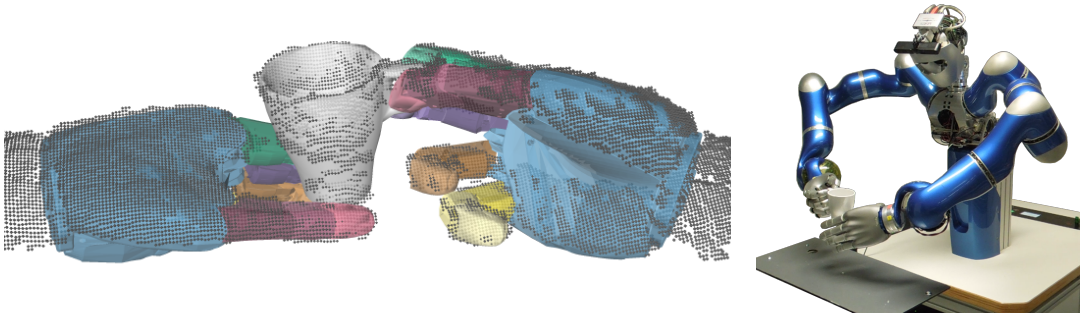
Depth-Based Tracking with Physical Constraints for Robot Manipulation

Tanner Schmidt[1], Katharina Hertkorn[2], Richard Newcombe[1], Zoltan Marton[2], Michael Suppa[2], Dieter Fox[1]



***Abstract*— This work integrates visual and physical constraints to perform real-time depth-only tracking of articulated objects, with a focus on tracking a robot's manipulators and manipulation targets in realistic scenarios. As such, we extend DART, an existing visual articulated object tracker, to additionally avoid interpenetration of multiple interacting objects, and to make use of contact information collected via torque sensors or touch sensors. To achieve greater stability, the tracker uses a switching model to detect when an object is stationary relative to the table or relative to the palm and then uses information from multiple frames to converge to an accurate and stable estimate. Deviation from stable states is detected in order to remain robust to failed grasps and dropped objects. The tracker is integrated into a shared autonomy system in which it provides state estimates used by a grasp planner and the controller of two anthropomorphic hands. We demonstrate the advantages and performance of the tracking system in simulation and on a real robot. Qualitative results are also provided for a number of challenging manipulations that are made possible by the speed, accuracy, and stability of the tracking system.**

I. Introduction

In order for a robot to successfully manipulate objects using a model-based planner, the position and orientation of the objects relative to the manipulator and the current values of all manipulator joint angles are needed. A common solution is to visually track the object within the frame of reference of some camera; the object pose is then related to the hand pose through a transform from the camera frame of reference to the robot base frame of reference followed by a transform from the robot base frame of reference to the hand frame of reference [1], [2]. The latter depends on a calibrated proprioceptive system, while the former depends on a an extrinsic camera calibration. Any small errors in these calibration procedures can easily add up to large errors in the relative hand-to-object pose, potentially leading to failure of intended manipulations. Knowledge of manipulator joint angles can also be inaccurate for tendon driven or uncalibrated robots, further complicating grasp planning.

Also key to successful manipulation is the stability of the object pose estimate. If a manipulation is planned based on the currently estimated poses, but the estimate changes significantly in the intervening time before the manipulation is executed, the plan may no longer be valid. A good vision system for manipulation must therefore extract stable and accurate estimates even from noisy input data.

We approach the problems caused by calibration errors by tracking both the object and the robot hands in the camera frame of reference. Both poses are then subject to the same intrinsic calibration errors and are already in the same frame of reference, removing the need for extrinsic calibration altogether, and allowing for more accurate relative pose estimation. We perform the tracking by extending our previous DART tracking framework [3], and show how our signed distance function representation of the tracked models lends itself easily to incorporating additional physics-based terms into the error function. Specifically, we incorporate terms which penalize object interpenetration and disagreement between touch sensor feedback and physical contact between objects.

The issue of estimate stability is handled by using a switching model, which combines several frames of data to converge to an accurate and more stable object pose estimate when the object is stationary, either relative to the camera (while on the table) or relative to one of the hands (while firmly grasped). Our switching model requires an automated method for detecting when an object enters and exits these stable states in order to remain robust to unexpected motion of the object. We found that the dense visual error term provides a strong signal for this detection.

In related work, the poses of manipulators and manipulation targets are generally estimated using either physical information, visual information, information from tactile sensors, or some combination of the three. Physical laws

[1] Dept. of Computer Science & Engineering, University of Washington, Seattle, USA. {tws10,newcombe,fox}@cs.washington.edu

[2] Institute of Robotics and Mechatronics, German Aerospace Center (DLR), Wessling, Germany. {firstname.lastname}@dlr.de

provide a rich source of information, as it is known with certainty that the state of the system must obey these laws. However, ensuring that all laws are obeyed becomes rather computationally complex in manipulation scenarios where the normal and frictional forces of contact are in effect, many contacts can occur (during grasping), and where impulse forces are common. Furthermore, it may appear that a state violates physical laws when in fact the problem might simply be due to the fact that the models are not perfect and the sensor measurements are noisy. In this paper, we close the action-perception loop for manipulation by jointly tracking the robot hands, the finger joint angles, and the object pose. We attempt to respect physical constraints as far as possible by penalizing estimates that imply interpenetration or that are inconsistent with the contact estimation. Otherwise, we allow the physical system itself to provide the physics 'simulation' for us, and use dense visual information to estimate what has happened. The estimated state is reported back to the robot in real-time allowing for online interaction.

This paper is organized as follows. After discussing related work, Section III introduces our tracking framework. Experiments are given in Section IV, followed by conclusions.

II. Related Work

Physics-based tracking: Lowrey et al. use a physics engine to estimate the state of a walking robot based on a variety of sensors within the robot as well as a fast and highly accurate external marker-based tracking system [4]. While walking, the robot's feet repeatedly make contact with the ground, and as such the simulator is required to reason about contact forces. To reduce noise, they recompute estimates in a sliding window whenever new data has arrived.

Vision-based tracking: Many techniques used in object and robot tracking systems rely on distinctive keypoints in the image data. Azad et al. propose a method for tracking (single, rigid) 3D models by instead rendering their edges and computing the overlap between the image edges [5]. Ulrich et al. similarly match 3D CAD models in monocular images [6]. In contrast, we use 2.5D sensor data to solve for model poses directly in the 3D space. As one part of the 3D visual tracking, we take advantage of the implicitly represented difference between free and occluded space present in the depth map. Earlier image-based approaches to tracking articulated models did this indirectly with background subtraction and by using multiple cameras to obtain a sense of the 3D layout [7]. Klingensmith et al. presented a visual servoing system using an articulated iterative closest point (ICP) variant to track a robot arm, demonstrating that using visual feedback in the control of a robot enables greater robustness to calibration errors [8]. Our work demonstrates the value of visual feedback for tracking manipulated objects in addition to the robot itself.

Krainin et al. also use articulated ICP, augmented with sparse feature matching and dense color information to track a robot arm and a previously unknown object it has grasped, and to simultaneously build a model of the object [9]. Our articulated ICP variant, accelerated by signed distance function lookups and a GPU implementation, allows us to achieve real-time performance while additionally reasoning about physics-based constraints. Schulman et al. use color and depth observations to induce 'observation forces' on a deformable object model and are thereby able to use a physics simulator to find the low-energy state which has the least disagreement with the observation [10]. This work has been developed specifically for deformable objects and it is not clear how it would perform in our setting, which requires much faster update rates and the incorporation of additional constraints.

Contact-sensing-based tracking: Haidacher and Hirzinger presented 'a blind man's approach to grasping' in which they search through a set of possible pairings of fingers and object faces to find the pose of an object using only contact detection on the fingers [11]. Koval et al. propose a particle filter that tracks manipulated objects using only tactile information by sampling particles from a manifold of state space that respects contact constraints [12]. However, it seems unlikely that either of these approaches could provide accuracy commensurate with that provided by a dense visual data term minimized by gradient descent, unless a very large number of particles is used such that one always lands on the correct solution. Furthermore, manipulation of objects involves states in which the hand is not in contact, in which case contact-only methods can provide no information about the location of the object.

Combined manipulated object tracking: Zhang and Trinkle presented an offline particle filter approach that combines visual information, physics, and tactile feedback to track manipulated objects [13]. While they also focus on tracking through occlusion induced by manipulators, the slow update rate makes the approach less applicable to real-time manipulation scenarios. Chalon et al., rather than tracking visually through occlusions, use a vision system to initialize the object pose and then rely on physics and potentially contact information to track a manipulation with a particle filter [14]. As they use the grasp matrix to update the state of the particle filter, manipulation of objects without grasping is not considered. Bimbo et al. also approach the occlusion problem by fusing tactile information with visual information, but they do not take intersections between object and model into account [15].

III. Tracking Framework

Our tracking framework is an extension of the DART tracking system [3]. DART enables tracking of articulated objects in real-time, i.e. 30 fps, by storing the models implicitly as a collection of rigid signed distance functions (SDFs) that move relative to each other according to a kinematic tree, and by optimizing a dense visual data term on the GPU. In this paper, we add physical constraints to the previously-presented objective function. The tracker takes as input a depth map, D, and tries to estimate a vector θ describing the tracked state as depicted in that depth frame. This state vector includes the position, orientation, and articulation of all models, as well as the location of contact points between

models (described in more detail in section III-B). This is done iteratively using gradient descent. The full error function to be minimized is as follows:

$$\begin{aligned} E(\theta; D) =& E_{\text{mod}}(\theta; D) + \lambda_{\text{obs}} E_{\text{obs}}(\theta; D) \\ &+ \lambda_{\text{int}} E_{\text{int}}(\theta) + \lambda_{\text{con}} E_{\text{con}}(\theta) \ , \end{aligned} \tag{1}$$

where E_{mod} and E_{obs} represent the error terms in the original DART framework, measuring the error induced by observed points in the model SDF and by predicted model points in the observation SDF, respectively. We will now describe the final two terms, which are contributions of this work.

A. Intersection Term

Based on the simplest physical principles, we know a priori that the union of physical space occupied by any pair of rigid bodies must be empty. The visual terms alone should be sufficient to ensure that the pose estimates satisfy this condition under full visibility, but in times of heavy occlusion, as are common in manipulation scenarios, this physical constraint becomes a useful source of information for estimating otherwise unconstrained degrees of freedom.

Suppose there are two rigid bodies, A and B, represented implicitly with continuous signed distance functions $f_a(x, y, z)$ and $f_b(x, y, z)$, which give for every point in 3D space the shortest distance to the surface of the respective rigid body. Noting that SDFs are *negative* inside a body, a natural way to penalize the intersection of these two bodies would be as follows:

$$\iiint \min(0, f_a(x, y, z)) \min(0, f_b(x, y, z)) \, \mathrm{d}x \, \mathrm{d}y \, \mathrm{d}z \, . \tag{2}$$

However, a triple integral over a discrete representation of a signed distance function would be quite expensive. We then note that taking interior parts of both models into account simultaneously is unnecessary, as no point on the interior of B can be inside of A without some point on the surface of B having penetrated first. We can then simplify the triple integral by replacing it with two surface integrals:

$$\oiint \min(0, f_a^2(x, y, z)) \, \mathrm{d}S_B + \oiint \min(0, f_b^2(x, y, z)) \, \mathrm{d}S_A \ , \tag{3}$$

where S_A and S_B are the surfaces of the two rigid bodies. Finally, we can discretize this surface integral by considering only a finite set of points on the surface of all rigid bodies. Thus, as a pre-processing step we store a collection of points X^m for each model m in our set of tracked models, M, such that all $x \in X^m$ lie on the surface of model m. This is done by sampling points on the mesh faces uniformly by surface area. The points are stored in the frame of reference of the rigid body from which they were generated, and then transformed into the global frame via the kinematic chain of the articulated model. The error induced by points on model m_a penetrating model m_b is given by:

$$e_{\text{int}}^{m_a, m_b}(\theta) = \sum_{x_i \in X^{m_a}} \min(0, SDF_{m_b}(T_{m_b, f_i}(\theta) x_i))^2 \ , \tag{4}$$

where f_i is the frame of reference in which point x_i moves rigidly, and T_{m_b, f_i} is the transform from that local frame of reference to the frame of reference of model m_b, as defined by the kinematic chain and the current parameter estimates θ. The full intersection term we then minimize is:

$$E_{\text{int}}(\theta) = \sum_{m_a \in M} \sum_{m_b \in M} e_{\text{int}}^{m_a, m_b}(\theta) \ . \tag{5}$$

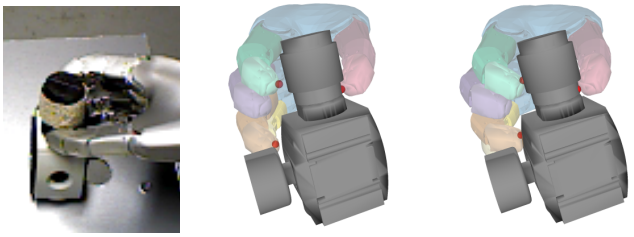
Note that (4) is directional, so we consider in (5) both the possibility of points from m_a intersecting m_b and vice versa. We also consider points from m_a self-intersecting other parts of model m_a, which can happen if it is an articulated model.

This formulation is almost identical to the error induced by observed points in the model SDF as presented in previous work, except for the truncation of the SDF to interior regions. We are thus able to compute first order derivatives for the error induced by intersecting points exactly as in [3].

B. Contact Term

While the intersection term applies generally to any rigid body, many robot hands can provide additionally helpful information by sensing when contact has been made with other surfaces. This is particularly useful in grasping scenarios, as many natural grasps involve the placement of fingers behind the object, where they are visually occluded.

Given the location of contact on the finger, perhaps from a high-resolution tactile sensor, the goal would be to simply minimize the squared distance from the contact point on the finger to the surface of the object, such that the pose estimate will reflect that contact is in fact being made. However, we are interested in a wider range of systems in which the existence of contact can be detected, perhaps via torque sensors, but not the location of the contact point; this location then becomes a hidden variable that needs to be estimated. While this point is minimally represented as a point on the two-dimensional manifold of the finger surface, we instead chose to simplify the optimization objective, and overparameterize the variable by representing the contact location as a 3D point in the frame of reference of the finger. We simply project the estimated point back onto the finger surface after every step of the gradient descent.



(a) (b) (c)

Fig. 1. (a) A grasp in which contacting fingers are occluded is executed. (b) and (c) show a rendering of this grasp from another view point; in (b) the contacts are first detected via the torque sensors in the hand, and in (c) the estimate of the object pose, hand pose, and the location of the contact points is updated such that the identified fingers touch the object (see red dots, best viewed when enlarged).

We define a variable c_i for each finger which takes on a value of 1 if contact was detected on that finger and a value of 0 otherwise. Our contact error term is then:

$$E_{\text{con}}(\theta) = \sum_i c_i \text{SDF}_{\text{obj}}(T_{obj,i}(\theta)\mathbf{p}_i(\theta))^2 \ , \qquad (6)$$

where $\mathbf{p}_i(\theta)$ is the estimate of the contact point on finger i. Once again, we have an error based on the lookup of an implicit point-to-surface distance of a point defined in a local frame of reference in a signed distance function. The derivatives of this term with respect to the hand and object poses are thus computed as in the intersection or observation to model error terms. For the derivatives with respect to each contact location, we have:

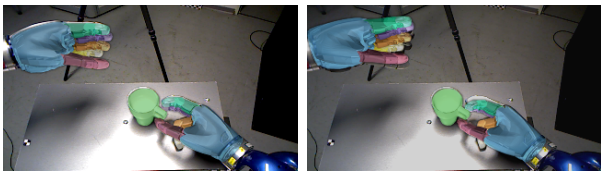
$$\frac{\partial}{\partial \mathbf{p}_i} E_{\text{con}}(\theta) = c_i \nabla \text{SDF}_{\text{obj}}(T_{obj,i}(\theta)\mathbf{p}_i(\theta)) \ , \qquad (7)$$

which is simply the gradient of the object SDF evaluated at the contact point. The effect of this term can be seen in Fig. 1.

C. Parameterization

When tracking robots that are capable of providing proprioceptive feedback, we do not have to start from scratch in pose estimation. However, the combination of calibration errors in the proprioceptive system and the camera means that the joint angles reported by a robot cannot be trusted exactly, as demonstrated in Fig. 2. We therefore use the reported joint angles, but do not rely on them entirely.

One approach for incorporating proprioceptive information into a visual, gradient-descent-based tracking framework is to treat reported joint angles as a prior in the Bayesian sense [9]. However, with our focus on occlusions, we found this approach to be suboptimal, due to a 'snap-back effect': when any degree of freedom becomes unobserved, it simply reverts back to the reported prior regardless of whether that prior had matched observed information in recent frames.



(a) Estimated state (b) Reported state

(c) Estimated state (d) Reported state

Fig. 2. In (a) and (c) we show tracking estimates overlaid on some exemplar frames. In (b) and (d), the right hand is shown where it has been estimated, while the left hand is positioned relative to the right hand according to the forward kinematics and reported joint angles of the left and right arms, and the fingers are shown positioned as reported. Note the errors that are particularly evident in the left thumb and right index fingers of (b) and (d).

We instead took the approach of Klingensmith et al., choosing parameters that represent relative offsets from the reported joint angles rather than absolute angles [8]. That is, at each time step, instead of finding the absolute joint angles θ that minimize the error function, we optimize over relative joint angles δ that minimize the error when added to reported joint angles $\hat{\theta}$:

$$\delta^* = \arg\min_{\delta} E(\hat{\theta} + \delta; D) \ , \qquad (8)$$

where $E(\theta; D)$ is the error function of equation (1). This parameterization has some nice properties, first and foremost being that if we assume the offset between the actual and reported joint angles is fairly constant, we can highly regularize our gradient descent steps to ensure the relative values change slowly and are less sensitive to sensor noise. As the amount of regularization approaches infinity, the tracker will follow the joint angles exactly, while a high but finite value allows for slow and steady adjustments to accommodate bias inherent to certain camera views and errors in system calibration (or lack thereof), and also allows the tracker to function when the offset is not, in fact, constant.

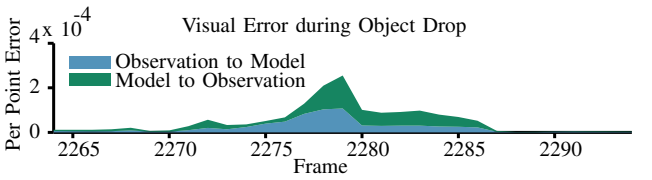
D. Switching Model

While we have a relatively strong signal as to where the hands are through the forward kinematics of the robot arms, we are not so lucky when it comes to the object position. To produce stable estimates of the relative transform between hand and object as needed by the grasp planner, we follow the work of Krainin et al. [9] in identifying three possible states of the object:

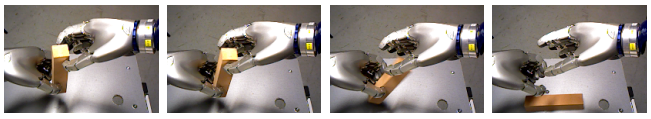
1) The object is at rest on a surface
2) The object is in an intermediate unstable state
3) The object is stably grasped by a hand

However, while Krainin et al. assume that they know the state based on the actions taken by the robot (i.e. the object is on the surface when the robot places it on the surface and stably grasped when the robot has performed a grasp), we make no such assumptions. Instead, we present an automatic switching model that detects state transitions based on observations of the model, making our tracker robust to failed grasps and slippage or unintended dropping of grasped objects, and also enables manipulations that don't involve grasping at all, such as pushing an object along a surface.

In order to detect state transitions, we first define our measure of the 'visual error' of a particular estimate as the average distance from all observed object points to the object model SDF surface, plus the average distance from all predicted object model points to the nearest unobserved space in the observation SDF. So, if O observed points are associated with the object and the predicted depth map has P object points, this metric will be $\frac{E_{\text{mod}}}{O} + \frac{E_{\text{obs}}}{P}$. The intuition here is that when the estimate is accurate, all predicted object points and all observed points data associated to the object should have low error, and when the object moves, there will be at least some predicted and observed points with high error, as shown in Fig. 3.



(a)



(b) Frame 2265 (c) Frame 2270 (d) Frame 2275 (e) Frame 2290

Fig. 3. Visual error for a sequence of tracked object pose estimates, before optimization. The object begins in a stable grasp state, transitions to the intermediate state (frame 2267) due to the high error indicating the object is not where it is expected to be based odometry update composed on thelast frame's estimate, and finally enters the stable rest state (frame 2287).

The object is always initialized in state 2, and transitions to one of the two stable states once the pre-optimization error metric falls below a pre-defined threshold. If the fingers are detecting contact when this happens, the object transitions to state 3, and otherwise transitions to state 1. The threshold is set just above the visual error typically observed when the object is at rest on a surface, which depends on the noise model of the sensor in use but is easily determined empirically.

The estimation of object state is useful in that we can assume that the object is stationary in the stable states — stationary relative to the world frame in state 1 and stationary relative to the palm frame in state 3. Based on this assumption, we could collect a series of frames and solve for the 6 parameters which minimize the sum of the error in all frames. We approximate this in a real-time and online manner by keeping a running estimate of the amount of information we have about each of the 6 degrees of freedom. We first note that our Hessian approximation $H = J^T J$ is the inverse of the covariance matrix, and therefore that entries along the diagonal are inversely proportional to the (squared) uncertainty of the corresponding variables, i.e. $H_{ii} = \frac{1}{\sigma_i^2}$. We therefore store a running sum, ι, of the inverse uncertainty of the object pose parameters in all the frames since entering a stable state. If the object is estimated to be in a stable state at time step t, we set:

$$\iota^{t+1} := \min(\iota^t + \alpha \text{diag}(H^t), \iota_{\text{max}}) \ , \tag{9}$$

where α is an 'accumulation rate' parameter set and ι_{max} limits the amount of regularization that can be applied to any particular degree of freedom. The exact value of α is not critical and is generally set anywhere between 0.2 and 0.8 in our experiments. The parameter ι_{max} actually has a unit, namely the inverse square of the unit of the corresponding degree of freedom, and is set accordingly. Then, we solve the following regularized normal equations for each Gauss-Newton step:

$$\Delta\theta^t = (J^T J + \iota^t)^{-1} eJ \ , \tag{10}$$

As a result, after the object has remained in a stable state for a number of frames, the estimate will change more slowly, depending on the amount of uncertainty in all frames, and should thus converge over time to a stable estimate of the true object pose and no longer be influenced by noise on a frame-to-frame basis. Crucially, each degree of freedom is regularized differently depending on our certainty of its state. Once we have detected a deviation from the stable state as previously described, we assume previous information is no longer valid and set $\iota = 0$ until we have re-entered another stable state.

IV. Experiments

As a platform for the proposed method, we use a shared autonomy system consisting of DLR's telepresence robot combined with online grasp planning [2], as described in Section IV-A. However, the only system-dependent assumption made is that the hands are capable of estimating whether each finger is in contact with an object, and the tracking approach should apply equally well to any system that satisfies this requirement. Grasps for two five-finger hands are planned online according to the current hand-to-object relative pose. This allows for planning of grasps in unexpected situations as the grasps are not restricted to those available in a grasp database. The shared autonomy setup poses additional challenges to the tracking system, as the human operator can move the arms freely (in contrast to executing a pre-planned motion), potentially bumping the objects or otherwise changing the hand-to-object relative pose, and the new pose is needed by the grasp planner before a new grasp can be planned. Following previous work [2], we use the static scene analysis as an initial guess for the presented tracking method, which, being a local gradient descent method, needs a rough initialization. We perform a quantitative evaluation of the tracking accuracy on real manipulations and show the benefits of the phsyical constraints using a synthetic manipulation as a baseline. In the accompanying video, we also present tracking estimates for a number of challenging manipulations for qualitative

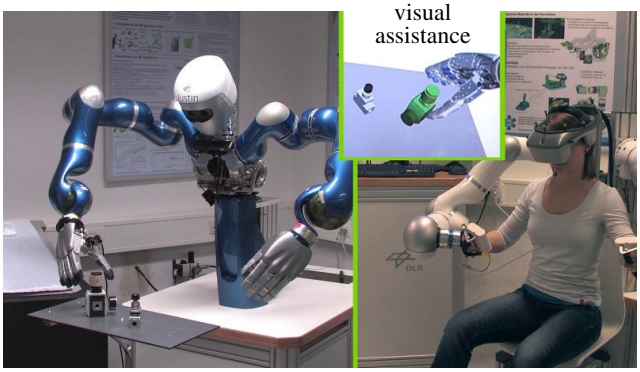


Fig. 4. Semi-autonomous grasping using DLR's telepresence system. It consists of the remote robot SpaceJustin (on the left) and the human-machine interface HUG (on the right). The visual assistance is displayed using the head-mounted display.

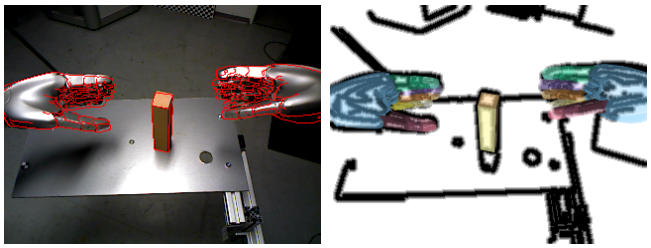


Fig. 5. Left: tracked joint edges overlayed with color image. Right: distance map of the color image's edges, with a cutoff radius of 5 pixels. Note missed edges at the right hand (insufficient contrast with background).

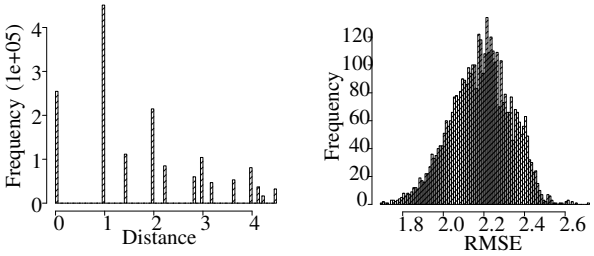


Fig. 6. Left: frequencies of distances between corresponding edge pixel positions in all frames. Right: distribution of mean per-frame RMSE of the distances between corresponding edge pixels.

analysis. All experiments were run with the error function weighted according to $\lambda_{\text{mod}} = 4$, $\lambda_{\text{int}} = 10$, and $\lambda_{\text{con}} = 25$.

A. System

The system consists of the multimodal human machine interface HUG [16] and the remote robot SpaceJustin, which is a modified version of DLR's humanoid robot Justin [17] (see Fig. 4). SpaceJustin has 17 actuated degrees of freedom (DoF) for its torso, head, and arms, and interacts with the environment with two DLR-HIT Hands II which have 15 DoF's each [18]. An Asus Xtion is mounted on the head and is used to track the hands, fingers, and the object. The robot arms of HUG and SpaceJustin are coupled in Cartesian space which allows the operator to control the movements of SpaceJustin's arms and experience realistic force feedback. A one DoF hand interface is used to trigger the grasping command once a stable grasp is planned. The operator perceives visual feedback by wearing a head-mounted display (HMD, NVisorSX60 from NVIS) showing the remote and estimated environment in 3D. In order to allow a high degree of immersion, the operator's head movements are also tracked, enabling control of the movement of SpaceJustin's head (and thus the position of the camera as well).

The robot system reports its current joint angles (47 in total) to the tracking system at a rate of 1 kHz. Additionally, the fingers provide a vector of binary inputs to the tracking system indicating whether each finger is currently in contact with a foreign object, as estimated by the torque sensors integrated in every finger. We do not estimate the contact point on the finger due to a lack of tactile sensing. The Asus Xtion provides depth information at a rate of 30 fps. Tracking and grasp planning run at this update rate. The difference between estimated joint angles and measured joint angles as calculated by the tracking system is sent back and taken into account by the control of the hands to correct for the errors in forward kinematics.

B. Accuracy

Since we use an uncalibrated system, the accuracy of the tracking is difficult to quantify without using a high-quality external tracking system and introducing markers with precisely known positions relative to the joints. Therefore, the accuracy of the proposed method is evaluated against an independent information source, namely the RGB image from the Asus Xtion, by comparing it to the projection of the models into the 2D image plane according to the pose estimate, as in Fig. 9.

Similar to [5], we measure the agreement between the edge maps computed on the color image and the estimate rendering by assuming nearest neighbors in the edge maps are in correspondence, up to some distance threshold. We report the root mean square error (RMSE) of the nearest neighbor distances in a frame as a quality-of-fit metric.

The drawback of the method is that it requires strong intrinsic calibration and edges that can be reliably detected in the color image, which we don't always have. However, the computed scores provide a more accurate measure than the proportion of overlapping edge pixels used by Azad et al., and give some intuition into the tracking quality.

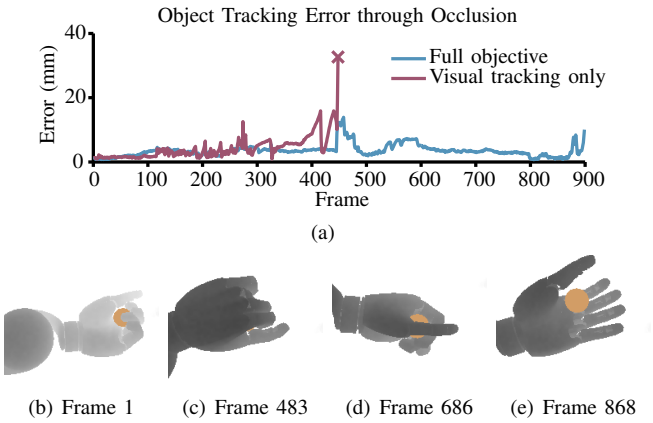
It is important to avoid incorrect correspondences by setting the cutoff distance to a reasonable level. As the tracking is qualitatively correct, we chose a strict threshold of 5 pixels in the 320×240 RGB images from the Xtion. As shown in Fig. 5, the edges of the tracked joints are within this limit, and we use the maximum value of 5 (rendered white) when they are not, or when an edge was not detected.

The result for a sequence of 4085 frames (including the ones in Fig. 3), shown in Fig. 6, indicate that most edges are overlapping or neighboring. The RMSE for each frame (which penalizes large errors more strongly) is around 2.2 pixels. Given the unverified intrinsic and extrinsic calibration of the color and depth sensors, and the other problems discussed above, the real error is probably lower, and the results are visually appealing.

C. Synthetic Experiment

To show the contribution of newly introduced components of the tracking system, we generated a synthetic manipulation of a small sphere using the Bullet physics engine[1] and a corresponding sequence of depth maps. The trajectory from the physics simulation gave us a baseline for comparison, although it should be noted that the simulation was not perfect. The rendered depth maps were corrupted with white noise with a standard deviation of 2 mm, sampled at a quarter of the depth map resolution, such that the noise in neighboring pixels is correlated, and discretized into 1 mm bins as in consumer depth sensors. The known true odometry is corrupted by a zero-mean Gaussian walk.

[1] http://bulletphysics.org/



(a)

(b) Frame 1 (c) Frame 483 (d) Frame 686 (e) Frame 868

Fig. 7. (a) The error in 3D of the estimate of the position of a sphere being manipulated by a robot hand in a synthetic sequence. (b)-(e) show a selection of frames from the sequence with the sphere highlighted in orange. Without physical constraints, tracking fails when the sphere becomes hidden.

We compare in Fig. 7 the tracking performance of the original vision-only DART system and of the presented system on this extremely challenging sequence, where the error metric is given by the Euclidean distance between the predicted and ground truth sphere center. As shown, the physical constraints lead to more stable estimates through the extreme occlusions in this sequence. Furthermore, once the ball becomes nearly entirely invisible to the camera (Fig. 7(c)), the vision-only system experiences a tracking failure from which it does not recover. With physical constraints, tracking is successfully maintained through the entire sequence, with a mean error of 3.5 mm.

D. Real Experiments

Due to the aforementioned difficulties in performing quantitative evaluation of markerless tracking methods, we also qualitatively demonstrate a number of manipulations we were able to perform due to the accurate and fast pose estimates provided by the presented tracking system, all of which may not even have been possible otherwise. The experiments are conducted using the same method for all objects and situations; no tuning of parameters is needed either in the tracking or the grasp planning.

The first challenging manipulation is the execution of a grasp with a narrow margin of error, which demands a high degree of accuracy, as any deviation in the estimated joint angles can cause the fingers to miss the small intended targets. Fig. 8 shows one such challenging grasp, in which three fingers are used to grasp a small handle on a coffee mug. Not only is the target small, but the mug surface is slippery and the center of mass is far from the contact points, all of which contribute to the difficulty of the grasp.

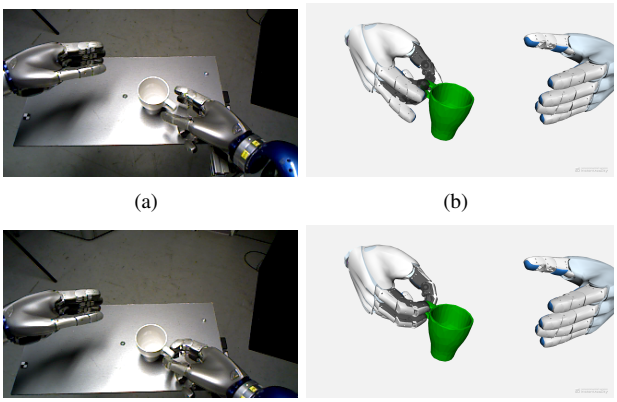
Another challenge for a tracking system is the movement of the camera, as we want the operator to be able to change her viewpoint to facilitate manipulation. Handling this using only the forward kinematics of the robot would require highly accurate knowledge of the camera position relative to the neck joints, as well as a very high quality intrinsic calibration of the camera. Instead, we use only a very rough initial estimate of the extrinsic calibration to predict the motion of the hands and object relative to the camera during neck motion, and then rely on the visual tracking to account for the errors. An example of our robustness to changing camera viewpoint can be seen in Fig. 9.

We were mostly interested in bi-manual manipulations, which pose additional challenges to the tracking system. In previous work on the shared autonomy system [2], the object pose was estimated while it lay stationary on the table, not occluded by either hand. When handing the object from one hand to the other, however, the grasp planner needs to know the relative transformation between the grasping hand and an object which is not on the table, is occluded by the fingers of the other hand, and could be moving.

Finally, we did not want to make the limiting assumption that every grasp attempted by the operator would be successful, and we did not want to have to restart the tracking system every time an object moved unexpectedly. We show in the accompanying video that while our tracking system is able to use information from multiple frames to stabilize object pose estimates, it is also able to detect unexpected motion and react quickly when the object moves during grasping or drops suddenly from the hand. In these cases, the operator is usually able to immediately begin correcting the failure, as an accurate estimate of the object pose is still available.

V. Conclusions

We presented the benefit of incorporating proprioceptive information and physical constraints into dense visual articulated model tracking, and applied it to jointly track robot hands and the objects they manipulate. The implicit signed distance function representation of the tracked models allows us to easily detect interpenetration of multiple interacting models, as well as self-intersection of a single model, and to correct these intersections by adding a new term to the visual error function. After introducing a hidden variable for the location of contact between models, we are also able to use our SDF representation to add a fourth term into the error function which favors pose estimates which



(a) (b)

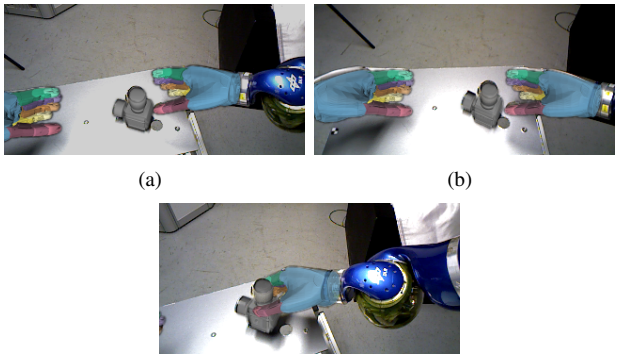
(c) (d)

Fig. 8. A grasp that demands high accuracy. (a) and (c) show the robot's point of view, (b) and (d) show the estimated joint angles and relative hand-to-object poses, as well as the planned grasp (grey fingers).

explain detected contacts. The intersection term is as trivially parallelizable as the visual error terms, and the contact term is processed at essentially no cost on the CPU as all other terms are being processed on the GPU. We are therefore able to use the model pose estimates for planning and control in a bi-manual shared autonomy system, achieving real-time update rates of 30 frames per second while tracking 48 pose parameters (two 21 degree of freedom hands plus a 6 degree of freedom object). We also introduced an automatic switching model that detects stationary states of the object and reacts accordingly, in order to provide tracking estimates that are stable and highly accurate when the object is not moving, yet keep up when the object does move quickly.

In the supplemental video, we show a number of manipulations that were enabled by using the estimates provided by the tracking system to plan and execute grasps. We sought to make the manipulations challenging by executing grasps with little room for error, such as lifting a coffee mug by a small handle, by moving the camera throughout the manipulation without accurate knowledge of the camera position relative to the kinematic tree, and by grasping objects that are already held (and thus occluded) in the other hand. Throughout the course of the manipulations, there were also failed grasps, unintentional drops, and non-grasping manipulations such as pushing the object across the table or using one hand to rotate an object held in the other hand, and we are able to maintain accurate tracking through these situations as well.

While we demonstrated all results on tracking manipulations executed by a shared autonomy system, the presented tracking method is equally applicable to fully autonomous systems; which we intend to investigate in future work. It would also be interesting to investigate algorithms for automatically positioning the camera to reduce uncertainty in the object pose parameters. Another extension of this work would be to do real-time model building, as in [9]. This could be done using a KinectFusion-style truncated signed distance function to represent the model, allowing the tracker to use the partial model for tracking with few changes to the method [19].



(a) (b)

(c)

Fig. 9. Tracking of the hands and object is maintained through changes in camera view.

ACKNOWLEDGMENTS

This work was funded in part by the Intel Science and Technology Center for Pervasive Computing (ISTC-PC) and by ONR grant N00014-13-1-0720.

REFERENCES

[1] A. Morales, T. Asfour, P. Azad, S. Knoop, and R. Dillmann, "Integrated grasp planning and visual object localization for a humanoid robot with five-fingered hands," in *Proc. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, 2006, pp. 5663–5668.

[2] K. Hertkorn, M. Roa, M. Brucker, P. Kremer, and C. Borst, "Virtual reality support for teleoperation using online grasp planning," in *Proc. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, 2013.

[3] T. Schmidt, R. Newcombe, and D. Fox, "Dart: Dense articulated real-time tracking," in *Proceedings of Robotics: Science and Systems*, July 2014.

[4] K. Lowrey, S. Kolev, Y. Tassa, T. Erez, and E. Todorov, "Physically-consistent sensor fusion in contact-rich behaviors," in *Proc. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, 2014.

[5] P. Azad, D. Munch, T. Asfour, and R. Dillmann, "6-DOF Model-based Tracking of Arbitrarily Shaped 3D Objects," 2011.

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

[7] J. Bandouch, O. C. Jenkins, and M. Beetz, "A self-training approach for visual tracking and recognition of complex human activity patterns," *International Journal of Computer Vision*, vol. 99, no. 2, pp. 166–189, 2012.

[8] M. Klingensmith, T. Galluzzo, C. Dellin, M. Kazemi, J. Bagnell, and N. Pollard, "Closed-loop servoing using real-time markerless arm tracking," in *Proc. IEEE Int. Conf. on Robotics and Automation*, May 2013.

[9] M. Krainin, P. Henry, X. Ren, and D. Fox, "Manipulator and object tracking for in-hand 3d object modeling," *Int. J. Robotics Research*, vol. 30, no. 11, pp. 1311–1327, 2011.

[10] J. Schulman, A. Lee, J. Ho, and P. Abbeel, "Tracking deformable objects with point clouds," in *Proc. IEEE Int. Conf. on Robotics and Automation*, 2013.

[11] S. Haidacher and G. Hirzinger, "Estimating finger contact location and object pose from contact measurements in 3d grasping," in *Proc. IEEE Int. Conf. on Robotics and Automation*, vol. 2, Sept 2003, pp. 1805–1810.

[12] M. Koval, M. Dogar, N. Pollard, and S. Srinivasa, "Pose estimation for contact manipulation with manifold particle filters," in *Proc. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, Nov 2013, pp. 4541–4548.

[13] L. Zhang and J. Trinkle, "The application of particle filtering to grasping acquisition with visual occlusion and tactile sensing," in *Proc. IEEE Int. Conf. on Robotics and Automation*, May 2012, pp. 3805–3812.

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

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

[16] T. Hulin, K. Hertkorn, P. Kremer, S. Schätzle, J. Artigas, M. Sagardia, F. Zacharias, and C. Preusche, "The DLR bimanual haptic device with optimized workspace," in *Proc. IEEE Int. Conf. on Robotics and Automation*, 2011, pp. 3441–3442.

[17] C. Borst, C. Ott, T. Wimböck, B. Brunner, F. Zacharias, B. Bauml, U. Hillenbrand, S. Haddadin, A. Albu-Schäffer, and G. Hirzinger, "A humanoid upper body system for two-handed manipulation," in *Proc. IEEE Int. Conf. on Robotics and Automation*, 2007, pp. 2766–2767.

[18] H. Liu, K. Wu, P. Meusel, N. Seitz, G. Hirzinger, M. Jin, Y. Liu, S. Fan, T. Lan, and Z. Chen, "Multisensory five-finger dexterous hand: The DLR/HIT hand II," in *Proc. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, 2008, pp. 3692–3697.

[19] R. Newcombe, S. Izadi, O. Hilliges, D. Molyneaux, D. Kim, A. Davison, P. Kohli, J. Shotton, S. Hodges, and A. Fitzgibbon, "Kinectfusion: Real-time dense surface mapping and tracking," in *IEEE ISMAR*. IEEE, October 2011.