

# Dynamic Projection Mapping of Deformable Stretchable Materials

Muhammad Twaha Ibrahim  
Donald Bren School of Information  
and Computer Sciences, University of  
California, Irvine, CA, USA  
muhammti@uci.edu

Gopi Meenakshisundaram  
Donald Bren School of Information  
and Computer Sciences, University of  
California, Irvine, CA, USA  
gopi@ics.uci.edu

Aditi Majumder  
Donald Bren School of Information  
and Computer Sciences, University of  
California, Irvine, CA, USA  
majumder@ics.uci.edu

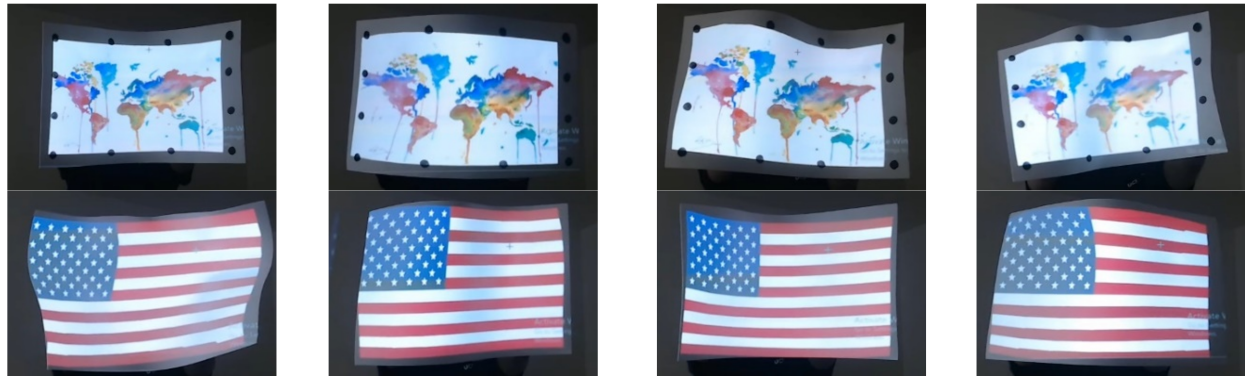


Figure 1: Dynamic projection mapping onto marker-based (top row) and marker-less (bottom row) surface.

## ABSTRACT

We present a method for dynamic projection mapping on deformable, stretchable and elastic materials (e.g. cloth) using a time of flight (ToF) depth camera (e.g. Azure Kinect or Pico-Flexx) that come equipped with an IR camera. We use Bezier surfaces to model the projection surface without explicitly modeling the deformation. We devise an efficient tracking method that tracks the boundary of the surface material using the IR-Depth camera. This achieves realistic mapping even in the interior of the surface, with simple markers (e.g. black dots or squares) or without markers entirely, such that the projection appears to be printed on the material. The surface representation is updated in real-time using GPU based computations. Further, we also show that the speed of these updates is limited by the camera frame rate and therefore can be adopted for higher speed cameras as well. This technique can be used to project on several stretchable moving materials to change their appearance.

## CCS CONCEPTS

• **Computing methodologies** → Computer graphics; • **Graphics systems and interfaces**; • **Mixed / augmented reality**; • **Artificial intelligence**; • **Computer vision**; • **Computer vision**



This work is licensed under a Creative Commons Attribution International 4.0 License.

VRST '20, November 01–04, 2020, Virtual Event, Canada  
© 2020 Copyright held by the owner/author(s).  
ACM ISBN 978-1-4503-7619-8/20/11.  
<https://doi.org/10.1145/3385956.3418970>

tasks; • **Scene understanding**; • **Human-centered computing**;  
• **Human computer interaction (HCI)**; • **Interaction devices**;  
• **Displays and imagers**;

## KEYWORDS

Dynamic Projection Mapping, Spatially Augmented Reality, Deformable Materials, Appearance Editing

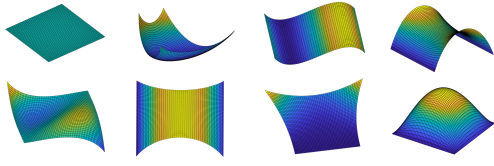
### ACM Reference Format:

Muhammad Twaha Ibrahim, Gopi Meenakshisundaram, and Aditi Majumder. 2020. Dynamic Projection Mapping of Deformable Stretchable Materials. In *26th ACM Symposium on Virtual Reality Software and Technology (VRST '20)*, November 01–04, 2020, Virtual Event, Canada. ACM, New York, NY, USA, 5 pages. <https://doi.org/10.1145/3385956.3418970>

## 1 INTRODUCTION

Projecting onto deformable, non-rigid surfaces can have many applications like visualization, surgical guidance, fashion, retail, design and entertainment. Most prior works avoid accurate reconstruction of deformable surface [2, 10, 14] or use expensive optimizations [21] to model per-frame deformation explicitly. Therefore, they are not real-time and cannot address stretchable or elastic materials.

In this paper, we take an alternate approach of using Bezier surfaces to model the projection surface without explicitly modeling the deformation. This allows us to update the surface representation and warp the projected image in real time using GPU based parallelization. We use an off-the-shelf time-of-flight (ToF) based depth camera (e.g. Azure Kinect, Pico Flexx), that come with a registered IR camera, to map an image to deformable material in real-time such that it appears to be printed on the material. We track only the boundary of the material using simple markers (e.g. black dots)



**Figure 2: Various surfaces that can be represented by a cubic Bezier patch, including (bottom-row) surfaces stretched down the middle, at the upper two corners and outward from the center.**

or depth-features leading to projection mapping with or without markers respectively in real-time on stretchable materials. Use of Bezier based representations also enable constrained projection mapping (e.g. length preserving mapping) to allow optimal view for multiple users.

### 1.1 Related Work

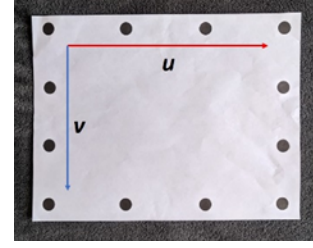
A large body of literature exists on projection mapping on static objects [22, 27–33, 35–37]. When considering dynamic objects, [2, 3, 5, 7, 10–18] handle rigid objects with known geometry in an accurately calibrated non-real-time system. [10, 13, 14] use specialized high-speed coaxial projector and camera to map specialized *planar* objects (e.g. printed with imperceptible IR ink) in real-time.

[1, 4, 6, 8, 9, 19–22, 34, 38–40] address non-stretchable deformable objects. [1] uses IR-camera to track embedded IR markers and recover the deformable shape on-the-fly which in turn is mapped by the projector. [22] can accommodate any number of projectors/cameras and any 3D geometry, but only at high latency for fast recalibration and not for dynamic objects. [34, 40] use an RGB camera for optical-flow based methods [38, 39] that create an enhanced image to alter the perceived motion or stiffness of the fabric. However, these methods can only handle known motions of the deformable object and do not capture the changing geometry of the deformable object. [4, 8, 9, 16, 20, 21] use multiple high-speed high-end, often customized, cameras, projectors, LED illuminators and fabric with printed IR markers for projection mapping on dynamic deformable materials with imperceptible lag. However, accuracy of projection is extremely low making the projections not stick to the objects and flicker continuously.

Prior work on tracking deformable object in real-time using an RGB-D camera [43–46] cannot be used with dynamic projection mapping since the projection on the surface interferes with the tracking. Therefore, IR camera-based tracking is typically used for dynamic deformable objects.

### 1.2 Our Contribution

Instead of modeling deformations [21], we model the surface itself using a Bezier patch. A Bezier patch can represent all kinds of deformable objects, including stretchable material, and is backward compatible to represent planar rigid surfaces (Figure 2). Further, they lend themselves to accurate approximation and computation on GPU for real time performance even while using an off-the-shelf ToF depth camera. Such a camera usually comes with a registered



**Figure 3: Marker dots printed on a deformable material. The  $(u,v)$  parameterization is shown explicitly with the top-left corner being  $(0,0)$  and bottom-right corner being  $(1,1)$ .**

IR camera and is used to create a calibrated projector-camera system. Using an IR-depth camera enables us to track simple markers (e.g. black dots) leading to a marker-based system, or depth features leading to a marker-less system. Like [16], we use tracking of boundary-based features (e.g. edges or corners), but in deformable materials. We present the first general framework for dynamic projection mapping of deformable materials, including stretchable elastics, using a projector-camera system.

## 2 PROPOSED SYSTEM

Our dynamic projection mapping system consists of a projector, an IR camera and a depth camera – all geometrically calibrated for their intrinsic and extrinsic parameters using the software implementation of [41, 42]. We assume our projection surface to be a smooth, deformable rectangle, that can be elastic as well. Each frame of the IR and depth camera is used to track feature points on the boundary of the deformable projection surface – either explicitly marked using color (e.g. black dots) or implicit depth features at the boundary of the projection surface – creating a marker-based (Section-2.1) or marker-less (Section-2.2) system respectively.

These features are used to compute the parameters of a Bezier patch representing the deformable surface geometry. Finally, the projection is warped according to the fitted patch and projected onto the surface conforming to the shape of the deformable shape.

### 2.1 Bezier Patch Based Model

We model the projection surface as a bi-cubic Bezier patch in 3-D with 16 control points to represent various shapes (Figure 2). It is expressed as:

$$P^d(u, v) = \epsilon(u)^T M C^d M \epsilon(v)$$

where

$$M = \begin{pmatrix} -1 & 3 & -3 & 1 \\ 3 & -6 & 3 & 0 \\ -3 & 3 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{pmatrix}, \quad \epsilon(s) = \begin{pmatrix} s^3 \\ s^2 \\ s \\ 1 \end{pmatrix}$$

$d \in (X, Y, Z)$ ,  $C^d \in \mathbb{R}^{4 \times 4}$  are the control points and  $P^X$ ,  $P^Y$ ,  $P^Z$  are the 3-D coordinates of a point on the Bezier patch. The surface is defined by parameters  $u$  and  $v$  in the horizontal and vertical direction respectively (Figure 3).

Given  $n \geq 16$  correspondences between 3D points ( $P_l^X, P_l^Y, P_l^Z$ ) and their parameters ( $u_l, v_l$ ), we can compute the control points ( $C_{ij}^X, C_{ij}^Y, C_{ij}^Z$ ) using a least square minimization. In every iteration

of our method, the key challenge is to establish these correspondences. We need at least 16 correspondences which we gather only from boundary-based features.

## 2.2 IR-Based Tracking of Boundary Markers

**2.2.1 Surface Design.** We define the boundaries of our surface by equally spaced markers along the border (Figure 3). We pre-compute the  $(u, v)$  parameters of the markers and establish correspondences between them and their  $(u, v)$  parameters on the Bezier. The markers are tracked in each frame to compute the Bezier control points. We use IR imagery in order to avoid interference caused by the visible projection. Dark dots are well-visible to the IR camera against the surface. We use KLT feature tracking [24–26] for fast and robust tracking of the markers.

**2.2.2 Correspondence Estimation.** In this step, we establish the  $(u, v)$  Bezier parameters for each marker and unmarked interior points. Assuming  $D$  boundary markers, let  $d_i = (p_i, k_i)$  denote the detected location of marker  $i$ , where  $p_i \in \mathbb{R}^2$  is the marker’s 2D location in the IR camera coordinates and  $k_i \in \mathbb{R}^3$  is its corresponding 3D depth camera coordinate. We compute the center as the average of all the detected markers  $d_i$  given by  $d_{avg} = (p_{avg}, k_{avg}) = \frac{1}{D}(\sum_{i=1}^D p_i, \sum_{i=1}^D k_i)$ . First, we sort the detected markers by angle along the  $uv$  plane between a reference vector defined by connecting  $p_{avg}$  to  $p_1$  and the vector connecting  $p_{avg}$  to  $p_i$ . Note that the angle between these vectors increases monotonically in the IR camera space as we move clockwise from one marker to the next. Then, we label the corner and edge points by computing the angle each marker makes with its neighbors.

To estimate correspondences in the interior of the surface, we compute the planes passing through every adjacent pair of markers along each of the four boundary curves and the COP of the IR camera. We interpolate the normals of corresponding planes on opposite edges for fixed values of  $(u, v)$  and find the lines of intersection of these planes. The intersections of these lines with the surface provide the 3D points for those  $(u, v)$  parameters.

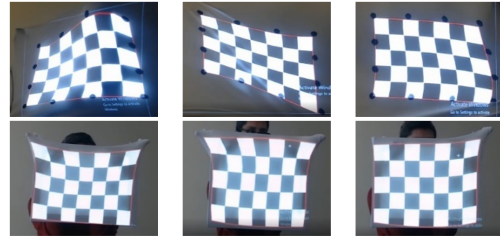
## 2.3 Depth-Based Marker-less Tracking

We extend the aforementioned method for marker-less tracking by leveraging the fact that the distance between each marker and the two corners along the same edge is uniform when no stretch is applied and the distance of points along the boundary from the display corners can be used as a feature. Assuming no occlusion, we segment the display surface using the depth camera from which we extract the *boundary mask*  $B$ . We find the corners of the display using the Harris corner detector [23] on the segmented mask, track them and establish their correspondence with the Bezier patch as mentioned in Section-2.2.

To compute the 3D point  $K$  along an edge parameterized by  $s$ , we minimize the following equation:

$$K(s) = \operatorname{argmin}_B (|D_0 - sL| + |D_1 - (1 - s)L|)$$

where  $D_0, D_1$  are the distances of points from the two corners comprising the edge and  $L$  is the length of that edge. The above method is used to compute points along the boundary for all four edges and



**Figure 4: A checkboard pattern projected onto the display surface experiencing various kinds of stretch. The top-row shows the projection adapting to the marker-based surface as it is stretched outward from behind it, stretched at opposite ends and pulled to the right. The bottom-row shows the projection adapting when the marker-less surface is stretched.**

establish their correspondences. The remaining correspondences i.e. those inside the border, are established as mentioned Section-2.1.

## 2.4 Projection Mapping

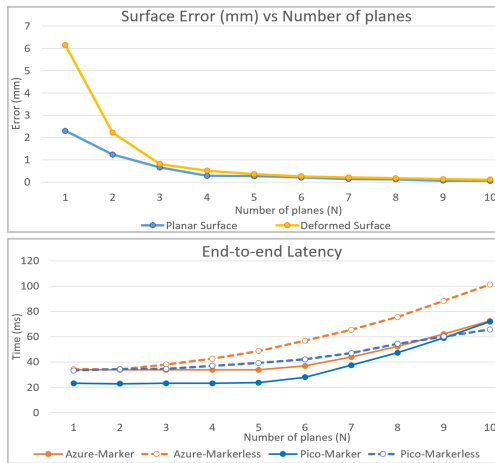
Using all the correspondences, we compute the control points of the Bezier patch (see Sec-2.1) that represents the surface. In order to render the image to be projected, we evaluate the Bezier patch at different  $(u, v)$  image coordinates to get the corresponding 3D surface point. Next, we project them back to the projector coordinates using bilinear interpolation. This mapping does not preserve distances. Therefore, for elastic surfaces, if a corner is pulled to one side, the projection expands to accommodate that stretch. The area of the stretched region also expands accordingly (Figure 4). In order to alleviate this effect, we can constrain the Bezier patch to enable a length-preserving mapping by moving the 3D point correspondences on each edge (except for the corners) such that the distance between all points along that edge becomes equal (please see the video).

**2.4.1 Implementation Details.** Our hardware setup consists of a standard desktop workstation with an Intel Core i7-9700K CPU @ 3.6GHz with 64GB of RAM and an NVIDIA GeForce RTX 2080 SUPER GPU. We used OpenCV’s CUDA implementation to track the markers, compute the distance maps and render the final display. Estimating correspondences and computing the Bezier control points is performed on the CPU. Both the marker-based and marker-less systems can run at close to camera frame rates.

In our implementation, we used Optoma DLP Projector. We tested our system using two different cameras: the Microsoft Azure Kinect (1280x720 @ 30fps) and the CamBoard Pico-Flexx (224x171 when run @ 45fps). Both cameras provide registered IR and depth images, while the Azure Kinect additionally provides a registered RGB image which we do not use.

Evaluating the Bezier patch for every image coordinate is expensive creating a bottleneck for our real-time system. Therefore, we render our final image as a piecewise planar approximation of the Bezier surface. We divide the original display image into a  $N \times N$  grid, evaluate the Bezier patch at the  $(u, v)$  coordinates of the grid





**Figure 5: Surface error (top) and end-to-end latency (bottom) against the number of planes/axis.**

corners and apply a homography, warping each image plane to the projector space for rendering the final display.

### 3 RESULTS AND DISCUSSION

Figure 1 shows the results of our projection mapping system on a marked and marker-less non-elastic surface undergoing different deformations. Note the projection conforms to the shape of the surface for all different deformations (please see video as well).

Figure 4 demonstrates that our method can adapt to stretchable materials for marker-based and marker-less surfaces (see video for applications of our system, including non-planar, curved cross-sections of CT/MRI scans).

We study the effect of the number of markers along the edges and the number of planes used for approximating the Bezier surface during texture mapping on (i) the end-to-end latency, and (ii) the error (in mm) between the Bezier patch and the actual surface geometry.

Increasing the number of markers along each edge does not significantly change the error of the Bezier patch since using four markers/edge provides the required 16 correspondences to compute the patch. Similarly, increasing the number of border markers does not impact the end-to-end latency since GPU compute time dominates and masks the small increase in CPU calculations due to more markers.

However,  $N$ , the number of planes/axis for approximating the Bezier surface, significantly affects both the error and the end-to-end latency. We measured the error as the average distance between the piece-wise planar representation of the Bezier surface and the corresponding points in point cloud when (i) the surface is planar and when (ii) it undergoes cubic deformations. As anticipated, the error for both planar and deformed surface reduces as  $N$  increases, but the error for the latter is much higher than the former for lower values of  $N$  (Figure 5 (top)). This is expected since a fewer number of planes cannot adequately represent a cubic surface.

Figure 5 (bottom) shows the end-to-end latency i.e. the total time between the projection of two successive frames, with increasing

values of  $N$  for both camera types, for marker-based and marker-less systems. As expected, increasing the number of planes increases the end-to-end latency/frame.

Figure 5 (bottom) also reveals another important point. Note that the camera frame latency becomes the bottleneck for both camera types. This is shown by the graph being flat at 33.33ms (30fps) till 5 planes for the Azure Kinect camera (solid orange lines) and at 22.22ms (45fps) till 5 planes for the Pico-Flexx camera (solid blue lines) for the marker-based system. In the marker-less system (Figure 5 (bottom), dotted lines), this advantage is offset by the higher computational overhead for handling automated feature detection in noisy depth data.

### 3.1 Limitations

One of the limitations of the proposed work is that it does not handle occlusion of the markers or the surface for both marker-based and marker-less implementation. One way to overcome this limitation is to compute a rigid transform between successive frames using the visible points and applying that transform to occluded points to estimate their location.

Additionally, while a bi-cubic Bezier patch can represent many deformations, it cannot accurately represent wrinkles or sharp folds on the surface. To improve this, a composite Bezier patch or Non-Uniform Rational B-Spline (NURBS) patch could be used to model the surface.

Finally, the proposed system cannot handle self-occlusion of the surface, an existing problem with projection mapping systems. This can be partially alleviated by extending the proposed system to use multiple projectors, so that light can reach the surface from multiple directions.

## 4 CONCLUSION AND FUTURE WORK

In summary, we present a general Bezier surface-based framework to achieve projection mapping on dynamic deformable stretchable materials using off-the-shelf ToF depth camera and registered IR camera accompanying it. Higher speed cameras can reduce any remaining perceptible latency in the system. We would like to extend the system further to handle non-diffused surfaces and fluids, handle occlusion and scale to multiple projectors.

## REFERENCES

- [1] Yuichiro Fujimoto, Ross T. Smith, Takafumi Taketomi, Goshiro Yamamoto, Jun Miyazaki, Hirokazu Kato, and Bruce H. Thomas. 2014. Geometrically-Correct Projection-Based Texture Mapping onto a Deformable Object. In *IEEE Transactions on Visualization and Computer Graphics*. 20. 540-9. DOI:10.1109/TVCG.2014.25
- [2] Tomohiro Sueishi, Hiromasa Oku, and Masatoshi Ishikawa. 2015. Robust high-speed tracking against illumination changes for dynamic projection mapping. In *IEEE Virtual Reality (VR), Arles, 2015*, pp. 97-104, doi: 10.1109/VR.2015.7223330.
- [3] Christian Siegl, Matteo Colaianni, Lucas Thies, Justus Thies, Michael Zollhöfer, Shahram Izadi, Marc Stamminger, and Frank Bauer. 2015. Real-time pixel luminance optimization for dynamic multi-projection mapping. In *ACM Trans. Graph.* 34, 6, Article 237 (November 2015), 11 pages. DOI:https://doi.org/10.1145/2816795.2818111
- [4] Gaku Narita, Yoshihiro Watanabe, and Masatoshi Ishikawa. 2015. 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 (VRST '15)*. Association for Computing Machinery, New York, NY, USA, 149–152. DOI:https://doi.org/10.1145/2821592.2821618
- [5] Yi Zhou, Shuangjiu Xiao, Ning Tang, Zhiyong Wei, and Xu Chen. 2016. Pmomo: Projection Mapping on Movable 3D Object. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems (CHI '16)*. Association



- for Computing Machinery, New York, NY, USA, 781–790. DOI:<https://doi.org/10.1145/2858036.2858329>
- [6] Christian Siegl, Vanessa Lange, Marc Stamminger, Frank Bauer, and Justus Thies. 2017. FaceForge: Markerless Non-Rigid Face Multi-Projection Mapping. In *IEEE Transactions on Visualization and Computer Graphics*, vol. 23, no. 11, pp. 2440–2446, Nov. 2017, doi: 10.1109/TVCG.2017.2734428.
  - [7] Vanessa Lange, Christian Siegl, Matteo Colaianni, Marc Stamminger, and Frank Bauer. 2017. Robust blending and occlusion compensation in dynamic multi-projection mapping. In *Proceedings of the European Association for Computer Graphics: Short Papers (EG '17)*. Eurographics Association, Goslar, DEU, 1–4. DOI:<https://doi.org/10.2312/egsh.20171000>
  - [8] Gaku Narita, Yoshihiro Watanabe, and Masatoshi Ishikawa. 2017. Dynamic Projection Mapping onto Deforming Non-Rigid Surface Using Deformable Dot Cluster Marker. In *IEEE Transactions on Visualization and Computer Graphics*, vol. 23, no. 3, pp. 1235–1248, 1 March 2017, doi: 10.1109/TVCG.2016.2592910.
  - [9] Leo Miyashita, Yoshihiro Watanabe, and Masatoshi Ishikawa. 2018. MIDAS projection: markerless and modelless projection mapping for material representation. In *ACM Trans. Graph.* 37, 6, Article 196 (November 2018), 12 pages. DOI:<https://doi.org/10.1145/3272127.3275045>
  - [10] Yuri Mikawa, Tomohiro Sueishi, Yoshihiro Watanabe, and Masatoshi Ishikawa. 2018. VarioLight: hybrid dynamic projection mapping using high-speed projector and optical axis controller. In *SIGGRAPH Asia 2018 Emerging Technologies (SA '18)*. Association for Computing Machinery, New York, NY, USA, Article 17, 1–2. DOI:<https://doi.org/10.1145/3275476.3275481>
  - [11] Hirota Asayama, Daisuke Iwai, and Kosuke Sato. 2018. Fabricating Diminishable Visual Markers for Geometric Registration in Projection Mapping. In *IEEE Transactions on Visualization and Computer Graphics*, vol. 24, no. 2, pp. 1091–1102, 1 Feb. 2018, doi: 10.1109/TVCG.2017.2657634.
  - [12] Philipp Kurth, Vanessa Lange, Christian Siegl, Marc Stamminger, and Frank Bauer. 2018. Auto-Calibration for Dynamic Multi-Projection Mapping on Arbitrary Surfaces. In *IEEE Transactions on Visualization and Computer Graphics*, vol. 24, no. 11, pp. 2886–2894, Nov. 2018, doi: 10.1109/TVCG.2018.2868530.
  - [13] Shingo Kagami, and Koichi Hashimoto. 2019. Animated Stickies: Fast Video Projection Mapping onto a Markerless Plane through a Direct Closed-Loop Alignment. In *IEEE Transactions on Visualization and Computer Graphics*, vol. 25, no. 11, pp. 3094–3104, Nov. 2019, doi: 10.1109/TVCG.2019.2932248.
  - [14] Lihui Wang, Hongjin Xu, Yunpu Hu, Satoshi Tabata, and Masatoshi Ishikawa. 2019. Dynamic Depth-of-Field Projection for 3D Projection Mapping. In *Extended Abstracts of the 2019 CHI Conference on Human Factors in Computing Systems (CHI EA '19)*. Association for Computing Machinery, New York, NY, USA, Paper INT021, 1–4. DOI:<https://doi.org/10.1145/3290607.3313246>
  - [15] Christoph Resch, Peter Keitler, and Gudrun Klinker. 2014. Sticky projections – A new approach to interactive shader lamp tracking. In *IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*, Munich, 2014, pp. 151–156, doi: 10.1109/ISMAR.2014.6948421.
  - [16] Naoki Hashimoto, Ryo Koizumi, and Daisuke Kobayashi. 2017. Dynamic Projection Mapping with a Single IR Camera. *Int. J. Comput. Games Technol.* 2017 (February 2017), 2. DOI:<https://doi.org/10.1155/2017/4936285>
  - [17] Christian Siegl, Matteo Colaianni, Marc Stamminger, and Frank Bauer. 2016. Stray-light compensation in dynamic projection mapping. In *SIGGRAPH ASIA 2016 Technical Briefs (SA '16)*. Association for Computing Machinery, New York, NY, USA, Article 21, 1–4. DOI:<https://doi.org/10.1145/3005358.3005364>
  - [18] Vanessa Lange, Christian Siegl, Matteo Colaianni, Philipp Kurth, Marc Stamminger, and Frank Bauer. 2016. Interactive Painting and Lighting in Dynamic Multi-Projection Mapping. In *De Paolis L., Mongelli A. (eds) Augmented Reality, Virtual Reality, and Computer Graphics. AVR 2016. Lecture Notes in Computer Science*, vol. 9769. Springer, Cham
  - [19] Daniel Saakes, Hui-Shyong Yeo, Seung-Tak Noh, Gyeol Han, and Woontack Woo. 2016. Mirror Mirror: An On-Body T-shirt Design System. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems (CHI '16)*. Association for Computing Machinery, New York, NY, USA, 6058–6063. DOI:<https://doi.org/10.1145/2858036.2858282>
  - [20] Amit H. Bermano, Markus Billeter, Daisuke Iwai, and Anselm Grundhöfer. 2017. Makeup Lamps: Live Augmentation of Human Faces via Projection. *Comput. Graph. Forum* 36, 2 (May 2017), 311–323.
  - [21] Jürgen Steimle, Andreas Jordt, and Pattie Maes. 2013. Flexpad: highly flexible bending interactions for projected handheld displays. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '13)*. Association for Computing Machinery, New York, NY, USA, 237–246. DOI:<https://doi.org/10.1145/2470654.2470688>
  - [22] Mahdi Abbaspour Tehrani, Gopi M, and Aditi Majumder. 2019. Automated Geometric Registration for Multi-Projector Displays on Arbitrary 3D Shapes Using Uncalibrated Devices. In *IEEE Transactions on Visualization and Computer Graphics*, doi: 10.1109/TVCG.2019.2950942.
  - [23] Chris Harris and Mike Stephens. 1988. A Combined Corner and Edge Detector. In *Alvey Vision Conference*.
  - [24] Bruce D. Lucas and Takeo Kanade. 1981. An iterative image registration technique with an application to stereo vision. *Proceedings of Imaging Understanding Workshop*, pages 121–130
  - [25] Carlo Tomasi and Takeo Kanade. 1991. Detection and Tracking of Point Features. *Carnegie Mellon University Technical Report CMU-CS-91-132*, April 1991
  - [26] Jianbo Shi and Carlo Tomasi. 1994. Good Features to Track. *IEEE Conference on Computer Vision and Pattern Recognition*, pages 593–600, 1994.
  - [27] Ramesh Raskar, Michael S. Brown, Ruigang Yang, Wei-Chao Chen, Greg Welch, Herman Towles, B. Seales, and Henry Fuchs. 1999. Multi projector displays using camera-based registration. *Proc. of IEEE Vis*, 1999
  - [28] Behzad Sajadi and Aditi Majumder. Automatic registration of multiple projectors on swept surfaces. 2010. In *ACM Virtual Reality and Software Technology*, 2010.
  - [29] Behzad Sajadi and Aditi Majumder. 2011. Auto-calibrating projectors for tiled displays on piecewise smooth vertically extruded surfaces. In *IEEE Transactions on Visualization and Computer Graphics* 2011
  - [30] Behzad Sajadi and Aditi Majumder. 2009. Markerless View Independent Geometric Registration of Multiple Distorted Projectors on Vertically Extruded Surfaces Using a Single Uncalibrated Camera. In *IEEE Transactions on Visualization and Computer Graphics*, 2009.
  - [31] Ramesh Raskar. 2000. Immersive planar displays using roughly aligned projectors. In *Proc. of IEEE VR*, 2000
  - [32] Michael S. Brown and W. Brent Seales. 2002. A practical and flexible tiled display system. In *10th Pacific Conference on Computer Graphics and Applications*, 2002. Proceedings., Beijing, China, 2002, pp. 194–203, doi: 10.1109/PCCGA.2002.1167859.
  - [33] Tyler Johnson and Henry Fuchs. 2007. Real-time projector tracking on complex geometry using ordinary imagery. In *IEEE CVPR Workshop on Projector/Camera Systems (PROCAMS)*, 2007
  - [34] Parinya Punpongsonon, Daisuke Iwai, and Kosuke Sato. 2020. FleXeen: Visually Manipulating Perceived Fabric Bending Stiffness in Spatial Augmented Reality. In *IEEE Transactions on Visualization and Computer Graphics*, vol. 26, no. 2, pp. 1433–1439, 1 Feb. 2020, doi: 10.1109/TVCG.2018.2871044.
  - [35] Ramesh Raskar, Greg Welch, Kok-Lim Low, and Deepak Bandyopadhyay. 2001. Shader Lamps: Animating Real Objects with Image-Based Illumination. In *Eurographics Workshop on Rendering Techniques EGSR 2001*, pp. 89–102.
  - [36] Ramesh Raskar, Jeroen van Baar, Paul Beardsley, Thomas Willwacher, Srinivas Rao, and Clifton Forlines. 2006. ILamps: geometrically aware and self-configuring projectors. In *ACM SIGGRAPH 2006 Courses (SIGGRAPH '06)*. Association for Computing Machinery, New York, NY, USA, 7–es. DOI:<https://doi.org/10.1145/1185657.1185802>
  - [37] Tyler Johnson, Florian Gyarfas, Rick Skarbez, Herman Towles, and Henry Fuchs. 2007. A Personal Surround Environment: Projective Display with Correction for Display Surface Geometry and Extreme Lens Distortion. In *Proceedings of IEEE Virtual Reality Conference, Virtual Reality 2007*, pp. 147–154.
  - [38] Katherine L. Bouman, Bei Xiao, Peter Battaglia, and William T. Freeman. 2013. Estimating the material properties of fabric from video. In *Proc. IEEE Int. Conf. Comput. Vis.*, 2013, pp. 1984–1991.
  - [39] Wenyan Bi and Bei Xiao. 2016. Perceptual constancy of mechanical properties of cloth under variation of external forces. In *Proceedings of the ACM Symposium on Applied Perception (SAP '16)*. Association for Computing Machinery, New York, NY, USA, 19–23. DOI:<https://doi.org/10.1145/2931002.2931016>
  - [40] Takahiro Kawabe, Taiki Fukiage, Masataka Sawayama, and Shin'ya Nishida. 2016. Deformation Lamps: A Projection Technique to Make Static Objects Perceptually Dynamic. *ACM Trans. Appl. Percept.* 13, 2, Article 10 (March 2016), 17 pages. DOI:<https://doi.org/10.1145/2874358>
  - [41] Daniel Moreno and Gabriel Taubin. 2012. Simple, Accurate, and Robust Projector-Camera Calibration. In *2012 Second International Conference on 3D Imaging, Modeling, Processing, Visualization & Transmission*, Zurich, 2012, pp. 464–471, doi: 10.1109/3DIMPVT.2012.77.
  - [42] <http://mesh.brown.edu/calibration/software.html>
  - [43] Julien Pilet, Vincent Lepetit and Pascal Fua. 2005. Real-time nonrigid surface detection. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, San Diego, CA, USA, 2005, pp. 822–828 vol. 1, doi: 10.1109/CVPR.2005.293.
  - [44] Ibai Leizea, Hugo Álvarez, Iker Aguinaga and Diego Borro. 2014. Real-time deformation, registration and tracking of solids based on physical simulation. In *IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*, Munich, 2014, pp. 165–170, doi: 10.1109/ISMAR.2014.6948423.
  - [45] Dat Tien Ngo, Jonas Östlund and Pascal Fua. 2016. Template-Based Monocular 3D Shape Recovery Using Laplacian Meshes. In *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 38, no. 1, pp. 172–187, 1 Jan. 2016, doi: 10.1109/TPAMI.2015.2435739.
  - [46] Antoine Petit, Vincenzo Lippello and Bruno Siciliano. 2015. Real-time tracking of 3D elastic objects with an RGB-D sensor. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Hamburg, 2015, pp. 3914–3921, doi: 10.1109/IROS.2015.7353928.