transactions on Visualization and Computer Graphics*, 25:3094–3104, 2019.

[22] T. Kawabe, T. Fukiage, M. Sawayama, and S. Nishida. Deformation lamps: A projection technique to make static objects perceptually dynamic. *In ACM Trans. Appl. Percept.*, 2016.

[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] V. Lange, C. Siegl, M. Colaianni, P. Kurth, M. Stamminger, and F. Bauer. Interactive painting and lighting in dynamic multi-projection mapping. *In Augmented Reality, Virtual Reality, and Computer Graphics*, 9769, 2016.

[26] V. Lange, C. Siegl, M. Colaianni, M. Stamminger, and F. Bauer. Robust blending and occlusion compensation in dynamic multi-projection mapping. In *Proc. of European Association for Computer Graphics: Short Papers*, pp. 1–4, 2017.

[27] P. Lincoln, G. Welch, and H. Fuchs. Continual surface-based multi-projector blending for moving objects. In *2011 IEEE Virtual Reality Conference*, pp. 115–118. IEEE, 2011.

[28] A. Majumder and R. Stevens. Perceptual photometric seamlessness in projection-based tiled displays. *ACM Trans. Graph.*, 24:118–139, jan 2005. doi: 10.1145/1037957.1037964

[29] 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.

[30] L. Miyashita, Y. Watanabe, and M. Ishikawa. Midas projection: markerless and modelless dynamic projection mapping for material representation. *ACM Trans. Graph.*, 2018.

[31] D. Miyazaki and N. Hashimoto. Dynamic projection mapping onto non-rigid objects with dot markers. In *2018 International Workshop on Advanced Image Technology (IWAIT)*, pp. 1–4. IEEE, 2018.

[32] 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.

[33] 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.

[34] 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.

[35] 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.

[36] P. Punpongsanon, D. Iwai, and K. Sato. Deforme: projection-based visualization of deformable surfaces using invisible textures. In *SIGGRAPH Asia 2013 Emerging Technologies*, pp. 1–3, 2013.

[37] P. Punpongsanon, D. Iwai, and K. Sato. Projection-based visualization of tangential deformation of nonrigid surface by deformation estimation using infrared texture. *In Virtual Reality*, 19, 2015.

[38] P. Punpongsanon, D. Iwai, and K. Sato. Flexeen: Visually manipulating perceived fabric bending stiffness in spatial augmented reality. *In IEEE Transactions on Visualization and Computer Graphics*, pp. 1433–1439, 2020.

[39] 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.

[40] 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.

[41] 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.

[42] R. Raskar, G. Welch, K.-L. Low, and D. Bandyopadhyay. Shader lamps: Animating real objects with image-based illumination. In *Rendering Techniques 2001: Proceedings of the Eurographics Workshop in London, United Kingdom, June 25–27, 2001 12*, pp. 89–102. Springer, 2001.

[43] C. Resch, P. Keitler, and G. Klinker. Sticky projections — a new approach to interactive shader lamp tracking. In *IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*, pp. 151–156, 2014.

[44] 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.

[45] 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*, 16(6):1623–1632, 2010. doi: 10.1109/TVCG.2010.128

[46] 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.

[47] B. Sajadi and A. Majumder. Auto-calibration of cylindrical multi-projector systems. In *2010 IEEE virtual reality conference (VR)*, pp. 155–162. IEEE, 2010.

[48] 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.

[49] 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.

[50] P. Sen, B. Chen, G. Garg, S. Marschner, M. Horowitz, M. Levoy, and H. Lensch. Dual photography. *ACM SIGGRAPH 2005 Papers*, 2005.

[51] 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.

[52] S. Shimazu, D. Iwai, and K. Sato. 3d high dynamic range display system. In *2011 10th IEEE International Symposium on Mixed and Augmented Reality*, pp. 235–236, 2011. doi: 10.1109/ISMAR.2011.6092393

[53] C. Siegl, M. Colaianni, M. Stamminger, and F. Bauer. Stray-light compensation in dynamic projection mapping. In *SIGGRAPH ASIA 2016 Technical Briefs*, 2016.

[54] 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. *In ACM Trans. Graph.*, 2015.

[55] C. Siegl, V. Lange, M. Stamminger, F. Bauer, and J. Thies. Faceforge: Markerless non-rigid face multi-projection mapping. *In IEEE Transactions on Visualization and Computer Graphics*, 23:2440–2446, 2017.

[56] 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.

[57] 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.

[58] M. A. Tehrani, M. T. Ibrahim, A. Majumder, and M. Gopi. 3d gamut morphing for non-rectangular multi-projector displays. *IEEE Transactions on Visualization and Computer Graphics*, pp. 1–15, 2023. doi: 10.1109/TVCG.2023.3277436

[59] M. Waldner, C. Pirchheim, and D. Schmalstieg. Multi projector displays using a 3d compositing window manager. *Emerging display technologies*, pp. 1–4, 2008.

[60] 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.

[61] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing*, 13(4):600–612, 2004.

[62] 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.

[63] 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.

[64] 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.

[65] R. Yang, D. Gotz, J. Hensley, H. Towles, and M. S. Brown. Pixelflex: A reconfigurable multi-projector display system. In *Proceedings Visualization, 2001. VIS'01.*, pp. 167–554. IEEE, 2001.

[66] Z. Zhang. A flexible new technique for camera calibration. *IEEE Transactions on pattern analysis and machine intelligence*, 22(11):1330–1334, 2000.

[67] 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.