Learning Driven Coarse-to-Fine Articulated Robot Tracking

Citation for published version:
Rauch, C, Ivan, V, Hospedales, T, Shotton, J & Fallon, M 2019, Learning Driven Coarse-to-Fine Articulated Robot Tracking. in 2019 IEEE International Conference on Robotics and Automation (ICRA). Institute of Electrical and Electronics Engineers (IEEE), pp. 6604-6610, 2019 IEEE International Conference on Robotics and Automation (ICRA), Montreal, Canada, 20/05/19. https://doi.org/10.1109/ICRA.2019.8794359

Digital Object Identifier (DOI):
10.1109/ICRA.2019.8794359

Link:
Link to publication record in Edinburgh Research Explorer

Document Version:
Peer reviewed version

Published In:
2019 IEEE International Conference on Robotics and Automation (ICRA)

Learning Driven Coarse-to-Fine Articulated Robot Tracking

Christian Rauch[1], Vladimir Ivan[1], Timothy Hospedales[1], Jamie Shotton[2], Maurice Fallon[3]

***Abstract*— In this work we present an articulated tracking approach for robotic manipulators, which relies only on visual cues from colour and depth images to estimate the robot's state when interacting with or being occluded by its environment. We hypothesise that articulated model fitting approaches can only achieve accurate tracking if subpixel-level accurate correspondences between observed and estimated state can be established. Previous work in this area has exclusively relied on either discriminative depth information or colour edge correspondences as tracking objective and required initialisation from joint encoders. In this paper we propose a coarse-to-fine articulated state estimator, which relies only on visual cues from colour edges and learned depth keypoints, and which is initialised from a robot state distribution predicted from a depth image. We evaluate our approach on four RGB-D sequences showing a KUKA LWR arm with a Schunk SDH2 hand interacting with its environment and demonstrate that this combined keypoint and edge tracking objective can estimate the palm position with an average error of 2.5cm without using any joint encoder sensing.**

I. Introduction

Traditional robot manipulation requires a precisely modelled articulated robot arm with accurate position and torque sensing to execute trajectories with high precision. This approach has been most successful in industrial automotive manufacturing but typically does not use any exteroception. In this work we focus on visually-driven manipulation where the scene is understood through visual object detection and fitting and the articulated robot arm is tracked visually. Many compliant robot arms suffer from structure bending and are not millimetre-precise while in some industrial scenarios manipulators are entirely devoid of sensing such as in nuclear decommissioning [1]. In these scenarios the vision-only manipulator tracking would be useful.

We explore model-based articulated arm tracking based entirely on RGB-D cameras passively detecting the arm. The goal is to determine the configuration of the robot arm model which best matches the observed state. One particular challenge is that a variety of different joint configurations can lead to visually similar observations. We are motivated by the work of [2], [3], [4] (in the field of human body tracking) to develop model fitting approaches which leverage learned discriminative information to fit kinematically plausible states to the observed data.

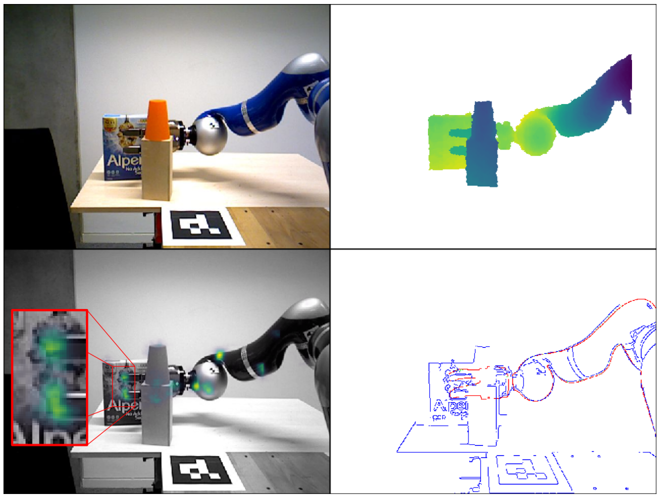

Fig. 1. Grasping in a cluttered environment. **Top left**: colour image, **top right**: colourised depth image of the manipulator and objects in the scene with the background removed. The combined tracking of keypoints (yellow/green, **bottom left**) and edges (blue, **bottom right**) enables precise tracking (red) during a grasping task even when parts of the manipulator are occluded. While keypoints provide sparse but stable visual cues for the fingers, edges provide pixel-accurate estimation of the upper arm. No joint encoder sensing was used here.

Approaches such as [5] and [6] use 2D keypoint estimation to predict the 2D pixel location of joints but do not leverage the kinematic and visual information provided by the manufacturer's 3D model of the robot. This kinematic information is only learned indirectly from a large training set and therefore needs to be explicitly enforced using an additional kinematic solver. Although keypoints can provide reliable constraints for the kinematic solver, they are only sparsely distributed. We therefore propose to use additional dense edge correspondences as a second tracking objective. These two objectives are visualised in Figure 1 for a cluttered environment. While edges provide densely distributed pixel-accurate correspondences, they are impaired by textured objects and are therefore unreliable as a sole tracking objective, in which case keypoints provide more stable cues.

Local optimisation algorithms leverage gradient information which, for kinematic models, can be easily obtained by differentiating the forward kinematics of the articulated model. However, initialising such a local optimisation solely from visual observations is challenging due to the visual ambiguity of shape symmetric robot links and the large range of possible joint motions. We therefore consider many possible candidates for initialising the optimisation. These

Christian Rauch is supported by Microsoft Research through its PhD Scholarship Programme. Maurice Fallon is supported by a Royal Society University Research Fellowship. This research is supported by the Engineering and Physical Sciences Research Council (EPSRC) through the ORCA Robotics Hub (EP/R026173/1).

[1] Institute of Perception, Action and Behaviour, School of Informatics, University of Edinburgh, UK Christian.Rauch@ed.ac.uk

[2] Microsoft, Cambridge, UK

[3] Oxford Robotics Institute, Department of Engineering Science, University of Oxford, UK mfallon@ox.ac.uk

candidates are drawn from a coarse robot state distribution that is predicted from a single depth image. Sampling from this distribution allows us to consider many candidates and select that one that provides the best visual cues.

In summary, we contribute:

1) a tracker initialisation strategy using a coarse joint position distribution predicted from a depth image
2) a combined tracking objective that uses stable and pixel-accurate cues from colour and depth images in a single unified framework

The combination of these stages makes our proposed tracking approach independent from joint encoder sensing and consecutively refines the state from the initially sampled configuration via keypoint tracking until the basin of convergence for pixel-accurate edge correspondences is reached.

We show that, while keypoints already provide a good performance for tracking a manipulator, the additional integration of edges can reduce the end-effector tracking error to 2.5cm for grasping scenarios.

II. Related Work

A large corpus of work [7], [8], [9], [1] has investigated visual tracking for robotic manipulation in recent years. Visual tracking approaches ought to be able to mitigate effects such as linkage elasticity or joint encoder inaccuracies and ought to enable a more precise manipulation accuracy and a more holistic representation of the manipulation scene, including the manipulandum and obstacles in the scene.

Joint distribution prediction: Inspired by previous work on predicting the state of an articulated model from images in [2], [10], we propose to predict a distribution over the articulated state space of a robot manipulator. Similar to [2], we represent this distribution as discretised bins. Instead of training discrete state regressors for each of these bins, we propose to directly sample from the distribution that is represented by these bins. Compared to the discrete states provided by the retrieval forest in [10], our proposed sampling approach provides continuous interpolated samples from the state space and hence also includes samples that are not exactly part of the training set.

Visual features: Different sparse and dense visual features have been used in tracking literature to establish correspondences between the observed and estimated state of a 3D model. Early work in this area used dense features like colour image edges [11], [1], [12] and depth images [9]. These correspondences are based on the local appearance of the estimated state and change with each iteration of the optimisation. This results in many local minima which can be mitigated by introducing discriminative information [13].

Sparse keypoint features, learned from data, are commonly used for human pose estimation [5], [6] and used to estimate the skeleton configuration from 2D images. These approaches do not resolve 3D ambiguity, nor do they provide the exact visual representation that is required for robotic grasping tasks. An additional 3D pose estimation stage [14] can regress from these keypoint locations to 3D joint coordinates. As proposed in [15], we resolve the ambiguity when mapping between a 2D keypoint to 3D pose by using a line of sight constraint and the camera intrinsics to constrain the optimisation state space.

Due to their stability, we propose to rely on keypoint tracking as the base objective. After initial optimisation, we then switch to dense but pixel-level accurate edge correspondence for accurate registration. Compared to our previous work [13], which used pixel-wise depth image segmentation as visual cues, the 2D keypoints proposed in this paper can be located behind occlusions (see Figure 1) while texture/colour edges provide sharper contours than the imprecise edges in typical depth images.

Kinematic optimisation: Model fitting approaches, such as [9], [12], [4], rely on accurate models to find the optimal state that is kinematically and visually plausible. While global optimisation methods are less prone to local minima, they are also more difficult to tune and are computationally expensive. Local optimisation approaches on the other hand are well established and make use of gradient information to quickly converge to a minimum. We use the optimisation toolbox EXOTica [16], which provides a modular way to exchange solvers and objectives, to fit our robot model using the keypoint and edge objectives. Inspired by [17], we first optimise the kinematic chain from the base of the robot to the palm or wrist, before optimising the smaller finger links. This makes sure that the optimisation of fingers, which have less visual features and are more likely occluded, is initialised from a reasonable state.

III. Method

A. Overview

To find kinematically plausible robot configurations which match the observed depth image, we propose a coarse-to-fine inverse kinematic optimisation in three stages (Figure 2). First, we sample from a predicted distribution of joint values to propose a set of possible initial configurations (Section III-C). These coarse samples are tested in the second stage to select the sample which minimises the keypoint tracking objective on the first image in a sequence. In the third stage, we minimise the combined keypoint and edge objective (Section III-F) consecutively on a sequence of images (Section III-G). All three stages use visual cues that we extract from the depth image using a multitask convolutional neural network (Section III-B).

B. Multitask Prediction

Predicting the joint position distributions and keyoint heatmaps is done in parallel on a depth image. Since these tasks share depth image features, they are commonly trained in a multitask setup (Figure 3). In our architecture, we use a ResNet-34 [18] to extract 256 feature maps that are used by the task specific branches. As the type of a keypoint relates to the link it belongs to, we train an additional segmentation task to segment the depth image into robot links and background.

The segmentation and keypoint heatmaps provide information in the image space about pixels being occupied by

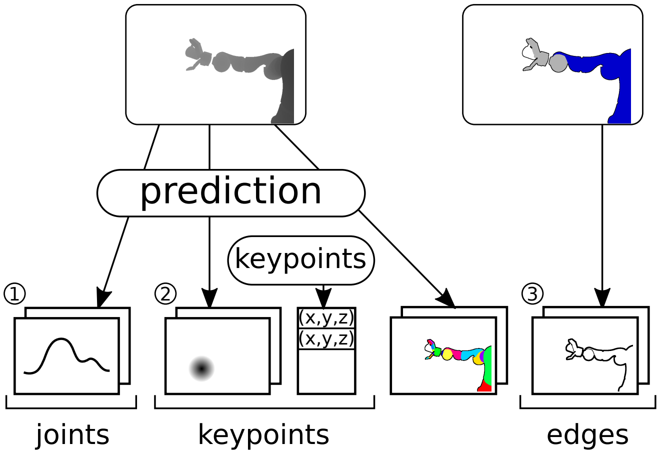

Fig. 2. An overview of the sources of information extracted from an observed RGB-D image pair. From left to right, they provide increasingly detailed visual cues for the tracking system.

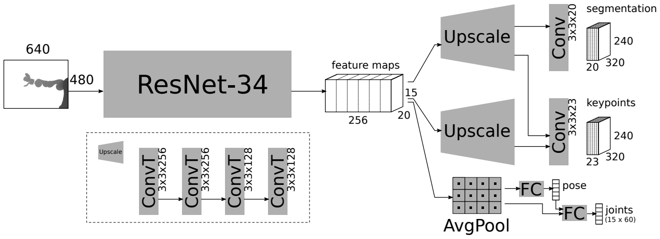

Fig. 3. Our approach uses multitask prediction to obtain segments, keypoints and joint estimates from a single depth image. All three tasks use the same depth features extracted by a ResNet-34 [18].

a link or a 2D keypoint, respectively. The feature maps are therefore individually upscaled by 3×3 transposed convolutions to 128 task specific feature maps of a quarter of the original depth image resolution. For the segmentation this is followed by a regular 2D 3×3 convolution and a softmax layer for providing the probabilities for the $N_L = 18$ robot links, the background and the object (20 classes in total). To reuse information about the location of links for the keypoint localisation, we concatenate the upscaled heatmap features with the segmentation features and apply 3×3 2D convolution with a sigmoid activation.

The 22 3D keypoints are manually placed on the surface of the 18 links. During training, they are transformed via the true state into the camera frame and projected onto the 2D image plane. 2D Gaussians with $\sigma = 3\text{px}$ are centred on each of the 22 keypoint pixel locations to obtain the final heatmaps [5]. An additional background heatmap is created to represent the probability of a pixel not being assigned to any keypoint. During prediction, we can only recover the line of sight from camera origin to the 2D keypoint on the image plane. This ambiguity is resolved during optimisation of point-to-line distances in a subsequent stage.

The third branch provides a joint state distribution to initialise the optimisation, and the 6D robot pose as support. Since a regular regression of the joint state only provides the state vector itself without a confidence measure, we train the network to predict a distribution of joint states. For each of the $N_J = 15$ joint positions, we place a 1D Gaussian with $\sigma = 0.1\text{rad}$ on the true joint position. This Gaussian is then discretised in the value range $[-\pi, \pi)\text{rad}$ into 60 bins which results in a resolution of $6\text{deg} \equiv 0.1\text{rad}$. All discretised joint positions are serialised into a single vector $\mathbb{R}^{900 \times 1}$ (15×60) and then reshaped into a matrix $\mathbb{R}^{15 \times 60}$ containing the score values of discretised joint positions. After prediction we can treat the scores of each joint as a probability distribution function (PDF) from which we can sample joint states. Since we are sampling from a continuous distribution, the resolution of the discretised bins does not directly affect the sampled states, as long as it is small enough to represent multiple modes of the joint positions.

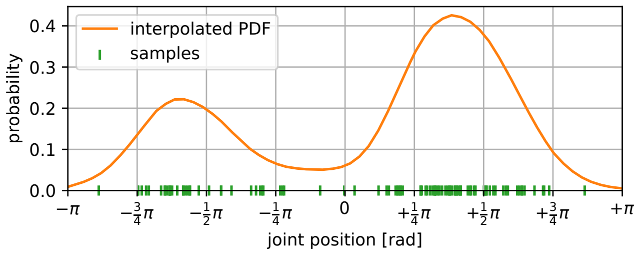

Fig. 4. Example distribution (orange) and samples (green) of a lower arm joint position, predicted and sampled from an image of the occluded bottle sequence (Figure 8). The distribution shows two strong modes at 1.2rad and −1.8rad since the link has a similar visual appearance at half rotations.

The inference for all tasks takes 0.07s per image.

C. Sampling of Initial Configuration

Robotic joints have a large range of possible joint positions and configurations that are far away in joint space can lead to the same global end-effector pose with very similar local appearance. Trying to directly predict the joint position from a single depth image is therefore an ambiguous task. Although we provide a single-mode 1D Gaussian joint position distribution as training target, it is likely that a multimodal distribution is predicted for visually similar appearing configurations of a link (Figure 4).

Since the mean or the strongest mode of such a distribution might not correspond to the observed link state, we propose to sample independently from each joint's distribution to initialise the optimisation several times. To obtain samples from the PDF represented by the score bins, we linearly interpolate the cumulative sum of the bin scores, which provides the cumulative distribution function (CDF), and sample uniformly in $\mathcal{U}(0, 1)$ from the inverse CDF.

D. Training

The segmentation, keypoint localisation and joint distribution prediction is trained on approximately 125000 synthetically rendered depth images in the manner described in our previous work [13]. These synthetic images show the robot at different configurations sampled from a wide range of states where the palm is inside the camera frustum. As proposed in [19], we randomly select one of 30 objects and place the object at a random pose inside the hand to simulate interaction with a manipulandum. We do not discriminate between these 30 objects, but treat them as a single object.

During training we minimise a weighted cross entropy for the segmentation task, the mean absolute error on the keypoint heatmaps and the mean squared error on the discretised joint scores. The cross entropy is weighted by median frequency balancing to increase the classification accuracy for smaller links such as the middle finger and finger tip.

E. Tracking Objective

The observed robot state is provided by the colour and depth images, and the predicted keypoints and joint position distribution. The estimated robot state is initially provided by the sampled configurations and thereafter from the optimisation on consecutive image frames. The visual representation of the estimated state is obtained by rendering the link meshes at their estimated pose.

The objective for the optimisation is to minimise the distance between observed 2D keypoints and edges and their corresponding estimated visual 3D representation. Since the depth of these keypoints and edges cannot be fully determined, e.g. a keypoint might be occluded, the objective is formulated using the line of sight.

The line of sight $\mathbf{l}$ of a 2D keypoint or 2D edge pixel is the ray that passes the camera origin and the 2D point on the image plane. These lines are defined in the camera frame and obtained via back-projection using the camera intrinsics. The start of this ray $\mathbf{l}_s$ can be constrained using corresponding depth information. The previously defined link keypoints and the link meshes are transformed from the link to the camera frame at each optimisation iteration and provide the corresponding estimated 3D visual representation.

Given the 3D line of sight $\mathbf{l}$ and a corresponding 3D point $\mathbf{p}$, the projection of $\mathbf{p}$ on $\mathbf{l}$, $\mathbf{p}_v$

$$\begin{aligned} t &= min\left(0, -\frac{(\mathbf{l}_s - \mathbf{p}) \cdot \mathbf{l}}{||\mathbf{l}||^2}\right) & (1) \\ \mathbf{p}_v &= \mathbf{l}_s + t\mathbf{l} & (2) \end{aligned}$$

is the point on the line closest to $\mathbf{p}$. The vector

$$\mathbf{d}_v = \mathbf{p}_v - \mathbf{p} \tag{3}$$

points from the estimated 3D point to its corresponding observed 3D point in the camera frame. $\mathbf{d}_v$ is the tracking objective for keypoints and edges which is to be minimised. We will denote this objective as Point-to-Line (P2L) task.

1) Keypoints: For each predicted heatmap, we select the pixel with the highest score and its associated depth reading to obtain the line of sight $\mathbf{l}$. The corresponding 3D keypoint is transformed from its local coordinate frame to the camera frame via forward kinematics during the optimisation. The P2L keypoint correspondences are established per observed image frame and stay constant during optimisation.

2) Edge Pixels: Estimated edge pixels are related to their closest observed edge pixel by computing a distance transform on the Canny [20] edges of the observed colour image. The estimated edge pixels and its 3D coordinate are provided by rendering the estimated state in the image frame. We iterate through these edge pixels and assign them to the closest observed edge pixel if the angle between their normals is smaller than 8 degree, i.e. if they point roughly in the same direction. This is similar to the orthogonal line search proposed in [12]. The edge-to-edge association provides multiple P2L tasks per link, which are updated at each iteration by rendering the new estimated state. To make these objectives more robust, we reject keypoints with scores smaller than 0.5 and reject edges pixels with more than 5cm distance.

F. Resolving Robot State Ambiguity

Each P2L task minimises the distance $\mathbf{d}_v$ between the observed line of sight rays in the camera frame and their corresponding 3D points $\mathbf{p}$ in the link frame (after transforming them into the same camera frame using the estimated joint configuration) with respect to the robot state $\mathbf{q}$. The gradient of this task is derived using the chain rule:

$$\frac{\partial \mathbf{d}_v}{\partial \mathbf{q}} = \left(\frac{\partial \mathbf{p}}{\partial \mathbf{q}} \cdot \frac{\mathbf{l}}{||\mathbf{l}||^2}\right) \mathbf{l} - \frac{\partial \mathbf{p}}{\partial \mathbf{q}} \quad . \tag{4}$$

$\frac{\partial \mathbf{p}}{\partial \mathbf{q}}$ is given by the kinematic Jacobian $J_{kin}(q) \in \mathbb{R}^{3\times 6+N_J}$, i.e. by differentiation of forward kinematics, where $J_{i,j} = \frac{\partial p_i}{\partial q_j}$ with $i \in [1, 3]$ (task space ϕ) and $j \in [1, 6 + N_J]$ (robot state space q), and N_J the number of joints.

The keypoint and edge correspondences provide N_T P2L tasks for each of the N_L links. The gradients $\left(\frac{\partial \mathbf{d}_{v,t}}{\partial \mathbf{q}}\right)$ and directions $(\mathbf{d}_{v,t})$ of these tasks $(t \in [1, N_T])$ are averaged per link l and objective type:

$$\begin{aligned} J_{P2L,l} &= \frac{1}{N_{T,l}} \sum_t^{N_{T,l}} \frac{\partial \mathbf{d}_{v,t}}{\partial \mathbf{q}} & (5) \\ \phi_{P2L,l} &= \frac{1}{N_{T,l}} \sum_t^{N_{T,l}} \mathbf{d}_{v,t} \quad . & (6) \end{aligned}$$

Since the tasks are shared among the keypoint (k) and edge (e) objective, their relative contribution is weighted by a link specific weight $\alpha \in \{0, 1\}$ and a global weight $w \in \{0, 1\}$:

$$\begin{aligned} J_{opt,l} &= \alpha_k J_{P2L,l,k} + w_e \alpha_e J_{P2L,l,e} & (7) \\ \phi_l &= \alpha_k \phi_{P2L,l,k} + w_e \alpha_e \phi_{P2L,l,e} \quad . & (8) \end{aligned}$$

The weights α are used to switch the tracking objective between keypoints ($\alpha_k = 1$, $\alpha_e = 0$) and edges ($\alpha_k = 0$, $\alpha_e = 1$) individually per link. The weight w_e is used to globally add or remove the edge tracking objective.

The distances are minimised iteratively with respect to $\mathbf{q}$

$$q_{i+1} = q_i + J_{opt}^{\dagger} \phi \tag{9}$$

using the pseudo-inverse $J_{opt}^{\dagger} \in \mathbb{R}^{6+N_J \times 3N_L}$ of all stacked J_{opt} and the stacked P2L distances $\phi \in \mathbb{R}^{3N_L \times 1}$.

Since the root link of the robot is rotational symmetric and often not observed in the depth image, we use the true 6D camera pose and do not optimise these state variables.

G. Tracking Pipeline

The tracking operates on a continuous sequence of depth and colour images. It is initialised once at the beginning by sampling 50 configurations from the predicted distribution (Section III-C) and selecting the state with the smallest keypoint objective (Section III-E.1), i.e. the forward kinematics state with the smallest average Euclidean distance between the 3D keypoints and their corresponding line of sight. The optimisation is then initialised at each new image pair using the previous solution and iterates for 10 iterations (0.37s).

The tracking objectives are switched at run time for each link individually. A link switches from keypoint to edge tracking ($\alpha_k = 0$, $\alpha_e = 1$), if all of its keypoint distances are closer than 2cm ($\|\mathbf{d}_v\| \leq 0.02$) and vice versa. Finger links always use the keypoint objective.

We initially only track the arm and palm and switch to full tracking when the upper links' keypoint error is smaller than 2cm, and switch back to arm and palm tracking when this error becomes larger than 3cm. This low and high threshold have been chosen to minimise hysteresis.

IV. Evaluation

We evaluate our tracking approach using a Kuka LWR4 7 DOF arm with a Schunk SDH2 7 DOF end-effector, which is observed by a fixed Asus Xtion PRO LIVE RGB-D camera. For further details on the experimental setup we refer to [13]. We evaluate our tracking approach on four sequences that show grasping of different objects and occlusions.

A. Sampling Robot States

First we will qualitatively evaluate the first stage of our proposed pipeline, which proposes initial robot joint configurations using the predicted distributions described in Section III-C. Figure 5 shows snapshots of two tracking sequences, each with three sampled configurations. For these visualisations, we sampled 50 configurations from the predicted distribution and automatically selected the three configurations with the smallest average edge-to-edge distance. These configurations coarsely align with the observed state and demonstrate that the predicted distribution provides reasonable robot states to initialise the local optimisation.

B. Tracking

We apply the proposed tracking approach on the four sequences as described in Section III-G. To evaluate the contribution of edge tracking, we apply tracking once with $w_e = 0$ (keypoint-only objective) and $w_e = 1$ (combined keypoint and edge objective) with the same sampled starting state. Apart from w_e, we use the same configuration for all sequences. Figures 6 to 9 report the position tracking error against forward kinematics for a forearm link and the palm (fifth and ninth link in the kinematic chain), with two snapshots of the sequence overlaid with contours of the state estimated using the combined keypoint and edge objective.

By using edges as an additional objective, the average palm position error in the non-occluded grasping sequences (Figures 6 and 7) has reduced from 3.7cm to 2.7cm and 3.1cm to 2.5cm, respectively. Although the *occluded bottle* sequence (Figure 8) shows an improved tracking performance of the forearm link, this is not propagated to the palm.

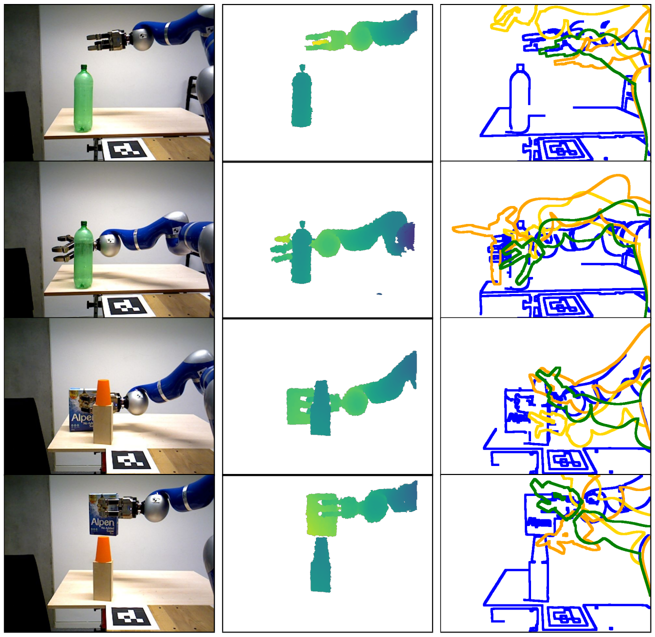

Fig. 5. Sampling from the joint position distribution. **Left**: colour image of the observed scene, **Middle**: depth image from which we predict the joint position distribution, **Right**: observed edges (blue) overlaid with the contours of three sampled configurations (green, orange, yellow).

In the *grasping behind occlusions* sequence (Figure 9), which is the most challenging of our sequences since it contains distractions of both types (manipulandum and occlusion), we are still able to track the palm with an average position error of 4.5cm, which is less than half of the palm length (9.38cm). The keypoint-only baseline performs slightly better in this case.

In summary, our proposed tracking approach is able to reliably track an occluded manipulator in grasping scenes, without making any assumptions on the presence of objects or the availability of joint encoder readings. This solves a common problem of articulated tracking approaches, which often need to be initialised from a known robot state. Our approach is therefore more generally applicable to scenarios where direct access to the robot is not available.

V. Conclusion

We presented a robotic manipulator tracking approach that solely relies upon visual cues to initialise tracking. It consecutively updates the estimated state using a combination of colour edge and depth keypoint correspondences. The proposed deep multitask network learns common depth image features that can efficiently be used in parallel for the coarse initialisation and the keypoint tracking objective. Dense colour image edges are then further used to refine the estimated state. Our approach only requires an accurate kinematic and visual model to generate training data and to provide the estimated visual representation during tracking. No real robot data was required to train the network.

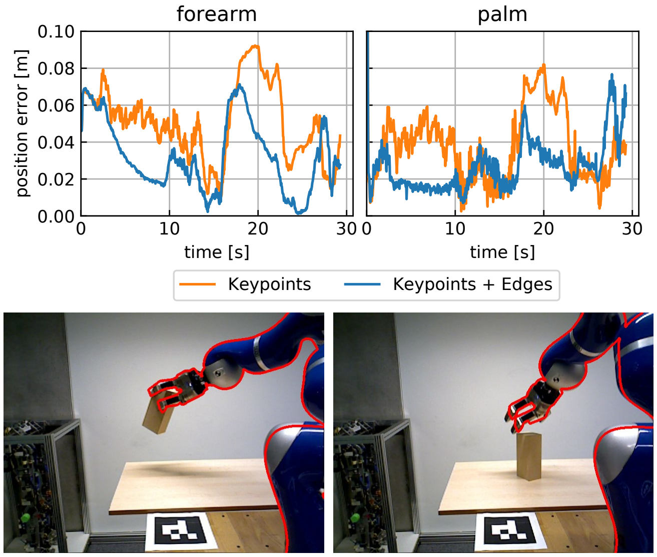

Fig. 6. *Grasping box*. Using the additional edge objective reduces average position error from 5cm to 3.1cm (forearm) and 3.7cm to 2.7cm (palm).

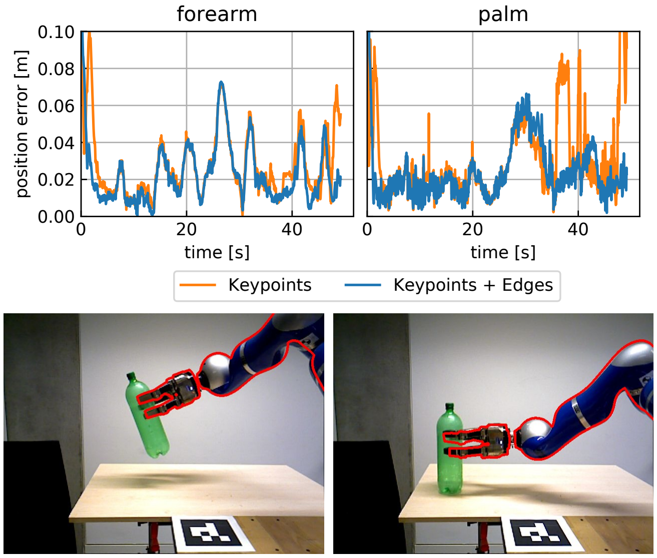

Fig. 7. *Grasping bottle*. Using additional edge objective reduces average position error from 2.7cm to 2.3cm (forearm) and 3.1cm to 2.5cm (palm).

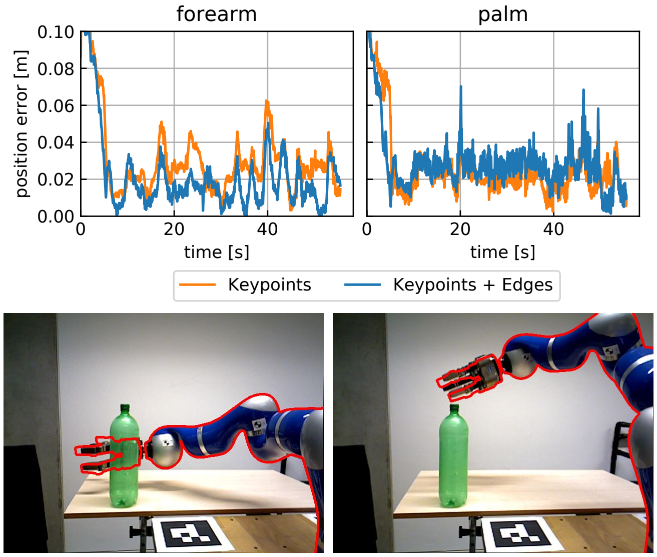

Fig. 8. *Occluded bottle*. Using additional edge objective reduces average position error for the forearm from 3.1cm to 2.1cm but increases the palm position error from 2.6cm to 2.8cm.

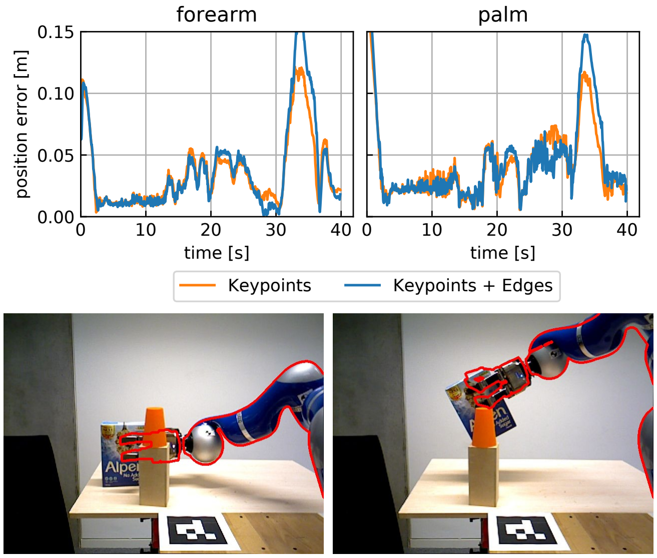

Fig. 9. *Grasping Alpen box behind occlusion*. The additional edge objective impairs tracking (average palm position error increased from 4.3cm to 4.5cm), when strong visual distractions are present.

We evaluated our approach on four sequences showing the grasping of different objects and varying occlusions, and found that the additional edge tracking objective improves tracking of grasping scenes compared to only keypoint tracking. Even in cases with strong distractions from a textured manipulandum and occlusions, which our previous approach cannot handle, we are able to purely visually track the palm with a position error of less than half its size. We note that we used the same tracking parameters for the sequences with and without occlusions and recommend tuning of the combination of the objectives depending on the expected amount of occlusions. In future work, we will investigate alternative dense correspondences like histogram of oriented gradients (HOG) to provide more robust pixel-level correspondences.

The proposed prediction of a joint position distribution provides samples that are sufficient to initialise tracking, but the discrete bin scores and their interpolated PDF are predicted and sampled independently which makes this sampling approach inefficient. The camera pose is currently not part of the predicted and sampled robot state distribution and is therefore assumed given and static. In future work we will investigate methods for sampling a holistic robot state that can additionally be used during tracking to detect tracking loss and reinitialise the tracker.

REFERENCES

[1] V. Ortenzi, N. Marturi, R. Stolkin, J. A. Kuo, and M. Mistry, "Vision-guided state estimation and control of robotic manipulators which lack proprioceptive sensors," in *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Oct 2016, pp. 3567–3574.

[2] T. Sharp, C. Keskin, D. Robertson, J. Taylor, J. Shotton, D. Kim, C. Rhemann, I. Leichter, A. Vinnikov, Y. Wei, D. Freedman, P. Kohli, E. Krupka, A. Fitzgibbon, and S. Izadi, "Accurate, robust, and flexible real-time hand tracking," in *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*, ser. CHI '15. New York, NY, USA: ACM, 2015, pp. 3633–3642. [Online]. Available: http://doi.acm.org/10.1145/2702123.2702179

[3] S. Sridhar, F. Mueller, A. Oulasvirta, and C. Theobalt, "Fast and robust hand tracking using detection-guided optimization," in *Proceedings of Computer Vision and Pattern Recognition (CVPR)*, June 2015.

[4] P. Krejov, A. Gilbert, and R. Bowden, "Guided optimisation through classification and regression for hand pose estimation," *Computer Vision and Image Understanding*, vol. 155, pp. 124 – 138, 2017.

[5] S. E. Wei, V. Ramakrishna, T. Kanade, and Y. Sheikh, "Convolutional pose machines," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016, pp. 4724–4732.

[6] A. Newell, K. Yang, and J. Deng, "Stacked hourglass networks for human pose estimation," in *Computer Vision – ECCV 2016*, B. Leibe, J. Matas, N. Sebe, and M. Welling, Eds. Cham: Springer International Publishing, 2016, pp. 483–499.

[7] K. Pauwels, V. Ivan, E. Ros, and S. Vijayakumar, "Real-time object pose recognition and tracking with an imprecisely calibrated moving RGB-D camera," in *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Sept 2014, pp. 2733–2740.

[8] J. Bohg, J. Romero, A. Herzog, and S. Schaal, "Robot arm pose estimation through pixel-wise part classification," in *2014 IEEE International Conference on Robotics and Automation (ICRA)*, May 2014, pp. 3143–3150.

[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 (ICRA)*, May 2015, pp. 119–126.

[10] J. P. C. Valentin, A. Dai, M. Nießner, P. Kohli, P. H. S. Torr, S. Izadi, and C. Keskin, "Learning to navigate the energy landscape," *CoRR*, vol. abs/1603.05772, 2016. [Online]. Available: http://arxiv.org/abs/1603.05772

[11] I. Oikonomidis, N. Kyriazis, and A. A. Argyros, "Full DOF tracking of a hand interacting with an object by modeling occlusions and physical constraints," in *2011 International Conference on Computer Vision*, Nov 2011, pp. 2088–2095.

[12] E. Muoz, Y. Konishi, V. Murino, and A. D. Bue, "Fast 6D pose estimation for texture-less objects from a single RGB image," in *2016 IEEE International Conference on Robotics and Automation (ICRA)*, May 2016, pp. 5623–5630.

[13] C. Rauch, T. Hospedales, J. Shotton, and M. Fallon, "Visual articulated tracking in the presence of occlusions," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*, May 2018, pp. 643–650.

[14] C. Zimmermann and T. Brox, "Learning to estimate 3D hand pose from single RGB images," in *IEEE International Conference on Computer Vision (ICCV)*, 2017. [Online]. Available: http://lmb.informatik.uni-freiburg.de/Publications/2017/ZB17a

[15] J. Brookshire and S. Teller, "Articulated pose estimation via over-parametrization and noise projection," in *Robotics: Science and Systems*, 2014.

[16] V. Ivan, Y. Yang, W. Merkt, M. P. Camilleri, and S. Vijayakumar, *EXOTica: An Extensible Optimization Toolset for Prototyping and Benchmarking Motion Planning and Control*. Cham: Springer International Publishing, 2019, pp. 211–240. [Online]. Available: https://doi.org/10.1007/978-3-319-91590-6_7

[17] D. Tang, J. Taylor, P. Kohli, C. Keskin, T. Kim, and J. Shotton, "Opening the black box: Hierarchical sampling optimization for estimating human hand pose," in *2015 IEEE International Conference on Computer Vision (ICCV)*, Dec 2015, pp. 3325–3333.

[18] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016, pp. 770–778.

[19] F. Mueller, D. Mehta, O. Sotnychenko, S. Sridhar, D. Casas, and C. Theobalt, "Real-time hand tracking under occlusion from an egocentric RGB-D sensor," in *Proceedings of International Conference on Computer Vision (ICCV)*, October 2017. [Online]. Available: https://handtracker.mpi-inf.mpg.de/projects/OccludedHands/

[20] J. Canny, "A computational approach to edge detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PAMI-8, no. 6, pp. 679–698, Nov 1986.