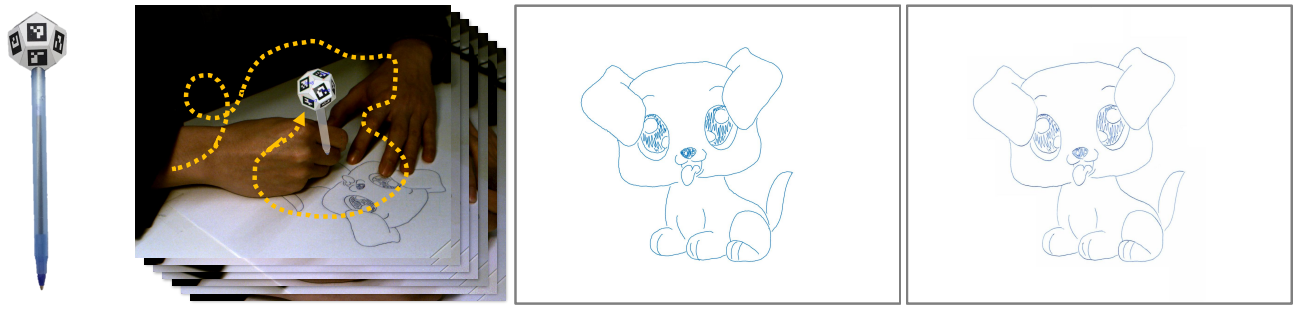
DodecaPen: Accurate 6DoF Tracking of a Passive Stylus

Po-Chen Wu[*†] **Robert Wang**[†] **Kenrick Kin**[†] **Christopher Twigg**[†] **Shangchen Han**[†]
Ming-Hsuan Yang[‡] **Shao-Yi Chien**[*]

[*]Media IC & System Lab, National Taiwan University [†]Oculus Research, Facebook Inc.
[‡]Vision and Learning Lab, University of California at Merced

[*]pcwu@media.ee.ntu.edu.tw [†]{rob.wang, kenrick.kin, chris.twigg, shangchen.han}@oculus.com
[‡]mhyang@ucmerced.edu [*]sychien@ntu.edu.tw



(a) DodecaPen (b) Monocular video (c) DodecaPen tracking (d) Ground-truth scan

Figure 1. Our proposed system can track the 6DoF pose of (a) a calibrated pen (the DodecaPen) from (b) a single camera with submillimeter accuracy. We show (c) a digital 2D drawing as the visualization of the tracking result, and compare with (d) a scan of the actual drawing.

ABSTRACT

We propose a system for real-time six degrees of freedom (6DoF) tracking of a passive stylus that achieves sub-millimeter accuracy, which is suitable for writing or drawing in mixed reality applications. Our system is particularly easy to implement, requiring only a monocular camera, a 3D printed dodecahedron, and hand-glued binary square markers. The accuracy and performance we achieve are due to model-based tracking using a calibrated model and a combination of sparse pose estimation and dense alignment. We demonstrate the system performance in terms of speed and accuracy on a number of synthetic and real datasets, showing that it can be competitive with state-of-the-art multi-camera motion capture systems. We also demonstrate several applications of the technology ranging from 2D and 3D drawing in VR to general object manipulation and board games.

ACM Classification Keywords

H.5.m. Information Interfaces and Presentation (e.g. HCI): Miscellaneous

UIST 2017, October 22–25, 2017, Quebec City, QC, Canada

 ISBN 978-1-4503-4981-9/17/10…$15.00

DOI: https://doi.org/10.1145/3126594.3126664

Author Keywords

6DoF pose tracking; binary square markers; mixed reality.

INTRODUCTION

Real-time six degrees of freedom (6DoF) tracking of a controller or stylus is the basis for interaction in many virtual and augmented reality systems with applications in gaming, digital 2D and 3D drawing, and general 3D manipulation. Current techniques typically combine a variety of electronic sensing components (inertial measurement, magnetometer, cameras, LED markers, laser scanning) to track a 6DoF controller with high accuracy and low latency. In this paper, we explore a simpler hardware setup that uses a minimal amount of electronics to achieve high accuracy tracking. We propose a system that requires only a single off-the-shelf camera and a passive 3D-printed fiducial with several hand-glued binary square markers printed from a laser printer, as shown in Figure 1. We show that off-the-shelf fiducial tracking with markers is insufficient for achieving the accuracy necessary for digital 2D drawing. Instead, our system consists of the following components:

- A 3D printed dodecahedron with hand-glued binary square markers mechanically designed for pose estimation,
- A one-time calibration procedure for the (imprecise) model using bundle adjustment,
- Approximate pose estimation from fiducial corners,
- Inter-frame fiducial corner tracking, and
- Dense pose refinement by direct model-image alignment.

We show that each step of the above system is essential to robust tracking and that the combined system allows us to achieve an absolute accuracy of 0.4mm from a single camera, which is comparable to state-of-the-art professional motion capture (mocap) systems. We rigorously evaluate the performance of the proposed system when we degrade the camera (with shot noise, spatial blur, and reduced spatial resolution). We conclude with demonstrations of this accurate and easy-to-setup 6DoF status tracking system for the application of drawing in 2D and 3D as well as object manipulation in a virtual reality (VR) environment.

RELATED WORK

5DoF and 6DoF tracking of pens have been an active area of research in the computer vision and human computer interaction communities. The IrCube [17] and IrPen [15] trackers rely on setting up a source localization problem involving a cluster of directed LEDs, achieving an accuracy of 10mm in a 20cm $\times$ 20cm area. The Lumitrack approach [34] uses laser projections of coded patterns and a linear optical sensor to track at 800Hz with an accuracy of 5mm. The Light chisel system [6] consists of two LEDs inside a diffuse cylinder fiducial tracked by stereo cameras at an accuracy of 2mm over a 56cm $\times$ 31cm $\times$ 33cm volume. A pen can also be tracked from a light-field camera [31] through a lenslet array with an accuracy of 3mm. Consumer solutions for 6DoF tracking typically combine micro-electromechanical systems (MEMS) inertial measurement with laser positioning [18], optical tracking [24, 29], or magnetic tracking [28].

Our proposed system has the distinct advantage of ease-of-construction and setup over electronically instrumented solutions. Because there are no electronics (including LEDs) on the stylus, threading wires or charging batteries are not a concern. Neither lasers nor active illumination is required. The only requirements are the use of a 2D office printer, a 3D printer, some glue, and a global shutter camera. Because we need only a single camera, it can be mounted casually on a tripod placed on the user's desk, without concern for recalibration of multiple cameras. Despite these constraints, we achieve an accuracy of 0.4mm at 60Hz over a 30cm $\times$ 40cm working area.

The easiest-to-construct and most popular 6DoF tracking solution has been the 2D binary square fiducial marker, which has been used extensively for both recognition and tracking. Libraries for efficient identification and localization of binary square markers have become a building block for many AR solutions [11, 12]. The typical output of such a library is a sparse set of corresponded corners on the recognized marker, which can then be used to solve for 6DoF position and orientation by Perspective-n-Point (PnP) algorithms [27].

We show that 6DoF tracking from sparse constraints from a binary square marker alone does not deliver sufficient accuracy or robustness for tracking a pen for writing and drawing. To improve the tracking accuracy, we resort to dense alignment methods, which directly optimize the match between the 3D model of the DodecaPen and the image pixels. As each binary square marker has many sharp edges and corners, it is well-suited for providing a precise pose using dense alignment methods. Crivellaro and colleagues [8] provide an excellent summary of modern techniques in this area. We use the forward-additive approach of the Lucas and Kanade (LK) method [20] with a backtracking line search scheme [23] to perform dense pose refinement.

Dense alignment has been successfully applied before to 6DoF tracking of rigid objects or scenes. Pauwels *et al*. [26] use a stereo camera to compute 3D points coordinates of the scene and densely aligns these 3D points between the model and the scene with the iterative closest point (ICP) method [4]. The DTAM [22] algorithm uses a forward compositional variant of LK for camera pose estimation, which can be shown to be equivalent to the forward-additive approach at first order [3].

Motion capture [21] is another widely used method for high-fidelity 6DoF tracking. Typically a large array of strobing cameras observes a set of passive retroreflective fiducials. Triangulation and tracking are used to obtain the absolute position and orientation of the tracked object at better than millimeter accuracy. We validate against and show comparable performance with a state-of-the-art 16-camera motion capture [21] solution for the task of 6DoF tracking of a pen for drawing.

PROBLEM FORMULATION

Given a target object O_t (the DodecaPen in this work) represented by a dense surface model (triangle mesh) and a camera image I_c, the task is to determine the 6DoF object pose $\mathbf{p}$ of O_t relative to the camera. Let $\mathbf{x}_i = [x_i, y_i, z_i]^\top, i = 1, \ldots, n, n \geq 3$ be a set of reference points in the local object-space of O_t, and let $\mathbf{u}_i = [u_i, v_i]^\top$ be the corresponding 2D image-space coordinates of I_c. The relationship between them can be obtained using camera projection,

$$\mathbf{u}_i(\mathbf{p}) \equiv \mathbf{u}_i(\mathbf{R}, \mathbf{t}) = \text{Proj}\left([\mathbf{R}|\mathbf{t}] \begin{bmatrix} x_i \\ y_i \\ z_i \\ 1 \end{bmatrix} \right), \qquad (1)$$

where $\mathbf{R} \in SO(3)$ and $\mathbf{t} \in \mathbb{R}^3$ are the object rotation matrix and translation vector, respectively. In this work, the pose $\mathbf{p}$ is formulated as a 6D vector consisting of the 3D axis-angle representation of $\mathbf{R}$ and the 3D translation vector $\mathbf{t}$. In (1), the $\text{Proj}(\cdot)$ is the known projection operator for the camera.

PROPOSED APPROACH

The proposed 6DoF pose tracking system comprises two phases: approximate pose estimation (APE) and dense pose refinement (DPR) (Figure 2). Once we have computed the 6DoF pose of the dodecahedron, we can recover the pen-tip trajectory and use it to reconstruct the drawing.

Dodecahedron Design

Although binary square fiducial markers are commonly attached to cubes [9, 14], pose recovery can fail when only a single marker is visible due to an ambiguity in the PnP problem [32]. By substituting a dodecahedron as the tracked object, we ensure that at least two planes are visible in most cases, eliminating the ambiguity. We used an off-the-shelf 3D printer

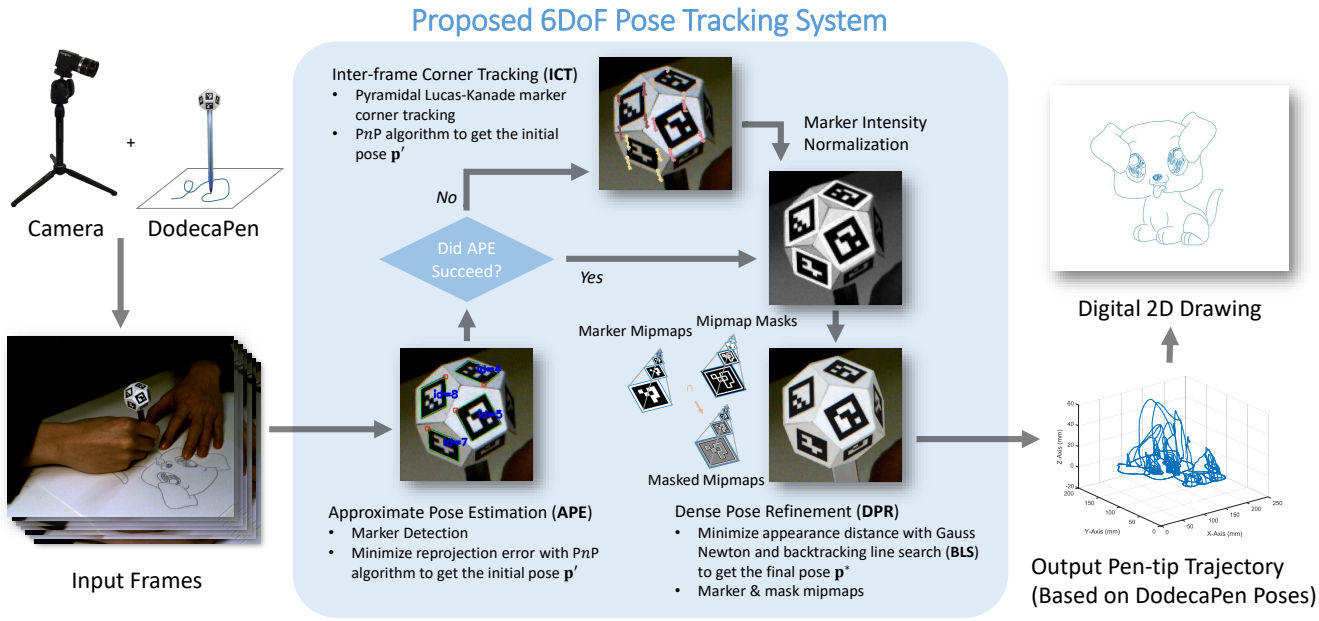


Figure 2. System overview. In the *approximate pose estimation* step, we detect the binary square fiducial markers in the input images, and estimate the 6DoF pose of the DodecaPen using the P*n*P algorithm. If fewer than two markers are detected, we use the LK method to track marker corners between frames. In the *dense pose refinement* step, the pose $\mathbf{p}'$ is refined by minimizing the appearance distance between the 3D model of the DodecaPen and image pixels to get the final pose $\mathbf{p}^*$. We generate the pen-tip trajectory in the 3D view from the computed 6DoF pose sequence, and visualize the 2D drawing by removing points where the pen tip is lifted off the page.

to create our trackable dodecahedron. Each edge of the resulting dodecahedron is 12.9 mm in length, while the markers glued on its surface have edges of length 10.8 mm and are printed with a laser printer. Each marker is generated with the ArUco library [12] and is encoded as a 6×6 grid where the external cells are set as black.

Approximate Pose Estimation

We first use the binary square fiducial marker detector provided in the ArUco library [11] to detect markers in input images. This gives us an image-space position and orientation of each marker on the dodecahedron. We use these to recover the 6DoF dodecahedron pose $\mathbf{p}$ by minimizing the *reprojection error*, the ℓ^2 difference between the projected object points $\mathbf{u}_i(\mathbf{p})$ and the observed image points $\hat{\mathbf{u}}_i$,

$$E_r(\mathbf{p}) = \frac{1}{n}\sum_{i=1}^{n} (\hat{\mathbf{u}}_i - \mathbf{u}_i(\mathbf{p}))^2 . \qquad (2)$$

This is a standard P*n*P problem, which we minimize using the Levenberg-Marquardt method [27]. To accelerate the marker detection process, we use a constant acceleration motion model to predict the dodecahedron pose and constrain ArUco's search region for the fiducial markers if the pose was successfully recovered in the frame. The predicted pose $\tilde{\mathbf{p}}_t$ in the current frame t is computed with the information from the last frame $t-1$,

$$\tilde{\mathbf{p}}_t = \mathbf{p}_{t-1} + \dot{\mathbf{p}}_{t-1} + \frac{1}{2}\ddot{\mathbf{p}}_{t-1}, \qquad (3)$$

where $\dot{\mathbf{p}}$ and $\ddot{\mathbf{p}}$ are the pose velocity and acceleration between frames, respectively. The search region for the current frame is set to be four times the area of the dodecahedron in the last frame to account for fast motion.

Inter-frame Corner Tracking

We occasionally find that the APE method fails due to motion blur or because most of the markers are strongly tilted relative to the camera. Because P*n*P cannot work reliably in the case where we detect fewer than two markers, we apply the inter-frame corner tracking (ICT) scheme to generate more constraints for P*n*P. We use the pyramidal LK optical flow tracker [5] to track the corners of the markers from frame to frame.

Square markers can be challenging for optical flow algorithms because different corners have a very similar appearance, and thus the pyramidal LK implementation frequently finds incorrect correspondences. Therefore, we perform the tracking in two rounds. In the first round, we track each visible marker separately in the camera frame and compute the velocity vectors of each marker by differencing with the previous frame. We reject markers whose velocity is further than three standard deviations from the mean. We then initialize the marker corner tracker using the trusted predictions from the first round and run the tracking for the four corners of each remaining marker a second time with similar outlier removal strategy. The resulting motion tracks are much more reliable.

Dense Pose Refinement

Unfortunately, the initial pose $\mathbf{p}'$ computed using P*n*P is too jittery to use in tracking the pen tip. We can substantially improve the pose accuracy using a *dense alignment*, minimizing the appearance distance between the image I_c and the object

O_t pixels across all of the visible marker points $\mathbf{x}_i$,

$$E_a(\mathbf{p}) = \frac{1}{n}\sum_{i=1}^{n}\left(I_c\left(\mathbf{u}_i(\mathbf{p})\right) - O_t\left(\mathbf{x}_i\right)\right)^2. \quad (4)$$

We solve this nonlinear least squares problem using Gauss-Newton iteration; to approximate how the image changes with respect to pose, we approximate it using a first-order Taylor series as follows,

$$\begin{aligned}\Delta\mathbf{p}^* &= \underset{\Delta\mathbf{p}}{\operatorname{argmin}}\frac{1}{n}\sum_{i=1}^{n}\left(I_c\left(\mathbf{u}_i\left(\mathbf{p}'+\Delta\mathbf{p}\right)\right) - O_t\left(\mathbf{x}_i\right)\right)^2 \\ &\approx \underset{\Delta\mathbf{p}}{\operatorname{argmin}}\frac{1}{n}\sum_{i=1}^{n}\left(I_c\left(\mathbf{u}_i\left(\mathbf{p}'\right)\right) + \left.\frac{\partial I_c}{\partial\mathbf{p}}\right|_{\mathbf{p}=\mathbf{p}'}\Delta\mathbf{p} - O_t\left(\mathbf{x}_i\right)\right)^2. \quad (5)\end{aligned}$$

To solve for $\Delta\mathbf{p}$ in each iteration, we set the first derivative of (5) equal to zero, and solve the resulting system of linear equations,

$$\mathbf{J}_c\Delta\mathbf{p} = \mathbf{O}_t - \mathbf{I}_c, \quad (6)$$

where $\mathbf{O}_t$ and $\mathbf{I}_c$ are vector forms of $O_t\left(\mathbf{x}_i\right)$ and $I_c\left(\mathbf{u}_i\right)$, respectively, and $\mathbf{J}_c$ is the Jacobian matrix of $\mathbf{I}_c$ with respect to $\mathbf{p}$ and is computed by the chain rule. We use the QR decomposition to solve (6).

Because our least squares problem is nonlinear, Gauss-Newton iteration does not always converge with a fixed step size. We thus perform a backtracking line search to scale the step size after each iteration of solving (6). We shrink $\Delta\mathbf{p}$ by $\Delta\mathbf{p} \leftarrow \alpha\Delta\mathbf{p}$ until it meets the Armijo-Goldstein condition below,

$$E_a(\mathbf{p}+\Delta\mathbf{p}) \le E_a(\mathbf{p}) + c\nabla E_a(\mathbf{p})^\top\Delta\mathbf{p}, \quad (7)$$

where $\nabla E_a(\mathbf{p})$ is the local function gradient. We set $\alpha = 0.5$ and $c = 10^{-4}$ empirically.

To ensure intensity invariance and to minimize the residual between the model and image, we normalize the intensity first before solving the dense alignment problem above. We observe that the primary variation in intensity is due to the normal direction of each plane (and marker) as shown in Figure 2. Therefore we normalize the intensity per local marker.

To avoid aliasing effects, we also need to ensure that the model fiducial markers are resampled to be the same size they appear in the image. We generate a mipmap of the binary square fiducial markers ahead of time to enable efficient sampling of the model points at approximately the same scale as the image.

There are large portions of the square marker that do not significantly contribute to the error term, notably in regions of uniform intensity where $\nabla I_c(\mathbf{u}_i) = 0$ and thus, $\frac{\partial I_c}{\partial\mathbf{p}} = 0$. We take advantage of this by selectively masking out flat regions ahead of time on our marker as shown in Figure 2, dropping regions where $\nabla O_t\left(\mathbf{x}_i\right) = 0$ and hence $\nabla I_c(\mathbf{u}_i)$ is likely to be zero as well. The white and black colors of the masks in Figure 2 represent the active and non-active regions, respectively. The gray color of the final masked markers represent the non-active regions. We show that we can significantly accelerate the algorithm without compromising tracking quality using this masking technique.

Dodecahedron Calibration

While square markers are easy to print and glue on to the dodecahedron, the manual nature of this process necessarily results in the model error, leading to inaccurate pose tracking results, as we show in Section 5. We perform dodecahedron calibration (DC) to determine the precise pose of each marker with respect to the dodecahedron $\mathbf{p}_j$. We first take several dodecahedron photos (24 in this work) and apply a one-time offline bundle adjustment, by minimizing the following cost function,

$$E_a(\{\mathbf{p}_j, \mathbf{p}_k\}) = \sum_i\sum_j\sum_k\left(I_c\left(\mathbf{u}_i\left(\mathbf{p}_j;\mathbf{p}_k\right)\right) - O_t\left(\mathbf{x}_i\right)\right)^2, \quad (8)$$

with respect to both marker poses $\mathbf{p}_j$ and dodecahedron poses with respect to the camera $\mathbf{p}_k$. Because the problem is ill-posed, we fix one of the marker poses and adjust other marker and dodecahedron poses simultaneously using Gauss-Newton iteration, similarly to how we solved (6) in Section 4.4. We initialize the marker poses $\mathbf{p}_j$ to their ideal positions on the dodecahedron, and we initialize the camera poses $\mathbf{p}_k$ with the APE approach.

Pen-tip Calibration

To recover a drawing, we need to know the position of the pen tip. Since the pen tip is a ball, we calibrate the position of the sphere center $\mathbf{c} = [x_c, y_c, z_c]^\top$ with respect to the coordinate frame of the dodecahedron. Given the 6DoF pose of the dodecahedron, we can get the world position of the pen tip (i.e., the ball center) $\mathbf{c}' = [x_c', y_c', z_c']^\top = \mathbf{R}\mathbf{c}+\mathbf{t}$, where $(\mathbf{R},\mathbf{t})$ is the pose of the dodecahedron. Finally, we can check if the distance between the pen-tip sphere center and the paper surface is less than the radius of the pen ball at runtime to determine if the pen is drawing.

To calibrate the position of the pen tip $\mathbf{c}$, we press the pen tip against a surface to keep it fixed, while moving the dodecahedron. We track the dodecahedron to obtain a number of its poses $(\mathbf{R}_k, \mathbf{t}_k)$, where $k \in [1, m]$. Since the pen-tip center is fixed in world space, we can write the equation $\mathbf{R}_{k_1}\mathbf{c}+\mathbf{t}_{k_1} = \mathbf{R}_{k_2}\mathbf{c}+\mathbf{t}_{k_2}$ for all k_1 and k_2. From m poses, we can obtain $\frac{m(m-1)}{2}$ linear equations, which can be solved to obtain the least squares estimate of the pen-tip position $\mathbf{c}$.

EXPERIMENTAL RESULTS

We evaluate the proposed method for the 6DoF DodecaPen pose tracking using both synthetic and real datasets, and compare it with an OptiTrack [21] motion caption system. Our system is run on a desktop computer with a 3.6 GHz CPU and 32 GB RAM. We use a Point Grey Flea3 1.3 MP color camera (60Hz, 1280×1024) with a Fujinon 12.5mm f/1.4 lens for an effective horizontal field of view 60 degrees.

Given the ground-truth rotation matrix $\hat{\mathbf{R}}$ and translation vector $\hat{\mathbf{t}}$, we compute the rotational error of the estimated rotation matrix $\mathbf{R}$ by $E_{\mathbf{R}}(^\circ) = acosd((\mathrm{Tr}(\mathbf{R}^\top\cdot\hat{\mathbf{R}})-1)/2)$, where $acosd(\cdot)$ represents the arc-cosine operation in degrees. The translation error of the estimated translation vector $\mathbf{t}$ is measured by the

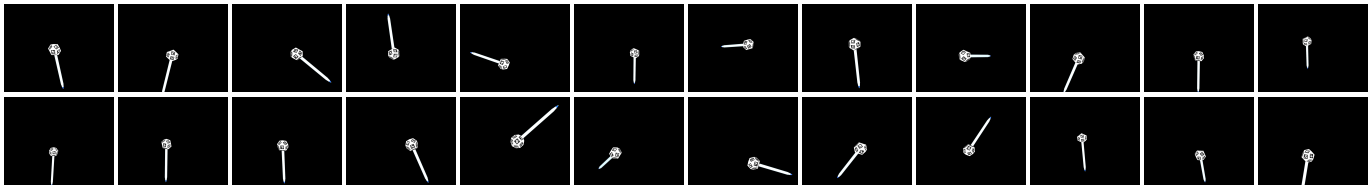


Figure 3. We generate synthetic image sequences with 24 motion patterns of the virtual DodecaPen for evaluation.

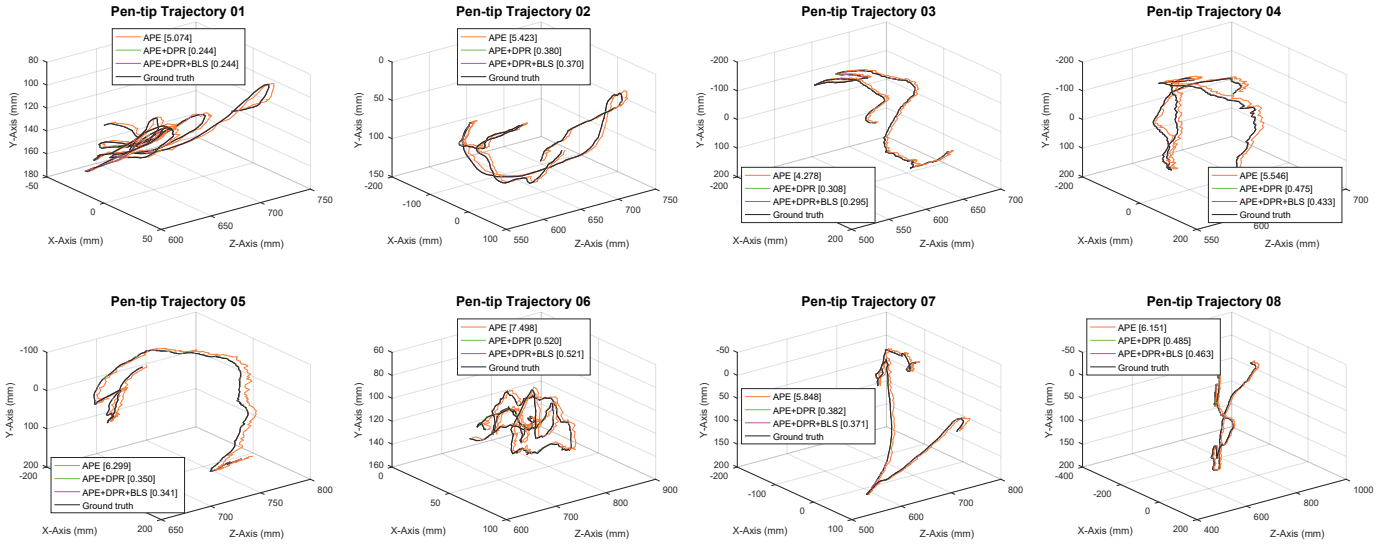


Figure 4. Pen-tip trajectories generated by different approaches. Average pen-tip errors (mm) **are shown in legends.**

ℓ^2 difference between $\hat{\mathbf{t}}$ and $\mathbf{t}$ defined as $E_{\mathbf{t}}(\text{mm}) = \|\hat{\mathbf{t}} - \mathbf{t}\|$. The pen-tip error E_{pen} is the ℓ^2 difference between two pen-tip positions transformed with either $(\mathbf{R}, \mathbf{t})$ or $(\hat{\mathbf{R}}, \hat{\mathbf{t}})$ in the camera coordinate system. The distance between the pen tip and the dodecahedron center is 143mm. The success rate (SR) is defined as the percentage of the successfully estimated poses within each sequence.

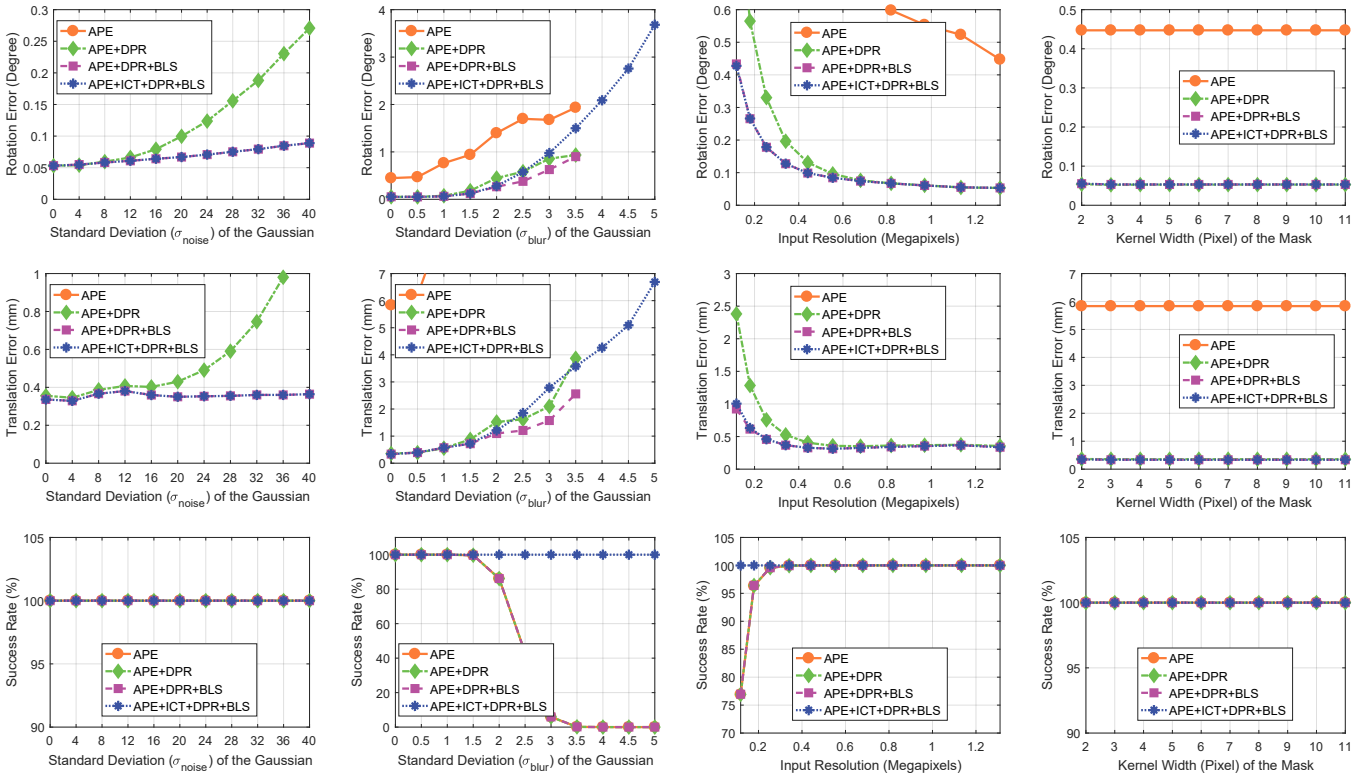
Approach	$E_{\mathbf{R}}$	$E_{\mathbf{t}}$	E_{pen}	Time	#Iter.
APE	0.447	5.835	5.854	1.100	–
APE+DPR	0.053	0.356	0.401	6.213	6.178
APE+DPR+BLS	0.053	0.336	0.386	6.140	3.834

Table 1. Evaluation results for different approaches on the synthetic dataset in terms of average rotation error $E_{\mathbf{R}}$ (°), translation error $E_{\mathbf{t}}$ (mm)**, pen-tip error E_{pen}** (mm)**, and runtimes per frame** (ms)**. The last column shows the average number of iterations for the DPR approach.**

Synthetic Data

We construct a synthetic dataset by generating 24 image sequences with different motion patterns of the virtual DodecaPen, as shown in Figure 3. The 6DoF DodecaPen pose sequence in each image sequence is obtained by recording poses of a rigid body with the OptiTrack motion capture system. Each sequence consists of 301 frames with the same resolution and intrinsics as our real camera. We initialize each tracking algorithm with the ground-truth pose for the first frame. Table 1 shows that the dense pose refinement (DPR) approach can achieve a significantly better accuracy than the approximate pose estimation (APE) approach from sparse constraints alone. With a backtracking linear search (BLS) scheme, we can take fewer Gauss-Newton iterations during the optimization process and also achieve more accurate results compared to not using a line search. It is also notable that even though the pen tip is far from the dodecahedron center, the pen-tip error is dominated by the translation error. We select some pen-tip trajectories generated by approaches in Figure 4 and compare them with ground-truth. The trajectories generated from the APE approach alone is visibly jittery, while those generated by approaches using the DPR approach are more stable and numerically closer to ground-truth. The average number of pixels (without considering masking) for the markers on the DodecaPen is 6136 over all of the sequences in the synthetic dataset.

We further evaluate the proposed approaches under varying shot noise, spatial blur, camera resolutions, and mask kernel widths to evaluate the sensitivity of the system to the most common types of degradation to allow practitioners to evaluate the feasibility of this system. There are several observations to note in the results in Figure 5. First, when the input frames are degraded with shot noise, the tracking results without the BLS scheme degrade more rapidly than those with it. We also find that sufficient shot noise can prevent direct alignment from converging without the line search. The BLS scheme is particularly effective when the small residual approximation of Gauss-Newton breaks down with noise.



(a) Shot noise (b) Spatial blur (c) Camera resolution (d) Mask kernel width

Figure 5. Experimental results on synthetic dataset under four various conditions with different degradation levels. Note that the standard deviation of the Gaussian shot noise is set for an intensity range of 0 to 255. Spatial blur sigma is in pixels for a 1280 $\times$ 1024 image.

Second, although the ArUco marker detector can detect markers well for images corrupted with high shot noise, it quickly fails for spatially blurry images. Hence, tracking success rate drops dramatically with spatial blur. In contrast, by adding the inter-frame corner tracking (ICT) scheme, our pose estimation can be quite robust (in terms of tracking success rate) to spatial blur, although accuracy suffers.

Third, the proposed approach still performs favorably even with VGA resolution sensors (i.e., 0.3 megapixels) while the execution time is reduced to 3.0 ms.

Finally, the accuracy seems to be empirically unaffected by different sizes of the mask kernel even when the number of valid pixels used in dense alignment drops from 6136 to 3941.

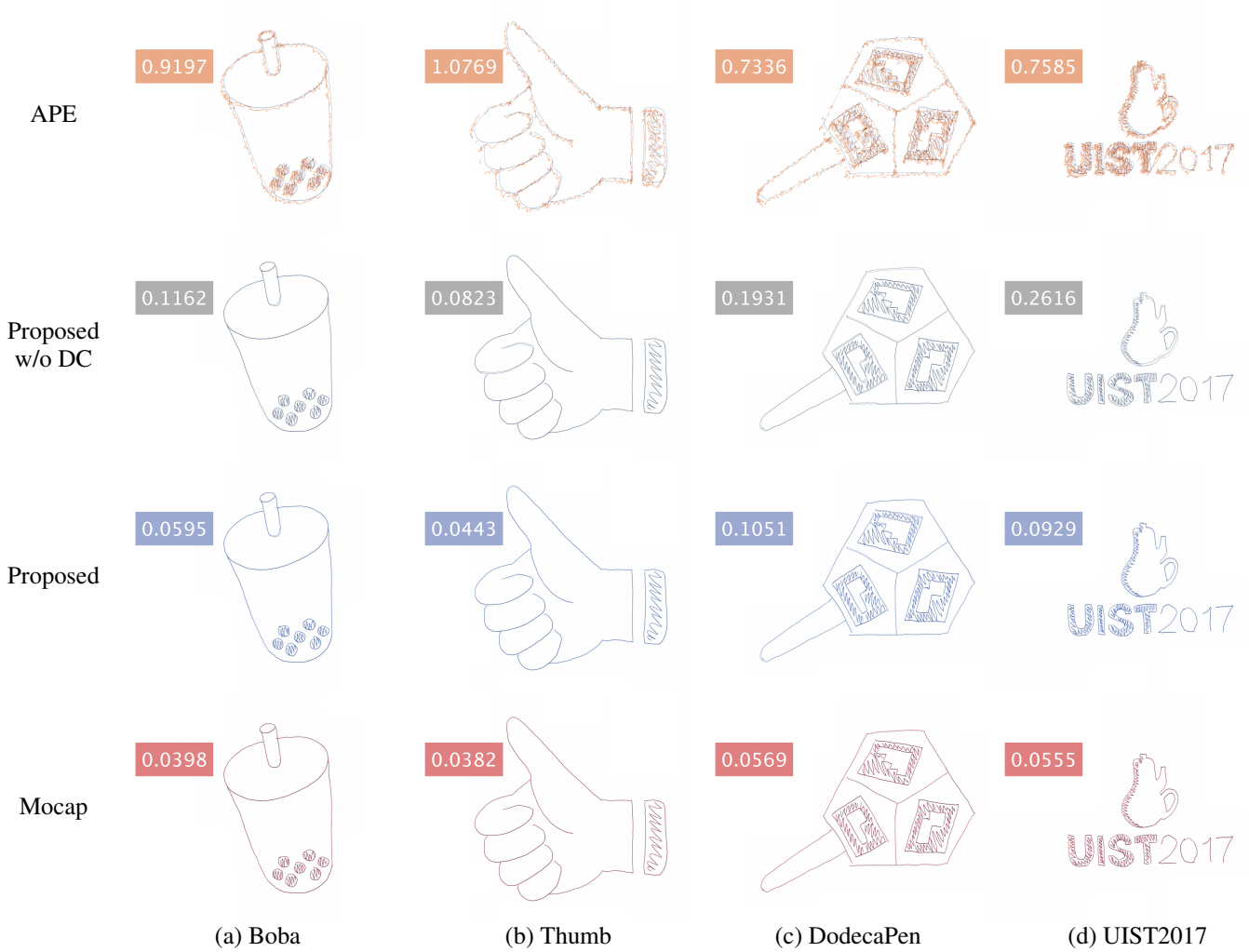
Real Data

Because the DodecaPen is an actual ball-point pen, we can evaluate the accuracy of our approach by comparing the resulting hand-drawn image and the digital 2D drawing produced by our technique. The ground-truth image (on a letter size paper) is obtained from a scanner, while the digital 2D drawing is generated by the built-in plot function in MATLAB. Both images are scaled to a resolution of 1650$\times$1275. The maximum rotation and translation speeds of the dodecahedron in the real dataset are around 80 degree/s and 200 mm/s, respectively. General drawing and writing are covered within these speeds. The relative rigid transformation between the camera and the drawing paper is resolved through calibration.

To compare the two drawings, we first binarize both drawings by Otsu's method [25] and obtain a 2D set of drawn points from each image. Next, we overlay these two binary images and find the nearest point in the other image for each point in both point sets according to their coordinates. The mean distances between each point and its nearest neighbor are regarded as the similarity metric.

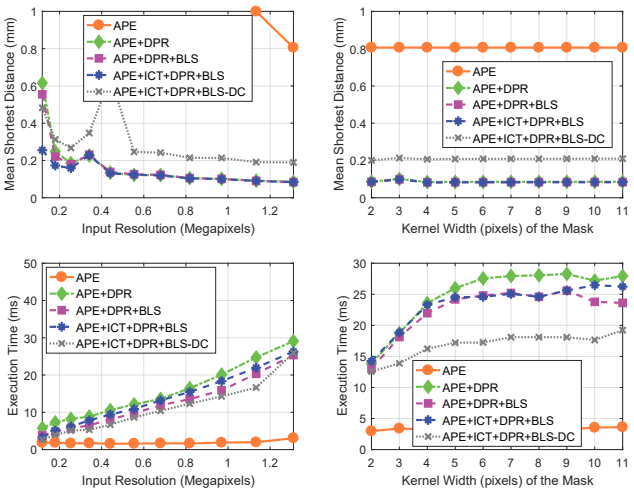
We collected four real drawings with different shapes, as shown in Figure 6. The proposed method can generate drawings virtually identical to ground-truth, while results from applying the APE approach alone are visually messy. Furthermore, without dodecahedron calibration (DC), distortions due to model error are clearly visible in the alignment with the ground truth. Figure 7 shows the accuracy and performance for various camera resolutions and mask kernel conditions. As we have already seen in Section 5.1, the proposed method can still perform well (0.5mm accuracy) even at VGA resolution. And masking does not seem to affect the tracking results, which makes it possible to run the proposed system at 60Hz by choosing the smallest mask kernel size.

In our final comparison, we compare the drawing results generated by the proposed DodecaPen system with those generated by a state-of-the-art motion capture system. The motion capture system is constructed with 16 OptiTrack Prime 17W (1.7 megapixels, 70 degrees field-of-view) cameras, as shown in Figure 8. The pen is augmented with eight more retroreflect-



(a) Boba (b) Thumb (c) DodecaPen (d) UIST2017

Figure 6. Hand-drawing results generated by different approaches. Each image is blended with the ground-truth drawing and augmented with a text box showing the mean shortest distance (in millimeters) between the generated and ground-truth drawing.



(a) Camera resolution (b) Mask kernel width

Figure 7. Experimental results on real dataset under various camera resolution and mask kernel conditions.

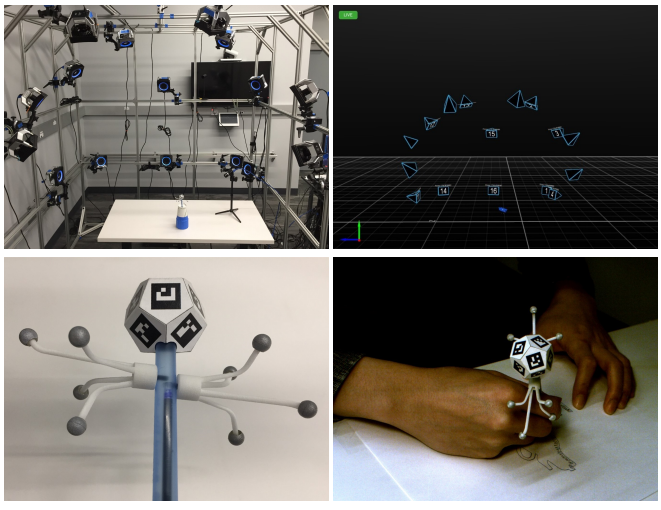


Figure 8. Experiments with OptiTrack motion capture system. Top tow: We use 16 OptiTrack cameras. Bottom row: We add eight retroreflective markers to the DodecaPen and shown a sample frame from the DodecaPen tracking camera.

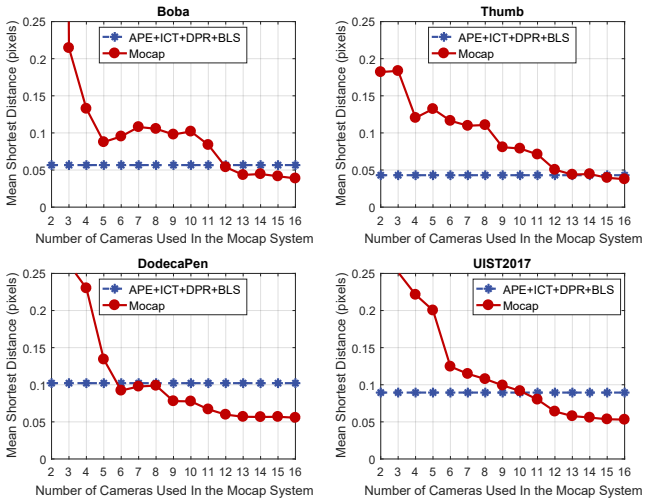


Figure 9. Experimental results of the motion capture system with different numbers of active cameras. The accuracy of the proposed method is comparable to a motion capture system with 10 active cameras.

ive balls as markers for the mocap system. After calibrating the mocap system, we record image sequences from all 16 motion capture cameras (with a combined 27MP of resolution) as well as the DodecaPen tracking camera (1.3MP) simultaneously. Because motion capture obtains the 3D position from triangulation from multiple cameras, it is interesting to see how accuracy degrades with fewer cameras. Since not every camera contributes to the pose computation on the same level, we make a best effort of selectively reducing the number of cameras in lowest priority order based on the distance to the pen as well as the percentage of the time the markers are blocked from that camera view. The results shown in Figure 9 reveal that the proposed method is comparable to a motion capture system with 10 active cameras (17MP). The drawing results generated by the mocap system with 16 cameras are also shown in Figure 6 and are virtually indistinguishable from the ground truth.

APPLICATIONS

The DodecaPen can provide low-cost writing and drawing capabilities to both 2D and 3D (e.g., VR) applications. We demonstrate both 2D and 3D drawing. Although a pen is typically used for writing and drawing, the pen (via the dodecahedron) can also serve as a handheld proxy for 3D objects.

2D Drawing

Our system can turn any flat surface into a digital writing and drawing surface, such as on a desk or whiteboard, as shown in Figure 10. Although the DodecaPen requires an external camera, the pen and surface do not require any electronics found in professional graphics tablets [33] and can digitize real graphite or ink without a textured pattern [2]. With 3D tracking we can utilize the space above the writing surface and enable hover-based interactions [13] as well as multi-layer interactions [30]. Instead of using an external camera, we could embed a camera with a global shutter to our existing devices (monitors, laptops, mobile devices) and create writable surfaces on the fly.

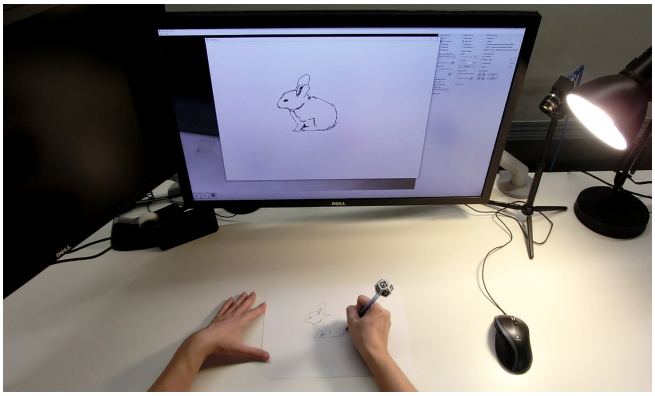
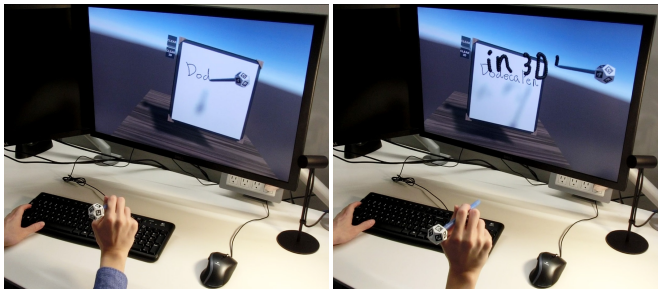


Figure 10. The DodecaPen can turn a flat surface into a digital drawing surface.



(a) Drawing on a 2D surface (b) Drawing in 3D space

Figure 11. In a VR environment, the DodecaPen can (a) draw on a midair 2D surface or (b) emit 3D ink when the spacebar is pressed.

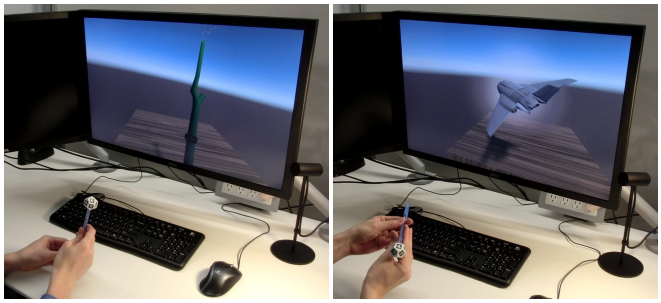
3D Drawing

In addition to drawing on a 2D surface in a 3D VR environment, we can use the DodecaPen to draw 3D curves, as shown in Figure 11. The pen can emit 3D ink for 3D annotation or be used as an instrument for content creation, such as a virtual sculpting tool [10]. For demonstration purposes, we use the spacebar to emit 3D ink, as shown in Figure 11 (b). The DodecaPen can also be used to digitize real 3D objects by specifying the 3D points of a surface (e.g., Ivan Sutherland's Volkswagen [7]) rather than scanning and then re-meshing [1].

General 6DoF Object Tracking

Although we focused on the specific application of tracking a pen, the dodecahedron can be used as a general 6DoF tracked object. We can use the dodecahedron to enable tangible input [19], either as a proxy for virtual 3D objects or to bring in other physical devices into VR. The form of the pen lends itself to represent cylindrical objects such as a VR wand or baton, as shown in Figure 12 (a). Additionally, it can represent more general objects to be inspected for educational or industrial (e.g., CAD models) purposes, as shown in Figure 12 (b). Furthermore, the proposed system can serve as a low-cost motion capture system for digital puppetry [16].

The tracked dodecahedron can be attached to physical objects other than a pen. In Figure 13 (a), we attach the dodecahedron to a physical keyboard to display in VR. The dodecahedron itself could be a tangible 12-sided VR die for use in a board game, as shown in Figure 13 (b).



(a) Cylindrical object (b) General object

Figure 12. The DodecaPen can (a) double as other cylindrical objects such as a VR wand or (b) provide general 6DoF object tracking.

(a) DodecaKeyboard (b) DodecaDie

Figure 13. The dodecahedron can (a) be attached to physical objects such as a keyboard for tracking in VR or (b) be used as a simple 12-sided VR die.

CONCLUSION

We have demonstrated a system for sub-millimeter-accurate 6DoF tracking using a set of readily available and easy-to-assemble components. Through design choices around the shape and appearance of the tracking fiducial as well as careful the application of computer vision algorithms for calibration and pose estimation, we show that single camera pose estimation can be fast enough and robust enough for drawing in 2D, 3D and in VR.

We have systematically validated each design decision of the system. We show that marker corner alignment is insufficient for robust and accurate tracking. A combination of inter-frame alignment and dense pose refinement is needed to achieve sufficient accuracy and robustness. A straightforward application of the Lucas and Kanade method is improved by adapting the step size with a backtracking line search. We show empirically that the algorithm can be accelerated by considering only the most relevant parts of the square marker for direct alignment. We also show that the bundle adjustment calibration of the handmade dodecahedron is essential and effective at correcting systematic errors in the model. Through a combination of simulation and experimentation, we characterize the system's sensitivity to shot noise, spatial blur, and image resolution to provide practitioners a useful guide for evaluating its applicability.

Limitations and Future Work

Despite the ease-of-construction and setup of our proposed system, it has some significant drawbacks. The proposed computer vision algorithm is slow by the standards of Lumitrack [34] or motion capture systems which can achieve a throughput of 300-800Hz. Because the algorithm is run on a PC, it incurs the latency of transferring the image to the host in addition to processing time. Although we show graceful degradation of the algorithm accuracy with camera resolution, the accuracy and the working volume of the system is ultimately limited by the angular resolution of the chosen camera system and the robustness of the binary square fiducial marker recognition software.

Since the tracking accuracy suffers from motion blur, we need to set the exposure time of the camera to a reasonable value for the application. From our experiments, we find that a maximum exposure time of 4ms is good for general writing or drawing. Therefore, if the imaging system is sufficiently sensitive to produce bright enough images in 4ms to detect markers, our tracking system works properly. If the input frame is too dark for the ArUco marker detector to detect markers, our system will not work. In this case, we need to either add more light or improve the imaging system (with a better sensor or a faster lens).

Our presented stylus contains no electronic components, but the proposed computer vision system can easily be augmented with buttons for discrete input and an inertial measurement unit to reduce latency and increase throughput. To simplify the VR setup, we could attach the DodecaPen camera to the headset instead of setting it on a desk, since the headset is also tracked. Although we have demonstrated that only part of the binary square fiducial marker is useful for dense alignment, we still transfer the entire image from the camera to the host. Integrating on-camera compute or new sensing modalities such as event cameras may further reduce latency and improve throughput. The proposed system cannot handle occlusion because it relies on a single camera, but occlusion can be addressed with the addition of more cameras at the cost of additional setup complexity and calibration.

ACKNOWLEDGMENTS

The work was performed during Po-Chen Wu's internship at Oculus Research. We would like to thank Joel Robinson for generating the dodecahedrons, Albert Hwang for doing the video voiceover, and Yuting Ye for her advice and support throughout this work. Ming-Hsuan Yang is supported in part by the NSF CAREER Grant #1149783.

REFERENCES

1. Pierre Alliez, David Cohen-Steiner, Olivier Devillers, Bruno Lévy, and Mathieu Desbrun. 2003. Anisotropic Polygonal Remeshing. In *Proceedings of ACM SIGGRAPH*.

2. Anoto. Accessed: 2017-08-08. *Anoto*. `http://www.anoto.com/`

3. Simon Baker and Iain Matthews. 2004. Lucas-Kanade 20 Years On: A Unifying Framework. *International Journal of Computer Vision* 56, 3 (2004), 221–255.

4. Paul J Besl and Neil D McKay. 1992. A Method for Registration of 3-D Shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 14, 2 (1992), 239–256.

5. Jean-Yves Bouguet. 2001. Pyramidal Implementation of the Lucas Kanade Feature Tracker: Description of the Algorithm. *Intel Corporation* 5, 1-10 (2001), 4.
6. Vojtech Bubník and Vlastimil Havran. 2015. Light Chisel: 6DOF Pen Tracking. *Computer Graphics Forum* 34, 2 (2015), 325–336.
7. Computer History Museum. Accessed: 2017-08-08. *Mapping Sutherland's Volkswagen*. http://www.computerhistory.org/revolution/computer-graphics-music-and-art/15/206/560
8. Alberto Crivellaro, Pascal Fua, and Vincent Lepetit. 2014. *Dense Methods for Image Alignment with an Application to 3D Tracking*. Technical Report.
9. Mark Fiala. 2005. ARTag, a Fiducial Marker System Using Digital Techniques. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*.
10. Tinsley A. Galyean and John F. Hughes. 1991. Sculpting: An Interactive Volumetric Modeling Technique. In *Proceedings of ACM SIGGRAPH*.
11. Sergio Garrido-Jurado, Rafael Muñoz-Salinas, Francisco José Madrid-Cuevas, and Manuel Jesús Marín-Jiménez. 2014. Automatic Generation and Detection of Highly Reliable Fiducial Markers Under Occlusion. *Pattern Recognition* 47, 6 (2014), 2280–2292.
12. Sergio Garrido-Jurado, Rafael Munoz-Salinas, Francisco José Madrid-Cuevas, and Rafael Medina-Carnicer. 2016. Generation of Fiducial Marker Dictionaries using Mixed Integer Linear Programming. *Pattern Recognition* 51 (2016), 481–491.
13. Tovi Grossman, Ken Hinckley, Patrick Baudisch, Maneesh Agrawala, and Ravin Balakrishnan. 2006. Hover Widgets: Using the Tracking State to Extend the Capabilities of Pen-operated Devices. In *Proceedings of ACM SIGCHI*.
14. Taejin Ha and Woontack Woo. 2010. An Empirical Evaluation of Virtual Hand Techniques for 3D Object Manipulation in a Tangible Augmented Reality Environment. In *IEEE Symposium on 3D User Interfaces*.
15. Jaehyun Han, Seongkook Heo, Hyong-Euk Lee, and Geehyuk Lee. 2014. The IrPen: A 6-DOF Pen for Interaction with Tablet Computers. *IEEE Computer Graphics and Applications* 34, 3 (2014), 22–29.
16. Robert Held, Ankit Gupta, Brian Curless, and Maneesh Agrawala. 2012. 3D Puppetry: A Kinect-based Interface for 3D Animation. In *Proceedings of ACM Symposium on User Interface Software and Technology*.
17. Seongkook Heo, Jaehyun Han, Sangwon Choi, Seunghwan Lee, Geehyuk Lee, Hyong-Euk Lee, SangHyun Kim, Won-Chul Bang, DoKyoon Kim, and ChangYeong Kim. 2011. IrCube Tracker: An Optical 6DOF Tracker based on LED Directivity. In *Proceedings of ACM Symposium on User Interface Software and Technology*.
18. HTC. Accessed: 2017-08-08. *HTC Vive*. https://www.vive.com/us/
19. Hiroshi Ishii and Brygg Ullmer. 1997. Tangible Bits: Towards Seamless Interfaces Between People, Bits and Atoms. In *Proceedings of ACM SIGCHI*.
20. Bruce D Lucas and Takeo Kanade. 1981. An Iterative Image Registration Technique with an Application to Stereo Vision. In *Proceedings of the IJCAI*.
21. NaturalPoint. Accessed: 2017-08-08. *OptiTrack*. http://optitrack.com/
22. Richard A Newcombe, Steven J Lovegrove, and Andrew J Davison. 2011. DTAM: Dense Tracking and Mapping in Real-Time. In *Proceedings of IEEE International Conference on Computer Vision*.
23. Jorge Nocedal and Stephen J Wright. 2006. Numerical Optimization. *Springer* (2006).
24. Oculus. Accessed: 2017-08-08. *Oculus Touch*. https://www.oculus.com/rift/
25. Nobuyuki Otsu. 1975. A Threshold Selection Method from Gray-Level Histograms. *Automatica* 11, 285-296 (1975), 23–27.
26. Karl Pauwels, Leonardo Rubio, Javier Diaz, and Eduardo Ros. 2013. Real-Time Model-Based Rigid Object Pose Estimation and Tracking Combining Dense and Sparse Visual Cues. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*.
27. Thomas Petersen. 2008. A Comparison of 2D-3D Pose Estimation Methods. *Aalborg University* (2008).
28. Razer. Accessed: 2017-08-08. *Razer Hydra*. https://www2.razerzone.com/au-en/gaming-controllers/razer-hydra-portal-2-bundle
29. Sony. Accessed: 2017-08-08. *PlayStation Move Motion Controller*. https://www.playstation.com/en-us/explore/accessories/playstation-move/
30. Sriram Subramanian, Dzimitry Aliakseyeu, and Andrés Lucero. 2006. Multi-layer Interaction for Digital Tables. In *Proceedings of ACM Symposium on User Interface Software and Technology*.
31. James Tompkin, Samuel Muff, James McCann, Hanspeter Pfister, Jan Kautz, Marc Alexa, and Wojciech Matusik. 2015. Joint 5D Pen Input for Light Field Displays. In *Proceedings of ACM Symposium on User Interface Software and Technology*.
32. Hung-Yu Tseng, Po-Chen Wu, Ming-Hsuan Yang, and Shao-Yi Chien. 2016. Direct 3D Pose Estimation of a Planar Target. In *Proceedings of IEEE Winter Conference on Applications of Computer Vision*.
33. Wacom. Accessed: 2017-08-08. *Wacom*. https://www.wacom.com/
34. Robert Xiao, Chris Harrison, Karl DD Willis, Ivan Poupyrev, and Scott E Hudson. 2013. Lumitrack: Low Cost, High Precision, High Speed Tracking with Projected m-Sequences. In *Proceedings of ACM Symposium on User Interface Software and Technology*.