Fusing Joint Measurements and Visual Features for In-hand Object Pose Estimation

Martin Pfanne[1], Maxime Chalon[1], Freek Stulp[1], and Alin Albu-Schäffer[2]

***Abstract*—For a robot to perform complex manipulation tasks, such as in-hand manipulation, knowledge about the state of the grasp is required at all times. Moreover, even simple pick-and-place tasks may fail because unexpected motions of the object during the grasp are not accounted for. This work proposes an approach which estimates the grasp state by combining finger measurements, i.e. joint positions and torques, with visual features that are extracted from monocular camera images. The different sensor modalities are fused using an extended Kalman filter. While the finger measurements allow to detect contacts and resolve collisions between the fingers and the estimated object, visual features are used to align the object with the camera view. Experiments with the DLR robot David demonstrate the wide range of objects and manipulation scenarios that the method can be applied to. They also provide insight into the strengths and limitations of the different, complementary types of measurements.**

***Index Terms*—Perception for Grasping and Manipulation, Dexterous Manipulation, Sensor Fusion**

I. Introduction

INSPIRED by the capabilities of the human hand, robotic manipulators have become more and more dexterous, compliant and sensitive. However, while the mechanics of these robotic hands open up new possibility for complex interactions with the environment, actually performing skilled manipulation largely remains an open problem. One of the aspects that make it so challenging to reliably manipulate an object is that it requires knowledge about the state of the grasp at all times. Indeed, even simple pick-and-place tasks may fail due to unreliable information about the location of the object inside the hand. When grasping, inaccuracies in the planning model or the execution often cause the object to move differently than was anticipated. Although the object may still settle in a stable grasp, if not accounted for, these deviations may negatively affect the outcome of a task. For example, the ketchup bottle in Fig. 1 would fall over if its unintended tilt during the grasp was not compensated before releasing it.

Manuscript received: February, 24, 2018; Revised May, 23, 2018; Accepted June, 19, 2018.

This paper was recommended for publication by Editor Han Ding upon evaluation of the Associate Editor and Reviewers' comments.

[1]Martin Pfanne, Maxime Chalon and Freek Stulp are with the Institute of Robotics and Mechatronics, German Aerospace Center, Wessling 82234, Germany martin.pfanne@dlr.de, maxime.chalon@dlr.de, freek.stulp@dlr.de

[2]Alin Albu-Schäffer is with the Institute of Robotics and Mechatronics, German Aerospace Center, Wessling 82234, Germany, and also with the Technical University of Munich, Munich 80805, Germany alin.albu-schaeffer@dlr.de

Digital Object Identifier (DOI): see top of this page.

Traditionally, object localization is performed from visual data, for example by Ulrich et al. [1], who used monocular camera images to recognize objects from their known 3D geometry. However, during object manipulation, these methods may suffer from occlusions of the object by the manipulator. Recognizing this problem, Bimbo et al. proposed several methods that combined visual information with other sensor modalities. While the iterative optimization algorithm in [2] fused information from vision, tactile sensors and joint encoders, the method presented in [3] estimated the object pose solely from tactile and force sensing, requiring visual data only for the initialization of the object pose. Similarly, in [4], Zhang et al. proposed a particle filter that incorporates tactile sensor data to improve the object tracking during periods of visual occlusion. At the same time they were able to estimate dynamic parameters of their model. [5], [6] built on this work, presenting probabilistic frameworks for the dynamic system state estimation. In addition to the object pose, Corcoran et al. [7] used a particle filter to estimate the shape of an unknown object through tactile sensing. In [8], information from the high resolution GelSight contact sensor was fused with RGB-D visual data to track objects using a point-cloud-based approach. Schmidt et al. [9], on the other hand, combined depth-only vision with physical constraints to solve the in-hand localization problem. Another popular approach was presented by Hebert et al. [10], who fused stereo vision with force-torque and joint position measurements to determine the object pose.

Our approach differs from these methods in that it only relies on joint position measurements as the minimal set of sensor inputs. Depending on the manipulation scenario, the estimation from this data may already be sufficient to solve

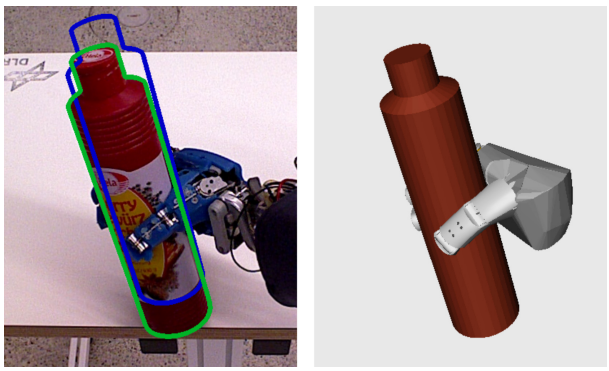
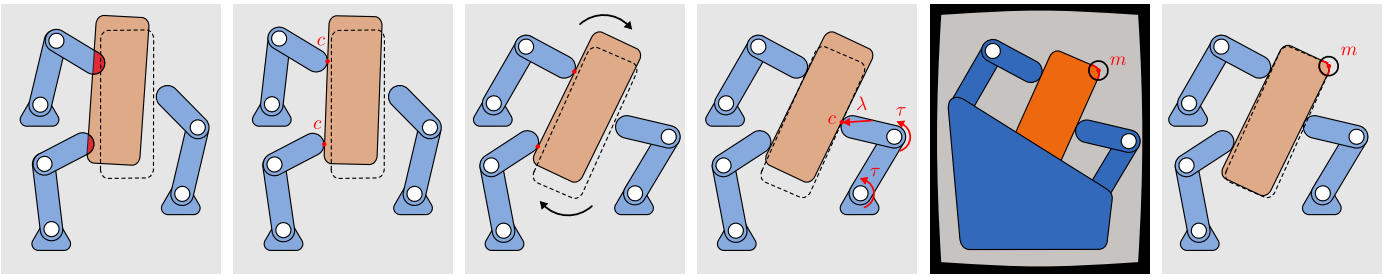


Fig. 1. Left: In-hand object pose estimation from joint position measurements (blue) and fused with visual features from a monocular camera image (green). Right: Knowledge about the scene.



(a) Initial estimation of an object grasp (b) Correction with finger positions (c) Prediction of object motion (d) Correction with finger torques (e) Feature extraction from camera image (f) Correction with visual feature

Fig. 2. (a) Initial estimation of an object grasp including ground truth (dashed). (b) Correction of the object pose to resolve the collisions. (c) Pose prediction from finger motions. (d) Corrected pose from an inferred contact from joint torques. (e) Camera view of the grasp and an extracted object feature m. (f) Corrected object pose to match the pixel coordinates of the visual feature.

a task that otherwise would fail. Estimating the pose of the tilted ketchup bottle in Fig. 1 from joint positions (blue) alone is enough to make sure that it does not fall when released.

At the same time, the proposed method allows the integration of additional sensor information. Joint torque measurements, if available, are used to infer contacts, without requiring additional tactile sensors. Visual features are extracted and fused from monocular camera images, offering a sensor-minimal option to integrate visual data, even if only small parts of the object are visible. Aligned with features from the camera image, the combined estimation (green) in Fig. 1 is able to precisely localize the ketchup bottle.

The proposed method continues our previous work on in-hand localization. Similar to [11], the formulation of linearizable motion and measurement models allow the probabilistic fusion of the sensor data with an extended Kalman filter. Compared to the preceding work in [12], which presented a particle filter solution, the EKF offers faster convergence at reduced computational cost.

The outline of the paper is as follows. Section II describes the estimation problem and gives an intuition about how the different types of measurements are used. Subsequently, Section III focuses on the implementation of these concepts. The experiments that were conducted to validate the method and compare the different types of measurements, are illustrated in Section IV. Finally, Section V gives a brief summary of the work, as well as concluding remarks.

II. Problem Description and Main Concept

This work is concerned with the estimation of the grasp state of a manipulated object. Primarily, this consists of identifying contacts between the manipulator and the object, as well as estimating the pose of the object. This section will give an intuition of how to use different types of measurements in order to inform this process.

Kinematic measurements: Fig. 2(a) illustrates a simple manipulation problem. An object is held by a manipulator using its three fingers. The assumed pose differs from the ground truth, represented by the dashed line. Given incorrect assumptions about the pose of the object and the finger links, the object is in collision with two of the fingers of the manipulator. Since this is physically not possible, without further information, it can be inferred that either the estimated pose of the object or the assumed pose of the finger links in collision are incorrect. An update to the estimated grasp state can resolve this collision by moving the object and/or the fingers, as illustrated in Fig. 2(b). While still not perfect, this correction moves the estimated object pose closer to the ground truth.

Furthermore, if the fingers are actively repositioned, the motion of the object can be predicted as well. Fig. 2(c) shows how the rotation of the object is inferred from the displacement of the fingers by mapping the finger motions to the object motion using the estimated contact points.

This example illustrates the information and assumptions that are necessary in order to estimate the grasp state from kinematic measurements. First, the object has to be rigid, of known geometry and an (inaccurate) initial estimate of the object pose has to be available. This, for example, can be provided by a vision system previous to the manipulation. During the manipulation, occlusions of the object by the hand make it much more challenging to localize purely from vision. Second, all parts of the hand have to be rigid, of known geometry and (inaccurate) measurements of the poses of the finger links have to be available. The poses of the finger links are calculated from the joint angles, using a kinematic description of the fingers.

Torque measurements: In the same way that kinematic measurements provide information about the position of contacts between the object and the manipulator, torque measurements allow the estimation of the forces that are applied through these contacts. Fig. 2(d) highlights the torques that are applied to the joints of one of the fingers when grasping the object. Given a kinematic description of the fingers and an estimation of possible contact positions, the direction and magnitude of the contact forces can be inferred. In addition to estimating the contact forces of known contacts, this also allows the detection of new contacts. Similarly to the kinematic contacts that were inferred from colliding bodies, these torque contacts can be used to correct the object and/or finger positions. If there is a contact between the object and a finger link, these two bodies have to be in touch. Therefore, as illustrated in Fig. 2(d), by satisfying this constraint, the estimated object and finger poses can be improved.

Visual features: Combining kinematic and torque measurements improved the estimated grasp state in Fig. 2(d) and brought it closer to the ground truth. Unfortunately, the

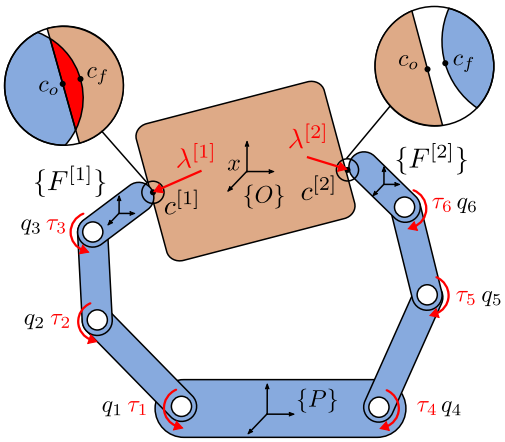


Fig. 3. Illustration and quantities of a two finger grasp of an object.

estimation of the object pose is not fully constrained by the finger measurements in the vertical direction. However, this aspect of the estimation can be improved already with very sparse visual information. If there are characteristic features of the object that can be detected on a camera image, the pixel coordinates of these features help to align the estimated object. For example, if the upper-right corner of the object in Fig. 2(e) is detected in a camera image, the estimated object pose can be corrected. These characteristic features have to be either known beforehand or automatically extracted and matched to a virtual camera image of the estimated scene. Fig. 2(f) shows the result of the correction.

In summary, the integration of visual features requires that a monocular camera feed is available and that object features can be extracted and matched to the estimated state of the scene. Additionally, the camera transformation has to be known in order to relate 2D pixel coordinates to 3D points on the object.

III. Implementation

After the previous section gave some intuition about how different types of measurements can inform the estimated grasp state, this section describes the actual implementation of these concepts, starting with the estimation from joint position measurements.

A. Kinematic Measurements

Definitions: Fig. 3 illustrates a manipulated object, while also highlighting the most important quantities of the grasp. The pose of the object is $x \in \mathbb{R}^6$. It represents the translation and rotation, described in Euler angles, of an object fixed frame $\{O\}$ w.r.t. a palm fixed frame $\{P\}$. Fixed to each finger link with index j is a frame $\{F^{[j]}\}$. The pose of these frames w.r.t. $\{P\}$ can be calculated from the vector of joint positions, $q \in \mathbb{R}^m$, using the forward kinematics of the fingers. The number of joints is denoted by m.

The position of a contact with index i between a finger link and the object is denoted by $c^{[i]} \in \mathbb{R}^3$ and expressed in $\{P\}$. If there is some distance between the object and the link, $c_o^{[i]} \in \mathbb{R}^3$ and $c_f^{[i]} \in \mathbb{R}^3$, both described in $\{P\}$, denote the two points on the surface of the object and on the link, respectively, that are closest. Similarly, $c_o^{[i]}$ and $c_f^{[i]}$ can be expressed for two colliding bodies, where $c_f^{[i]} - c_o^{[i]}$ is the smallest possible displacement to resolve the collision.

Contact detection: Section 2 described how the estimated grasp state should be corrected if finger links are penetrating the object. However, before this correction is possible, these collisions have to be identified. A wide range of methods have been proposed to determine if two geometric bodies are in collision. For this work, a modified version of the Gilber-Johnson-Keerthi (GJK) algorithm was used [13]. Given the poses and 3D meshes of the two bodies, this version of the GJK algorithm not only returns the information if the bodies are touching. It also calculates the two contact points, $c_o^{[i]}$ and $c_f^{[i]}$, as they were previously introduced. Therefore, the distance between these two points, $d^{[i]} = \|c_f^{[i]} - c_o^{[i]}\|$, describes either the smallest distance or penetration depth of the two bodies. A finger link is considered to be in contact with the object if $d^{[i]} < 0$. Distance calculations of non-convex geometries are realized by decomposing them into convex pieces, since the core GJK algorithm is only applicable to convex bodies.

Extended Kalman filter: The central part of the proposed method is an extended Kalman filter (EKF) [14]. It allows the recursive incorporation of new measurements in a probabilistic manner. At any time step it provides the current best estimate of the grasp state including its covariance. Furthermore, it accounts for inaccuracies in the measurements and the initial estimate of the grasp state.

The most important quantities of the EKF are the mean, y, and covariance, P, of the estimated grasp state. For the estimation from kinematic measurements y at time t is comprised of two components:

$$y_t = \begin{pmatrix} x_t \\ \tilde{q}_t \end{pmatrix} \tag{1}$$

namely the pose of the object, x_t, and a vector of joint position biases, $\tilde{q}_t \in \mathbb{R}^m$. The biases are used to estimate constant errors in the joint position measurements.

The initial value y_0 at $t = 0$ has to be provided, including its initial uncertainty, P_0. The initial object pose, x_0, can for example be obtained by a visual localization system, which is able to detect the object before being hindered by hand occlusions. The initial covariances of the object pose accounts for inaccuracies in this method.

The joint biases $\tilde{q}_0$ may be initialized to zero if no additional information is available. However, the respective covariance should be non-zero if inaccurate joint measurements are to be expected.

The first part of the EKF loop is the prediction step. In this case, its purpose is to relate motions of the fingers to those of the object. The appropriate control vector, u, for this task is the vector of joint velocities:

$$u_t = \dot{q}_t \tag{2}$$

The joint velocities can either be measured or calculated discretely. In the EKF framework, the state at time t is

described by the sum of a motion model, f, and zero mean Gaussian motion disturbances, w:

$$y_t = f(y_{t-1}, u_t) + w_t \tag{3}$$

Using these definitions, the mean and covariance of the state can be predicted as follows [14]:

$$\overline{y}_t = f(y_{t-1}, u_t) \tag{4}$$

$$\overline{P}_t = F_{t-1} P_{t-1} F_{t-1}^{\mathrm{T}} + Q_t \tag{5}$$

where Q_t is the covariance of w_t and:

$$F_t = \left.\frac{\partial f}{\partial y}\right|_{y_{t-1}, u_t} \tag{6}$$

Joint velocities can be related to object twists using the grasp matrix $G \in \mathbb{R}^{6\times 3n}$ and the hand Jacobian $J \in \mathbb{R}^{3n\times m}$ for hard-finger contacts [15], with n being the number of contacts that have been identified between the fingers and the object using the GJK algorithm:

$$\dot{x}_t = W G^{+} J \dot{q}_t \tag{7}$$

where G^{+} is the Moore-Penrose inverse of G and W is the matrix that maps the object twist to changes in its position and Euler angles.

The joint biases, $\tilde{q}_t$, remain unaffected by the prediction. Therefore, the complete motion model can be expressed as follows:

$$f(y_{t-1}, u_t) = y_{t-1} + \begin{pmatrix} W G^{+} J \\ 0^{m\times 1} \end{pmatrix} u_t \Delta t \tag{8}$$

where Δt is the time between two EKF steps.

This model is able to predict the object motion, while assuming that the position of the contact points on the surface of the object and the finger do not change. Of course, this assumption is not correct for rolling or slipping contacts. The errors caused by this assumption can be reduced by using more complex contact and prediction models. In any case, it will be corrected in the update step.

The update step is the second part of the EKF loop. Its purpose is to resolve incorrect alignments between the estimated poses of the object and finger links. Collisions between these bodies can be caused by errors in the initial object pose or inaccuracies in the finger measurements. Additionally, the prediction may cause the object and fingers to penetrate or separate by ignoring the rolling and slipping of contacts.

The magnitude of these misalignments can be described by the distance or penetration depth of the bodies. As illustrated in Fig. 3, this can be expressed by the difference between $c_o^{[i]}$ and $c_f^{[i]}$. If the object and a finger link are touching at contact i, then the difference between $c_o^{[i]}$ and $c_f^{[i]}$ has to be zero:

$$0 = c_f^{[i]} - c_o^{[i]} \tag{9}$$

This constraint shall be enforced by the update step of the EKF. In the EKF framework, measurements are described by the sum of a measurement model, z, and zero mean Gaussian measurement disturbances, v:

$$z_t = h(y_t) + v_t \tag{10}$$

The measurement model is then used in the update equations to correct the mean and covariance of the state [14]:

$$y_t = \overline{y}_t + K_t(z_t - h(\overline{y}_t)) \tag{11}$$

$$P_t = (I - K_t H_t)\overline{P}_t \tag{12}$$

with:

$$H_t = \left.\frac{\partial h}{\partial y}\right|_{y_t} \tag{13}$$

$$K_t = \overline{P}_t H_t^{\mathrm{T}} (H_t \overline{P}_t H_t^{\mathrm{T}} + R_t) \tag{14}$$

where R_t is the covariance of v_t.

Constraints, such as the one that is expressed in Eq. (9), can be modeled as perfect measurements in the EKF. If we assume that there is a contact of index i, where the object touches a link, then the measurable distance between $c_o^{[i]}$ and $c_f^{[i]}$ has to be zero. Therefore, it follows that $z_t^{[i]} = 0$. Expressed for all n contacts, the complete measurement vector can be written as:

$$z_t = 0^{3n\times 1} \tag{15}$$

The value of the measurement model at time t is expressed by the difference between the two contact vectors:

$$h(y_t) = c_f - c_o \tag{16}$$

The consequence of this formulation is that H_t in Eq. (12) cannot be directly obtained by deriving h_t w.r.t. y_t. While c_o and c_f are determined by the current estimate of the grasp state, they are not obtained from a differentiable function. Instead they are the result of the iterative GJK algorithm. However, as was described in the EKF prediction step, the velocities of contact points on the object can be related to the object twist using the grasp matrix G:

$$\frac{\partial c_o}{\partial x} = G W^{-1} \tag{17}$$

Similarly, the velocities of contact points on the finger can be related to the joint velocities using the hand Jacobian J. It follows that this relation also applies w.r.t. changes in the joint biases:

$$\frac{\partial c_f}{\partial \tilde{q}} = J \tag{18}$$

The complete H_t matrix combines both partial derivatives:

$$H_t = \begin{pmatrix} -G W^{-1} & J \end{pmatrix} \tag{19}$$

B. Torque Measurements

This subsection describes how the algorithm can be improved by inferring additional contacts from torque measurements in the finger joints.

Definitions: In addition to the kinematic quantities, Fig. 3 also shows the torques that are applied to each joint. The vector of all joint torques is denoted by $\tau \in \mathbb{R}^m$. The force that is applied to the object through contact i is denoted by $\lambda^{[i]} \in \mathbb{R}^3$.

Contact detection: If a finger link is touching the object, it applies a force to it, transmitted through the contact. The

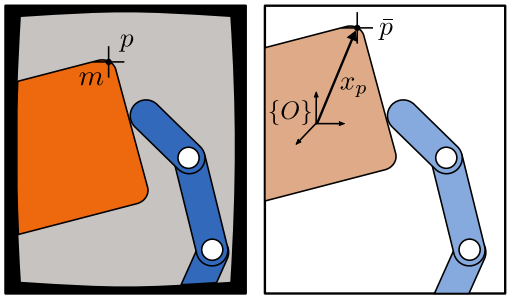


Fig. 4. Left: Illustration of a partial camera view of the grasp. Right: Matching virtual view of the estimated state of the scene.

magnitude and direction of the vector of all forces can be estimated from the measured joint torques, using the hand Jacobian, J:

$$\overline{\lambda} = (J^T)^+ \tau \tag{20}$$

Whether to consider contact i in the EKF is decided by two factors. First, $\overline{\lambda}^{[i]}$ has to lie inside the friction cone of the contact. If the angle of the friction cone is not known, it can be conservatively chosen, for example 45°. Second, the magnitude of the force is considered in order to avoid false positives. Only contacts with a force that is greater than an empirical threshold, for example 5 N, are included in the estimation. Tactile sensors could be included in a similar way to provide additional contact information.

C. Visual Features

The integration of position and torque measurements constrain the pose of the object. However, these measurements alone are not always sufficient to fully constrain the pose estimation. In such cases, already very sparse visual information helps to improve the assumed location of the object.

Definitions: Fig. 4 shows two similar scenes. The left figure illustrates the image from a monocular camera that partially observes an object grasp. On the right is the estimated view of the scene. A visual feature m, located at pixel coordinates $p \in \mathbb{N}^2$, was extracted from the camera image. It is matched to a corresponding feature in the estimated scene. The pixel coordinates in the virtual image, $\bar{p} \in \mathbb{N}^2$, can be related to a 3D point on the object at $x_p \in \mathbb{R}^3$ described in $\{O\}$, using the pinhole camera model.

Feature extraction: Many methods have been proposed in literature to extract and match visual features from images. The experiments that were conducted as part of this work utilized the following two procedures:

Visual markers: AprilTags [16] that are rigidly attached to the object provide a simple way to extract features, as long as they are not occluded. The detection of an AprilTag on the object provides a set of four pixel coordinates, representing the four corners of the tag. Since the pose of the tag w.r.t. the object is known, the 3D coordinates of the features can be easily obtained as well.

Feature detection + shape matching: Computer vision algorithms can be used to extract and match features from images without the need for dedicated markers. The procedure that was used with the proposed method comprised of the following steps:

1) Application of the Canny detector to the virtual and real image to extract edge images
2) Extraction of corners from the virtual and real edge images using the Harris corner detector
3) Extraction of characteristic contour pieces around the corners
4) Matching of the two sets of contours pieces using the generalized Hough transform
5) Filtering of outliers using the RANSAC algorithm

The result of this procedure is a list of features with corresponding pixel coordinates in both the virtual and real image. The 3D position of the features in the virtual images can be calculating using ray tracing or a depth map of the 3D scene.

Extended Kalman filter: For a feature of index k, the difference between the pixel coordinates in the real image, $p^{[k]}$, and those in the virtual image, $\bar{p}^{[k]}$, describes the misalignment of the object w.r.t. this feature. Therefore, the corrected estimation should satisfy:

$$0 = p^{[k]} - \bar{p}^{[k]} \tag{21}$$

To realize this, the visual features can be included as an additional measurement in the EKF. The measurement vector, z_t in Eq. (15), is extended to include the vector of the feature pixel coordinates in the camera image:

$$z_t = \begin{pmatrix} 0^{3n\times 1} \\ p_t \end{pmatrix} \tag{22}$$

The measurement model, $h(y_t)$ in Eq. (16), is modified to account for the vector of pixel coordinates in the virtual image:

$$h(y_t) = \begin{pmatrix} c_f - c_o \\ \bar{p}_t \end{pmatrix} \tag{23}$$

In order to find the derivative of the measurement model, H_t, $\bar{p}_t$ has to expressed w.r.t. the pose of the object.

The pixel coordinates of a feature in the virtual image, $\bar{p}^{[k]}$, can be related to $x_p^{[k]}$, the estimated 3D position of the feature on the object. The camera matrix, $C \in \mathbb{R}^{3\times 4}$, projects ${}^c x_p^{[k]}$, the position described in the camera frame $\{C\}$, to $\bar{p}^{[k]}$:

$$s \begin{pmatrix} \bar{p}^{[k]} \\ 1 \end{pmatrix} = C \begin{pmatrix} {}^c x_p^{[k]} \\ 1 \end{pmatrix} \tag{24}$$

where s is a scalar factor.

The projection of the feature position described in the object frame, $\{O\}$, can be expressed as follows:

$$s \begin{pmatrix} \bar{p}^{[k]} \\ 1 \end{pmatrix} = C T_c^{-1} T_o \begin{pmatrix} x_p^{[k]} \\ 1 \end{pmatrix} \tag{25}$$

where T_c denotes the transformations of $\{C\}$ w.r.t. the palm frame, $\{P\}$, and T_o denotes the transformation of $\{O\}$ w.r.t. $\{P\}$, which is calculated from x. Equation (25) can be derived to obtain the partial derivative of $\partial \bar{p}^{[k]} / \partial x$.

IV. Validation

The proposed method was validated in a series of experiments utilizing the DLR humanoid robot David [17]. In total, five objects were manipulated using different grasps.

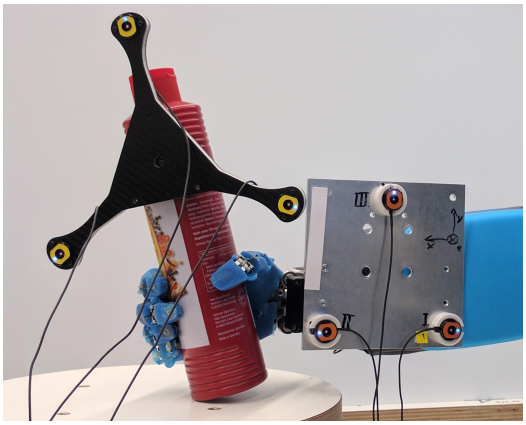


Fig. 5. Experimental setup: Tracking markers on the object and the wrist provide a ground truth of the object pose.

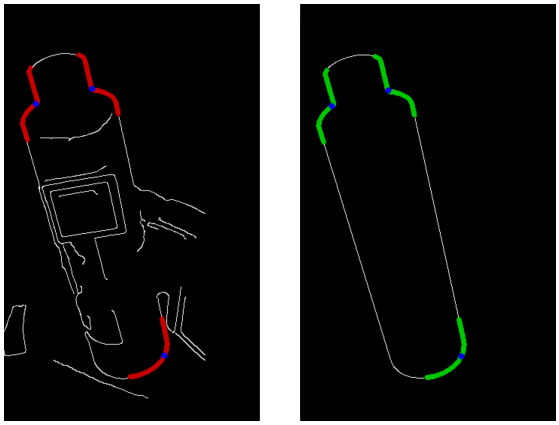


Fig. 6. Feature extraction: First, corner features (blue) are extracted from both the camera (left) and virtual (right) edge images. Second, surrounding contour pieces are used to identify matches between the features.

A. Experimental Setup

The ground truth for the experimental validation was provided by the K610 visual tracking system from Nikon. Fig. 5 illustrates the setup. All five experiments used the same set of parameters to configure the algorithm. If not stated otherwise, the initial pose of the object was provided by the tracking system. The covariance of the state was initialized with 2 cm in position, 5° in orientation and 5° in the joint biases. During execution, the algorithm processed new measurements at a rate of 10 Hz.

In each scenario, different types of measurements were incorporated into the estimation in order to compare their respective performances. Fig. 7 summarizes the results of all experiments. For each experiment, it illustrates both the initial and final state of the manipulation scenario, and highlights the quality of the estimation from different combinations of measurements. The center graphs show the estimated changes in translation and rotation of the objects and compare them to the ground truth from the tracking system. The terminal errors in position and orientation at the end of the experiments are summarized in the right graphs in Fig. 7, as well as in Table I.

In total, four combinations of measurements were evaluated. The estimation from kinematic data, represented by the red line, used only joint position measurements. The combination of kinematic and torque measurements is shown in blue. The orange line illustrates the fusion of both finger measurements and visual features extracted from AprilTags. Finally, green represents the estimation from finger measurements and automatically extracted visual features. The feature extraction method is illustrated in Fig. 6.

A statistical analysis of the performance of the method, for example to evaluate the effect of variations in the initial pose, was not examined in the context of this paper. Future work will extend in this direction.

B. Experiments

Exp. I: Power grasp of a ketchup bottle: The first experiment consisted of a power grasp of a ketchup bottle, where the fingers were commanded to close until a preset torque limit was reached. During this grasp, the bottle tilted inside the hand. The initial estimate of the object pose was purposely set 20 mm higher, to illustrate the strengths and limitations of different types of measurements.

The supplementary material of this work includes a video[1] of a pick-and-place scenario using the same ketchup bottle and grasp. The video illustrates the effect of no in-hand localization, the estimation from joint position measurements and the fusion with visual features, respectively, on the success of the task execution.

Exp. II: In-hand manipulation of a brush: For the second experiment, a brush, held in a fingertip grasp, was actively rotated inside the hand, causing the contacts to roll on the surface of the fingers and the object.

Exp. III: Push and grasp of a water bottle: The third experiment was a power grasp of a water bottle. It highlights two additional aspects of the proposed method. First, before grasping the object, it was pushed on the table, as can be seen in the change of y in the respective graph in Fig. 7. Second, the water bottle is mostly transparent. Combined with occlusions by the fingers, this makes it very challenging to estimate the object pose solely using RGB camera vision.

Exp. IV: Grasp of a free-form shampoo bottle: While the previous objects were either cylindrical or prismatic, the shampoo bottle in the fourth experiment has a less common shape. Furthermore, the grasp of the object was neither a power grasp, nor a fingertip grasp.

Exp. V: Fingertip grasp and manipulation of a pen: The final experiment consisted of the fingertip grasp and manipulation of a pen on a table. After grasping, the pen was rotated twice back and forth. This object was chosen to illustrate the application of the proposed method to small objects. Since the pen was to small to attach the ground truth markers or an AprilTag, the respective figures in Fig. 7 only show three lines and do not include an illustration of the terminal estimation errors. The initial pose of the pen was manually set, placing it flat on the table and roughly aligning it with the view from the robot camera.

[1]In the video, the robot arm is first executing a preset trajectory, which ensures that the ketchup bottle is properly placed inside the hand, even if the initial object pose was not well known. Next, the hand is torque controlled to robustly grasp the object, however, tilting it in the process. Based on the estimated object pose, the arm is then moved to place the object. Relying only on the initial assumption of the pose causes the object to fall.

TABLE I
TERMINAL ABSOLUTE ERRORS IN POSITION AND ORIENTATION

	Joint Positions	Positions +Torques	AprilTag Features	Extracted Features
Exp. I: Ketchup				
Position in [mm]	29.3	26.6	20.2	11.1
Orientation in [deg]	22.9	9.0	7.0	11.2
Exp. II: Brush				
Position in [mm]	37.6	22.7	13.5	13.7
Orientation in [deg]	26.5	10.8	10.1	11.1
Exp. III: Water				
Position in [mm]	28.6	22.4	16.0	17.4
Orientation in [deg]	19.9	9.5	11.2	26.9
Exp. IV: Shampoo				
Position in [mm]	34.4	15.9	8.4	8.5
Orientation in [deg]	22.1	14.7	6.8	10.4

C. Discussion

Both Exp. I and Exp. III show that the estimation from kinematic data is able to approximately track the motion of objects during power grasps. While not as precise as the combination with other measurements, the estimated pose can be sufficient to ensure the successful execution of a task, i.e. compensating the tilt of a bottle before placing it. However, for the grasps in Exp. II and Exp. IV, which rely more on fingertip contacts, joint position measurements alone are insufficient. Since these grasps are much less kinematically constrained than power grasps, the inclusion of torque measurements, which improves the contact detection and maintenance, greatly reduces the estimation error.

The estimated joint position biases, when using both finger measurements, were in the range of +/- 20°.

The limitation of both types of finger measurements is their inability to constrain the object pose in all degrees of freedom. For example, both the ketchup bottle in Exp. I and the water bottle in Exp. III are not constrained by the fingers along (z) or around (ψ) the vertical symmetry axis of these objects. Therefore, neither DOF is observable from finger measurements alone. However, the fusion with visual features is able to complement the estimation in this regard. All experiments show significant improvements from the incorporation of visual information, both from AprilTags and automatically extracted features. Exp. III and Exp. V also demonstrate how even partial visual information can help to inform the estimation. In both cases, only features on the cap of the bottle or the pen could be reliably extracted. One shortcoming of the proposed method for automatic feature extraction is that it only considers contour features. Particularly in Exp. III, the resulting lack of observability of ψ causes a significant error in orientation.

V. CONCLUSION

This paper proposed a new method for the estimation and tracking of the grasp state of a manipulated object by combining position and torque measurements from the manipulator with visual features that are extracted from a monocular camera image. By fusing these different measurement modalities using an extended Kalman filter, the algorithm is able to resolve collisions in the estimated grasp configuration, infer the contact configuration and align the estimated object with the camera view of the robot. Experiments with the DLR robot David and a variety of different objects illustrated the performance of the method in general, as well as the strengths and limitations of the different types of measurements.

Future work will continue in several directions. First, the scope of the algorithm can be extended. Torque measurements not only allow the detection of contacts, they also make it possible to continuously estimate the contact forces. Second, the availability of a reliable estimate of the grasp state enables research on more complex manipulation scenarios, such as in-hand manipulation.

REFERENCES

[1] M. Ulrich, C. Wiedemann, and C. Steger, "CAD-based recognition of 3d objects in monocular images." in *2009 IEEE International Conference on Robotics and Automation*, vol. 9, pp. 1191–1198.

[2] J. Bimbo, L. D. Seneviratne, K. Althoefer, and H. Liu, "Combining touch and vision for the estimation of an object's pose during manipulation," in *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 4021–4026.

[3] J. Bimbo, P. Kormushev, K. Althoefer, and H. Liu, "Global estimation of an objects pose using tactile sensing," *Advanced Robotics*, vol. 29, no. 5, pp. 363–374, 2015.

[4] L. Zhang and J. C. Trinkle, "The application of particle filtering to grasping acquisition with visual occlusion and tactile sensing," in *2012 IEEE International Conference on Robotics and Automation*, pp. 3805–3812.

[5] L. Zhang, S. Lyu, and J. Trinkle, "A dynamic bayesian approach to real-time estimation and filtering in grasp acquisition," in *2013 IEEE International Conference on Robotics and Automation*, pp. 85–92.

[6] S. Li, S. Lyu, and J. Trinkle, "State estimation for dynamic systems with intermittent contact," in *2015 IEEE International Conference on Robotics and Automation*, pp. 3709–3715.

[7] C. Corcoran and R. Platt, "A measurement model for tracking hand-object state during dexterous manipulation," in *2011 IEEE International Conference on Robotics and Automation*, pp. 4302–4308.

[8] G. Izatt, G. Mirano, E. Adelson, and R. Tedrake, "Tracking objects with point clouds from vision and touch," in *2017 IEEE International Conference on Robotics and Automation*, pp. 4000–4007.

[9] T. Schmidt, K. Hertkorn, R. Newcombe, Z. Marton, M. Suppa, and D. Fox, "Depth-based tracking with physical constraints for robot manipulation," in *2015 IEEE International Conference on Robotics and Automation*, pp. 119–126.

[10] 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 *2011 IEEE International Conference on Robotics and Automation*, pp. 5935–5941.

[11] M. Pfanne and M. Chalon, "EKF-based in-hand object localization from joint position and torque measurements," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 2464–2470.

[12] M. Chalon, J. Reinecke, and M. Pfanne, "Online in-hand object localization," in *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 2977–2984.

[13] E. G. Gilbert, D. W. Johnson, and S. S. Keerthi, "A fast procedure for computing the distance between complex objects in three-dimensional space," *IEEE Journal on Robotics and Automation*, vol. 4, no. 2, pp. 193–203, 1988.

[14] H. Durrant-Whyte and T. Bailey, "Simultaneous localization and mapping: part i," *IEEE Robotics & Automation Magazine*, vol. 13, no. 2, pp. 99–110, 2006.

[15] B. Siciliano and O. Khatib, *Springer Handbook of Robotics*. Springer, 2008, ch. 28. Grasping, pp. 671–700.

[16] E. Olson, "Apriltag: A robust and flexible visual fiducial system," in *2011 IEEE International Conference on Robotics and Automation*, pp. 3400–3407.

[17] M. Grebenstein, A. Albu-Schäffer, T. Bahls, M. Chalon, O. Eiberger, W. Friedl, *et al.*, "The DLR hand arm system," in *2011 IEEE International Conference on Robotics and Automation*, pp. 3175–3182.

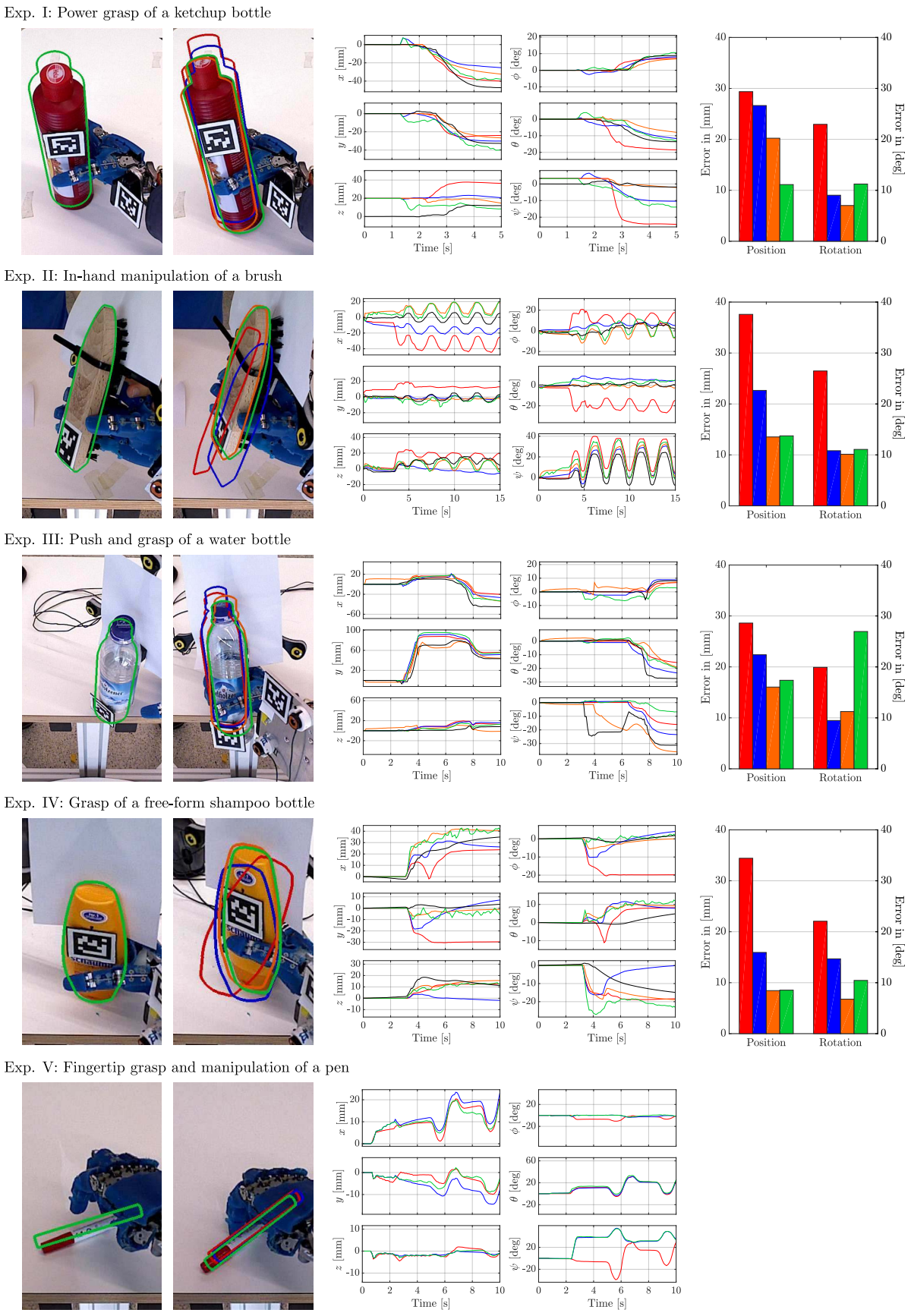


Fig. 7. Left image: Camera view of the object before the grasp including the initial estimate (green). Right image: Final view and estimation results. Center graphs: Change in position and orientation of the ground truth (black) compared to the estimation. Right graph: Terminal absolute errors in position and orientation at the end of the experiment. The colors denote the different combinations of measurements as follows: joint positions (red), joint positions and torques (blue), both joint measurements and AprilTag features (orange), and both joint measurements and automatically extracted features (green).