Self Calibrating Lenticular Markers and Articulated Pose Estimation

Abstract—**We explore new approaches to pose estimation of articulated objects based on chromo-coded lenticular whose color depends on the angle from which they are viewed. We solve the self-calibration problem for a collection of chromo-coding lenticular markers. This is a version of the Structure From Motion problem where the points are made from a lenticular material whose apparent color constrains its relative rotation. We derive geometric constraints to support calibration from markers arbitrarily placed on a surface, and then show how they can be used to measure the pose of a simple articulated object.**

I. INTRODUCTION

Pose estimation has a key role in computer vision because it is helpful to know where things are and how they are oriented. Most pose estimation algorithms are based on detecting or tracking fiducial markers with known absolute or relative 3D positions. Solving for the position and rotation of an object depends on sensitive calculations of the relative positions of points, so the accuracy of pose-estimation especially for skinny objects where the fiducial markers are nearly co-linear, is limited by the resolution of the camera.

Recently, alternative fiducial markers have been introduced where the observed color of the fiducial marker encodes constraints on the rotation of the marker relative to the camera [10]. This allows more flexibility in marker placement and the ability to solve for the pose of skinny objects. However, like all fiducial markers, it requires that the position of the fiducial marker is known, and additionally requires that the rotation and the relationship between color and relative angle is also known. This creates a barrier to using chromo coding fiducial markers.

The contribution of this paper is to derive and demonstrate self-calibration approaches to simplify the of chromo coded markers outside the laboratory environment. We consider a work-flow where someone wants to use chromo-coded markers to track an object, so they stick a collection of stickers on the object surface and then move the object through a range of approximately known poses. From this data, we solve a version of the structure from motion problem that solves for the relative position, surface normal and orientation of each chromo-coded marker, and solve for the mapping between their observed color and the angle from which they are viewed. We characterize how this self-calibration depends on variation in placing the markers and show demonstrations where it is used to solve for simply articulated pose-estimation in AR scenarios.

II. BACKGROUND

When observed in an image, finding points in an image that have known 3D relationships give clean geometric constraints that define the position and location of the object in the scene [5]. Fiducial markers are often used to simplify the process of finding known 3D points. The most common fiducial markers are based on libraries including ARtoolkit [6], ARTag [3], [4] and CALTag [1]. These create identifiable black and white that are easy to detect and identify. The most common approach automatically prints out a collection of different tags on a sheet of paper; given a pattern with multiple fiducial markers whose relative 3D coordinates are known.

A few prior works have explicitly created fiducial markers whose relative appearance depends on the direction from which they are viewed. Agam fiducials [2] were the first to create micro-textured surfaces whose facets have different colors or intensities and inspired a collection of ways where observed patterns code explicitly for relative orientation [14], [16], [13], [15], [8], or color variations which encode constraints on rotation [7]. Constraints from such markers have been explicitly derived for both pose-estimation [11], [9] and camera calibration applications to augmented reality [12].

Fiducial markers are used in many ways. Our paper particularly considers the use of fiducial markers in settings where they are placed on objects in ad-hoc configurations. This specific problem has been considered for standard fiducial markers for tangible interaction devices, in work which uses structure from motion to self-calibrate the relative position of hand-placed markers [17]. This work is a first step in the same direction for fiducial markers whose color gives an additional rotational constraint.

III. GEOMETRIC BACKGROUND

The geometry of chromo-coded fiducial markers was derived in the context of pose-estimation [10]. We use similar markers made from lenticular arrays, constructed so that they appear different colors when viewed from different orientations. These offer geometric constraints, as illustrated in Figure 1. The apparent color depends on how far the viewing direction is rotated around the axis of the cylindrical lenses of the lenticular array. Thus, when a marker is observed in a given color, the viewing ray is constrained to lie in a particular plane. For a particular hue, we define n_{hue} as the surface normal to the plane containing all viewing directions of that hue.

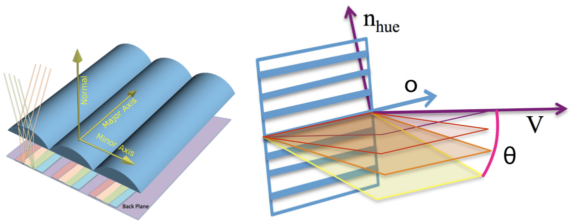

Fig. 1. (a) Lenticular arrays focus light from different directions onto strips behind each cylindrical lens, with the right backplane pattern, the apparent color depends on viewing direction. (b) The apparent color geometry depends on that axis o parallel to the cylindrical lenses, the angle of the viewing direction θ around that axis. All viewing directions in a particular plane are the same color, and we describe the plane for a particular hue by its surface normal $\boldsymbol{n}_{hue}$.

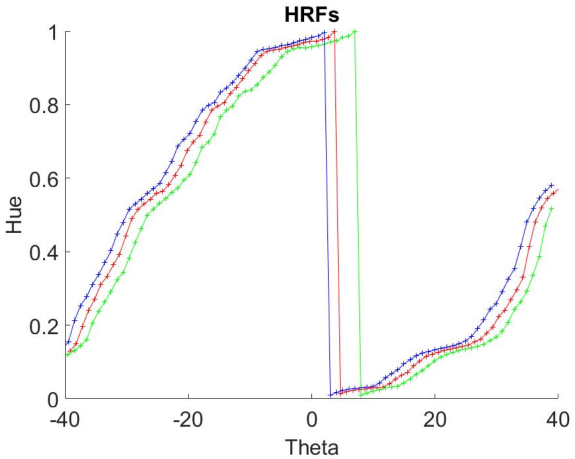

Fig. 2. Example HRFs measured on an optical turntable show a consistent shape with a slight offset, consistent with challenges in manufacturing chromo-coded lenticular markers.

A. Marker Model and Color-Angle Relationship

Figure 1 shows the local geometry of a chromo coded marker made from a lenticular array. The major axis lies along the direction of the cylinders, a minor axis is perpendicular, and a surface normal. When viewed from a particular direction, the viewing angle, in this coordinate system, determines the color. We use the observed hue of the marker as more robust measurement that a richer description of the color, encode the relationship of the hue and viewing angle through a function we call the hue response function (HRF). For a given viewing direction $\boldsymbol{V}$, we expect to observe a color whose hue is $H(\boldsymbol{V})$. In addition, the hue changes only as a function of the angle of $\boldsymbol{V}$ around the major axis of the marker, θ. Then $H(\boldsymbol{V})$ becomes $H(\theta)$.

While previous work calibrates every marker completely independently [12], we find that variation between markers (produced in the same production run) arises from slight mismatches in the period of the back-plane texture and the lenticular spacing. For example, on a lenticular array with 2 lenticular lenses per millimeter printed over a 10 cm patch, a stretch of 0.1 mm in the printing would cause the color-angle relationship to vary over the 10cm patch. If that 10 cm patch is cut into many markers, this creates a linear offset in the HRF of one marker relative to the others.

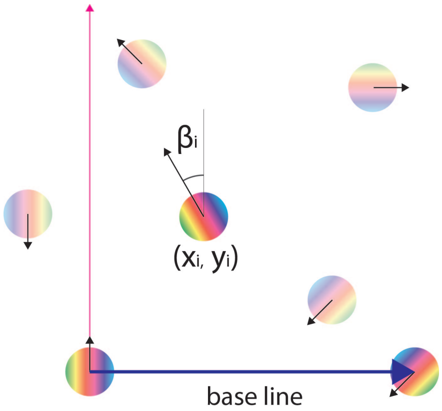

Fig. 3. A cartoon showing the baseline marker that define the origin and x-axis of the object local coordinate system. The position and in-plane rotation of each marker is described with (x_i, y_i) and β_i.

In producing our chromo-coded markers, all circular markers are cut from a 10×10 cm lenticular card by a laser cutter. The radius of each marker is 2 cm. At the scale of one marker, the colors appear visually consistent, and across markers we observe a consistent shift. Figure 2 demonstrates this shift under laboratory conditions. Therefore, we here model the HRF as an unknown function mapping the angle θ to a hue. We model this function using spline fitting method and assume it is consistent across markers except for a constant shift.

Thus, the HRF for marker i can be described as a general function that takes a viewing direction $\boldsymbol{V}$, computes the angle θ of the corresponding $\boldsymbol{V}$ of that marker and then apply a marker specific angle offset to that angle:

$$H_i(V) = H(\theta + \text{offset}(i)) \tag{1}$$

B. Relative Marker Positions

When using several markers, it is necessary to characterize their relative position and orientation. We use two markers to define a coordinate system; the first marker serves as the origin and the second marker is defined to lie at location (1,0,0). Thus the distance between the two markers is defined as 1 unit; and the x-axis is defined to be along the vector from the first to the second marker. The choice of baseline markers is arbitrary and they can be selected in any way which is convenient for describing other markers. All other markers are defined in this coordinate system, with the position of marker i denoted as $\boldsymbol{d}_i = (x_i, y_i)$, shown in Figure 3.

The orientation of a marker also affects its apparent color. We characterize the orientation of a marker as the angle its major axis makes with the y-axis of the baseline coordinate system; for marker i we denote this as β_i.

C. Relative Pose of Camera and Object

In a given frame, the appearance of the markers depends on their orientation, and the parameters of the camera that take a picture. We assume the camera calibration matrix K is known, so the parameters that vary from frame to frame are the rotation matrix R and the translation matrix T that relate the coordinate system of the camera to the coordinate system of the object.

Based on a pinhole camera model, an image pixel $\boldsymbol{p}$ views the world along ray $K^{-1}\boldsymbol{p}$. If this pixel is viewing fiducial marker i, then to determine the color we need to express this viewing ray in the coordinate system of marker i, which requires rotating the ray into the coordinate system of the marker. That marker is rotated relative to the object coordinate system by an angle β_i and the object coordinate system is rotated relative to the camera by a rotation R_0, leading to an overall rotation as $R_i = R_0 R_z(\beta_i)$, where $R_z(\beta_i)$ is the rotation matrix characterizing a rotation around the z-axis by an angle β_i. This leads to the complete mapping of the pixel of the center of marker i, $\boldsymbol{p}_i$, into a viewing direction as:

$$\boldsymbol{V}_i = {R_i}' K^{-1} \boldsymbol{p}_i \tag{2}$$

In later sections we use these geometric constraints to solve for the camera pose based on multiple frames of video. In these cases, the geometric constraints are indexed by the frame number f, as:

$$\boldsymbol{V}_i^f = {R_i^f}' K^{-1} \boldsymbol{p}_i^f, \tag{3}$$

which captures the fact that a fiducial marker might be observed at a different location at each frame $\boldsymbol{p}_i^f$ because the object may have a different rotation at each frame R_i^f, but the camera calibration K and the rotation of the marker relative to the object coordinate system β_i remains consistent.

IV. OPTIMIZING FOR MARKER SELF-CALIBRATION

Based on multiple viewpoints of a set of chromo-coded markers, we optimize for positions, orientations and color-responses of all markers. The input data is derived from each image (indexed by f) and for each marker (indexed by i), and we describe the position on the image where the marker was observed as $\boldsymbol{p}_i^f$ and the hue of that marker in that frame as h_i^f.

Our goal is to solve for the relative positions and orientations of each marker on the object (the $\boldsymbol{d}_i$ and β_i), the fitting function of the HRF for all markers and the offset that characterizes the color-response of each specific marker. As a side effect, we also optimize of the rotation R^f and the translation T^f that characterize the relative position of the camera to the object plane in each frame f. The constraints on these parameters can be broken into several forms; constraints based on the measured hue of the markers, constraints on the measured image position of the markers and constraints on the consistency of the HRF model. This section presents a collection of constraints, the full algorithms that uses these constraints to optimize over all unknowns is presented in the next section.

A. Constraints on rotation from hue

The observed color of a fiducial marker depends on the angle from which it is viewed, through the constraint defined in Equation 3. We define a constraint that compares the predicted hue value for a given estimate (of the rotations, the marker orientations, and the HRF parameters) as:

$$E_{hue} = \sum_{i,f} \| H(\boldsymbol{V}_i^f) - h_i^f \|_* \tag{4}$$

Hue values are between 0 and 1, with both 0 and 1 representing a hue of red, so we define the norm ($\|a-b\|_*$) as $min(|a-b|, 1-|a-b|)$ to reflect that a hue very close to 0 is close to a hue very close to 1.

B. Constraints on rotation from $\boldsymbol{n}_{hue}$

A complement to hue constraints is the constraints on the n_{hue} as shown in Figure 1. If the pose of marker i is correct, $\boldsymbol{n}_{hue_i}$ should be perpindicular to the corresponding viewing direction $\boldsymbol{V}_i$. Therefore the magnitude of the dot product between $\boldsymbol{n}_{hue_i}^f$ and $\boldsymbol{V}_i^f$ should be zero. For a given viewing direction $\boldsymbol{V}_i^f$ derived from estimates of the rotation, the marker orientations, and the HRF parameters, gives an error term as:

$$E_{\boldsymbol{n}_{hue}} = \sum_{i,f} |\boldsymbol{n}_{hue_i}^f \cdot \boldsymbol{V}_i^f| \tag{5}$$

C. Constraints from observed marker location

Standard fiducial markers define constraints based on their position. The prediction position of a marker depends on the camera calibration, rotation, translation, and the position of the marker in the object coordinate system. Given these parameters, we can predict the position of marker i in frame f as:

$$p_i^f = K(R^f \boldsymbol{d_i} + T^f),$$

and use this to define a re-projection error. This is summed over all points in all frames:

$$E_{pts} = \sum_{i,f} |\boldsymbol{p}_i^{f'} - \boldsymbol{p}_i^f| \tag{6}$$

where $\boldsymbol{p}_i^{f'}$ is the measured pixel coordinate of the centroid of marker i in frame f.

D. Relative Weighting

These three errors functions have different scales, E_{hue} and $E_{\boldsymbol{n}_{hue}}$ range between 0 and 1, and E_{pts} is a potentially arbitrarily large pixel-coordinate difference. Near optimal solutions, we found the average error the hue for each marker in each frame is 0.02, the average dot error for each marker in each frame is 0.02 and the center points reprojection error for each marker in each frame is approximately 1 pixels. Therefore, in all subsequent work we add multiply the error in E_{pts} by 0.02 so that all terms are approximately equally weighted.

V. Optimization Algorithm

Given a collection of F images of k markers, we seek to optimize for all the parameters that specify the marker positions and orientation. We will use the error functions from the previous section with the following initialization:

- $\boldsymbol{\beta} = \beta_1 \dots \beta_k$, the orientations of each marker, are initialized as an alternating series of $0, 90, \dots$ and users are asked to place markers with orientations that are approximately consistent with this.
- $\boldsymbol{D} = \boldsymbol{d}_3 \dots \boldsymbol{d}_k$, the positions of k markers ($\boldsymbol{d}_1$ and $\boldsymbol{d}_2$ define the origin and the x-axis so they have no free parameters). Initial estimates of $\boldsymbol{d}_3 \dots \boldsymbol{d}_k$ are computing by choosing one calibration image at random, assuming the object is fronto-parallel, and applying the similarity transform so that $\boldsymbol{d}_1$ is at the origin and $\boldsymbol{d}_2$ has coordinates (1,0).
- $\mathrm{off} = \mathrm{offset}(1), \dots \mathrm{offset}(k)$, the offsets of the HRF for each marker are all initialized to be 0.
- $\boldsymbol{R} = R_1, \dots R_t$, the rotation parameters for each frame are initialized so that the orientation of the object coordinate system is always fronto-parallel.
- $\boldsymbol{T} = T_1, \dots T_t$, the translation parameters for each frame are initialized so that the object is assumed to be 10 unit from the camera (where the unit is defined by the distance between the two baseline markers).
- $H(\theta)$ is the spline function handle of the HRF function optimized after each iteration.

Given this initialization, we perform several steps of alternating non-linear optimization, with 3 different error functions.

1) Hue error iteration
   a) Fix $\boldsymbol{R}, \boldsymbol{\beta}, \boldsymbol{D}, H(\theta), \mathrm{off}$ and optimize $\boldsymbol{T}$ to minimize E_{pts}
   b) Fix $H(\theta), \mathrm{off}$ and optimize $\boldsymbol{R}, \beta, \boldsymbol{D}, \boldsymbol{T}$ to minimize $E_{hue} + E_{pts}$.
   c) Fix $\boldsymbol{R}, \boldsymbol{\beta}, \boldsymbol{D}, \boldsymbol{T}$ and optimize $H(\theta), \mathrm{off}$ to minimize the sum of E_{hue}
2) Dot error iteration
   a) Fix $\boldsymbol{R}, \boldsymbol{\beta}, \boldsymbol{D}, H(\theta), \mathrm{off}$ and optimize $\boldsymbol{T}$ to minimize E_{pts}
   b) Fix $H(\theta), \mathrm{off}$ and optimize $\boldsymbol{R}, \beta, \boldsymbol{D}, \boldsymbol{T}$ to minimize $E_{\boldsymbol{n}_{hue}} + E_{pts}$.
   c) Fix $\boldsymbol{R}, \boldsymbol{\beta}, \boldsymbol{D}, \boldsymbol{T}$ and optimize $H(\theta), \mathrm{off}$ to minimize the sum of E_{hue}
3) Cross iteration
   a) Fix $\boldsymbol{R}, \boldsymbol{\beta}, \boldsymbol{D}, H(\theta), \mathrm{off}$ and optimize $\boldsymbol{T}$ to minimize E_{pts}
   b) Fix $H(\theta), \mathrm{off}$ and optimize $\boldsymbol{R}, \beta, \boldsymbol{D}, \boldsymbol{T}$ to minimize $E_{hue} + E_{pts}$.
   c) Fix $H(\theta), \mathrm{off}$ and optimize $\boldsymbol{R}, \beta, \boldsymbol{D}, \boldsymbol{T}$ to minimize $E_{\boldsymbol{n}_{hue}} + E_{pts}$.
   d) Fix $\boldsymbol{R}, \boldsymbol{\beta}, \boldsymbol{D}, \boldsymbol{T}$ and optimize $H(\theta), \mathrm{off}$ to minimize the sum of E_{hue}

This somewhat complicated optimization process is motivated as follows. The relationship between hue and angle is complicated because the same hue is visible at multiple angle. The first alternating minimization step uses E_{hue} which defines an error over the observed color, and is therefore robust to initializations that may be far from the correct rotation or marker orientation. The second optimization uses $E_{\boldsymbol{n}_{hue}}$ because this gives a direct angular constraint. The third optimization is one that we discovered empirically to further refine the results. In all following tests, we apply each of these three iteration methods 5 times.

VI. Experimental Results

We show three experiments with a real world implementation of the color coded lenticular markers. The first uses a marker configuration shown in Figure 4, which includes a checkerboard to allow standard pose estimation tools to provide an alternative estimate of the pose. The second and third use a hinge and a hinged ruler to explore articulation of points within a plane and rotating out of the plane.

A. Rigid Configuration Experiment

The first experiment is a controlled scenario that includes a checkerboard to enable comparison to standard approaches for pose estimation. The experiment captures 50 images of the object with 4 fiducial markers (and the checkerboard which is not used on our algorithm), as the object is rotation through approximately -60 to 60 degrees around the vertical axis. All variables are initialized and the optimization is run as described in the last section. Figure 4 shows results highlighting the convergence of the HRF (the mapping from color to angle), the improvements in estimating the rotation of the object and the improvements in the estimation of the marker locations visible in the improved reprojection errors.

In this controlled case, we also test the robustness of the calibration algorithm, to explore sensitivity to the initialization. We explore the sensitivity to whether the marker orientations are orthogonal with each other, and whether the images of the object are sampled smoothly over the different possible object orientations. The proposed end-user calibration is as follows:

While taking a video with the camera, users should rotate the object approximately around the y-axis with approximately equal rotational velocity, with a range of approximately -60 to 60 degrees. Then they should rotation around the x-axis over the same range. This data acquisition protocol offers simple initialization for the rotation estimates.

In the robustness test, we consider reasonable affordances in how carefully a user might follow directions. First, instead of rotating the object from -60 to 60 degrees, they might rotate through a different angle. Figure 5 shows the average error in estimating object rotation for initial guesses that the object was rotation by plus/mines 50 60 70 80 degrees, and the figure shows minimal effects on the outcome.

Next, we simulate the error occur when user rotate the object around something other than the vertical axis. Using an initial condition where the rotation axis was wrong by an error δa_x

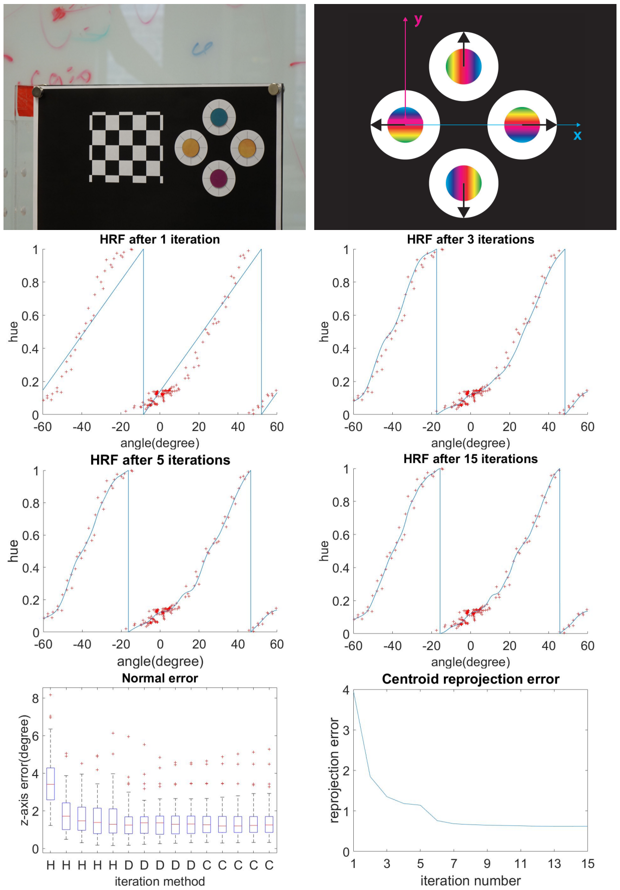

Fig. 4. Top row: the marker configuration for the rigid test. The middle 2 rows show the refinement of the hue-response function after 1, 3, 5, 15 iterations. Red points have locations that depend on their angle as measured by the checkerboard (x) and their measured hue (y). The bottom left figure shows the angular error in estimating the surface normal as a function of iteration. The x-axis labels highlight that the "Hue" error iterations was done first for 5 iterations, then the "Dot" error iterations then the "Cross" error iterations. The bottom right figure shows the points reprojection error of each marker in each frame.

that differs from the y-axis by plus/mines 5 10 15 20 degrees has some effect, but overall errors remain less than 2 degrees.

Finally, when the user place the markers in an orientation they may not place it exactly parallel or orthogonal to the original marker. We add ±5 and ±10 degree error on the orientation initial guess, and highlight in Figure 5 that this has very minimal effects.

B. In Plane Articulation

In the second experiment we explore the effect of different marker locations and configurations. To do this, we attach 4 markers to an adjustable ruler, which can rotate in plane, as shown in Figure 6. When articulating, the top two markers rotate relative to the bottom two, and the ruler markings allows us to understand the relative rotation.

Keeping the ruler closed in a rigid configuration, we run the user-enabled calibration, rotating the ruler around the x-axis and y-axis from around 60 degrees to -60 degrees by hand.

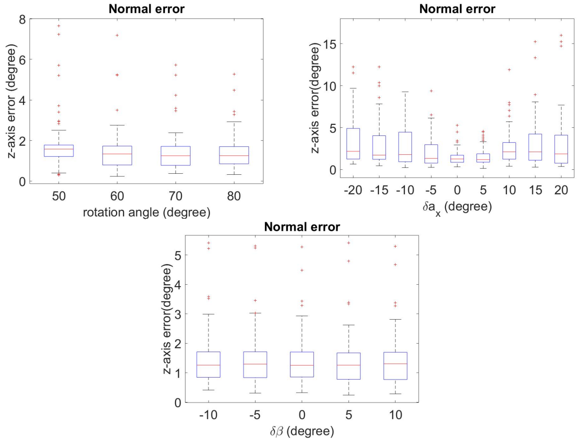

Fig. 5. Top left figure: the final normal error of all calibration frames with rotation angle around the y-axis/x-axis from -50 to 50, -60 to 60, -70 to 70, -80 to 80. Top right figure: the final normal error of all calibration frames with error δa_x in the rotation initial condition. The bottom figure shows the final normal error of all calibration frames with error in orientation angle initial condition

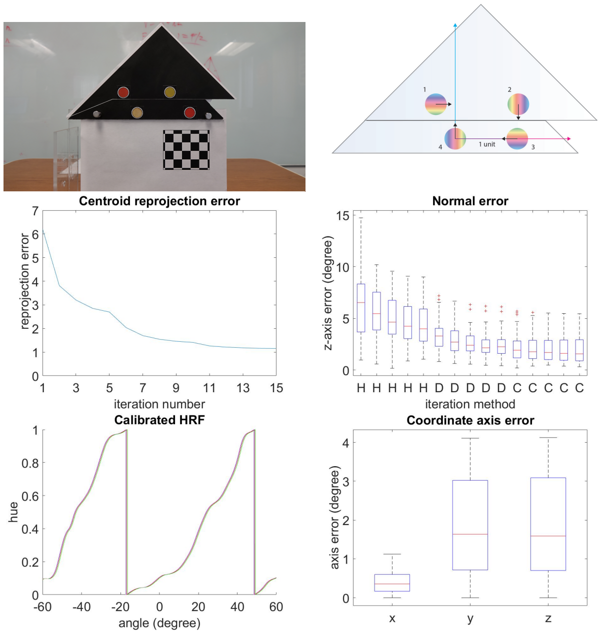

Fig. 6. Top row: an image of an articulated compas ruler and a cartoon of the approximate marker coordinates. Middle row: the reprojection error and normal error after each optimization iteration. The bottom left shows the calibrated HRF. The bottom right figure shows the rotation estimation error of object plane with single image after HRF and calibration

Figure 6 shows the error in the estimated rotation, highlighting that the rotational estimate is good across a large variety of poses.

The calibration step gives estimates for $H(\theta), \text{off}, \boldsymbol{R}, \boldsymbol{\beta}, \boldsymbol{D}, \boldsymbol{T}$. We then explore the ability to measure the articulation. By construction marker $1, 2$ move in a rigid configuration, marker $3, 4$ move in another rigid

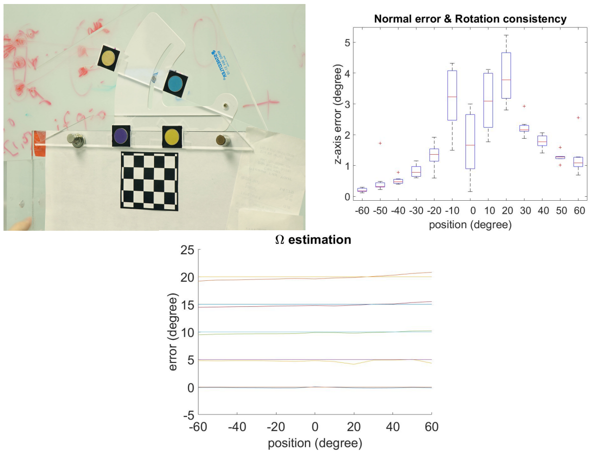

Fig. 7. The top left figure shows the articulated ruler in another configuration. The top right shows the error for rotation estimation as a function of the true orientation. The bottom shows the estimates of the articulation angle Ω across all directions (which should be constant at 0, 5, 10, 15, 20 degrees).

configuration, but two different configurations are connected through a pivot forcing an in-plane rotation. It only needs one new parameter, the opening angle Ω, to describe the new configuration.

We solve for R, T, Ω using the cross-iteration method. We use the same experimental setup as the previous experiment, where the rulers are set to a fixed angle, held against a wall and a camera observes them from different angles. We spread the ruler apart in increments of 5 degrees from 0 to 20. Figure 7 shows the computed joint angle as a function of angle from which the rulers are observed (on the x-axis) and the degrees of articulation.

C. Hinge structure test

The third experiment is to do pose estimation with markers on a hinge structure, an initial exploration to estimate pose of a non-planar object. We put marker $1, 2$ in a rigid plane, marker $3, 4$ move in another rigid plane. These two planes can have a hinge angle α shown in Figure 8.

First, we keep the hinge structure planar and calibrate $H(\theta), \text{off}, \boldsymbol{\beta}, \boldsymbol{D}$. Then, we change α with 0, 15 and 30 degree, and rotate the structure around the y-axis from -60 to 60 degree with 10 degree increment. In each frame we solve for $\boldsymbol{R}, \boldsymbol{T}, \alpha$ during the single image pose estimation.

VII. DISCUSSION

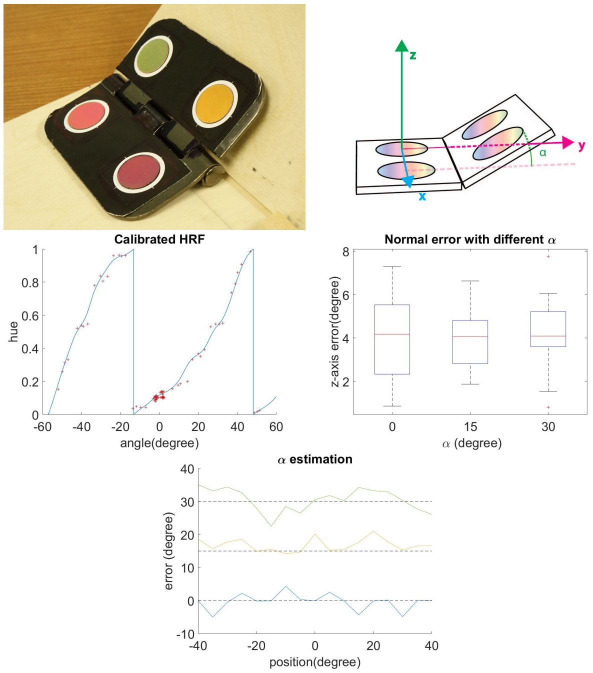

Fig. 8. Top row: the layout and marker's local configuration. The middle left figure is the calibrated HRF. The middle right figure is normal error of the rotation estimation with hinge angle 0, 15, 30 degree. The bottom figure is the α estimation in different pose

We offer an initial exploration of algorithms to self-calibrate a collection of chromo-coded fiducial markers. These markers offer an additional geometric constraint beyond the position where they are seen, because their color depends on their angle of view. Challenges in manufacturing chromo-coded markers require them to be individually calibrated, but we found that a simple offset sufficed to capture changes from one marker to the next. With this calibration the additional constraint that comes from color makes it possible to design simple algorithms to self-calibrate the relative position of multiple markers, and to use them for pose-estimation of single rigid objects, or articulated objects that have multiple rigid parts. Included in the supplemental video is an example of this articulated pose estimation on a longer video; some frames of this video are shown in Figure 9.

REFERENCES

[1] B. Atcheson, F. Heide, and W. Heidrich. Caltag: High precision fiducial markers for camera calibration. In *International Workshop on Vision, Modeling and Visualization*, volume 10, pages 41–48. Citeseer, 2010.

[2] A. M. Bruckstein, R. J. Holt, T. S. Huang, and A. N. Netravali. New devices for 3d pose estimation: Mantis eyes, agam paintings, sundials, and other space fiducials. *International Journal of Computer Vision*, 39(2):131–139, 2000.

[3] M. Fiala. Artag, a fiducial marker system using digital techniques. 2005.

[4] M. Fiala. Designing highly reliable fiducial markers. 32(7):1317–1324, July 2010.

[5] R. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2003.

[6] H. Kato and M. Billinghurst. Marker tracking and hmd calibration for a video-based augmented reality conferencing system. In *IEEE and ACM International Workshop on Augmented Reality*, 1999.

[7] E. J. Larsen. Tracking head position and orientation. US Patent 8922644, 2014.

[8] A. Mohan, G. Woo, S. Hiura, Q. Smithwick, and R. Raskar. Bokode: imperceptible visual tags for camera based interaction from a distance. *ACM Transactions on Graphics (TOG)*, 28(3):98, 2009.

[9] I. Schillebeeckx, J. Little, B. Kelly, and R. Pless. The geometry of colorful, lenticular fiducial markers. In *International Conference on 3D Vision*, 2015.

[10] I. Schillebeeckx, J. Little, and R. Pless. The geometry of colorful, lenticular fiducial markers. In *Proc. IEEE International Conference on 3D Vision (3DV)*. IEEE, 2015.

[11] I. Schillebeeckx and R. Pless. Structured light field design for correspondence free rotation estimation. In *International Conference on Computational Photography*, 2015.

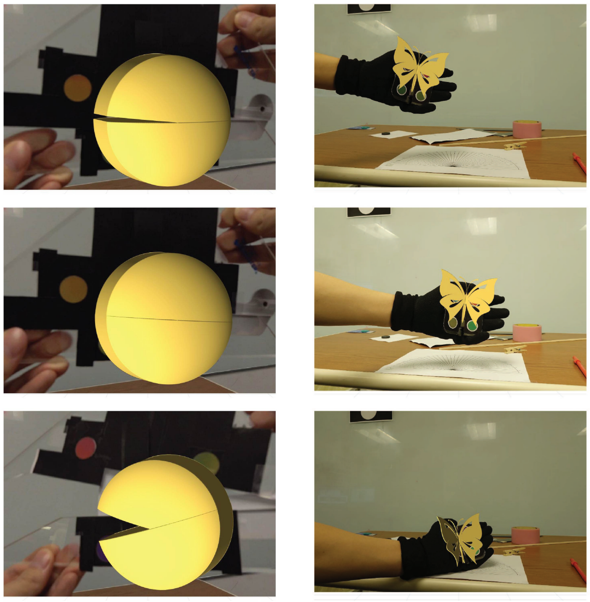

Fig. 9. AR effect: 3D Pac-man and 3D butterfly. Please see the supplemental video.

[12] I. Schillebeeckx and R. Pless. Single Image Camera Calibration with Lenticular Arrays for Augmented Reality. *IEEE Conference on Computer Vision and Camera Calibration*, 2016.

[13] H. Tanaka, Y. Sumi, and Y. Matsumoto. A high-accuracy visual marker based on a microlens array. In *IEEE/RSJ International Conference on Intelligent Robotics and Systems*, 2012.

[14] H. Tanaka, Y. Sumi, and Y. Matsumoto. A visual marker for precise pose estimation based on lenticular lenses. In *IEEE International Conference on Robotics and Automation*, 2012.

[15] H. Tanaka, Y. Sumi, and Y. Matsumoto. Further stabilization of a microlens-array-based fiducial marker. In *IEEE International Symposium on Mixed and Augmented Reality*, 2013.

[16] H. Tanaka, Y. Sumi, and Y. Matsumoto. A solution to pose ambiguity of visual markers using moire patterns. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2014.

[17] A. van Rhijn and J. D. Mulder. Optical tracking and calibration of tangible interaction devices. pages 41–50, 2005.