Vision-Aided Dynamic In-Hand Localization Under Rolling Contact via Constrained Kalman Filtering

Ling Tang*[1], Xuye Wu*[2], Haniyeh Fekrmandi[2] and Yan-Bin Jia[2]

***Abstract*— Dynamic in-hand manipulation is challenging but highly useful — particularly when the manipulated object is a tool executing some task. Success of such a maneuver heavily hinges on the robotic hand's ability to constantly estimate the object's relative pose and motion, based on tactile and/or visual input, and to utilize the information instantly for control purpose. This paper presents an extended Kalman filter (EKF) that exploits the dynamics of in-hand manipulation, obeys the kinematic constraint of rolling contact, improves the object's raw pose estimate from vision, and tracks its velocity and angular velocity. Experiments with an anthropomorphic hand, equipped with a control policy, have not only achieved a high estimation accuracy but also demonstrated success in several precision-required tasks which would have failed often otherwise.**

I. Introduction

One of the long-standing challenges in grasping [1], [2] and in-hand manipulation [3]–[5] is to localize the manipulated object relative to the robotic hand, either before or during the operation. Localization is essential for manipulating a tool to execute tasks during which force control [6]–[8] — most effective at dealing with contact compliance and reacting to a constant flow of sensory data — heavily depends on tracking not only the object's pose but also its motion relative to the hand.

This paper presents an EKF for in-hand pose and motion estimation. The filter takes as input (rough) pose estimates generated using a vision-based technique and polishes them by exploiting dynamics of the task and kinematics of rolling. The estimation method bears several distinct characteristics:

1) *Dynamic estimation*. The work copes with that the object is not still but being dynamically manipulated. Thus, estimation cannot be executed as one shot but has to be performed repeatedly like tracking.
2) *Rolling contact and curved surface.* The method does away with the common assumption about invariant hand-object contact, which is too strict and whose realization was not a matter of concern in previous estimation works. Instead, we consider rolling contacts at the fingertips, which can be realized under force control [3] and experimentally demonstrated [9]. Consequently, rolling kinematics [10], [11] is incorporated into the system dynamics to design the EKF.
3) *Constrained Kalman filtering.* This is due to the use of unit quaternions to represent the object's orientation, and the constraint of point coincidence at every finger-object contact.
4) *Integration of random sample consensus (RANSAC) into filtering.* At each step of EKF, robust measurement (of the object's pose) is constructed from the image frame by RANSAC, which utilizes the prior state estimate to initialize its ICP subroutine.

II. Related Work

Early research considered model-based recognition and localization of stationary (and predominantly polyhedral) objects, often employing constraint satisfaction to find the most consistent matching between sensory data and candidate models [12]–[15]. The interpretation tree method motivated one of the earliest work on in-hand pose estimation [16]. Hypothesis-and-test was applied to localize grasped motionless objects with more complex geometry to reach a consistent hypothesis with [17] or without [18] collision checking.

Camera pose estimation, also known as calibration, is often achieved via solution of the perspective-n-point (PnP) problem [19], [20], which assumes n correspondences, each between a feature point on the same object and a point in the image plane. This typically involves depth recovery followed by the use of the iterative closest point (ICP) method [21]. A more robust approach [23] includes a preprocessing step that combines P3P with RANSAC [22] to filter out outlier points.

Object tracking and interception were demonstrated using optical flow combined with nonlinear filtering, trajectory planning, and control [24]. In-hand vision-only pose tracking was performed [25] based on dynamics by repeatedly hypothesizing trajectories (beliefs) and choosing the best one. A single RGB image could be combined with the noisy joint angles of the grasping human hand as input to a convolutional neural network (CNN) to refine the pose [26]. However, the above vision-based methods were not adequate for dexterous manipulation due to no modeling of the hand-object contact.

Despite no issue of camera occlusion, tactile sensing has long been plagued by small and low-resolution data. The interest in tactile localization has been rekindled with the appearance of GelSight [27], which allows the preconstruction of a tactile map [28] to localize an object in contact through registration [29], [30]. Not limited by GelSight, the pose could be determined using the principal component analysis

*Equal leading contributions.

[1]Ling Tang is with Amazon, Seattle, WA 98109, USA. tangling@amazon.com

[2]Xuye Wu, Haniyeh Fekrmandi, and Yan-Bin Jia are with the Department of Computer Science, Iowa State University, Ames, IA 50011, USA. wuxuye, haniyehf, jia@iastate.edu

We thank Amazon Robotics AI for its generous donation of the Shadow Dexterous Hand that was used for this research.

(PCA) to find the best matching patch on the object with the tactile data [31]. The above methods, assuming the object to be motionless while touched by the hand, cannot be directly incorporated into in-hand manipulation.

In-hand estimation of 3D or 6D pose was also carried out by fusing GelSight data and RGB images, with feature extraction by CNNs and pose estimation by extra neural nets [37], [38], either through optimization [39] or under a similarity metric [40]. Most of these methods again assumed a still object during estimation. Pose changes, if happening, were not continuous as part of a motion.

There has been a recent surge in the use of neural nets to unify visual, proprioceptive, and tactile information, and then the application of reinforcement learning to develop control policies trained on video data [32], [33], [34]. Such learning-based methods, however, are not suitable for high precision tasks due to large pose errors. For example, the orientation error was within 0.4 radian for only 45% of the test objects in [35] which, together with [36] on visuotactile grasping, cited the need for incorporating manipulator dynamics, contact modeling, and/or hardware properties to reduce the sim-to-real gap and improve performance. A tremendous amount of training time (e.g., four days) also hampers manipulation of new objects whose models and properties can be quickly acquired.

With the ability to deal with high nonlinearities, particle filtering and Bayesian methods, have been applied to tactile localization but are mostly limited to motionless objects, relative to the environment or robotic hand, due to their high computational costs. Examples include tracking of the contact state between an object and a tactile end-effector [41], estimation of the 3D pose of a fixed object [42] by repetitively refining some high probability region, and 6-DOF localization of still objects by recursively driving particles toward consistent pose regions [43].

In the case where the object moves relative to the hand, the apparatus used for estimation includes nonlinear observers [44] based on contact motions on fingertips and Kalman filters [45], [46], [47] that minimize the deviations of the predicted contacts from measurements of vision or tactile sensing.

Combinatorial contact modes consider which fingers are in contact with which vertices or facets of the object. They were estimated using a Bayesian method [48] or particle filtering [49]. In the former work, the continuous evolution of the 6D pose was tracked by an EKF based on force/torque measurements, hand joint angles, and stereo vision. In the latter work, dynamics was first approximated using a linear complementarity formulation before Kalman filtering was applied. Both approaches ignored the smooth nature of a contact movement on the manipulator by assuming the object to be either in a firm grasp or change between a small number of contact modes.

III. Dynamical System

Dynamic in-hand manipulation is most effectively carried out under a control policy that respects contact compliance, exploits force feedback, and has knowledge about the object's (instantaneous) configuration [3], [50], [51]. This section sets up a dynamical system describing the object's motion and evolution of all finger contacts, and arrives at a complete description of the system's state.

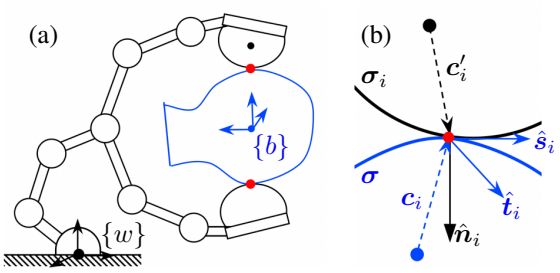


Fig. 1: (a) Dynamic manipulation by a robotic hand and arm. (b) Contact between fingertip i and the object, where $\hat{\boldsymbol{t}}_i$ and $\hat{\boldsymbol{s}}_i$ are two vectors in the tangent plane at the contact on the object, and $\hat{\boldsymbol{n}}_i$ is the contact normal on the fingertip.

A. Object Motion

As shown in Fig. 1(a), we place the world frame $\{w\}$ at the base of the robot. The object with mass m has a body frame $\{b\}$ located at its center of mass and aligned with its principal axes, so its inertia tensor I_{cm} is a diagonal matrix. Let ${}^w\boldsymbol{p}$ be the position of the object's center of mass in $\{w\}$, and R be the matrix describing the rotation from $\{w\}$ to $\{b\}$. Denote by $\boldsymbol{v}$ and $\boldsymbol{\omega}$ the object's linear and angular velocities expressed in $\{b\}$, respectively[3].

The object's dynamics is described by Newton-Euler equations in $\{b\}$ as

$$\begin{pmatrix} \boldsymbol{f}_h \\ \boldsymbol{\tau}_h \end{pmatrix} = \begin{pmatrix} mI_3 & 0 \\ 0 & I_{cm} \end{pmatrix} \begin{pmatrix} \dot{\boldsymbol{v}} \\ \dot{\boldsymbol{\omega}} \end{pmatrix} + \begin{pmatrix} -mR^\top \boldsymbol{g} \\ \boldsymbol{\omega} \times (I_{cm}\boldsymbol{\omega}) \end{pmatrix}, \quad (1)$$

where the external force $\boldsymbol{f}_h$ and torque $\boldsymbol{\tau}_h$ are applied by the hand and expressed in $\{b\}$, and the gravitational acceleration $\boldsymbol{g} = (0, 0, -9.81)^\top$. The object is being manipulated by k fingers of the hand, whose tips exert the contact forces ${}^w\boldsymbol{f}_1, \ldots, {}^w\boldsymbol{f}_k$, respectively. We have

$$\boldsymbol{f}_h = \sum_{i=0}^{k} R^\top {}^w\boldsymbol{f}_i \quad \text{and} \quad \boldsymbol{\tau}_h = \sum_{i=0}^{k} R^\top [R\boldsymbol{c}_i]_\times {}^w\boldsymbol{f}_i,$$

where $\boldsymbol{c}_i$ is the position of contact i in $\{b\}$, and the operator $[\cdot]_\times$ yields the antisymmetric matrix that turns the cross product of its vector operand with any vector into a matrix product.

B. Contact Configuration

Fig. 1(b) shows the contact between the fingertip i and the object. The object's surface is described as a patch $\boldsymbol{\sigma}$ in $\{b\}$, so that $\boldsymbol{c}_i = \boldsymbol{\sigma}(u_i, v_i)$. In the body frame of the fingertip i, its surface $\boldsymbol{\sigma}_i$ locates the same contact as $\boldsymbol{c}_i' = \boldsymbol{\sigma}_i(\xi_i, \delta_i)$. The contact is completely specified by $(u_i, v_i, \xi_i, \delta_i)$. The configuration of the k contacts is described by a total of $4k$ variables.

[3]From now on, a term with the superscript ^{w} is expressed in the world frame $\{w\}$ and one with no superscript is expressed in the body frame $\{b\}$.

We adopt a control policy from [9] to manipulate the objects in hand, so that the fingers roll on the object without slipping. The contact configuration thus evolves under the rolling kinematics [10], [11]. At the i-th contact, the two points $\boldsymbol{c}_i$ and $\boldsymbol{c}'_i$ must have the same position and velocity in $\{w\}$, namely,

$$({}^w\boldsymbol{p} + R\boldsymbol{c}_i) - ({}^w\boldsymbol{p}_i + R_i\boldsymbol{c}'_i) = \mathbf{0}_3, \quad (2)$$

$$({}^w\dot{\boldsymbol{p}} + R(\boldsymbol{\omega} \times \boldsymbol{c}_i)) - ({}^w\dot{\boldsymbol{p}}_i + R_i(\boldsymbol{\omega}_i \times \boldsymbol{c}'_i)) = \mathbf{0}_3, \quad (3)$$

where the i-th fingertip's position ${}^w\boldsymbol{p}_i$, angular velocity $\boldsymbol{\omega}_i$, and rotation matrix R_i are calculated from the kinematics, and $\mathbf{0}_3$ is a 3-vector with zero entries. Differentiating (2) with respect to time and subtracting (3) from the resulting equation, we get $R\dot{\boldsymbol{c}}_i = R_i\dot{\boldsymbol{c}}'_i$. Letting $\boldsymbol{u}_i = (u_i, v_i)^\top$ and $\boldsymbol{\xi}_i = (\xi_i, \delta_i)^\top$, we rewrite this equation as

$$\begin{pmatrix} R\frac{\partial \boldsymbol{c}_i}{\partial \boldsymbol{u}_i} & -R_i\frac{\partial \boldsymbol{c}'_i}{\partial \boldsymbol{\xi}_i} \end{pmatrix} \begin{pmatrix} \dot{\boldsymbol{u}}_i \\ \dot{\boldsymbol{\xi}}_i \end{pmatrix} = \mathbf{0}_3. \quad (4)$$

At the same contact, the tangential plane to $\boldsymbol{\sigma}$ must be perpendicular to the normal $\hat{\boldsymbol{n}}_i$ on $\boldsymbol{\sigma}_i$. Let $\hat{\boldsymbol{t}}_i$ and $\hat{\boldsymbol{s}}_i$ be the unit vectors along the directions of $\partial\boldsymbol{\sigma}/\partial u_i$ and $\partial\boldsymbol{\sigma}/\partial v_i$ respectively. We have

$$(R\hat{\boldsymbol{t}}_i) \cdot (R_i\hat{\boldsymbol{n}}_i) = 0, \quad (5)$$

$$(R\hat{\boldsymbol{s}}_i) \cdot (R_i\hat{\boldsymbol{n}}_i) = 0. \quad (6)$$

Differentiating the above equations and combining the resulting equations into one, we obtain

$$\begin{pmatrix} U_1 \\ U_2 \end{pmatrix} \begin{pmatrix} \dot{\boldsymbol{u}}_i \\ \dot{\boldsymbol{\xi}}_i \end{pmatrix} = \begin{pmatrix} W_1 \\ W_2 \end{pmatrix} \begin{pmatrix} \boldsymbol{\omega} \\ \boldsymbol{\omega}_i \end{pmatrix}, \quad (7)$$

where, from (5),

$$U_1 = \left((R_i\hat{\boldsymbol{n}}_i)^\top R\frac{\partial \hat{\boldsymbol{t}}_i}{\partial \boldsymbol{u}_i}, \quad (R\hat{\boldsymbol{t}}_i)^\top R_i\frac{\partial \hat{\boldsymbol{n}}_i}{\partial \boldsymbol{\xi}_i} \right),$$

$$W_1 = \left((R_i\hat{\boldsymbol{n}}_i)^\top [R\hat{\boldsymbol{t}}_i]_\times R, \quad (R\hat{\boldsymbol{t}}_i)^\top [R_i\hat{\boldsymbol{n}}_i]_\times R_i \right),$$

and U_2 and W_2 are similarly derived from (6). Equations (4) and (7), for $1 \le i \le k$, together describe how the grasp configuration evolves as the fingers roll on the object.

C. System Dynamics

The camera's position ${}^w\boldsymbol{p}_c$ and orientation R_c are calibrated offline, causing measurement errors that may compromise the estimation accuracy. We incorporate the camera extrinsics into the system state, so they get polished together with the measurements.

In summary, the state of the object under the manipulation of the k fingers is given by

$$\boldsymbol{x} = ({}^w\boldsymbol{p}^\top, {}^w\boldsymbol{r}^\top, \boldsymbol{v}^\top, \boldsymbol{\omega}^\top, \boldsymbol{u}_1^\top, \boldsymbol{\xi}_1^\top, \cdots, \boldsymbol{u}_k^\top, \boldsymbol{\xi}_k^\top, {}^w\boldsymbol{p}_c^\top, {}^w\boldsymbol{r}_c^\top)^\top, \quad (8)$$

where the orientations of the object and camera are compactly represented by unit quaternions ${}^w\boldsymbol{r}$ and ${}^w\boldsymbol{r}_c$. The state $\boldsymbol{x}$ has a total of $n = 20 + 4k$ variables. It evolves according to the equation

$$\dot{\boldsymbol{x}} = \boldsymbol{e}(\boldsymbol{x}), \quad (9)$$

where, among the components of $\boldsymbol{e}$,

1) ${}^w\dot{\boldsymbol{p}} = R\boldsymbol{v}$;
2) ${}^w\dot{\boldsymbol{r}} = ({}^w\boldsymbol{r} \otimes \boldsymbol{\omega})/2$;[4]
3) the accelerations $\dot{\boldsymbol{v}}$ and $\dot{\boldsymbol{\omega}}$ are obtained from (1);
4) the contact velocities $\dot{\boldsymbol{u}}_i$ and $\dot{\boldsymbol{\xi}}_i$, $1 \le i \le k$, are solved from (4) and (7);
5) ${}^w\dot{\boldsymbol{p}}_c = 0$ and ${}^w\dot{\boldsymbol{r}}_c = 0$ since the camera remains stationary.

IV. EKF With Quadratic Constraints

For the nonlinear system (9), a hybrid EKF [56, pp. 403–407] with continuous-time dynamics and discrete-time measurements can be derived below.

A. The EKF

To obtain an EKF, the dynamics and measurement model are written as below:

$$\dot{\boldsymbol{x}} = \boldsymbol{e}(\boldsymbol{x}) + \boldsymbol{w}, \quad (10)$$

$$\boldsymbol{y} = \boldsymbol{h}(\boldsymbol{x}) + \boldsymbol{z}, \quad (11)$$

where $\boldsymbol{w}$ and $\boldsymbol{z}$ are the continuous-time process noise and discrete-time measurement noise, respectively. The output $\boldsymbol{y}$, observed as $({}^c\boldsymbol{p}^\top, {}^c\boldsymbol{r}^\top)^\top$, can be described as a function of the state $\boldsymbol{x}$ below:

$$\boldsymbol{h}(\boldsymbol{x}) = \begin{pmatrix} {}^w\boldsymbol{r}_c^* \otimes ({}^w\boldsymbol{p} - {}^w\boldsymbol{p}_c) \otimes {}^w\boldsymbol{r}_c \\ {}^w\boldsymbol{r}_c^* \otimes {}^w\boldsymbol{r} \end{pmatrix}.$$

Measurement of $\boldsymbol{y}$ from range data will be described in Section IV-C.

Denote by $\tilde{\boldsymbol{x}}_l^-$ and $\tilde{\boldsymbol{x}}_l^+$ the prior (before the measurement) and posterior (after the measurement) state estimates, respectively, at time step l. The system of (10) and (11) is linearized at l with its partial derivatives, $\partial\boldsymbol{f}/\partial\boldsymbol{x}$ and $\partial\boldsymbol{y}/\partial\boldsymbol{x}$, both evaluated at the estimate $\tilde{\boldsymbol{x}}_{l-1}^+$. The EKF algorithm [56, p. 404] minimizes the trace of the expected error covariance P_l^+. It first propagates the posterior state estimate $\tilde{\boldsymbol{x}}_{l-1}^+$ to the prior estimate $\tilde{\boldsymbol{x}}_l^-$ using (10), and its covariance P_{l-1}^+ to P_l^- based on the differential Riccati equation. A new measurement $\boldsymbol{y}_l$ is then incorporated into $\tilde{\boldsymbol{x}}_l^-$ by projecting the discrepancy between $\boldsymbol{y}_l$ and the predicted value $\boldsymbol{h}(\tilde{\boldsymbol{x}}_l^-)$ into the state space through a gain matrix K. Then the expected error covariance P_l^+ is updated by adapting the priori P_l^- to this correction.

B. Constraining the Gain Matrix

When updating the system state, the object's pose, the contact configuration, and the robot kinematics must be consistent with each other. Such consistency, however, is not guaranteed due to the numerical errors introduced during the state propagation from $\tilde{\boldsymbol{x}}_{l-1}^+$ to $\tilde{\boldsymbol{x}}_l^-$, and the noises introduced in (10) and (11). Therefore, we need to enforce the contact constraints explicitly in the derivation of the EKF.

Let $\boldsymbol{d}(\boldsymbol{x})$ be a vector-valued function obtained by stacking the left hand sides of equations (2), (5) and (6), for $1 \le i \le k$, resulting a total output size of $5k$. We linearize the contact

[4] See Appendix B in [52] for a derivation.

constraints at the current prior estimate $\tilde{\boldsymbol{x}}_l^-$ (from forward propagation) as follows [57]:

$$\boldsymbol{d}(\tilde{\boldsymbol{x}}_l^-) + \left.\frac{\partial \boldsymbol{d}}{\partial \boldsymbol{x}}\right|_{\tilde{\boldsymbol{x}}_l^-} (\boldsymbol{x} - \tilde{\boldsymbol{x}}_l^-) = \boldsymbol{0}.$$

Rewriting the above vector equation and then extracting the scalar equation components, we have, for $1 \le j \le 5k$,

$$\eta_j(\boldsymbol{x}) \equiv \boldsymbol{b}_j^\top \boldsymbol{x} + d_j = 0, \tag{12}$$

where $\boldsymbol{b}_j^\top$ denotes the j-th row vector in $\partial \boldsymbol{d}/\partial \boldsymbol{x}$, and the constant scalar $d_j = [\boldsymbol{d}(\tilde{\boldsymbol{x}}_l^-)]_j - \boldsymbol{b}_j^\top \tilde{\boldsymbol{x}}_l^-$ with the operator $[\cdot]_j$ extracting the j-th element of the vector.

In addition, the two quaternions ${}^w\boldsymbol{r}$ and ${}^w\boldsymbol{r}_c$ as state variables in (8) must be unit vectors to describe the orientations of the object and camera, respectively. This adds two constraints ${}^w\boldsymbol{r}^\top {}^w\boldsymbol{r} = 1$ and ${}^w\boldsymbol{r}_c^\top {}^w\boldsymbol{r}_c = 1$. Rewriting them in the quadratic form, we obtain

$$\eta_j(\boldsymbol{x}) \equiv \boldsymbol{x}^\top A_j \boldsymbol{x} - 1 = 0, \tag{13}$$

for $j = 5k+1$ and $5k+2$, where

$$A_{5k+1} = \text{diag}(\boldsymbol{0}_3, \boldsymbol{1}_4, \boldsymbol{0}_3, \boldsymbol{0}_3, \boldsymbol{0}_{4k}, \boldsymbol{0}_3, \boldsymbol{0}_4),$$
$$A_{5k+2} = \text{diag}(\boldsymbol{0}_3, \boldsymbol{0}_4, \boldsymbol{0}_3, \boldsymbol{0}_3, \boldsymbol{0}_{4k}, \boldsymbol{0}_3, \boldsymbol{1}_4).$$

The gain matrix K is solved via constrained minimization:

$$\min_K \ \text{tr}(P_l^+) \ \ \text{s.t. } \eta_j(\boldsymbol{x}_l^+) = 0 \ \ \text{for } 1 \le j \le 5k+2.$$

Here we introduce $5k+2$ Lagrange multipliers to exert the constraints. The stationary condition $\partial L/\partial K = \boldsymbol{0}$ of the resulting Lagrangian has high dimensionality and nonlinearity. We solve it using the interior point optimization solver (IPOPT) [58] with a small tolerance set for the root.

C. Measurements From Vision

At every time instant, a Kinect Azure camera captures a frame of image of the object and its surroundings. The image has depth information for its pixels to be converted into points with three coordinates in the camera frame. This results in a range image, as a set C of points, only part of which are from the object's surface. First, we obtain a subset $Q \subseteq C$ containing points that are from the surface with high confidence. Next, we apply the RANSAC algorithm [22], known to be robust to data noise, to register the points in Q onto the object's mesh model. This produces the pose measurement $\boldsymbol{y}$ for the EKF at the current time instant k. More details about these operations are given below.

To secure a large enough point set Q, we match the mesh model against the range data. This requires a rough estimate of the object's pose. Here we make use of the posterior state estimate $\boldsymbol{x}_{k-1}^+$ by the EKF at the previous time instant $k-1$, extracting the pose component $({}^w\boldsymbol{p}_{k-1}^+, {}^w\boldsymbol{r}_{k-1}^+)$ and motion component $(\boldsymbol{v}_{k-1}^+, \boldsymbol{\omega}_{k-1}^+)$. A prior pose estimate $({}^w\boldsymbol{p}_k^-, {}^w\boldsymbol{r}_k^-)$ at time instant k is obtained via integration of the motion over one time step. We include in Q those points from C that lie close enough to the mesh model transformed under the pose estimate.

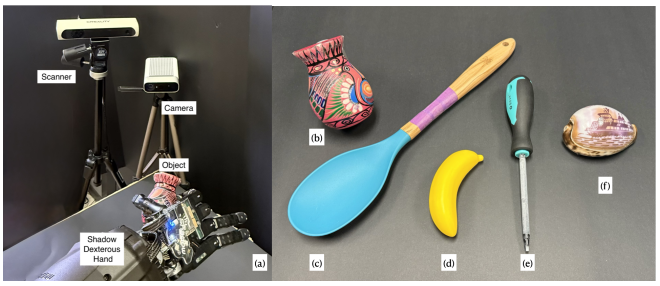


Fig. 2: Hardware and objects in the experiments: (a) a Shadow Dexterous Hand mounted on a Barrett WAM Arm (not shown in the figure), a Kinect Azure camera for capturing RGB-D images, a Creality 3D scanner for validation; five objects to be manipulated: (b) a vase, (c) a spoon, and (d) a banana toy; (e) a screwdriver; (f) a seashell.

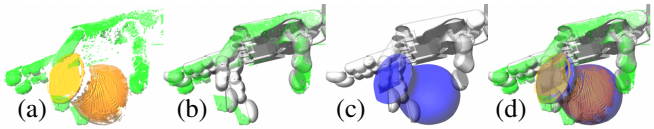


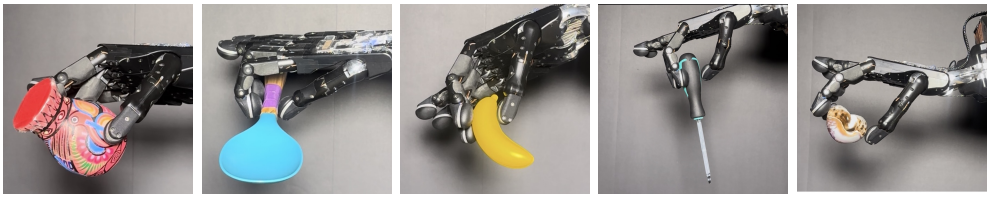
Fig. 3: Pose validation. (a) Mesh data of the Shadow Hand (green) and the vase (yellow) in Fig. 2 generated from scanning with data separation carried out by hand. (b) Registration of the hand data onto the hand's kinematic model using the ICP algorithm to localize the scanner frame $\{s\}$. (c) Vase rendered in its estimated pose. (d) Overlay of the ground truth configuration in (a) and the predicted configuration in (c). Pose error is obtained by invoking the ICP a second time.

RANSAC is then conducted at the time instant k. In every iteration, it randomly draws a very small subset $S \subset Q$ of points, with $|S| = 3$ or 4. It then registers the points onto the model to obtain a transformation and apply it to the remaining points in Q, adding to S those close enough to the model. The algorithm terminates with a large enough S achieved during one iteration or the largest S after a fixed number of iterations. The transformation is polished using all the points in S and its inverse becomes the pose measurement $\boldsymbol{y}$ for the EKF.

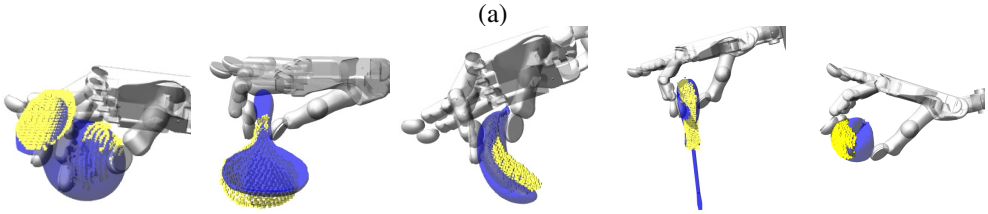
V. Experiments

For validation we conducted experiments with a 20-DOF Shadow Dexterous Hand driven by a 4-DOF Barrett WAM Arm manipulating five objects: a vase, a spoon, a banana toy, a screwdriver, and a seashell, as shown in Fig. 2. Mesh models of these objects were acquired using either a NextEngine or Creality 3D scanner. Spline surfaces were constructed over these mesh models via fitting.

A control policy [9] was adopted to realize rolling finger contacts. The total force and torque to be exerted on an object was calculated according to the dynamics equation (1) using a proportional–integral–derivative (PID) control servo with its desired trajectory and estimated pose and velocities from the EKF. This total wrench was distributed to each finger using the grasp matrix with adequate internal components added to keep contact forces inside their respective friction cones. Then the desired contact position of fingertip i was



(a)



(b)

RANSAC	avg posn. err. (m)	0.0282	0.0135	0.0267	0.0286	0.0303
estimate	avg orient. err. (rad)	0.0827	0.0951	0.3434	0.0911	0.1803
EKF	avg posn. err. (m)	0.0108	0.0070	0.0076	0.0184	0.0124
estimate	avg orient. err. (rad)	0.0836	0.0828	0.2420	0.0711	0.1516

(c)

Fig. 4: Estimation results on five objects: a vase, a spoon, a banana toy, a screwdriver, and a seashell. (a) Scenes at the ends of a maneuver, one of three on each object. (b) Corresponding simulated scenes using MuJoCo [60]. (c) Errors of the RANSAC and EKF pose estimates (each with position and orientation components) for the objects from (a) in the same order, determined using the validation method in Section V-B. (The orientation error on a symmetric object such as the vase excludes the component about the axis of symmetry.) The RANSAC estimate is the measurement provided to the last iteration of the EKF. It is computed over the last image frame using the procedure described in Section IV-C. The EKF estimate is part of the posterior state estimate, hence polished over the RANSAC estimate. Listed in the table are the average errors, each over three maneuvers of a different object.

obtained by ${}^w\boldsymbol{p}_i + kf_{ei}$, where ${}^w\boldsymbol{p}_i$ was the contact configuration estimated by the EKF, f_{ei} represents the contact force error which is scaled by some compliance factor k. Finally, we utilized an iterative inverse Jacobian solver of inverse kinematics to map the desired contact positions into the joint angles of the arm and hand.

A. Calibrating the Shadow Hand

During the experiments with the Shadow Hand, we observed a discrepancy between the measured fingertip position and the value calculated using the hand's forward kinematics. Hypothesizing constant fingertip offsets in the respective distal phalange's body frame, we used the 3D scanner to capture the robotic hand as meshes in a number of configurations. The ICP algorithm [21] was employed to match each mesh with the predicted hand configuration from forward kinematics. The resulting error was recorded under the ADD-S metric [54], which uses the average distance from a point on the object's surface in its estimate pose to the surface in its real pose (i.e., the ground truth). The pose offsets of all the fingertips (in the body frames of the respective distal links) were determined from minimizing the sum of the prediction errors over all captured configurations via the use of the IPOPT [58] using numerical differentiation. After the calibration, the discrepancy in the fingertip positions determined from the scanned mesh and the simulated model was reduced from 1.9 mm to 0.6 mm under the ADD-S metric.

B. Validation of Estimated Poses

Full validation of the estimation method during a dynamic manipulation would require the availability of a time series of "ground truth" poses, which would be unrealistic for in-hand manipulation. Instead, validation was conducted over the "snapshot" at the end of a maneuver by the hand, when it was possible for us to collect data needed for generating the ground truth.

We used the Creality 3D scanner to capture a mesh of the robotic hand together with the object in the ending snapshot (when the latter's pose had changed). The mesh data was in the scanner frame $\{s\}$ and had to be transformed into the world frame $\{w\}$. This was done by registering the portion of the mesh data corresponding to the hand[5] onto the robotic hand model in current joint angle readings under forward kinematics. Matching, conducted using the ICP algorithm,

[5]We separated this portion from that representing the object manually.

generated a transformation which was immediately applied to the mesh data portion representing the object to obtain the "ground truth". The transformed object data was then registered onto the full object model in its estimated pose, by invoking the ICP algorithm again. The inverse of the returned transformation, described by translation $\boldsymbol{p}_e$ and rotation matrix R_e, gave the position and orientation errors, respectively, as $||\boldsymbol{p}_e||$ and $||\log(R_e)||$ [59], where $\log(R_e)$ returns the skew-symmetric matrix representing the axis-angle of the rotation and $||\cdot||$ gives the rotation angle. Fig. 3 shows images representing the above steps in one localization instance.

C. Estimation Results

Fig. 4 summarizes dynamic in-hand localization of five objects. Real and simulated snapshots (at the last time instants) are displayed in (a) and (b), respectively. Three movements were performed on each object and the average position and orientation errors at the ends of the movements were computed and summarized in the table in (c).

The average errors on position and orientation from RANSAC-based measurements (utilizing the posterior state estimate of the EKF at the previous time instant) are in the ranges of 13–30 mm and 0.08–0.34 radian, respectively. They have been reduced by EKF to be within the ranges of 7–18 mm and 0.07–0.24 radian, respectively. For a rotationally symmetric object such as the vase, the error does not account for any rotation about the axis of symmetry (which cannot be observed in the range image). Both orientation errors for the banana toy stand out largely because of the object's small size and quite large portion of occluded surface area.

VI. Summary and Future Work

The presented EKF is capable of tracking the pose and motion of an object under in-hand manipulation, combining vision, dynamics, and rolling kinematics (which is realized through force control). The integration of RANSAC (and ICP) into the filter generates robust pose measurements from image frames that are repeatedly polished by the latter. Despite lack of a full-extent validation due to vision and hardware limitations, experiments have shown high accuracies in estimated poses.

Estimation performance is affected by the size and curvedness of the object's visible surface region. This depends on occlusions by the fingers manipulating the object. Incorporation of tactile data into the EKF will improve the estimation accuracy. This issue of sensor fusion awaits our investigation.

An advantage of the presented estimation method is its readiness for integration into manipulation. In the near future, we will exploit its use in force control during tool manipulation to demonstrate human-level dexterity. One challenge comes with finger gaits, which involve contact additions and removals that will change the system dynamics and thus the form of EKF. They may also increase the likelihood of an unexpected configuration change caused, for instance, by slippage to disrupt the EKF estimation. This change can be captured by vision in the current setup. We will look into immediate re-initialization of the EKF based on the current image frame and the most recent state estimate.

The use of RANSAC in generating raw pose measurements incurs a high computation cost that state estimation is currently conducted offline over stored image frames of the action. The measurement algorithm as presented in Section IV-C still needs a speedup by a factor within 50% for online estimation to be possible. This has temporarily prevented demonstration with high-precision manipulation tasks that require real-time estimates. With continuing improvements on the algorithm and a planned migration of the computing platform to a much faster NVIDIA GeForce GTX 745 workstation, we are quite optimistic about real-time state estimation and its use in high-precision manipulation in the coming weeks.

References

[1] V.-D. Nguyen, "Constructing force-closure grasps," *Int. J. Robot. Res.*, vol. 7, no. 3, pp. 3–16, 1988.

[2] A. Bicchi and V. Kumar, "Robotic grasping and contact: A review," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2000, pp. 348–353.

[3] A. Cole, J. Hauser, and S. Sastry, "Kinematics and control of multifingered hands with rolling contact," in *Proc. IEEE Int. Conf. Robot. Automat.*, 1988, pp. 228–233.

[4] Z. Li, J. F. Canny, and S. S. Sastry, "On motion planning for dextrous manipulation, part I: the problem formulation," in *Proc. IEEE Int. Conf. Robot. Automat.*, 1989, pp. 775–780.

[5] A. M. Okamura, N. Smaby, and M. R. Cutkosky, "An overview of dexterous manipulation," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2000, pp. 255–262.

[6] N. Hogan, "Impedance control: An approach to manipulation: Parts I — III," *ASME J. Dyn. Sys. Meas. Control*, vol. 107, pp. 1–24, 1985.

[7] M. Raibert, *Legged Robots That Balance*, The MIT Press, Cambridge, MA, 1986.

[8] L. Villani and J. De Schutter, "Force control," in B. Siciliano and O. Khatib, ed., *Handbook of Robotics, Part A*, Springer, 2008, pp. 195–219.

[9] Y. Xue, "Robotic manipulation of hand tools: The case of screwdriving," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2024, pp. 13883–13890.

[10] D. J. Montana, "The kinematics of contact and grasp," *Int. J. Robot. Res.*, vol. 7, no. 3, pp. 17–32, 1988.

[11] A. A. Cole, P. Hsu, and S. Sastry, "Kinematics and control of multifingered hands with rolling contact," *IEEE Trans. Automa. Control*, vol. 34, no. 4, pp. 398–404, 1989.

[12] W. E. L. Grimson, *Object Recognition by Computer: the Role of Geometric Constraints*, The MIT Press, Cambridge, 1990.

[13] B. K. P. Horn and K. Ikeuchi, "The mechanical manipulation of randomly oriented parts," *Sci. Am.*, vol. 251, no. 2, pp. 100-111, 1984.

[14] W. E. L. Grimson and T. Lozano-Pérez, "Model-based recognition and localization from sparse range or tactile data," *Int. J. Robot. Res.*, vol. 3, no. 3, pp. 3–35, 1984.

[15] P. C. Gaston and T. Lozano-Pérez, "Tactile recognition and localization using object models: The case of polyhedra on a plane," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 6, no. 3, pp. 257–266, 1984.

[16] David M. Siegel, "Finding the pose of an object in a hand," in *Proc. IEEE Int. Conf. Robot. Automat.*, 1991, pp. 406–411.

[17] B. Wen, C. Mitash, S. Soorian, A. Kimmel, A. Sintov, and K. E. Bekris, "Robust, occlusion-aware pose estimation for objects grasped by adaptive hand," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2020, pp. 6210–6217.

[18] S. Haidacher and G. Hirzinger, "Estimating finger contact location and object pose from contact measurements in 3-D grasping," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2003, pp. 1805–1810.

[19] L. Quan and Z. Lan. "Linear n-point camera pose determination," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 21, no. 8, pp. 774–780, 1999.

[20] V. Lepetit, F. Moreno-Noguer, and P. Fua. "EPnP: an accurate $O(n)$ solution to the PnP problem," *Int. J. Comput. Vis.*, vol. 81, pp. 155–166, 2009.

[21] P. J. Besl and N. D. McKay, "A method for registration of 3-D shapes," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 14, no. 2, pp. 239–256, 1992.

[22] M. A. Fischler and R. C. Bolles. "Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography," *Commu. ACM*, vol. 24, no. 6, pp. 381-395, 1981.

[23] L. Kneip, D. Scaramuzza, and R. Siegwart. "A novel parametrization of the perspective-three-point problem for a direct computation of absolute camera position and orientation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2011, pp. 2969—2976.

[24] P. K. Allen, A. Timcenko, B. Yoshimi, and P. Michelman, "Automated tracking and grasping of a moving object with a robotic hand-eye system," *IEEE Trans. Robot. Automat.*, vol. 9, no. 2, pp. 152–165, 1993.

[25] J. Liang, A. Handa, K. Van Wyk, V. Makoviychuk, O. Kroemer, and D. Fox, "In-hand object pose tracking via contact feedback and GPU-accelerated robotic simulation," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2020, pp. 6203–6209.

[26] M. Kokic, D. Kragic, and J. Bohg, "Learning to estimate pose and shape of hand-held objects from RGB images," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Sys.*, 2019, pp. 3980–3987.

[27] M. Johnson and E. Adelson, "Retrographic sensing for the measurement of surface texture and shape," in *Proc. IEEE Int. Conf. Comput. Vis. Pattern Recognit.*, 2009, pp. 1070–1077.

[28] R. Platt, F. Permenter, and J. Pfeiffer, "Using Bayesian filtering to localize flexible materials during manipulation," *IEEE Trans. Robot.*, vol. 27, no. 3, pp. 586–598, 2011.

[29] R. Li, R. Platt Jr., W. Yuan, A. ten Pas, N. Roscup, M. A. Srinivasan, and E. Adelson, "Localization and manipulation of small parts using Gelsight tactile sensing," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Sys.*, 2014, pp. 3988–3993.

[30] M. Bauza, O. Canal, and A. Rodriguez, "Tactile mapping and localization from high-resolution tactile imprints," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2019, pp. 3811–3817.

[31] J. Bimbo, S. Luo, K. Althoefer, and H. Liu, "In-hand object pose estimation using covariance-based tactile to geometry matching," *IEEE Robot. Autom. Lett.*, vol. 1, no. 1, pp. 570–577, 2016.

[32] Y. Yuan, H. Che, Y. Qin, B. Huang, Z.-H. Yin, K.-W. Lee, Y. Wu, S-C. Lim, and X. Wang. "Robot synesthesia: in-hand manipulation with visuotactile sensing," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2024, pp. 6558–6564.

[33] T. Lin, Y. Zhang, Q. Li, H. Qi, B. Yi, S. Levine, and J. Malik. "Learning visuotactile skills with two multifingered hands," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2025, pp. 5637–5643.

[34] Q. Ye, Q. Liu, S. Wang, J. Chen, Y. Cui, K. Jin, H. Chen, X. Cai, G. Li, and J. Chen. "Visual-tactile pretraining and online multitask learning for humanlike manipulation dexterity," *Sci. Robot.*, vol. 11, no. 110, eady2869, 2026.

[35] T. Chen, M. Tippur, S. Wu, V. Kumar, E. Adelson, and P. Agrawal. "Visual dexterity: in-hand reorientation of novel and complex object shapes" *Sci. Robot.*, vol. 8, no. 84, eadc9244, 2023.

[36] Y. Han, K. Yu, R. Batra, N. Boyd, C. Mehta, T. Zhao, Y. She, S. Hutchinson, and Y. Zhao. "Learning generalizable vision-tactile robotic grasping strategy for deformable objects via transformer," *IEEE/ASME Trans. Mechatronics*, vol. 30, no. 1, pp. 554–566, 2025.

[37] T. Anzai and K. Takahashi, "Deep gated multi-modal learning: in-hand object pose changes estimation using tactile and image data," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Sys.*, 2020, pp. 9361–9368.

[38] S. Dikhale, K. Patel, D. Dhingra, I. Naramura, A. Hayashi, S. Iba, and N. Jamali, "Visuotactile 6D pose estimation of an in-hand object using vision and tactile sensor data," *IEEE Robot. Autom. Lett.*, vol. 7, no. 2, pp. 2148–2155, 2022.

[39] A. N. Chaudhury, T. Man, W. Yuan, and C. G. Atkeson, "Using collocated vision and tactile sensors for visual servoing and localization," *IEEE Robot. Autom. Lett.*, vol. 7, no. 2, pp. 3427–3434, 2022.

[40] Y. Gao, S. Matsuoka, W. Wan, T. Kiyokawa, K. Koyama, and K. Harada, "In-hand pose estimation using hand-mounted RGB cameras and visuotactile sensors," *IEEE Access*, vol. 11, pp. 17218–17232, 2023.

[41] C. Corcoran and R. Platt Jr., "A measurement model for tracking hand-object state during dexterous manipulation," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2010, pp. 4302–4308.

[42] A. Petrovskaya and O. Khatib, "Global localization of objects via touch," *IEEE Trans. Robot.*, vol. 27, no. 3, pp. 559–585, 2011.

[43] G. Vezzani, U. Pattacini, G. Battistelli, L. Chisci, and L. Natale, "Memory unscented particle filter for 6-DOF tactile localization," *IEEE Trans. Robot.*, vol. 33, no. 5, pp. 1139–1155, 2017.

[44] Y.-B. Jia and M. Erdmann., "Pose and motion from contact," *Int. J. Robot. Res.*, vol. 18, no. 5, pp. 466–490, 1999.

[45] M. Pfanne and M. Chalon, "EKF-based in-hand object localization from joint position and torque measurements," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Sys.*, 2017, pp. 2464–2470.

[46] M. Pfanne, M. Chalon, F. Stulp, and A. Albu-Schäffer, "Fusing joint measurements and visual features for in-hand object pose estimation," *IEEE Robot. Autom. Lett.*, vol. 3, no. 4, pp. 3497–3504, 2018.

[47] Q. Lin, C. Yan, Q. Li, Y. Ling, Y. Zheng, W. Lee, Z. Wan, B. Huang, and X. Liu, "Tactile-based object pose estimation employing extended Kalman filter," in *Proc. IEEE Int. Conf. Adv. Robot. Mechatron.*, 2023, pp. 118–123.

[48] P. Hebert, N. Hudson, J. Ma, and J. Burdick, "Fusion of stereo vision, force-torque, and joint sensors for estimation of in-hand object location," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2011, pp. 5935–5941.

[49] S. Li, S. Lyu, and J. Trinkle, "State estimation for dynamic systems with intermittent contact," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2015, pp. 3709–3715.

[50] M. Cherif and K. K. Gupta, "Global planning for dexterous reorientation of rigid objects: finger tracking with rolling and sliding," *Int. J. Robot. Res.*, vol. 20, no. 1, pp. 57-84, 2001.

[51] A. Bhatt, A. Sieler, S. Puhlmann, and O. Brock, "Surprisingly robust in-hand manipulation: An empirical study," *Proc. Robot.: Sci. Sys.*, 2021.

[52] Y.-B. Jia, "Quaternions [online notes for foundations of robotics and computer vision]," https://faculty.sites.iastate.edu/jia/files/inline-files/quaternion.pdf, 2024.

[53] M. Gardner, "Pose and motion estimation of free-flying objects: aerodynamics, constrained filtering, and graph-based feature tracking," *IEEE Trans. Robot.*, vol. 38, no. 5, pp. 3187–3202, 2022.

[54] Y. Xiang, T. Schmidt, V. Narayanan and D. Fox, "PoseCNN: a convolutional neural network for 6D object pose estimation in cluttered scenes," In *Robotics: Sci. Syst.*, 2018.

[55] S. Garrido-Jurado, R. Muñoz-Salinas, and F. J. Marín-Jiménez, "Automatic generation and detection of highly reliable fiducial markers under occlusion," *J. Pattern Recog.*, vol. 47, no. 6, pp. 2280–2292, 2014.

[56] D. Simon, "Optimal State Estimation," *Hoboken, NJ, USA: Wiley*, 2006.

[57] D. Simon, "Kalman filtering with state constraints: a survey of linear and nonlinear algorithms," *IET Control Theory & Applications*, 4(8), pp. 1303-1318, 2010.

[58] A. Wächter and L. T. Biegler, "On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming," *Math. Program.*, vol. 106, no. 1, pp. 25–57, 2006.

[59] D. Q. Huynh, "Metric for 3D rotations: Comparisons and analysis," *J. Math. Imaging Vis.*, vol. 35, pp. 155–164, 2009.

[60] E. Todorov, T. Erez, and Y. Tassa, "MuJoCo: A physics engine for model-based control," *IEEE/RSJ Int. Conf. Intell. Robot. Syst.*, 2012, pp. 5026–5033.