Optics EXPRESS

Automated calibration of multi-camera-projector structured light systems for volumetric high-speed 3D surface reconstructions

MARC E. DEETJEN[1,2] AND DAVID LENTINK[1,3]

[1] *Mechanical Engineering Department, Stanford University, Stanford, CA 94305, USA*
[2] *mdeetjen@stanford.edu*
[3] *dlentink@stanford.edu*

Abstract: It is challenging to calibrate multiple camera-projector pairs for multi-view 3D surface reconstruction based on structured light. Here, we present a new automated calibration method for high-speed multi-camera-projector systems. The method uses printed and projected dot patterns on a planar calibration target, which is moved by hand in the calibration volume. Calibration is enabled by automated image processing and bundle-adjusted parameter optimization. We determined the performance of our method by 3D reconstructing a sphere. The accuracy is -0.03 ± 0.09 % as a percentage of the diameter of the calibration volume. Applications include quality control, autonomous systems, engineering measurements, and motion capture, such as the preliminary 3D reconstruction of a bird in flight we present here.

1. Introduction

Structured light systems are widely used to generate 3D reconstructions of objects because they are accurate, non-contact, efficient, and can reconstruct the surface geometry at high temporal and spatial resolution. These systems use a camera to record light patterns generated by a projector that are reflected by the surface of a 3D object. Depending on the type of light pattern used, even rapidly deforming objects can be 3D reconstructed. Temporally encoded light patterns, such as binary coding [1, 2] and phase-shifted coding [3, 4], limit 3D reconstruction to motionless or slowly moving objects, because they require comparison of consecutive frames. Spatially encoded light patterns on the other hand, are robust to inter-frame movement because each frame can be 3D reconstructed individually [5]. Some structured light systems use grayscale patterns which cannot be projected at high frame rates [6–9], while other systems are limited by using low-speed (< 100 Hz) projectors [10, 11]. However, by using a high-speed camera-projector system and a 1-bit light pattern, such as psuedo-random dots [12, 13] or psuedo-random intersecting lines [14], 3D reconstructions of deforming geometries can be achieved at any frame rate with sufficient lighting. Regardless of which projection pattern is used, computation of the 3D surface geometry is based on triangulation algorithms that rely on image processing and camera-projector calibration.

The calibration of cameras is a well-studied problem, which involves calibrating the intrinsic camera parameters that define the imaging sensor geometry, called intrinsic parameters. If there are multiple cameras, this also includes calibrating the relative position and orientation between cameras, called extrinsic parameters. Early calibration methods used precisely manufactured 3D targets with pre-determined marker locations [15, 16]. Recently, simpler targets have become preferred, because they can more easily be adapted and deployed in different calibration volumes. A common method used in commercial motion capture systems relies on a wand as a calibration target, which consists of two or more reflective spheres at pre-determined distances [17]. Another popular calibration target in vision studies is a 2D planar surface with a checkerboard pattern of pre-determined dimensions [18, 19]. Both of these calibration targets are manually moved through the imaging volume in different orientations. The most commonly applied model for the

#347097 https://doi.org/10.1364/OE.26.033278
Journal © 2018 Received 1 Oct 2018; revised 13 Nov 2018; accepted 15 Nov 2018; published 6 Dec 2018

camera's intrinsic parameters is the pinhole model [20]. This simple model can be improved using parameters that compensate for camera distortion [21]. Other models such as polynomial fitting are also used [22, 23].

Although a projector can usually be modeled simply as the inverse of a camera, the calibration of projectors is more cumbersome. The calibration must be completed with the help of cameras, because a projector cannot observe the calibration target. Therefore, information about how light rays are projected is found by projecting light rays onto a light reflecting surface, which is then captured by a camera. Several approaches for camera-projector calibration rely on self-calibration where the calibration is adjusted concurrently with 3D reconstruction [24–30]. Because of this, self-calibration is operational for only a couple reconstruction methods. However, calibrating before reconstruction is usually preferred because it is more stable and accurate in most cases [31], and it also works for any reconstruction method including high-speed applications.

There exist three primary approaches to calibrate a structured light system: the camera and projector are calibrated during consecutive separate steps, they are calibrated simultaneously, or some combination of both calibration approaches is used. Each approach typically relies on a planar calibration target that is moved through the calibration volume. The calibration target requires a pre-determined pattern on it for each device: a printed pattern of pre-determined dimensions for the camera calibration and a projected pattern captured by the camera for the projector calibration. The patterns are designed carefully so that they can be easily separated from each other and so that keypoints (a visual landmark) on them can be easily identified. The keypoints are used to optimize the calibration model by minimizing reprojection error. Reprojection error is the difference between the locations of the keypoints that are predicted by the calibration model and the locations identified in the camera image.

The most common approach for calibrating a structured light system uses separate steps: first the camera is calibrated, then the projector. The camera is calibrated using the printed pattern, after which the now-calibrated camera is used to calibrate the projector. The projector calibration step requires projecting a pre-determined pattern on the calibration target in multiple positions, which differ from the positions used to calibrate the camera [32–37]. Whereas this temporal separation eliminates image processing issues that could be caused by spatially overlapping patterns, the separate calibration steps can cause errors to propagate.

Another approach for calibrating a structured light system uses a combination of multiple calibration approaches: calibration of the two devices in separate steps, and calibration of the two devices simultaneously. These approaches use a robot or apparatus to repeatably move a calibration target to multiple, pre-determined positions. The same positions are used for both the camera calibration (printed pattern) and the projector calibration (projected pattern), but a different calibration target is used so that the patterns do not overlap [31, 38]. In this manner, the patterns are still temporally separated, but approximately share the same spatial location for increased accuracy, by mathematically optimizing the calibration simultaneously. One downside of this method is that it requires an expensive mechanism to move the calibration target. This mechanism needs to be adapted to different calibration volumes (e.g. small versus large), and new errors are introduced with the magnitude depending on the repeatability and precision of the mechanism.

Finally, a more integrated approach for calibrating a structured light system calibrates the camera and projector simultaneously. This strategy prevents excess errors from separate calibration steps. In this case, the multiple positions of the calibration target are simultaneously used for both the camera and projector calibration. To identify each of the spatially overlapping patterns, both printed and projected, the camera image needs to be filtered. Filtering is simplified by carefully designing each pattern so that it can isolated with image processing steps [39, 40].

Currently, volumetric 3D surface reconstructions of objects are usually accomplished using a rotation table to give structured light systems a 360 degree view [41–43]. However, for (rapidly) deforming objects, multiple view angles are required simultaneously. For multiple cameras, such calibration techniques exist [17,44–46], but 3D reconstruction solely using cameras is not precise for uniform surfaces. For multiple cameras and projectors, robust calibration methods remain to be developed. An existing method relies on self-calibration using fringe projection [25], but self-calibration is less stable and accurate as discussed above, and the reconstruction method used does not work at high-speeds. Another approach to calibrating a multi-camera-projector system is to calibrate each camera-projector pair first, then calibrate all cameras separately [47]. Established methods exist for both steps. However, these sequential steps cause errors to propagate, and the additional calibration steps are cumbersome.

To automatically and accurately calibrate multi-camera-projector structured light systems for volumetric high-speed 3D surface reconstruction, we developed a new method. This method is based on an integrated calibration target, image acquisition procedure, and algorithmic solution. Automation is achieved with sparse dot calibration patterns and corresponding image processing techniques. High accuracy is achieved by calibrating all camera and projector devices simultaneously using established calibration models. This new technique can then be combined with the single-shot monochrome 3D reconstruction method we recently developed [14]. Used together, we demonstrate the technique by reconstructing the volumetric 3D surface of a sphere and freely flying bird at 3200 Hz. We have thus established high-speed volumetric 3D reconstruction for rapidly deforming objects.

2. Materials and methods

2.1. High-speed multi-view structured light system

To create a high-speed multi-view structured light system to 3D reconstruct deforming volumes, we set up four camera-projector pairs to image an object from four view angles [Fig. 1]. The four high-speed cameras (Phantom Miro M310; Vision Research, Wayne, NJ, USA) and four high-speed projectors (DLP® LightCrafter™ E4500MKII™; EKB Technologies, Bat-Yam, Israel) were synchronized temporally using an external trigger box (9600+ Series Digital Delay Pulse Generator; Quantum Composers, Inc., Bozeman, MT, USA). The calibration pipeline computed the calibration parameters based on imaging at 1000 Hz. Subsequently, these calibration parameters were used to make frame-by-frame 3D reconstructions based on 3200 Hz imaging of stationary sphere and a flying bird. The frame rate was reduced during calibration to enable the projectors to switch colors as well as patterns. A sphere (3-1/4 Gray PVC Ball; Precision Plastic Ball Co., Franklin Park, IL, USA) of known size (diameter 82.55 ± 0.13 mm) and a flying bird were 3D reconstructed to demonstrate the structured light system. The bird was a 4-year-old female near-white Pacific parrotlet [*Forpus coelestis* (Lesson 1847), 27-29 g, 0.23 m wingspan], which was trained using positive reinforcement to fly between perches 0.5 m apart. All birds received food and water ad libitum, their cages were enriched, and all training and experimental procedures were approved by Stanford's Administrative Panel on Laboratory Animal Care.

The four cameras were placed approximately 0.8 m away from the center of the calibration volume while the four projectors were placed approximately 0.5 m away. To reduce cross-talk between the four camera-projector pairs, every camera-projector pair used a specific projector center wavelength (either blue: 455 nm, green: 530 nm, or red: 625 nm) in combination with a custom light filter for the camera (72.0 mm diameter custom filters; blue: low-pass filter with 475 nm cutoff, green: band-pass filter centered around 540 nm with 80 nm full width at half maximum, red: high-pass filter with 600 nm cutoff; Andover Corporation, Salem, NH, USA). The three RGB wavelengths were selected based on the power specifications of our off-the-shelf high-speed projectors to maximize projected light intensity. Because we defined three color channels for four camera-projector pairs, we temporally separated the two pairs that relied on the same color. We did this by alternating the reconstruction frames, sacrificing their reconstruction speed by a factor of two. Further work would be required to develop algorithms that remove

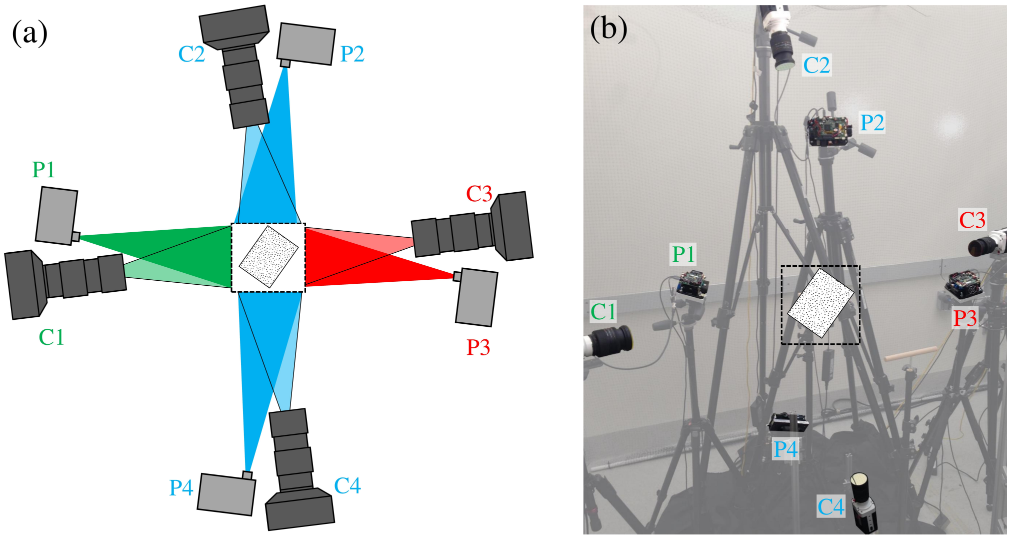

Fig. 1. The multi-view structured light system. a) Schematic showing the configuration of four camera and projector pairs. To reduce cross-talk between the four projected patterns, we use three different colors matched by camera color filters for pairs 1-3. The color of the 4th pair is duplicated from the 2nd pair, and the signals are separated temporally by alternating reconstruction frames. The other signals are temporally synchronized frame-by-frame. The calibration volume is indicated by the dashed contour, while the 2D calibration target that is moved through the volume is indicated by the rectangle with dots on it. b) Actual setup with the four pairs of high-speed cameras and projectors that we used to reconstruct the flying bird from four camera view angles. The calibration volume and target are shown as in (a). The setup used to reconstruct the sphere and in the majority of the calibration accuracy results had similar spacing and the same camera-projector configuration.

crosstalk without trading in image speed. Future systems could also take advantage of other light source and filter combinations to increase the number of camera-projector pairs.

2.2. Automated technique for calibrating multiple cameras and projectors

To enable 3D surface reconstruction with multi-view structured light systems, the intrinsic and extrinsic parameters of the cameras and projectors need to be calibrated. To calibrate the system in a simple automated manner, we designed a 2D calibration target that can be easily adapted for different calibration volumes. Unlike most calibration targets, this calibration target is two-sided so that every camera and projector can simultaneously image it or project onto it respectively. All camera and projector parameters are calibrated simultaneously by the user moving the calibration target through the volume, and by switching between the pattern printed on the target (for camera calibration) and the pattern projected onto the target (for both camera and projector calibration) at high-speed. Afterwards, we apply a linear correction for the small displacement of the calibration target between frames. As a result, each spatial position of the calibration target throughout the calibration procedure is approximately identical between every camera and projector, while each pattern is separated temporally. Because we image the calibration patterns simultaneously on both the front and back side of the calibration target, and the two sides are related by a rigid transformation, the relative positions of both printed and projected pattern dots captured by all cameras can be compared. This enables accurate extrinsic parameter computation between all devices.

The calibration target was made by printing two unique gray pseudo random dot patterns onto white sticker paper, which were adhered to either side of a flat aluminum plate. For camera

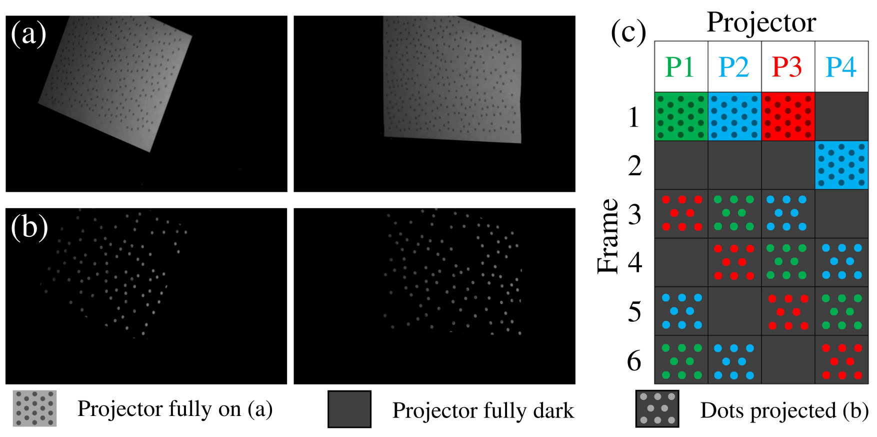

Fig. 2. Multiple-view structured light calibration technique. a) An aluminum plate covered on both sides with printed dots is moved around the calibration volume by the user (two calibration target positions recorded by a single camera are shown). The gray, pseudo-random dots are printed on two paper stickers which are adhered to each side of the calibration target, while black tape covers the edges of the target. The printed dots for camera calibration are recorded by each high-speed camera when its paired projector is fully on (each pixel is illuminated). The images are 12-bit and thus have much better contrast than can be conveyed in print. b) The same two calibration target positions are shown as in (a), but now pseudo-random dots are projected onto the target and recorded by the high-speed camera for projector calibration. c) A sequence of 6 frames is projected and recorded at 1000 Hz every 0.5 seconds for our specific implementation. Each projector projects four different frames imaged by different cameras. First, a projector projects uniform light (a fully on frame) in the color of the associated camera filter to make the gray dots on the calibration target visible to the paired camera. Second, a pseudo-random dot pattern is projected in all three colors so they can be seen by the other cameras to determine the connectivity of the multi-view system.

Table 1. Calibration pattern descriptions. Each calibration pattern consists of uniform dots spread in a pseudo-random pattern with locations determined by a custom Matlab code. The printed dots are used for camera calibration while the projected dots are used for both camera and projector calibration. There are two separate printed dot patterns, one for each side of the calibration target.

	Printed side 1	Printed side 2	Projected
Dot diameter	3.61 mm	3.61 mm	15 pixels
Dot min. spacing	4.57 mm	4.57 mm	20 pixels
Number of dots	334	323	308
Pattern width	190.5 mm	190.5 mm	912 pixels
Pattern height	254.0 mm	254.0 mm	1140 pixels

calibration, the printed calibration dots on the target are made visible by projecting uniform light with the projectors [Fig. 2(a)]. The projectors are calibrated by imaging a pseudo random dot pattern projected on the calibration target [Fig. 2(b)]. The printed and projected dots may overlap, and in order to simplify the image processing steps needed to separate the patterns, we carefully designed the dot patterns. The printed dots are gray rather than black, and both patterns are somewhat sparse with specifications in Table 1. Further, the local uniqueness of the dot pattern enables us to use pattern recognition when only parts of the dot pattern are captured by the camera, such as when the hand of the user blocks the view. For single camera calibration, a locally unique pattern is not required. However, for multiple cameras or any number of projectors, a locally unique pattern is needed to match corresponding dots that are projected or captured by different devices. Finally, we used black tape to cover the edges of the plate to eliminate reflections. Combined, these solutions enable us to automatically identify and associate the printed and projected dots on both sides of the calibration target as imaged from multiple views.

To increase accuracy and reduce the total processing time, we designed a specific order for illuminating the calibration target for multi-view camera and projector calibration [Fig. 2(c)]. First, we image the printed pattern for each camera. This is done by projecting uniform light on the calibration target at the specific wavelength of the filter attached to the projector's paired camera [Figs. 2(a) and 2(c)]. Next, we increase the accuracy of the calibration by maximizing the number of interconnections between cameras and projectors. This is accomplished by directing every projector to sequentially project the calibration pattern in each wavelength [Figs. 2(b) and 2(c)], so that it can be imaged by every camera to the degree that they are observable. Because the calibration target moves slightly in between frames, we condense the sequence to as few frames as possible by projecting in different wavelengths simultaneously. For applications in which the calibration target is close to motionless in between frames, image processing can be simplified by projecting an inverse dot pattern as well. However, to make the calibration procedure more simple to implement, we decided to move the calibration target by hand. Consequently, we need to correct for small movements between frames mathematically in Section 2.6. In order to reduce the total processing time of the resulting 6 frame sequences, we only recorded this sequence every 0.5 seconds. This allows the calibration target to move substantially between sequences. Whereas our example implementation is based on combining four cameras and four projectors, our calibration strategy can be generalized and applied to larger systems in a straightforward manner.

2.3. Image processing algorithm for dot detection

Once the printed and projected dot patterns are recorded by all of the cameras, the dots in each frame are automatically detected using image processing. Regardless of whether the dots are printed or projected, the same image processing procedure is used. All data processing was conducted in MATLAB R2017b (MathWorks, Inc., Natick, MA, USA).

To begin, all possible dot locations are identified based on the grayscale contrast in the original image captured by the camera [Figs. 3(a) and 3(e)]. The contrast in the original image is first normalized against the 99th percentile of its intensity so that all images have a similar grayscale distribution. The 99th percentile is used so that any stray light from incorrect projectors does not skew results. A custom code is then used to binarize the image, converting it from grayscale to black and white. The custom code takes the locally adaptive threshold of the image [48] [background of Figs. 3(b) and 3(f)] in 50×50 pixel segments with 50 % overlap. We set the minimum variance threshold of the pixel segments to 1e-8 (each pixel intensity ranges from 0-1) to identify all potential dots, even in regions with low lighting contrast. Because of this, excess blobs are identified in regions of low contrast and finite camera noise [e.g. Fig. 3(f)], but subsequent image filtering processes filter out these excess blobs. Regions of grouped black pixels, or blobs, are identified as possible printed or projected dots. If a global threshold is used instead, details of entire portions of the image might be eliminated. This happens when image brightness varies across the entire image, which is typical for most experimental settings.

Once the set of all possible dot locations is identified, filtering techniques are used to identify the blobs that are most likely printed or projected dots. For each blob, four parameter descriptors are computed and normalized around a mean of zero and a standard deviation of one: blob area, ellipticity, x position, and y position. The Alpha Shapes algorithm [49] is first used to eliminate clear outlier blobs after which k-means clustering [50] is used to separate regions of blobs within

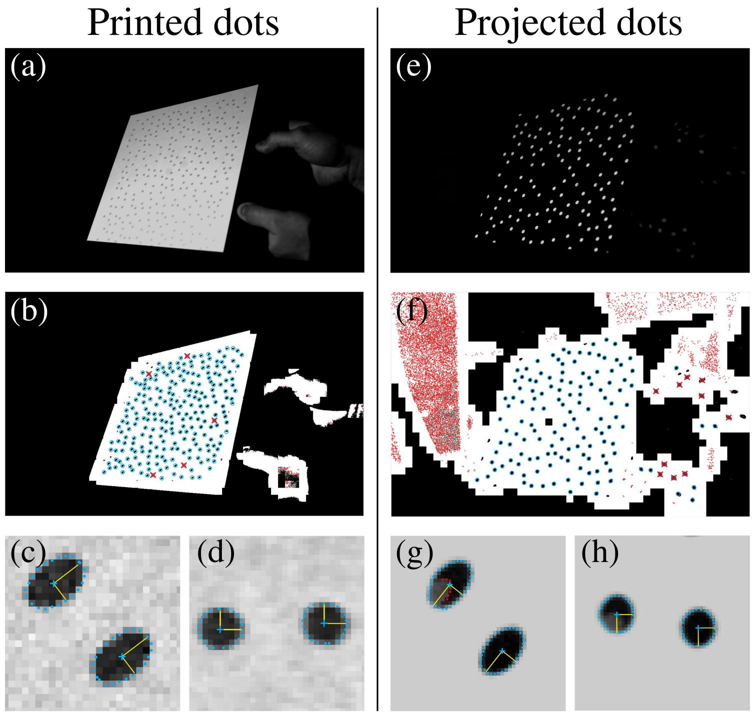

Fig. 3. Image processing steps to detect dots. a-d) Identification of the printed dots captured by the high-speed camera. e-h) Identification of the projected dots captured by the high-speed camera. a,e) Original image as captured by the camera. The images captured by the cameras are 12-bit and thus have much better contrast than can be conveyed in print. b,f) Black and white image after local adaptive thresholding, which we designed to detect dots even in low contrast lighting conditions leading to excess dot detection in low contrast regions with finite camera noise. Black regions of pixels are denoted as blobs and recorded as possible locations of printed or projected dots. Red points (•) are blobs eliminated by K-means clustering based on the following blob parameters: number of similar blobs, area, ellipticity, x position, and y position. Red crosses (X) indicate elliptical blobs that have been eliminated in a second round of filtering in which they were identified as ellipses that are substantially dissimilar from the rest. Blue circles (O) are the final set of detected dots. c,g) A subpixel edge detector is used to identify the edges of elliptical blobs (blue points). We then eliminate outlier edges due to overlapping printed and projected dots (labeled in red in g) using the RANSAC algorithm applied to an elliptic fit. The fitted ellipses are used to compute and display the center, major and minor axis, and orientation of each elliptical blob. d,h) Finally, we apply skew correction to the image (c,g) based on the median orientation and major to minor axis ratio of the detected elliptical dots. This correction transforms the ellipses back into circles and ensures that the spacing between circles is proportional to the spacing between printed or projected dots.

the 4-dimensional parameter space. Next each region of blobs is rated using Eq. (1)

$$\mathrm{S_b} = \mathrm{n_b}\left(\overline{\mathrm{E_b}}\right)^{-2}\left(\overline{\mathrm{A_b}}\right), \tag{1}$$

where for region of blobs b, S_b is the score, n_b is the number of blobs, $\overline{\mathrm{E_b}}$ is the mean ellipse fit error, and $\overline{\mathrm{A_b}}$ is the mean area of all the blobs in that region. A higher score indicates a higher likelihood that the region of interest contains blobs corresponding to the printed or projected dots. K-means clustering is run 10 distinct times to cluster the blobs into regions. This starts with a single region on the first distinct run, and ends with 10 regions. The best region score over the 10 distinct runs is chosen. The parameter boundaries defining the best region are then expanded slightly for each of the four parameters (area, ellipticity, x position, and y position) in case the clustering was too limiting. The blobs rejected by these steps are shown as red points (•) in Figs. 3(b) and 3(f) and the remaining set of blobs we will now call elliptical blobs.

Once a set of elliptical blobs has been identified as possible printed or projected dots, the subpixel edges of each elliptical blob are detected in order to compute its subpixel center. To enable effective subpixel edge detection, image contrast is equalized by using two linear fits of the image intensity. One fit equalizes the background brightness in the region of elliptical blobs while the other fit equalizes the brightness of the elliptical blobs themselves. We compute linear polynomial image-intensity surfaces fitted to the average brightness levels surrounding each elliptical blob and internal to each elliptical blob, using the RANSAC algorithm to remove outliers [51]. The more uniform contrast enables the edge detector [52] to robustly find the edges of the elliptical blobs with subpixel accuracy [Figs. 3(c) and 3(g)]. To eliminate outliers in the identified edges, in particular due to overlapping printed and projected dots, we apply the RANSAC algorithm to an elliptical fit of the detected edges. From this elliptical fit we find the center, major and minor axis, and orientation of each elliptical blob [Figs. 3(c) and 3(g)].

As a final filtering step, most elliptical blobs that do not lie on the calibration target are eliminated. This is done by using the Alpha Shapes algorithm to eliminate elliptical blobs dissimilar from the rest. Elliptical blobs are grouped based on how similar they are to each other by comparing the major axis, minor axis, and orientation angle. This grouping is used because all dots projected onto the same planar surface, like the calibration target, will share similar sizes and orientations. In contrast, dots projected onto a non-planar surface, such as the user's hand, will have varied sizes and orientations. In this way, elliptical blobs on the calibration target are mostly retained while many of the elliptical blobs that were projected onto other surfaces are eliminated. The elliptical blobs eliminated by this final filter are labeled with red crosses (X), while the remaining detected dots are labeled with blue circles (O) in Figs. 3(b) and 3(f).

2.4. *Algorithm for matching detected dots with pre-determined printed and projected dots*

After generating a list of the centers of detected dots [Figs. 3(b) and 3(f) and Fig. 4(a)], these dots need to be matched to the pre-determined printed dots or pre-determined projected dots [Fig. 4(b)]. Both matching procedures are the same, although the printed dots have two distinct pre-determined patterns, one for each side of the calibration target, so only one pattern should match. Before we can match detected dots with pre-determined dots, the pose of the elliptical detected dots needs to be corrected. This removes dot pattern skew, due to camera perspective and calibration target orientation. Skew can be corrected by rotating the positions of the dots by the median orientation angle, and then scaling the positions in the x and y axis by the median major and minor axis. These newly calculated positions approximate the original pattern [Figs. 3(d) and 3(h) and Fig. 4(a)].

In order to locally match the detected dot pattern to pre-determined printed or projected dots, each detected and pre-determined dot and its immediate surroundings is normalized [Figs. 4(c) and 4(d)]. The individual dot that has to be matched is placed at position (0,0), its nearest neighbor is placed at (1,0), and the position of its other nearest neighbor dots scale based on these two positions. For the detected dots, the seven closest dots are normalized in this fashion, and for the pre-determined printed or projected dots, the five closest pre-determined dots are normalized. By chance, it can happen that there are multiple nearest neighbors to be placed at (1,0), so we perform an equivalent normalization for those specific neighbors.

The next step of the matching algorithm is based on error minimization of the small groups of normalized detected dots compared with the small groups of normalized pre-determined printed or projected dots. When matching the small groups of normalized dots, points (0,0) and (1,0) will always match by design. The goodness of local pattern overlap is therefore determined by

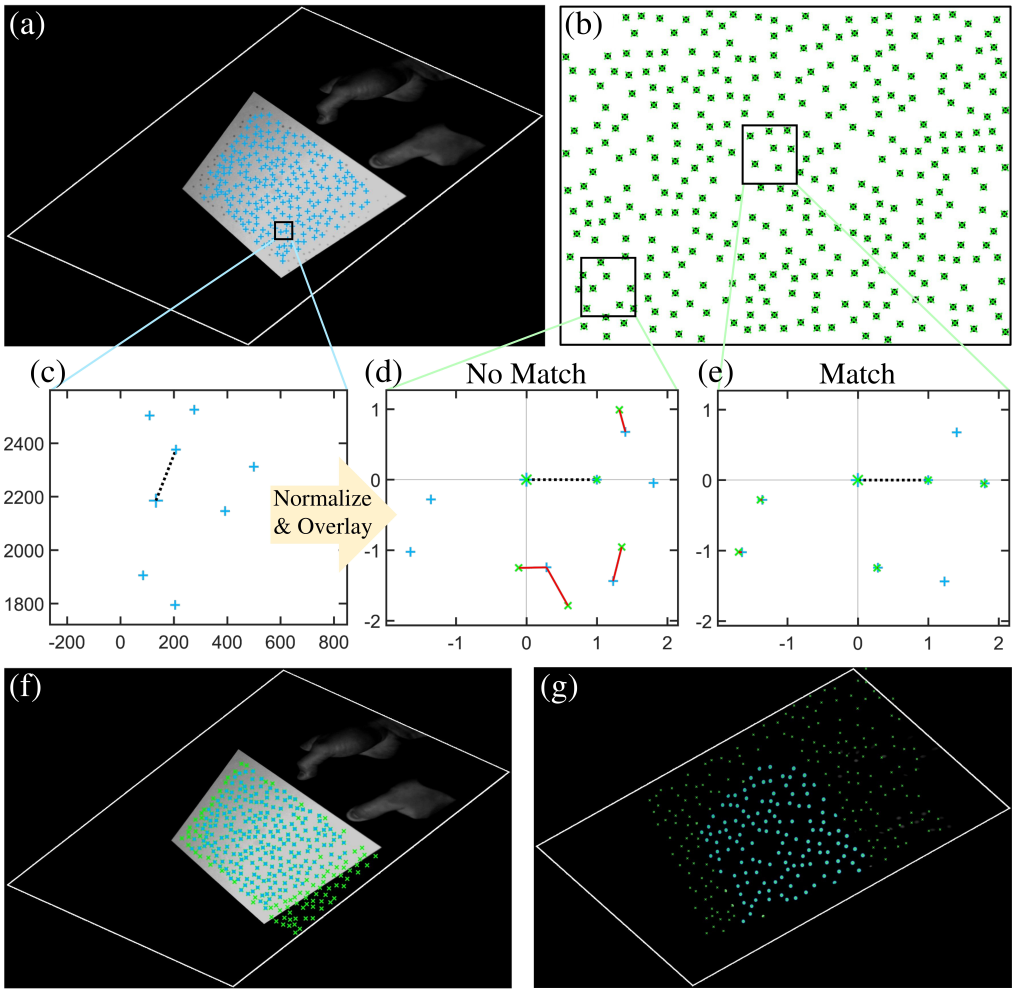

Fig. 4. Automated matching of detected dots identified using image processing (blue) with pre-determined projected or printed dots (green). a) Detected dots overlaid on a skew-corrected image captured by the high-speed camera. The skewed pose of the elliptical detected dots has been corrected so that the spacing matches the original spacing when the dots were circles. b) Pre-determined printed dots. c) Detected dots are matched with pre-determined printed dots using the same procedure as they are matched with pre-determined projected dots. A sample detected dot is selected along with its seven nearest neighbor dots. The dot is connected to its nearest neighbor with a dashed black line. d) All eight dots from (c) are then normalized according to the transformations needed to move the dot at the center to (0,0) and its nearest neighbor to (1,0). The same is done for a random pre-determined printed dot and its five nearest neighbor dots. In this case, the overlay of the two sets of dots shows a bad match, because the red lines that connect pairs of closest overlapping dots are long. This indicates high error for this particular match between two sets of dots. e) The best match of the set of dots in (c) with a subset of the dots from (b) occurs when the red error lines are barely visible. The matched dots in (e) are used to iteratively propagate matches across the list of all detected dots identified using image processing. f,g) The outcome of the propagated matching process is that the pre-determined dot positions (green) can be overlaid on top of the detected dot positions (blue) for both the printed dots (f) and projected dots (g). For visualization purposes, the green dots displayed here are shown to the full extent of the printed or projected area, even if a matching dot was not detected with image processing.

the four next-closest paired detected and pre-determined dots for a total overlap of six paired dots. The error for each pair of dots (detected vs. pre-determined) is visualized as red lines shown in Fig. 4(d). The total matching error equals the sum of squares of paired line distances.

After minimization, iterative planar homography mapping is used to match all detected dots with pre-determined dots. The six matching dots with the lowest error [Fig. 4(e)] are used to compute a planar homography **H**, between detected dots **P** and pre-determined printed or projected dots $\tilde{\mathbf{P}}$. **H** is found by minimizing Eq. (2)

$$\min_{\mathbf{H}} \left\| \mathbf{P} - \mathbf{H} \cdot \tilde{\mathbf{P}} \right\|_2, \tag{2}$$

where **H** is a 3×3 matrix, $\tilde{\mathbf{P}}$ and **P** are $3 \times n$ 2D homogeneous coordinates, and n is the number of dots. The double vertical bars indicate the use of a Euclidean norm. As Eq. (3) shows

$$\mathbf{KX} = \begin{bmatrix} a_1 & a_2 & \cdots & a_n \\ b_1 & b_2 & \cdots & b_n \\ c_1 & c_2 & \cdots & c_n \end{bmatrix} = \begin{bmatrix} a_1/c_1 & a_2/c_2 & \cdots & a_n/c_n \\ b_1/c_1 & b_2/c_2 & \cdots & b_n/c_n \\ 1 & 1 & \cdots & 1 \end{bmatrix} \begin{bmatrix} c_1 & & & \\ & c_2 & & \\ & & \ddots & \\ & & & c_n \end{bmatrix} = \mathbf{P}\lambda, \tag{3}$$

the homogeneous coordinate constraint requires that the first and second rows are normalized by the third row so that the entire third row equals one. Any subsequent equations follow the same notation where **X** indicates 3D coordinates and **P** indicates 2D homogeneous pixel coordinates normalized by a corresponding $n \times n$ diagonal matrix labeled λ. The intrinsic calibration matrix, **K**, is used to convert between 3D coordinates and 2D pixel coordinates. In general, the value of λ is not used beyond fulfilling the homogenous constraint. Matching dots are iteratively added to the six initial matches if their error is sufficiently low, and dots only cease to be added when no changes are detected from the previous iteration. In this way, all detected dots that can be matched to pre-determined printed or projected dots are matched. We require a minimum of 25 matched dots in order to use the data from a captured frame in future calibration steps [Figs. 4(f) and 4(g)].

2.5. Initialization of calibration parameters using device pairs

The intrinsic and extrinsic calibration parameters of each camera and projector are initialized based on the matched dots, using toolboxes from a calibration toolbox for MATLAB [19]. This approximate initialization step first calibrates the intrinsic parameters of each individual camera and projector separately, before the extrinsic parameters are calculated for camera-projector and camera-camera pairs.

First, the intrinsic camera parameters are calibrated based on the printed dots identified on the calibration target. In general, calibration amounts to an error minimization problem between detected and pre-determined dots with parameters defined by the calibration model. In the case of intrinsic camera calibration, the error function is defined by Eq. (4)

$$\min_{{}^{c_i}\mathbf{R}^{b_{k:i}},\, {}^{c_i}\mathbf{T}^{b_{k:i}}_{c_i},\, \mathbf{K}_{c_i}} \sum_{k=1}^{K} \left\| {}_{c_i b_k}\tilde{\mathbf{P}}^{\bigcirc}_{c_i} - {}_{c_i b_k}\mathbf{P}^{\bigcirc}_{c_i} \right\|_2, \tag{4}$$

which is the squared sum of pixel error between the detected dots, ${}_{c_i b_k}\tilde{\mathbf{P}}^{\bigcirc}_{c_i}$, and the pre-determined printed dots reprojected on the camera image, ${}_{c_i b_k}\mathbf{P}^{\bigcirc}_{c_i}$. The tilde ($\sim$) is used to distinguish between detected and pre-determined dots, while the symbols $\bigcirc$ and $\otimes$ are used distinguish between printed and projected dots respectively (please note we do not use these symbols as mathematical operators in this work). The index $\mathrm{c_i}$ is the ith camera, $\mathrm{p_i}$ is the ith projector, and $\mathrm{b_{k:i}}$ is the kth of K calibration target (or board) positions seen by the ith camera. The left subscript is used to indicate the context of the dots, which camera sees the dots and on which calibration target position, and the right subscript is the reference frame. All other notation follows in the same manner and is additionally listed in Table 4. The error function in Eq. (4) can be minimized by optimizing the values of ${}^{\mathrm{c_i}}\mathbf{R}^{\mathrm{b_{k:i}}}$, ${}^{\mathrm{c_i}}\mathbf{T}^{\mathrm{b_{k:i}}}_{\mathrm{c_i}}$, and $\mathbf{K}_{\mathrm{c_i}}$. The 3D rotation matrix, ${}^{\mathrm{c_i}}\mathbf{R}^{\mathrm{b_{k:i}}}$, is from reference frame $\mathrm{b_{k:i}}$ to $\mathrm{c_i}$, the 3D translation vector, ${}^{\mathrm{c_i}}\mathbf{T}^{\mathrm{b_{k:i}}}_{\mathrm{c_i}}$, is from reference frame $\mathrm{b_{k:i}}$ to $\mathrm{c_i}$ written in the $\mathrm{c_i}$ reference frame, and $\mathbf{K}_{\mathrm{c_i}}$ is the intrinsic calibration matrix for the ith camera. $\mathbf{K}$ is defined generally in Eq. (5) for either cameras or projectors

$$\mathbf{K} = \begin{bmatrix} \alpha & 0 & u_0 \\ 0 & \beta & v_0 \\ 0 & 0 & 1 \end{bmatrix}, \tag{5}$$

where constants α and β are proportional to the focal length, and (u_0, v_0) is principal point in pixels. Whereas the detected dots, ${}_{\mathrm{c_i b_k}}\tilde{\mathbf{P}}^{\bigcirc}_{\mathrm{c_i}}$, in Eq. (4) are already known, we need to compute the pre-determined printed dots reprojected on the camera image, ${}_{\mathrm{c_i b_k}}\mathbf{P}^{\bigcirc}_{\mathrm{c_i}}$. This is done by using Eq. (6)

$${}_{\mathrm{c_i b_k}}\mathbf{P}^{\bigcirc}_{\mathrm{c_i}}\lambda = \mathbf{K}_{\mathrm{c_i}} \cdot {}_{\mathrm{c_i b_k}}\mathbf{X}^{\bigcirc}_{\mathrm{c_i}} = \mathbf{K}_{\mathrm{c_i}}\left({}^{\mathrm{c_i}}\mathbf{R}^{\mathrm{b_{k:i}}} \cdot {}_{\mathrm{c_i b_k}}\mathbf{X}^{\bigcirc}_{\mathrm{b_{k:i}}} + {}^{\mathrm{c_i}}\mathbf{T}^{\mathrm{b_{k:i}}}_{\mathrm{c_i}} \cdot \mathbf{J}_{1,\mathrm{n}}\right), \tag{6}$$

where $\mathbf{X}$ are the 3D coordinates. Further, λ is used as in Eq. (3), to satisfy the constraint for 2D homogeneous coordinates $\mathbf{P}$, and $\mathbf{J}_{1,\mathrm{n}}$ is a matrix of all ones with size $1 \times n$. Here we use the dot $(\cdot)$ throughout this paper to denote multiplication (not the dot product). By minimizing the error function in Eq. (4), ${}^{\mathrm{c_i}}\mathbf{R}^{\mathrm{b_{k:i}}}$, ${}^{\mathrm{c_i}}\mathbf{T}^{\mathrm{b_{k:i}}}_{\mathrm{c_i}}$, and $\mathbf{K}_{\mathrm{c_i}}$ can be computed for each camera by using a calibration toolbox for MATLAB [19].

Next, the intrinsic calibration for each projector is computed based on the projected dots identified on the calibration target. The error function that characterizes the projector calibrations is defined in Eq. (7)

$$\min_{\mathbf{K}_{\mathrm{p_i}},{}^{\mathrm{p_i}}\mathbf{R}^{\mathrm{b_{k:i}}},{}^{\mathrm{p_i}}\mathbf{T}^{\mathrm{b_{k:i}}}} \sum_{k=1}^{K} \left\| {}_{\mathrm{c_i p_i b_k}}\tilde{\mathbf{P}}^{\otimes}_{\mathrm{p_i}} - {}_{\mathrm{c_i p_i b_k}}\mathbf{P}^{\otimes}_{\mathrm{p_i}} \right\|_2, \tag{7}$$

which is similar to the error function in Eq. (4) but written in projector pixel coordinates. While we already have pre-determined projected dots ${}_{\mathrm{c_i p_i b_k}}\mathbf{P}^{\otimes}_{\mathrm{p_i}}$ in these coordinates, we need to transform the detected dots from camera pixels ${}_{\mathrm{c_i p_i b_k}}\tilde{\mathbf{P}}^{\otimes}_{\mathrm{c_i}}$ to projector pixels ${}_{\mathrm{c_i p_i b_k}}\tilde{\mathbf{P}}^{\otimes}_{\mathrm{p_i}}$. As a middle step, we first find the 3D locations of detected dots ${}_{\mathrm{c_i p_i b_k}}\tilde{\mathbf{X}}^{\otimes}_{\mathrm{b_{k:i}}}$ in each paired camera reference frame. These are based on the known calibration target locations defined by ${}^{\mathrm{c_i}}\mathbf{R}^{\mathrm{b_{k:i}}}$ and ${}^{\mathrm{c_i}}\mathbf{T}^{\mathrm{b_{k:i}}}_{\mathrm{c_i}}$, which we computed in the previous camera calibration initialization. We can start by rewriting Eq. (6) as Eq. (8)

$${}_{\mathrm{c_i p_i b_k}}\tilde{\mathbf{P}}^{\otimes}_{\mathrm{c_i}}\lambda = \mathbf{K}_{\mathrm{c_i}} \cdot {}_{\mathrm{c_i p_i b_k}}\tilde{\mathbf{X}}^{\otimes}_{\mathrm{c_i}} = \mathbf{K}_{\mathrm{c_i}}\left({}^{\mathrm{c_i}}\mathbf{R}^{\mathrm{b_{k:i}}} \cdot {}_{\mathrm{c_i p_i b_k}}\tilde{\mathbf{X}}^{\otimes}_{\mathrm{b_{ki}}} + {}^{\mathrm{c_i}}\mathbf{T}^{\mathrm{b_{k:i}}}_{\mathrm{c_i}} \cdot \mathbf{J}_{1,\mathrm{n}}\right), \tag{8}$$

which uses projected dots instead of printed dots. The third row of ${}_{\mathrm{c_i p_i b_k}}\tilde{\mathbf{X}}^{\otimes}_{\mathrm{b_{k:i}}}[3,:] = 0$, because all of these points lie on the calibration target, so we can simplify this expression to

$${}_{\mathrm{c_i p_i b_k}}\tilde{\mathbf{P}}^{\otimes}_{\mathrm{c_i}}\lambda = \mathbf{K}_{\mathrm{c_i}} \begin{bmatrix} {}^{\mathrm{c_i}}\mathbf{R}^{\mathrm{b_{k:i}}}[:,1] & {}^{\mathrm{c_i}}\mathbf{R}^{\mathrm{b_{k:i}}}[:,2] & {}^{\mathrm{c_i}}\mathbf{T}^{\mathrm{b_{k:i}}}_{\mathrm{c_i}} \end{bmatrix} \begin{bmatrix} {}_{\mathrm{c_i p_i b_k}}\tilde{\mathbf{X}}^{\otimes}_{\mathrm{b_{k:i}}}[1,:] \\ {}_{\mathrm{c_i p_i b_k}}\tilde{\mathbf{X}}^{\otimes}_{\mathrm{b_{k:i}}}[2,:] \\ 1 \end{bmatrix}, \tag{9}$$

where the first and second columns of ${}^{\mathrm{c_i}}\mathbf{R}^{\mathrm{b_{k:i}}}$ are ${}^{\mathrm{c_i}}\mathbf{R}^{\mathrm{b_{k:i}}}[:,1]$ and ${}^{\mathrm{c_i}}\mathbf{R}^{\mathrm{b_{k:i}}}[:,2]$ respectively. This enables us to solve for the 3D coordinates on the calibration target, ${}_{\mathrm{c_ip_ib_k}}\tilde{\mathbf{X}}^{\otimes}_{\mathrm{b_{k:i}}}$, in Eq. (10)

$$\begin{bmatrix} {}_{\mathrm{c_ip_ib_k}}\tilde{\mathbf{X}}^{\otimes}_{\mathrm{b_{k:i}}}[1] \\ {}_{\mathrm{c_ip_ib_k}}\tilde{\mathbf{X}}^{\otimes}_{\mathrm{b_{k:i}}}[2] \\ 1 \end{bmatrix} = \left(\mathbf{K}_{\mathrm{c_i}} \begin{bmatrix} {}^{\mathrm{c_i}}\mathbf{R}^{\mathrm{b_{k:i}}}_1 & {}^{\mathrm{c_i}}\mathbf{R}^{\mathrm{b_{k:i}}}_2 & {}^{\mathrm{c_i}}\mathbf{T}^{\mathrm{b_{k:i}}}_{\mathrm{c_i}} \end{bmatrix}\right)^{-1} \left({}_{\mathrm{c_ip_ib_k}}\tilde{\mathbf{P}}^{\otimes}_{\mathrm{c_i}}\lambda\right). \tag{10}$$

The transformation from the calibration target to the projector is similar to Eq. (6) and is written in Eq. (11)

$${}_{\mathrm{c_ip_ib_k}}\tilde{\mathbf{P}}^{\otimes}_{\mathrm{p_i}}\lambda = \mathbf{K}_{\mathrm{p_i}} \cdot {}_{\mathrm{c_ip_ib_k}}\tilde{\mathbf{X}}^{\otimes}_{\mathrm{p_i}} = \mathbf{K}_{\mathrm{p_i}} \left({}^{\mathrm{p_i}}\mathbf{R}^{\mathrm{b_{k:i}}} \cdot {}_{\mathrm{c_ip_ib_k}}\tilde{\mathbf{X}}^{\otimes}_{\mathrm{b_{k:i}}} + {}^{\mathrm{p_i}}\mathbf{T}^{\mathrm{b_{k:i}}}_{\mathrm{p_i}} \cdot \mathbf{J}_{1,\mathrm{n}}\right). \tag{11}$$

Now that the projector pixels ${}_{\mathrm{c_ip_ib_k}}\tilde{\mathbf{P}}^{\otimes}_{\mathrm{p_i}}$ have been computed, we can minimize the error function in Eq. (7) to compute $\mathbf{K}_{\mathrm{p_i}}$ for each projector by using a calibration toolbox for MATLAB [19].

Lastly, the extrinsic parameters between camera-projector pairs and between neighboring cameras are computed. For camera-projector pairs, the rotation matrices ${}^{\mathrm{c_i}}\mathbf{R}^{\mathrm{p_i}}$ and translation vectors ${}^{\mathrm{c_i}}\mathbf{T}^{\mathrm{p_i}}_{\mathrm{c_i}}$ are computed by minimizing the error function in Eq. (12)

$$\min_{\mathbf{K}_{\mathrm{c_i}},\mathbf{K}_{\mathrm{p_i}},{}^{\mathrm{c_i}}\mathbf{R}^{\mathrm{p_i}},{}^{\mathrm{c_i}}\mathbf{T}^{\mathrm{p_i}}_{\mathrm{c_i}}} \sum_{k=1}^{K} \left\| {}_{\mathrm{c_ip_ib_k}}\tilde{\mathbf{P}}^{\otimes}_{\mathrm{p_i}} - {}_{\mathrm{c_ip_ib_k}}\mathbf{P}^{\otimes}_{\mathrm{p_i}} \right\|_2, \tag{12}$$

which uses projected dot data. This calibration is similar to the intrinsic projector calibration in Eq. (7), but directly relates the camera and projector reference frames. The calibration reference frame is eliminated from the optimization by the relation in Eqs. (13) and (14)

$${}^{\mathrm{c_i}}\mathbf{R}^{\mathrm{p_i}} = {}^{\mathrm{c_i}}\mathbf{R}^{\mathrm{b_{k:i}}} \cdot \left({}^{\mathrm{p_i}}\mathbf{R}^{\mathrm{b_{k:i}}}\right)^{\mathrm{T}}, \tag{13}$$

$${}^{\mathrm{c_i}}\mathbf{T}^{\mathrm{p_i}}_{\mathrm{c_i}} = {}^{\mathrm{c_i}}\mathbf{T}^{\mathrm{b_{k:i}}}_{\mathrm{c_i}} - {}^{\mathrm{c_i}}\mathbf{R}^{\mathrm{p_i}} \cdot {}^{\mathrm{p_i}}\mathbf{T}^{\mathrm{b_{k:i}}}_{\mathrm{p_i}}, \tag{14}$$

where $^{\mathrm{T}}$ indicates the transpose of a matrix, which is equivalent to the inverse for a rotation matrix. For neighboring cameras, the rotation matrices ${}^{\mathrm{c_i}}\mathbf{R}^{\mathrm{c_j}}$ and translation vectors ${}^{\mathrm{c_i}}\mathbf{T}^{\mathrm{c_j}}_{\mathrm{c_i}}$ are computed by minimizing the error function in Eq. (15)

$$\min_{{}^{\mathrm{c_i}}\mathbf{R}^{\mathrm{c_j}},{}^{\mathrm{c_i}}\mathbf{T}^{\mathrm{c_j}}_{\mathrm{c_i}}} \sum_{k=1}^{K} \left\| {}_{\mathrm{c_ib_k}}\tilde{\mathbf{P}}^{\bigcirc}_{\mathrm{c_i}} - {}_{\mathrm{c_jb_k}}\tilde{\mathbf{P}}^{\bigcirc}_{\mathrm{c_i}} \right\|_2. \tag{15}$$

This uses the printed dot data transformed to one of the two camera reference frames. Similarly, to the camera-projector pairs, Eqs. (16) and (17) are used to relate calibration target positions to different camera positions

$${}^{\mathrm{c_i}}\mathbf{R}^{\mathrm{c_j}} = {}^{\mathrm{c_i}}\mathbf{R}^{\mathrm{b_{k:i}}} \cdot \left({}^{\mathrm{c_j}}\mathbf{R}^{\mathrm{b_{k:i}}}\right)^{\mathrm{T}}, \tag{16}$$

$${}^{\mathrm{c_i}}\mathbf{T}^{\mathrm{c_j}}_{\mathrm{c_i}} = {}^{\mathrm{c_i}}\mathbf{T}^{\mathrm{b_{k:i}}}_{\mathrm{c_i}} - {}^{\mathrm{c_i}}\mathbf{R}^{\mathrm{c_j}} \cdot {}^{\mathrm{c_j}}\mathbf{T}^{\mathrm{b_{k:i}}}_{\mathrm{c_j}}. \tag{17}$$

For neighboring camera calibrations, frames where the same printed dots are seen in multiple cameras are logged. The $I-1$ camera pairs with the most such frames are used to compute rotations and translations between all cameras, where $I = 4$ is the number of cameras we used in our particular setup. The camera-projector and camera-camera extrinsic parameters are computed by minimizing the error functions defined in Eqs. (12) and (15) using the stereo calibration routine section of a calibration toolbox for MATLAB [19]. Because the toolbox can only compute extrinsic calibrations for two devices at a time, the errors between devices on opposite sides of the setup buildup. This calibration is only approximate, because the optimization and connectivity of the calibration of all cameras and projectors simultaneously, requires additional optimization after this initialization step.

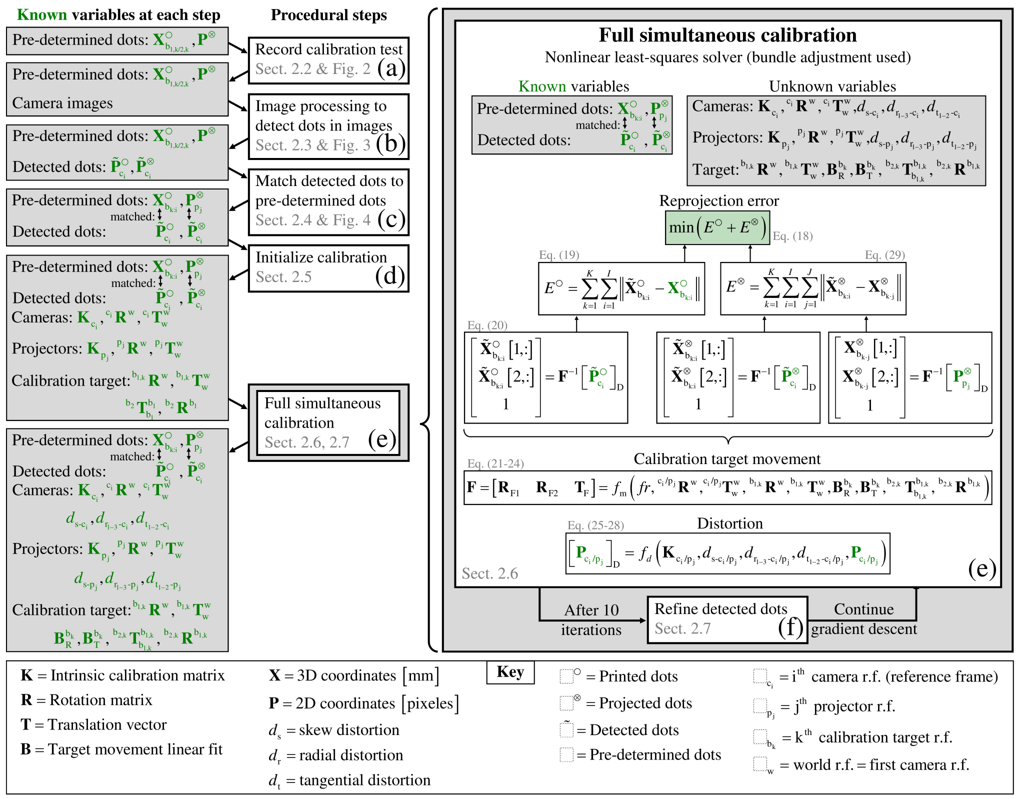

Fig. 5. Flow chart of the entire calibration method. At the beginning of the calibration process, the only known variables are the pre-determined printed and projected dot patterns. At the end, dots have been detected, matched, and all calibration parameters optimized. The variables that are known at each procedural calibration step are labeled in green throughout and summarized in the leftmost vertical set of boxes. The notation of select variables has been shortened in the figure, and the full notation can be found in the equations referenced in the figure. a-e) Each procedural calibration step is shown in the next column of boxes with greater detail shown in Figs. 2-4 for the first three steps. e) The large box outlined in gray on the right side of this figure is a zoomed-in view of the full simultaneous calibration algorithm detailed in Section 2.6 of the text. Within this optimization, the calibration parameters are computed by minimizing the reprojection error of the 3D locations of the pre-determined dots compared to the detected dots. f) Part of the way through the full optimization (e), the positions of the detected dots are refined as detailed in Section 2.7 of the text. The key at the bottom is a quick reference with further detail given in the text and summarized in Table 4. The meanings of superscripts and subscripts, which can be applied to multiple variables, are generalized by replacing the main variable with a dashed box.

2.6. Calibration optimization of all devices simultaneously

After the intrinsic and extrinsic calibration parameters are initialized for all the cameras and projectors, a bundle-adjusted nonlinear least-squares solver is used to fine-tune the calibration. During this fine-tuning process, accuracy is increased by solving the calibration error minimization problem with all dot data simultaneously. The accuracy is further increased by including parameters that account for the movement of the calibration target in between frames and the distortion of the sensor and lens geometry of the cameras and projectors. Additionally, the accuracy is further increased by refining the center positions of detected dots using the almost-complete calibration itself (procedure in Section 2.7). The calibration optimization algorithm is visualized in Fig. 5 and expanded upon below.

All of the calibration parameters (listed in Table 2) are optimized by minimizing reprojection error [Fig. 5(e)]. This error function is defined as

$$\min_{\text{All terms in Table2}} \left(E^{\bigcirc} + E^{\otimes}\right), \tag{18}$$

and consists of the error in millimeters between pre-determined dots and detected dots reprojected onto the calibration target. For this minimization problem, we use the trust-region-reflective approach [53] as applied in the lsqnonlin function in MATLAB. The reprojection error has two components: printed dot error, $E^{\bigcirc}$, and projected dot error, $E^{\otimes}$. To compute the contribution to the error by the printed dots, $E^{\bigcirc}$, the pre-determined printed dot positions, ${}_{\mathrm{c_i b_k}}\mathbf{X}^{\bigcirc}_{\mathrm{b_{1,k}}}$, are compared to the detected dot positions

$$E^{\bigcirc} = \sum_{i=1}^{I}\sum_{k=1}^{K}\left\| {}_{\mathrm{c_i b_k}}\tilde{\mathbf{X}}^{\bigcirc}_{\mathrm{b_{1,k}}} - {}_{\mathrm{c_i b_k}}\mathbf{X}^{\bigcirc}_{\mathrm{b_{1,k}}}\right\|_2. \tag{19}$$

The detected dots reprojected onto the calibration target, on calibration target side number one are labeled ${}_{\mathrm{c_i b_k}}\tilde{\mathbf{X}}^{\bigcirc}_{\mathrm{b_{1,k}}}$.

In order to compute the total error of the printed dots in Eq. (19), we need to solve for the detected dot positions reprojected onto the calibration target, ${}_{\mathrm{c_i b_k}}\tilde{\mathbf{X}}^{\bigcirc}_{\mathrm{b_{1,k}}}$, by using the detected dot positions in pixels, ${}_{\mathrm{c_i b_k}}\tilde{\mathbf{P}}^{\bigcirc}_{\mathrm{c_i}}$. We do so similarly to the transformation made in the initialization step in Eq. (10), but also including distortion and movement correction into the solution

$$\begin{bmatrix} {}_{\mathrm{c_i b_k}}\tilde{\mathbf{X}}^{\bigcirc}_{\mathrm{b_{1,k}}}[1,:] \\ {}_{\mathrm{c_i b_k}}\tilde{\mathbf{X}}^{\bigcirc}_{\mathrm{b_{1,k}}}[2,:] \\ 1 \end{bmatrix} = \begin{bmatrix} \mathbf{R}_{\mathrm{F}}[:,1] & \mathbf{R}_{\mathrm{F}}[:,2] & \mathbf{T}_{\mathrm{F}} \end{bmatrix}^{-1} \left[{}_{\mathrm{c_i b_k}}\tilde{\mathbf{P}}^{\bigcirc}_{\mathrm{c_i}}\right]_{\mathrm{D}}. \tag{20}$$

The variables $\mathbf{R}_{\mathrm{F}}$ and $\mathbf{T}_{\mathrm{F}}$ are linear fits to correct for the movement of the calibration target, detailed in Eqs. (21)-(24), and $\left[{}_{\mathrm{c_i b_k}}\tilde{\mathbf{P}}^{\bigcirc}_{\mathrm{c_i}}\right]_{\mathrm{D}}$ is ${}_{\mathrm{c_i b_k}}\tilde{\mathbf{P}}^{\bigcirc}_{\mathrm{c_i}}$ adjusted for device distortion and normalized by $\mathbf{K}_{\mathrm{c_i}}$, detailed in Eqs. (25)-(28).

First, we define the linear fits to correct for the small rotational and translational movements of the calibration target in the calibration space. $\mathbf{R}_{\mathrm{F}}$ and $\mathbf{T}_{\mathrm{F}}$ and are defined in Eqs. (21) and (22) as follows

$$\mathbf{R}_{\mathrm{F}} = {}^{\mathrm{c_i}}\mathbf{R}^{\mathrm{w}} \cdot \mathrm{rod}^{-1}\left(\mathrm{rod}\left({}^{\mathrm{w}}\mathbf{R}^{\mathrm{b_{1,k}}}\right) + \mathbf{B}^{\mathrm{b_k}}_{\mathrm{R}}\left(fr(i) - \frac{F+1}{2}\right)\right)\mathbf{R}^{*}, \tag{21}$$

$$\mathbf{T}_{\mathrm{F}} = {}^{\mathrm{c_i}}\mathbf{R}^{\mathrm{w}} \cdot \left({}^{\mathrm{w}}\mathbf{T}^{\mathrm{b_{1,k}}}_{\mathrm{w}} + \mathbf{B}^{\mathrm{b_k}}_{\mathrm{T}}\left(fr(i) - \frac{F+1}{2}\right) - {}^{\mathrm{w}}\mathbf{T}^{\mathrm{c_i}}_{\mathrm{w}} - \mathbf{T}^{*}\right), \tag{22}$$

where the Rodrigues transformation [54] function is denoted as rod( ) and transforms a 3×3 matrix into a 3×1 vector. The inverse function is denoted as rod^{-1} (). The value of $fr(i)$ is the frame number out of $F = 6$ total frames per sequence of frames, as shown in Fig. 2(c) for our particular setup. ${}^{\mathrm{w}}\mathbf{R}^{\mathrm{b_{1,k}}}$ and ${}^{\mathrm{w}}\mathbf{T}^{\mathrm{b_{1,k}}}_{\mathrm{w}}$ are the average rotational and translational position of the calibration target in the k^th sequence of frames. Finally, $\mathbf{B}^{\mathrm{b_k}}_{\mathrm{R}}$ and $\mathbf{B}^{\mathrm{b_k}}_{\mathrm{T}}$ are the linear fitting parameters to account for movement between frames. A linear fit is the best first order approximation of movement for rotation, because the Rodrigues function is linear in the limit of small angular displacements. The world reference frame, denoted with index w, is equivalent to the reference frame of the first camera. We define the rigid transformations between both sides of the calibration target as $\mathbf{R}^{*}$ and $\mathbf{T}^{*}$, which completes Eqs. (21) and (22). We can do this because the printed dot patterns on both side of the calibration target do not move relative to each other.

Table 2. Summary of variables that are optimized during simultaneous calibration of all cameras and projectors. These variables are optimized by minimizing the reprojection error function in Eq. (18) using a nonlinear least-squares solver. The number of variables that are optimized for our particular setup are listed for each category of variables (cameras, projectors, and calibration target positions). The * indicates that the rotation and translation of the first camera are set to the identity matrix and origin respectively by design. Including more optimization variables correlates with higher computational cost, so we see that the primary expense lies in solving for the calibration target positions (as exemplified for this particular calibration trial). While these calibration target positions are not the primary goal of the calibration, they are necessary building blocks to compute the primary parameters of interest: the camera and projector calibration variables.

		Parameters per item	Total parameters
Cameras $I = 4$ (setup specific)			
Intrinsic	$\mathbf{K}_{\mathrm{c_i}}$	4	16
Rotational*	${}^{\mathrm{c_i}}\mathbf{R}^{\mathrm{w}}$	3	9
Translational*	${}^{\mathrm{c_i}}\mathbf{T}^{\mathrm{w}}_{\mathrm{w}}$	3	9
Distortion $D = 6$	$d_{\mathrm{s-c_i}}, d_{\mathrm{r1-c_i}}, d_{\mathrm{r2-c_i}}, d_{\mathrm{r3-c_i}}, d_{\mathrm{t1-c_i}}, d_{\mathrm{t2-c_i}}$	6	24
Total camera parameters			**58**
Projectors $J = 4$ (setup specific)			
Intrinsic	$\mathbf{K}_{\mathrm{p_j}}$	4	16
Rotational	${}^{\mathrm{p_j}}\mathbf{R}^{\mathrm{w}}$	3	12
Translational	${}^{\mathrm{p_j}}\mathbf{T}^{\mathrm{w}}_{\mathrm{w}}$	3	12
Distortion $D = 6$	$d_{\mathrm{s-p_j}}, d_{\mathrm{r1-p_j}}, d_{\mathrm{r2-p_j}}, d_{\mathrm{r3-p_j}}, d_{\mathrm{t1-p_j}}, d_{\mathrm{t2-p_j}}$	6	24
Total projector parameters			**64**
Calibration Target Positions $K = 181$ (calibration specific)			
Rotational	${}^{\mathrm{b_{1,k}}}\mathbf{R}^{\mathrm{w}}$	3	543
Translational	${}^{\mathrm{b_{1,k}}}\mathbf{T}^{\mathrm{w}}_{\mathrm{w}}$	3	543
Rotational linear fit	$\mathbf{B}^{\mathrm{b_k}}_{\mathrm{R}}$	3	543
Translational linear fit	$\mathbf{B}^{\mathrm{b_k}}_{\mathrm{T}}$	3	543
Board side 1 to 2	${}^{\mathrm{b_{2,k}}}\mathbf{T}^{\mathrm{b_{1,k}}}_{\mathrm{b_{1,k}}}, {}^{\mathrm{b_{2,k}}}\mathbf{R}^{\mathrm{b_{1,k}}}(\theta)$		4
Total target parameters			**2176**
Total parameters			**2298**

*1st camera position and orientation pre-defined

In the case that the ith camera sees side one of the calibration target, these transformations are equal the identity matrix and zero respectively. When side two of the calibration target is visible, Eqs. (23) and (24) are used

$$\mathbf{R}^* = {}^{\mathrm{b_{2,k}}}\mathbf{R}^{\mathrm{b_{1,k}}}(\theta), \tag{23}$$

$$\mathbf{T}^* = {}^{\mathrm{w}}\mathbf{R}^{\mathrm{b_{1,k}}} \cdot {}^{\mathrm{b_{2,k}}}\mathbf{T}^{\mathrm{b_{1,k}}}_{\mathrm{b_{1,k}}}, \tag{24}$$

where ${}^{\mathrm{b_{2,k}}}\mathbf{R}^{\mathrm{b_{1,k}}}$ and ${}^{\mathrm{b_{2,k}}}\mathbf{T}^{\mathrm{b_{1,k}}}_{\mathrm{b_{1,k}}}$ are the rotation matrix and translation vector from side one to side two of the target. The rotation and translation does not change for different values of k. Since the two sides of the board are parallel to each other, ${}^{\mathrm{b_{2,k}}}\mathbf{R}^{\mathrm{b_{1,k}}}$ is only a function of a single angle θ.

Second, we define the device distortion of a pixel, $\left[{}_{\mathrm{c_ib_k}}\tilde{\mathbf{P}}^{\bigcirc}_{\mathrm{c_i}}\right]_{\mathrm{D}}$, as used in Eq. (20) to incorporate the calibration parameters that account for distortion of the sensor and lens geometry of the cameras and projectors. In order to simplify the notation, we define x and y as

$$\left[{}_{\mathrm{c_ib_k}}\tilde{\mathbf{P}}^{\bigcirc}_{\mathrm{c_i}}[:,m]\right]_{\mathrm{D}} = \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}, \tag{25}$$

for the m^{th} of n points in ${}_{\mathrm{c_ib_k}}\tilde{\mathbf{P}}^{\bigcirc}_{\mathrm{c_i}}$. The distortion equations are nonlinear, so we must solve for x and y by minimizing the following equation

$$\min_{x,y} \left\| {}_{\mathrm{c_ib_k}}\tilde{\mathbf{P}}^{\bigcirc}_{\mathrm{c_i}}[:,m]\,\lambda - \breve{\mathbf{K}}_{\mathrm{c_i}}\mathbf{D}(x,y) \right\|_2, \tag{26}$$

where $r^2 = x^2 + y^2$, $\mathbf{D}(x,y)$ is defined in Eq. (27), and $\breve{\mathbf{K}}$ is defined in Eq. (28)

$$\mathbf{D}(x,y) = \begin{bmatrix} \left(1 + d_{\mathrm{r1}}r^2 + d_{\mathrm{r2}}r^4 + d_{\mathrm{r3}}r^6\right)x + 2d_{\mathrm{t1}}xy + d_{\mathrm{t2}}\left(r^2 + 2x^2\right) \\ \left(1 + d_{\mathrm{r1}}r^2 + d_{\mathrm{r2}}r^4 + d_{\mathrm{r3}}r^6\right)y + 2d_{\mathrm{t2}}xy + d_{\mathrm{t1}}\left(r^2 + 2y^2\right) \\ 1 \end{bmatrix}, \tag{27}$$

$$\breve{\mathbf{K}} = \begin{bmatrix} \alpha & d_s\alpha & u_0 \\ 0 & \beta & v_0 \\ 0 & 0 & 1 \end{bmatrix}. \tag{28}$$

Here, d_s is the skew distortion coefficient, $d_{\mathrm{r1}}, d_{\mathrm{r2}}, d_{\mathrm{r3}}$ are the radial distortion coefficients, and $d_{\mathrm{t1}}, d_{\mathrm{t2}}$ are the tangential distortion coefficients [19]. For some cameras and projectors, the skew distortion is negligible and thus d_s can be set to zero. Often, d_{r3} or additionally d_{r2} may be set to zero to avoid overfitting the data. Now that the movement correction and distortion are defined, we can compute the values of ${}_{\mathrm{c_ib_k}}\tilde{\mathbf{X}}^{\bigcirc}_{\mathrm{b_{1,k}}}$ needed to compute the printed dot reprojection error, $E^{\bigcirc}$, in Eq. (19). However, the reprojection error of the projected dots, $E^{\otimes}$, remains to be computed.

The projected dot error, $E^{\otimes}$, can be computed similarly to the printed dot error, $E^{\bigcirc}$, both of which are needed to optimize all of the calibration parameters simultaneously. For this computation, both the detected projected dots, ${}_{\mathrm{c_ip_jb_k}}\tilde{\mathbf{P}}^{\otimes}_{\mathrm{c_i}}$, and the pre-determined projected dots, ${}_{\mathrm{c_ip_jb_k}}\mathbf{P}^{\otimes}_{\mathrm{p_j}}$, must be reprojected onto the calibration target as ${}_{\mathrm{c_ip_jb_k}}\tilde{\mathbf{X}}^{\otimes}_{\mathrm{b_{1,k}}}$ and ${}_{\mathrm{c_ip_jb_k}}\mathbf{X}^{\otimes}_{\mathrm{b_{1,k}}}$ respectively, and compared. The reprojection error is defined as

$$E^{\otimes} = \sum_{i=1}^{I}\sum_{j=1}^{J}\sum_{k=1}^{K} \left\| {}_{\mathrm{c_ip_jb_k}}\tilde{\mathbf{X}}^{\otimes}_{\mathrm{b_{1,k}}} - {}_{\mathrm{c_ip_jb_k}}\mathbf{X}^{\otimes}_{\mathrm{b_{1,k}}} \right\|_2, \tag{29}$$

where $J = 4$ is the number of projectors we used in our particular setup. Since the projector is modeled in the same way as the camera, Eqs. (20)-(28) are used in a similar manner to solve for ${}_{\mathrm{c_ip_jb_k}}\tilde{\mathbf{X}}^{\otimes}_{\mathrm{b_{1,k}}}$ and ${}_{\mathrm{c_ip_jb_k}}\mathbf{X}^{\otimes}_{\mathrm{b_{1,k}}}$. This differs from the above computation where the printed dots ${}_{\mathrm{c_ib_k}}\mathbf{X}^{\bigcirc}_{\mathrm{b_{1,k}}}$ are known and only ${}_{\mathrm{c_ib_k}}\tilde{\mathbf{X}}^{\bigcirc}_{\mathrm{b_{1,k}}}$ needs to be computed from ${}_{\mathrm{c_ib_k}}\tilde{\mathbf{P}}^{\bigcirc}_{\mathrm{c_i}}$ using Eqs. (20)-(28).

Finally, the total reprojection error in Eq. (18) can be evaluated and minimized while implementing bundle adjustment for increased speed. Bundle adjustment is a technique that uses the pattern of the dot data that is fed into the nonlinear solver to reduce the computational time required [55]. Most nonlinear solvers compute the Jacobian at each refinement step. Bundle adjustment eliminates the need to compute any Jacobian value that is known to equal zero based on the pattern of the data. The total number of variables which are optimized are summarized in Table 2. The variables in Table 2 vary based on the number of cameras, I, the number of projectors, J, and the number of calibration target positions, K. They additionally vary based on the type of fit used to account for target position motion in between frames, and the number of distortion parameters fit. Prior to bundle adjustment, the number of elements in the Jacobian that need to be computed equals the size of the Jacobian matrix, N_{Jacobian}. This is the number of optimized variables times the number of total dots, as Eq. (30) shows

$$N_{\text{Jacobian}} = 2\left((16I-6)+(16J)+(12K+4)\right)\left(N^{\bigcirc}+N^{\otimes}\right), \tag{30}$$

where $N^{\bigcirc}$ and $N^{\otimes}$ are the number of detected printed and projected dots respectively. After bundle adjustment, the number of elements that need to be computed, $N_{\text{Jacobian-adj}}$, drops significantly as Eq. (31) shows

$$N_{\text{Jacobian-adj}} = 2\left(16\left(N^{\bigcirc}+N^{\otimes}\right)-6N_{\text{c}_1}\right)+2\left(16N^{\otimes}\right)+2\left(12\left(N^{\bigcirc}+N^{\otimes}\right)+4N_{\text{b}_2}\right). \tag{31}$$

N_{c_1} is the number of dots that the first camera detects, and N_{b_2} is the number of dots detected on the second side of the calibration target. If many distinct calibration target positions are used, then the number of dots seen by each camera becomes similar, $N_{\text{c}_1} \approx I^{-1}\left(N^{\bigcirc}+N^{\otimes}\right)$ and $N_{\text{b}_2} \approx 0.5\left(N^{\bigcirc}+N^{\otimes}\right)$, enabling Eq. (31) to be simplified to Eq. (32)

$$N_{\text{Jacobian-adj}} \approx 2\left(30-\frac{6}{I}\right)\left(N^{\bigcirc}+N^{\otimes}\right)+2\left(16N^{\otimes}\right). \tag{32}$$

Thus, for our specific implementation and calibration trial where $N^{\bigcirc} = 78756$ and $N^{\otimes} = 66055$, the approximate increase in speed gained by computing the Jacobian using bundle adjustment is $N_{\text{Jacobian}}/N_{\text{Jacobian-adj}} = 64$ times faster. To further decrease the computational time, we use parallel processing to solve the total reprojection error in Eq. (18). We also use an analytical solution to calculate the Jacobian to account for distortion in Eqs. (25)-(28).

2.7. Dot center location calculation with subpixel accuracy

To increase the accuracy of the detected dot centers, a final refinement and filtering step is used part of the way through the final optimization (Section 2.6). In order to balance low computational time with high accuracy, this final refinement step is conducted after 10 iterations of the gradient decent in the final optimization. We chose 10 iterations after observing multiple different calibration trials. In all of the trials, the change in error was deemed sufficiently minimal before 10 iterations had been completed. After the refined dot centers are computed, the final optimization continues to convergence, as shown in Figs. 5(e) and 5(f). We define convergence as the point when the minimization function, Eq. (18), changes by less than 1e-06 mm between iterations.

The detection of dot centers needs to be refined for two reasons. First, the circles are skewed non-elliptically during the calibration. Thus, the subpixel edges of detected dots cannot be accurately approximated by an ellipse, as assumed in Section 2.3. Wu *et al.* [31] explained this effect by pointing out that when circles are projected and viewed through distorted lenses and sensors, they skew non-linearly. Second, the circle fitting accuracy is decreased by the sequential steps used to fit each ellipse. Ellipses were fit to the edges identified with the subpixel edge detector [52]. A better method is to fit each ellipse and its edges simultaneously, so that information is not lost in between steps, which we apply here.

We refine the computation of the centers of the detected dots by transforming data between reference frames and fitting dots in the reference frame where they are circular. This is achieved by using the nearly-complete calibration to transform the camera image pixel intensities of printed and projected detected dots into the calibration target and projector reference frames respectively

$$\left(u_{\mathrm{c_i}}, v_{\mathrm{c_i}}, \mathbf{I}_{\mathrm{c_i}}\left(u_{\mathrm{c_i}}, v_{\mathrm{c_i}}\right)\right) \rightarrow \left(a_{\mathrm{b_k/p_j}}, b_{\mathrm{b_k/p_j}}, \mathbf{I}_{\mathrm{c_i}}\left(u_{\mathrm{c_i}}, v_{\mathrm{c_i}}\right)\right). \tag{33}$$

The coordinates in camera pixels are u, v, the coordinates in either the calibration target or projector pixels are a, b, and the grayscale intensity of camera images is $\mathbf{I}_{\mathrm{c_i}}$. In this reference frame, a different edge detection technique can be used to more accurately fit each circle and its edges simultaneously. For projected dots, any pixels that are found to overlap printed dots are deleted, to ensure consistent reflective characteristics. The overlap is easy to identify since the approximate locations of all printed and projected dots have been identified, in contrast to the initial effort to do so in Section 2.3. After the detected dot pixels are transformed to either the calibration target or projector reference frame, a circular sigmoid function is used to fit the circular dot in Eq. (34)

$$\min_{a,b,R,c_1,c_2,c_3} \sum_{\text{Dot pixels}} \left\| \frac{c_1}{1+e^{c_3(r(a,b)-R)}} + c_2 - \mathbf{I}_{\mathrm{c_i}}\left(u_{\mathrm{c_i}}, v_{\mathrm{c_i}}\right) \right\|_2, \tag{34}$$

$$r(a,b) = \sqrt{\left(a_{\mathrm{b_{ki}/p_j}} - a\right)^2 + \left(b_{\mathrm{b_k/p_j}} - b\right)^2}. \tag{35}$$

This equation has fitting parameters a, b, R, c_1, c_2, c_3 where r is defined as a function of a and b in Eq. (35). A similar technique for robustly fitting edges of known shapes is used in [56]. The refined center of the detected dot is at a, b, the radius is R, the difference in brightness between the inside and outside of the dot is c_1, the brightness outside the dot is c_2, and c_3 scales with image blur.

The detected dot characteristics defined by these six fitting parameters can additionally be used as a final filtering step to eliminate blurry or noisy detected dots. In order to generalize the filters for any dot size and image intensity, limits are placed on three dimensionless numbers, which are defined using the fitting parameters. First, to reject noisy data points, we required the standard deviation of the error in Eq. (34), scaled by the contrast, c_1, to be less than 0.35. This was true for 98.5 % of printed dots and 97.3 % of projected dots for our particular setup and calibration. Second, to reject blurry data points, we required $c_3 \big/ R_{\mathrm{b/p}} > 2.5$, where $R_{\mathrm{b/p}}$ is the known radius of printed or projected dots. This was true for 99.8 % of printed dots and 99.1 % of projected dots. Lastly, to reject poorly fit data points, we required $0.8R_{\mathrm{b/p}} < R < 1.2R_{\mathrm{b/p}}$, which was true for 99.8 % of printed dots and 93.6 % of projected dots. In total, 98.2 % of printed dots and 89.6 % of projected dots remained. These remaining refined dot centers are then transformed back into the camera reference frame in pixels, where they are used to complete the final optimization in Section 2.6.

3. Results and discussion

We present a novel calibration technique to accurately and automatically calibrate multiple cameras and projectors simultaneously. Volumetric 3D surface reconstructions of deforming objects can be achieved at high-speed, by combining the calibration technique presented here, with the binary single-shot algorithm published in Deetjen *et al.* 2017 [14] for a single camera-projector pair. Here we present the essential hardware and algorithmic solutions needed to combine and calibrate multiple synchronized high-speed camera-projector pairs. During 3D

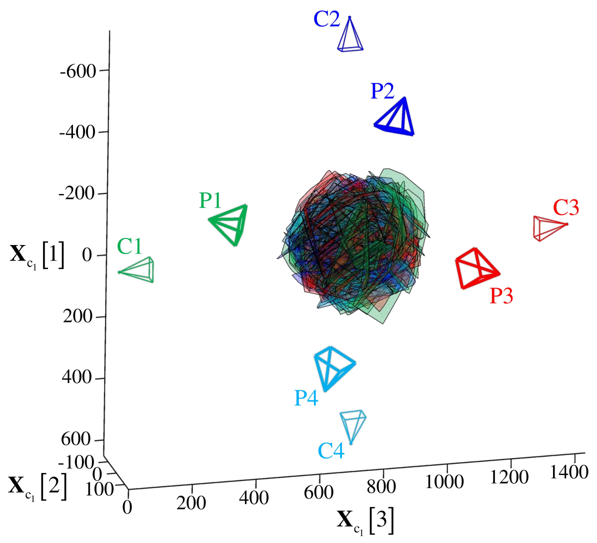

Fig. 6. Final calibration of four cameras and four projectors that are calibrated simultaneously [Fig. 1]. Three different colors are used for the four camera-projector pairs (the two different shades of blue used in this figure are solely used to graphically distinguish the pairs using the same blue projection color and filter). The 3D locations of the detected dots and the corresponding detected calibration target positions are displayed in the color corresponding to the camera in which the dot was detected. The coordinate system is relative to the first camera, subscript c1, and all units are in millimeters. The mean absolute value of the reprojection error is 0.094 mm, which is 0.023 % of the diameter of the calibration volume.

surface reconstruction [Fig. 1], the cross-talk between cameras and projectors is controlled by projecting using specific colors. These colors are matched by the camera color filters for each camera-projector pair. During calibration, each projector projects in each color sequentially [Fig. 2] to increase connectivity between cameras and projectors.

In order to easily and automatically calibrate all devices simultaneously, we designed a patterned 2D calibration target, a projection pattern, and corresponding image processing methods. Gray printed dots are printed in a pseudo-random pattern on both sides of the calibration target for camera calibration. The dots are gray rather than black to enable detection of projected dots overlapping with printed dots on the same target. The user moves the calibration target around the calibration volume, while all cameras record a quick burst of images of the illuminated target (separated by periodic intervals to reduce data). During a burst, each camera first records an image of the printed dot pattern illuminated by its paired projector. Next, the cameras record a dot pattern from each visible projector, by directing each projector to sequentially project dots in multiple colors [Fig. 2]. The dot patterns captured by the cameras [Fig. 5(a)] are processed with algorithms to find the dots on the target and ignore the rest [Fig. 3 and Fig. 5(b)]. Dot detection is aided by relatively sparse dot patterns and printing gray rather than black dots. Once the subsequent algorithm is used to match the detected dots with the pre-determined printed or projected dots [Fig. 4 and Fig. 5(c)], all cameras and projectors can be calibrated simultaneously [Fig. 5(e)].

To assess the importance of different steps in the calibration process, we compare the accuracy of the final calibration against the less complete calibration intermediate models. The calibration we used for our analysis is based on 181 calibration target positions, for which we automatically detected the printed and projected dots to compute the camera and projector calibration parameters [Fig. 6]. When we calibrate each device and the connections between them separately, as is done in the initialization step [Fig. 5(d)], the mean absolute value of the reprojection error for

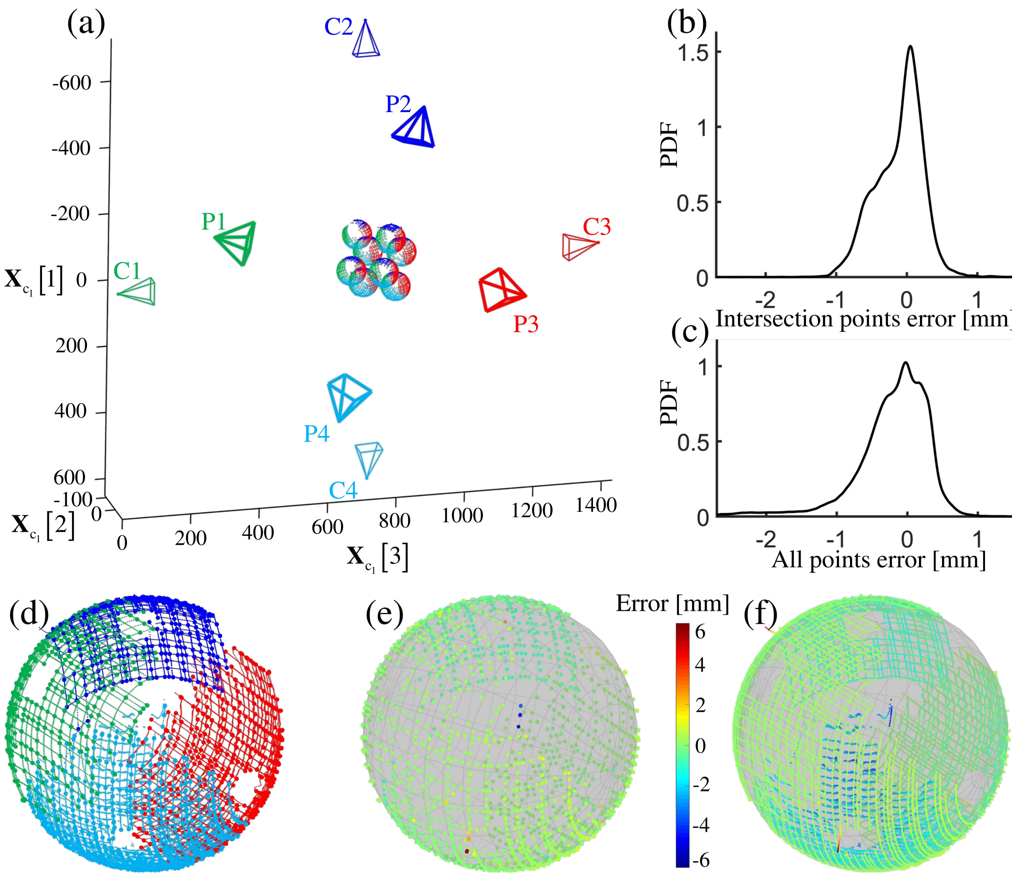

Fig. 7. We demonstrate the accuracy of our structured light system by automatically 3D reconstructing a sphere of known dimensions (structured light system: Fig. 1; calibration: Fig. 6). a) The 3D reconstructed surface of a sphere based on 8 separate images plotted on top of each other for 8 locations in the calibrated volume [Fig. 6]. b-c) The best accuracy is achieved at the stripe intersections, because they correspond to two known projection planes. The remaining points are reconstructed based on one projection plane, which reduces accuracy. The mean absolute value of the error of the 3D surface reconstruction is -0.11 ± 0.38 mm for intersection points (b) and -0.25 ± 0.79 mm for all of the points combined (c). d) Surface reconstruction of the sphere in position 2 (Table 3) based on 8 imaging frames. Filled circles indicate 3D reconstructed stripe intersections of the hash pattern. e) Error in the reconstruction of the sphere in (d) based on the more accurate stripe intersections. f) Error in the reconstruction for all pixel positions along each stipe.

the printed and projected dots together is 0.97 mm. This initialization step does not include parameters to correct for distortion or calibration target movement. When we add connectivity between all of the devices by calibrating everything simultaneously [Fig. 5(e)], the reprojection error drops to 0.22 mm. For the final calibration, we optimized across all parameters in Table 2, including parameters to correct for distortion and calibration target movement [Fig. 5(f)]. The mean absolute value of the reprojection error for this final calibration drops to 0.094 mm, which translates to an average of 0.17 pixels in the camera images. Normalized by the diameter of the calibration volume (406 mm; diameter of a sphere with equal volume to the convex geometry of all calibration points), the mean absolute value of the reprojection error is 0.023 % of the calibration volume.

We 3D reconstructed the volumetric surface of a well-defined gray sphere to evaluate the

Table 3. Accuracy of the 3D reconstruction of a sphere of known dimensions (diameter 82.55 ± 0.13 mm) averaged over 8 imaging frames for each of the eight locations in Fig. 7(a). Stripe intersection points are more accurate than the rest of the points, because they are reconstructed based on two projection planes rather than one. The accuracy is the mean error ± precision (s.d.). The absolute error is given in mm and the relative error as a percentage of the average calibration volume diameter 406 mm.

Position	Intersection points error		All points Error	
1	-0.19 ± 0.30 mm	-0.05 ± 0.07 %	-0.25 ± 0.35 mm	-0.06 ± 0.09 %
2	-0.00 ± 0.37 mm	-0.00 ± 0.09 %	-0.21 ± 0.65 mm	-0.05 ± 0.16 %
3	-0.31 ± 0.37 mm	-0.08 ± 0.09 %	-0.47 ± 0.62 mm	-0.12 ± 0.15 %
4	-0.06 ± 0.25 mm	-0.01 ± 0.06 %	-0.29 ± 1.23 mm	-0.07 ± 0.30 %
5	-0.13 ± 0.31 mm	-0.03 ± 0.08 %	-0.19 ± 0.38 mm	-0.05 ± 0.09 %
6	0.05 ± 0.46 mm	0.01 ± 0.11 %	-0.08 ± 0.72 mm	-0.02 ± 0.18 %
7	-0.19 ± 0.44 mm	-0.05 ± 0.11 %	-0.42 ± 0.81 mm	-0.10 ± 0.20 %
8	-0.05 ± 0.38 mm	-0.01 ± 0.09 %	-0.11 ± 1.29 mm	-0.03 ± 0.32 %
All	-0.11 ± 0.38 mm	-0.03 ± 0.09 %	-0.25 ± 0.79 mm	-0.06 ± 0.20 %

accuracy and precision of the calibration method. To determine the surfaces we use our binary single-shot high-speed 3D reconstruction algorithm [Fig. 7], which we previously developed [14]. The reconstruction is based on a locally-unique "hash" projection pattern of perpendicularly intersecting stripes. We use the calibration to compute and merge the 3D surface reconstructions for each camera-projector pair. To quantify how performance varies throughout the calibration volume, we placed the sphere in eight different locations and reconstructed it in each position. To compensate for the reduced reflectivity of the gray sphere, we lengthened the exposure time to 4.9 ms to image it in each position. We then manually removed the background of the sphere in each image for convenience (the background of the sphere was not optimized for automatic removal in long-exposure conditions).

To account for image processing limitations in the 3D reconstruction algorithm, "voxel carving" [57], "incorrect stripe matching", and "low triangulation angle" filters were added to the technique in [14]. The voxel carving filter uses the calibration, together with the multiple camera views, to remove some of the incorrectly reconstructed points. It does this by only retaining 3D points which lie on the object being reconstructed when reprojected into each 2D camera view. The incorrect stripe matching filter identifies stripes in the "hash" projection pattern which are incorrectly matched in the camera images. This is done by comparing the 3D error between the best and the next-best stripe matches between each camera image and the projected pattern. The best match should have an error that is significantly lower than the next-best match; if this is not the case, the stripe is removed. The low triangulation angle filter removes reconstructed points that are disproportionally sensitive to image processing and calibration errors. Triangulation angle is the angle between the projection plane produced by a stripe, and the camera vector produced by a pixel on that stripe. Using the triangulation angle and the calibration accuracy, we compute the predicted error of 3D reconstructed points for each pixel on each stripe. To compute predicted error for all these points, we calculate the predicted error at all intersections of perpendicular stripes, and then subsequently interpolate between these locations. The predicted error can be computed at stripe intersections, because they have the highest 3D reconstruction accuracy, thanks to the extra information that is available for reconstruction (two stripes rather

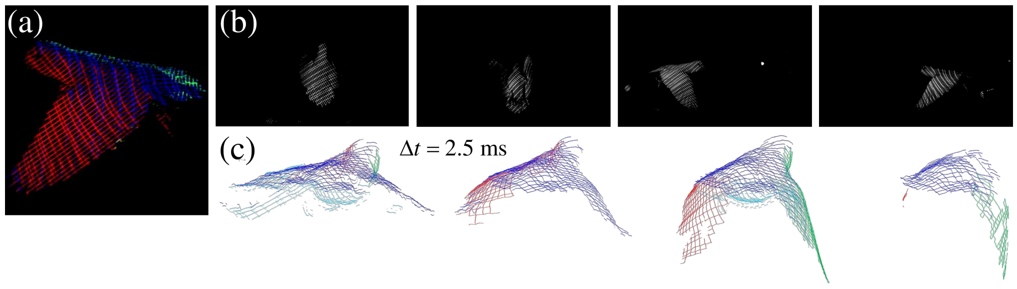

Fig. 8. Snapshot of a preliminary 3200 fps 3D-surface reconstruction of a Pacific Parrotlet during takeoff, flapping its morphing wings at 20 Hz. a) Color recording of the four projected color patterns on the bird (recorded with a 5th camera not shown in Fig. 1; Phantom LC-310; Vision Research, Wayne, NJ, USA). b) The images captured by the four cameras recording the structured light projected on the bird's surface (contrast enhanced). From left to right, the views show the camera looking down at the bird (blue light), looking up at the bird (blue light), looking at the right side of the bird (green light), and looking at the left side of the bird (red light). c) A preliminary 3D surface reconstruction of a downstroke of the Pacific Parrotlet in flight. The image processing artifacts and gaps seen in the reconstructions are similar to others found in this region of the downstroke. The bottom surface of the bird (light blue) is reconstructed from frames that are delayed by 1/3200 seconds from the other camera-projector pairs, so that the two blue camera-projector pairs do not overlap. The third reconstructed frame in (c) corresponds to all of the frames in (a) and (b). The first three frames in (c) correspond to 73 %, 87 %, and 100 % of the first downstroke after takeoff, and the final frame corresponds to 13 % of the next upstroke.

than the normal one stripe). Predicted error at stripe intersections is computed for both stripe directions, by comparing the 3D reconstruction using both stripes, with the 3D reconstruction using only one of the stripes. When implementing this low triangulation angle filter, we chose to remove points on stripes with a predicted error higher than 3.0 mm (spanning 0.74 % of the calibration volume diameter). This length-based cutoff was selected to eliminate obvious outliers, while closely maintaining the shape of the probability density function of the error shown in Fig. 7(c).

After filtering the outliers, the accuracy of the multi-view structured light system is quantified by 3D reconstructing a sphere in 8 locations of the calibration volume [Fig. 7]. The accuracy of the reconstructed data points varies depending on whether they lie at stripe intersections in the "hash" structured light pattern, or at other pixels along a stripe. Points at stripe intersections have two stripes rather than the normal one stripe, so the accuracy is higher. This difference in accuracy can be understood as follows. Each of the two stripes form a projection plane, while together the two projection planes form a projection vector. The 3D reconstruction of stripe intersection points is thus computed as the closest 3D point between the projection vector and the camera vector. This is the standard method for triangulation in non-high-speed structured light applications [5]. The remaining points on the stripes are reconstructed with a lower accuracy, because our monochrome "hash" pattern design sacrifices accuracy for the ability to 3D reconstruct based on a single image. We quantify the accuracy as the error between the 3D reconstructed data and the spherical fit of that data, using the known sphere diameter (82.55 ± 0.13 mm). The accuracy of the 3D reconstruction algorithm is -0.25 mm (mean error) and the precision is ± 0.79 mm (s.d. of error) for all 874,282 3D reconstructed points across all eight sphere locations. The associated 45,056 stripe intersection points are more accurately resolved, with an accuracy and precision of -0.11 ± 0.38 mm. The accuracy and precision for each of the eight sphere locations is given in Table 3 and shows the performance is consistent across the volume.

Finally, we present snapshots of a partial 3D volumetric surface reconstruction of a freely flying bird by combining the calibration method presented here with the structured light system we previously developed [14] and used for the sphere [Fig. 7]. The bird flapped its wings at 20 Hz during takeoff and we reconstructed its first downstroke [Fig. 8]. The calibration used to reconstruct the 3D surface of the bird had a mean absolute value reprojection error of 0.11 mm across the 95 different calibration target positions. The reconstruction demonstrates how the method presented here can be used to 3D reconstruct a rapidly moving and deforming, complex, surface. However more cameras and projectors are needed to fully reconstruct the body and wings. Further, a new algorithm is needed to reconstruct the depth edges of the bird, which are present at the leading and trailing edges of the wings and tail.

In conclusion, this novel calibration technique is the first automated method to calibrate multiple cameras and projectors simultaneously, which is particularly relevant for high-speed volumetric imaging of rapidly moving and deforming objects. The simultaneous rather than sequential calibration of all devices increases accuracy, and the unique printed and projected patterns enable full automation. The only calibration hardware required is an easily manufactured calibration target that can be moved around by the user in a wide range of calibration volumes. The image acquisition is straightforward and requires a single recording of ~1000 frames for each camera imaging the calibration target in about ~180 positions. For our setup, in which we moved the calibration target by hand, this took about 2 minutes. When combined with our published algorithm for 3D reconstructing the surface of an object using a single camera-projector pair [14], the entire pipeline for 3D reconstructing object surfaces from multiple views at high-speed is automated. The relative accuracy and precision of -0.03 ± 0.09 % of the diameter of a sphere, shows the promise of our method for a diverse set of measurement applications. Although further work is needed to fully 3D reconstruct the deforming 3D surfaces of particularly complex geometries, such as flying birds, we demonstrated the ability in principle. Further, other structured light methods [10–13] can now be adapted for automated high-speed usage to achieve similar results using our automated calibration method for multiple cameras and projectors. While we demonstrated the calibration method for four camera-projector pairs, it can be used to calibrate multi-view structured light systems in general with larger numbers of cameras and projectors. The applications range from 3D motion capture for computer generated movies, to consumer products, autonomous systems, industrial quality control, and scientific research in general.

Appendix: Notation

Table 4. Variable definitions.

Variable/Function	Description/Equation	Size
Intrinsic device calibration		
$\mathbf{K}_{\mathrm{c_i/p_i}}$	Intrinsic device (camera or projector) calibration matrix for the $\mathrm{i^{th}}$ camera or $\mathrm{i^{th}}$ projector	3×3
α, β	Device focal length (x, y), $\alpha = \mathbf{K}[1,1]$, $\beta = \mathbf{K}[2,2]$	1
u_0, v_0	Device principal point (x, y), $u_0 = \mathbf{K}[1,3]$, $v_0 = \mathbf{K}[2,3]$	1
$d_{\mathrm{r1}}, d_{\mathrm{r2}}, d_{\mathrm{r3}}$	1st, 2nd, and 3rd radial distortion coefficient	1
$d_{\mathrm{t1}}, d_{\mathrm{t2}}$	1st and 2nd tangential distortion coefficient	1
d_{s}	Skew distortion coefficient	1
Extrinsic device calibration		

${}^{\mathrm{A}}\mathbf{R}^{\mathrm{B}}$	Rotation matrix from reference frame B to reference frame A where ${}^{\mathrm{A}}\mathbf{R}^{\mathrm{B}} = {}^{\mathrm{A}}\mathbf{R}^{\mathrm{C}} \cdot {}^{\mathrm{C}}\mathbf{R}^{\mathrm{B}}$	3×3
${}^{\mathrm{A}}\mathbf{T}^{\mathrm{B}}_{\mathrm{C}}$	Translation vector from reference frame B to reference frame A written in reference frame C	3×1
$\mathbf{B}^{\mathrm{b_k}}_{\mathrm{R}}$	Linear fitting parameters to account for rotation of the kth calibration target position between frames	3×1
$\mathbf{B}^{\mathrm{b_k}}_{\mathrm{T}}$	Linear fitting parameters to account for translation of the kth calibration target position between frames	3×1
Notation		
${}_{\mathrm{c_i b_k / c_i p_j b_k}}\tilde{\mathbf{P}}^{\bigcirc/\otimes}_{\mathrm{c_i/p_i/b_k}}$	Notation for dots **P** or **X** Left subscript: where dots originate/are seen Right subscript: reference frame Right superscript: printed, $\bigcirc$, or projected, $\otimes$, dots Top: detected ($\sim$) vs. pre-determined (no accent) dots	$3 \times n$
$\mathrm{c_i}, \mathrm{w} = \mathrm{c_1}$	Subscript: ith camera. World reference frame is the 1st camera reference frame	N/A
$\mathrm{p_i}$	Subscript: ith projector	N/A
$\mathrm{b_{k:i}}, \mathrm{b_{k\cdot i}}, \mathrm{b_{2,k}}, \mathrm{b_k}$	Subscript: kth calibration target with designation based on how the target was seen. $\mathrm{b_{k:i}}$: seen by the ith camera; $\mathrm{b_{k\cdot i}}$: seen by the ith projector; $\mathrm{b_{2,k}}$: target side 2; $\mathrm{b_k} = \mathrm{b_{1,k}}$	N/A
$\mathbf{X}$	3D coordinates	$3 \times n$
$\mathbf{P}$	2D homogeneous coordinates where $\mathbf{P}[1,:]$ and $\mathbf{P}[2,:]$ are pixel coordinates on the camera or projector and $\mathbf{P}[3,:] = 1$	$3 \times n$
λ	Diagonal matrix of constants used to convert 3D coordinates to 2D homogeneous coordinates so that the $\mathbf{P}[3,:] = 1$ constraint is satisfied	$n \times n$
$\mathbf{A}[:, i]$	ith column of **A** matrix	$\_ \times 1$
$\mathbf{A}[i, :]$	ith row of **A** matrix	$1 \times \_$
$\breve{\mathbf{K}}$	This added symbol indicates that the intrinsic device calibration matrix definition includes the skew distortion coefficient	3×3
Constants		
I	Number of cameras	1
J	Number of projectors	1
K	Number of calibration target positions	1
F	Number of sequential frames captured by cameras in a single burst	1
$fr(i)$	ith sequential frame number out of F frames	1
$N^{\bigcirc/\otimes}$	Total number of detected printed, $\bigcirc$, or projected, $\otimes$, dots	1

N_{Jacobian}	Total number of values computed in the Jacobian of the full calibration minimization function without bundle adjustment	1
$N_{\text{Jacobian-adj}}$	Total number of values computed in the Jacobian of the full calibration minimization function with bundle adjustment	1
Circular sigmoid dot fit		
a, b	Fitting parameters 1-2: center of the dot being fit	1
R	Fitting parameter 3: radius of the dot being fit	1
c_1	Fitting parameter 4: contrast between outside and inside dot	1
c_2	Fitting parameter 5: brightness outside dot	1
c_3	Fitting parameter 6: measure of blurriness	1
$\mathbf{I}_{\mathrm{c_i}}\left(u_{\mathrm{c_i}}, v_{\mathrm{c_i}}\right)$	Grayscale intensity of camera image at the coordinates $\left(u_{\mathrm{c_i}}, v_{\mathrm{c_i}}\right)$	1
$\left(a_{\mathrm{b_{ki}/p_j}}, b_{\mathrm{b_{ki}/p_j}}\right)$	Coordinates in either projector pixels or the calibration target reference frame transformed from $\left(u_{\mathrm{c_i}}, v_{\mathrm{c_i}}\right)$	1
Other		
$E^{\bigcirc/\otimes}$	Sum of reprojection error between pre-determined printed, $\bigcirc$, or projected, $\otimes$, dot positions and detected dots	1
$\mathbf{H}$	2D planar homography between detected dots and printed or projected dots	3×3
$\mathbf{R}_{3\times1} = \mathrm{rod}\left(\mathbf{R}_{3\times3}\right)$	Rodrigues transformation to transform a 3×3 rotation matrix into a 3×1 vector	N/A
$\mathbf{R}_{3\times3} = \mathrm{rod}^{-1}\left(\mathbf{R}_{3\times1}\right)$	Inverse Rodrigues transformation to transform a 3×1 vector into a 3×3 rotation matrix	N/A
$\mathbf{J}_{1,\mathrm{n}}$	Matrix of all ones	$1 \times n$

Funding

National Science Foundation (NSF) (DGE-114747, 1552419); Army Research Laboratory (ARL) (831324).

Acknowledgments

National Science Foundation (NSF) Graduate Research Fellowship Award DGE-114747 to M.D., Army Research Lab (ARL) cooperative agreement W911NF-08-2-0004, BAE Systems award number 831324 with the Micro Autonomous Systems and Technology (MAST) Collaborative Technology Alliance, and NSF CAREER Award 1552419 to D.L. We thank Ben Hightower for reviewing the mathematics and Diana Chin for helping train the bird shown in Fig. 8.

Disclosures

Stanford Office of Technology Licensing (P).

References

1. J. D. Ackerman, K. Keller, and H. Fuchs, "Surface reconstruction of abdominal organs using laparoscopic structured light for augmented reality," Three-Dimensional Image Capture Appl. **4661**, 39–46 (2002).
2. R. T. McKeon and P. J. Flynn, "Three-dimensional facial imaging using a static light screen (SLS) and a dynamic subject," IEEE Transactions on Instrumentation Meas. **59**, 774–783 (2010).
3. W. Lohry and S. Zhang, "High-speed absolute three-dimensional shape measurement using three binary dithered patterns," Opt. Express **22**, 26752 (2014).
4. Y. Wang, J. I. Laughner, I. R. Efimov, and S. Zhang, "3D absolute shape measurement of live rabbit hearts with a superfast two-frequency phase-shifting technique," Opt. Express **21**, 6631–6636 (2013).
5. J. Salvi, S. Fernandez, T. Pribanic, and X. Llado, "A state of the art in structured light patterns for surface profilometry," Pattern Recognit. **43**, 2666–2680 (2010).
6. C. Guan, L. Hassebrook, and D. Lau, "Composite structured light pattern for three-dimensional video," Opt. express **11**, 406–417 (2003).
7. J. Lenar, M. Witkowski, V. Carbone, S. Kolk, M. Adamczyk, R. Sitnik, M. van der Krogt, and N. Verdonschot, "Lower body kinematics evaluation based on a multidirectional four-dimensional structured light measurement," J. biomedical optics **18**, 56014 (2013).
8. R. Sagawa, K. Sakashita, N. Kasuya, H. Kawasaki, R. Furukawa, and Y. Yagi, "Grid-based active stereo with single-colored wave pattern for dense one-shot 3D scan," in *IEEE 2nd Joint 3DIM/3DPVT International Conference on 3D Imaging, Modeling, Processing, Visualization and Transmission,* (2012), pp. 363–370.
9. W.-H. Su and H. Liu, "Calibration-based two-frequency projected fringe profilometry: a robust, accurate, and single-shot measurement for objects with large depth discontinuities," Opt. express **14**, 9178–9187 (2006).
10. H. Sarbolandi, D. Lefloch, and A. Kolb, "Kinect range sensing: structured-light versus time-of-flight Kinect," Comput. Vis. Image Underst. **139**, 1–20 (2015).
11. L. Keselman, J. I. Woodfill, A. Grunnet-Jepsen, and A. Bhowmik, "Intel RealSense stereoscopic depth cameras," CoRR pp. 1–10 (2017).
12. S. Ryan Fanello, C. Rhemann, V. Tankovich, A. Kowdle, S. Orts Escolano, D. Kim, and S. Izadi, "Hyperdepth: learning depth from structured light without matching," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition,* (2016), pp. 5441–5450.
13. T. Wolf and R. Konrath, "Avian wing geometry and kinematics of a free-flying barn owl in flapping flight," Exp. Fluids **56,** 28 (2015).
14. M. E. Deetjen, A. A. Biewener, and D. Lentink, "High-speed surface reconstruction of a flying bird using structured light," J. Exp. Biol. **220**, 1956–1961 (2017).
15. D. C. Brown, "Close-range camera calibration," Photogramm. Eng. **37**, 855–866 (1971).
16. I. Sobel, "On calibrating computer controlled cameras for perceiving 3-D scenes," Artif. Intell. **5**, 185–198 (1974).
17. D. H. Theriault, N. W. Fuller, B. E. Jackson, E. Bluhm, D. Evangelista, Z. Wu, M. Betke, and T. L. Hedrick, "A protocol and calibration method for accurate multi-camera field videography," J. Exp. Biol. **217**, 1843–1848 (2014).
18. Z. Zhang, "A flexible new technique for camera calibration," IEEE Transactions on Pattern Analysis Mach. Intell. **22**, 1330–1334 (2000).
19. J. Y. Bouguet, "Camera calibration toolbox for Matlab," (2000).
20. O. D. Faugeras, *Three-Dimensional Computer Vision: a Geometric Viewpoint* (MIT University, 1993).
21. M. Sonka, V. Hlavac, and R. Boyle, *Image Processing, Analysis, and Machine Vision* (PWS Publishing, 1999), ii ed.
22. R. Sitnik, "Phase scaling using characteristic polynomials," Proc. SPIE **5532**, 211–217 (2004).
23. I. Léandry, C. Brèque, and V. Valle, "Calibration of a structured-light projection system: development to large dimension objects," Opt. Lasers Eng. **50**, 373–379 (2012).
24. S. J. Maybank and O. D. Faugeras, "A theory of self-calibration of a moving camera," Int. J. Comput. Vis. **8**, 123–151 (1992).
25. W. Schreiber and G. Notni, "Theory and arrangements of self-calibrating whole-body three-dimensional measurement systems using fringe projection technique," Soc. Photo-Optical Instrumentation Eng. **39**, 159–169 (2000).
26. P. Beardsley and R. Raskar, "A self-correcting projector," IEEE Comput. Soc. Conf. on Comput. Vis. Pattern Recognit. **2**, 504–508 (2001).
27. S. Y. Chen and Y. F. Li, "Self recalibration of a structured light vision system from a single view," IEEE Int. Conf. on Robotics & Autom. pp. 2539–2544 (2002).
28. J. Yuan, Q. Wang, and B. Li, "A flexile and high precision calibration method for binocular structured light scanning system," Sci. World J. **2014**, 1–8 (2014).
29. B. Zhao and Z. Hu, "Camera self-calibration from translation by referring to a known camera," Appl. Opt. **54**, 7789–7798 (2015).
30. Q. Sun, X. Wang, J. Xu, L. Wang, H. Zhang, J. Yu, T. Su, and X. Zhang, "Camera self-calibration with lens distortion," Optik **127**, 4506–4513 (2016).

31. D. Wu, T. Chen, and A. Li, “A high precision approach to calibrate a structured light vision sensor in a robot-based three-dimensional measurement system,” Sensors **16**, 1–18 (2016).
32. F. Sadlo, T. Weyrich, R. Peikert, and M. Gross, “A practical structured light acquisition system for point-based geometry and texture,” IEEE Eurographics Symp. on Point-Based Graph. pp. 89–98 (2005).
33. J. Liao and L. Cai, “A calibration method for uncoupling projector and camera of a structured light system,” IEEE/ASME Int. Conf. on Adv. Intell. Mechatronics pp. 770–774 (2008).
34. K. Yamauchi, H. Saito, and Y. Sato, “Calibration of a structured light system by observing planar object from unknown viewpoints,” IEEE Int. Conf. on Pattern Recognit. pp. 1–4 (2008).
35. W. Gao, L. Wang, and Z.-Y. Hu, “Flexible calibration of a portable structured light system through surface plane,” Acta Autom. Sinica **34**, 1358–1362 (2008).
36. R. Legarda-Saenz, T. Bothe, and W. P. Juptner, “Accurate procedure for the calibration of a structured light system,” Opt. Eng. **43**, 464–471 (2004).
37. Z. Li, “Accurate calibration method for a structured light system,” Opt. Eng. **47**, 053604–1–8 (2008).
38. X. Chen, J. Xi, Y. Jin, and J. Sun, “Accurate calibration for a camera-projector measurement system based on structured light projection,” Opt. Lasers Eng. **47**, 310–319 (2009).
39. S. Zhang and P. S. Huang, “Novel method for structured light system calibration,” Opt. Eng. **45**, 083601–1–8 (2006).
40. M. Ren, J. Liang, B. Wei, and W. Pai, “Novel projector calibration method for monocular structured light system based on digital image correlation,” Optik **132**, 337–347 (2017).
41. P. Kuehmstedt, G. Notni, W. Schreiber, and J. Gerber, “Full-hemisphere automatic optical 3D measurement system,” SPIE **3100**, 261–265 (1997).
42. A. Asundi and W. Zhou, “Mapping algorithm for 360-deg profilometry with time delayed integration imaging,” Opt. Eng. **38**, 339–344 (1999).
43. L. Nie, Y. Ye, and Z. Song, “Method for calibration accuracy improvement of projector-camera-based structured light system,” Opt. Eng. **56**, 074101–1–9 (2017).
44. T. Svoboda, D. Martinec, and T. Pajdla, “A convenient multicamera self-calibration for virtual environments,” Presence: Teleoperators Virtual Environ. **14**, 407–422 (2005).
45. B. Li, L. Heng, K. Koser, and M. Pollefeys, “A multiple-camera system calibration toolbox using a feature descriptor-based calibration pattern,” IEEE Int. Conf. on Intell. Robots Syst. pp. 1301–1307 (2013).
46. ARPG, “Autonomous robotics perception group: visual-inertial calibration tool: vicalib,” (2018).
47. A. Grunnet-jepsen, P. Winer, A. Takagi, J. Sweetser, K. Zhao, T. Khuong, D. Nie, and J. Woodfill, “Using the RealSense D4xx depth sensors in multi-camera configurations,” (2018).
48. D. Bradley and G. Roth, “Adaptive thresholding using the integral image,” J. Graph. Tools **12**, 13–21 (2007).
49. H. Edelsbrunner, D. G. Kirkpatrick, and R. Seidel, “On the shape of a set of points in the plane,” IEEE Transactions on Inf. Theory **29**, 551–559 (1983).
50. S. P. Lloyd, “Least squares quantization in PCM,” IEEE Transactions on Inf. Theory **28**, 129–137 (1982).
51. M. A. Fischler and R. C. Bolles, “Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography,” Commun. ACM **24**, 381 – 395 (1981).
52. A. Trujillo-Pino, K. Krissian, M. Alemán-Flores, and D. Santana-Cedrés, “Accurate subpixel edge location based on partial area effect,” Image Vis. Comput. **31**, 72–90 (2013).
53. T. F. Coleman and Y. Li, “An interior trust region approach for nonlinear minimization subject to bounds,” SIAM J. on Optim. **6**, 418–445 (1996).
54. E. Trucco and A. Verri, *Introductory Techniques for 3-D Computer Vision* (Prentice Hall, 1998).
55. B. Triggs, P. F. McLauchlan, R. I. Hartley, and A. W. Fitzgibbon, “Bundle adjustment - a modern synthesis,” Vis. Algorithms: Theory Pract. **1883**, 298–372 (2000).
56. S. W. Lee, S. Y. Lee, and H. J. Pahk, “Precise edge detection method using sigmoid function in blurry and noisy image for TFT-LCD 2D critical dimension measurement,” Curr. Opt. Photonics **2**, 69–78 (2018).
57. K. N. Kutulakos and S. M. Seitz, “A theory of shape by space carving,” Proc. Seventh IEEE Int. Conf. on Comput. Vis. **3**, 197–216 (1999).