4-DoF Tracking for Robot Fine Manipulation Tasks

Mennatullah Siam*, Abhineet Singh*, Camilo Perez and Martin Jagersand
Faculty of Computing Science
University of Alberta, Canada
mennatul,asingh1,caperez,jag@ualberta.ca

***Abstract*—This paper presents two visual trackers from the different paradigms of learning and registration based tracking and evaluates their application in image based visual servoing. They can track object motion with four degrees of freedom (DoF) which, as we will show here, is sufficient for many fine manipulation tasks. One of these trackers is a newly developed learning based tracker that relies on learning discriminative correlation filters while the other is a refinement of a recent 8 DoF RANSAC based tracker adapted with a new appearance model for tracking 4 DoF motion.**

Both trackers are shown to provide superior performance to several state of the art trackers on an existing dataset for manipulation tasks. Further, a new dataset with challenging sequences for fine manipulation tasks captured from robot mounted eye-in-hand (EIH) cameras is also presented. These sequences have a variety of challenges encountered during real tasks including jittery camera movement, motion blur, drastic scale changes and partial occlusions. Quantitative and qualitative results on these sequences are used to show that these two trackers are robust to failures while providing high precision that makes them suitable for such fine manipulation tasks.

***Keywords*-visual tracking; visual servoing; robot manipulation;**

I. INTRODUCTION

2D Object tracking is a core component in visual servoing [1] where visual feedback is used to guide the robot to perform certain tasks. One category of these tasks involves manipulation of objects [2] ranging from simple pick and place to more advanced ones. Fine manipulation [3] in particular, where small objects are handled, can be quiet challenging. Though a lot of research has been done using depth cameras like Kinect for manipulation tasks in general, yet these sensors are not suitable for fine manipulation. This is due to limitations in their range, resolution and accuracy. Most current depth cameras have an operation range of 0.8 to 5 m which is not enough for EIH configurations [3]. Some have used a Kinect sensor for initial positioning, followed by manual tele-operation to grasp objects [4]. However, a more versatile solution is to use image based visual servoing (IBVS) by itself for such scenarios.

In this paper, we provide a solution to visual tracking for performing fine manipulation tasks using IBVS with low cost cameras. The presented trackers are shown to work well with EIH configuration within very small ranges and with better accuracy than several state of the art trackers. They provide 4 DoF motion information, sufficient for high precision tasks to work, while being robust to occlusions, illumination changes and motion blur. We have integrated them as part of a tracking library called Modular Tracking Framework (MTF) [5] along with all trackers tested here so the results are easily reproducible. MTF also provides a ROS interface for effortless integration in robotics applications.

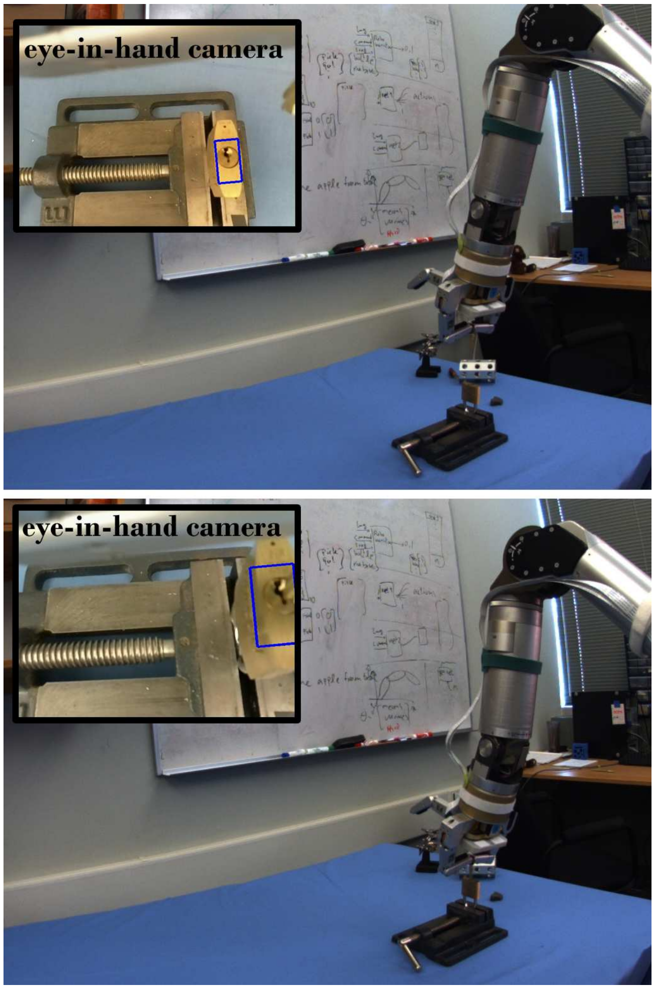


Figure 1: Fine manipulation task of inserting key into lock. Images from eye-to-hand and eye-in-hand configuration

To summarize, following are the main contributions of this work:

- A new 4 DoF correlation based tracker Rotation and Scale Space Tracker (RSST) is introduced.
- A state of the art RANSAC based tracker (RKLT) [7] is adapted to be more robust to appearance changes by incorporating an illumination invariant similarity metric and limiting it to 4 DoF motion estimation.
- A new benchmark [6] is made publicly available with 24 video sequences captured from both a fixed camera in eye-to-hand (ETH) configuration and a robot end effector mounted EIH stereo camera that are annotated (Fig. 1). We call it Tracking for Fine Manipulation Tasks (**TFMT**) dataset.
- Detailed experimental analysis for these trackers is presented in the context of fine manipulation tasks. Several new insights are obtained and discussed regarding the differences between tracking for manipulation tasks and general 2D object tracking. Experimental results that motivate our choice of 4 DoF are provided showing its advantages over both lower and higher DoF tracking.

Rest of this paper is organized as follows: section II provides an overview of the related work in 2D visual tracking followed by details about the two trackers presented here in sections III and IV. Section V presents the experimental analysis, first on general manipulation tasks, then on specific fine manipulation tasks. This is followed by tests on general object tracking scenarios. Finally section VI presents the conclusions.

II. Related Work

The general 2D object tracking problem has been extensively researched in past decades. Trackers can be categorized as discriminative trackers and generative. In the former category [8][9][10], the tracking task is posed as a binary classification problem. A discriminative classifier is then learned online from patches containing the object and the background. These learning based approaches are able to cope to some extent with illumination variations, partial occlusions, and viewpoint changes.

Generative trackers, on the other hand, learn a model to represent the object and then use it to search the current frame for the object. They can learn the model online [11] or can have a static model as in most registration based trackers [12][13]. These latter, in a sense, represent a different paradigm of object tracking where precise pose of the object is needed as opposed to a rough bounding box. Their goal is to estimate the optimal warp parameters between the current patch and the reference one. Since several of these use gradient based methods for computing the warp parameters, they are also computationally efficient. However, they often tend to fail under occlusions or other appearance changes, working, as they do, under the assumption that changes in the appearance are solely due to the warping. A relatively recent tracker in [7] combined a set of simple 2 DoF Lucas Kanade (LK) feature trackers with RANSAC to estimate 8 DoF motion. RANSAC rejected lost trackers as outliers thus increasing its robustness. It, however, used the sum of squared differences (SSD) of image intensities as the similarity measure which made it vulnerable to failure in the presence of illumination changes and partial occlusions. Also, its use of 8 DoF motion model made it more prone to getting stuck in local optima.

In recent research, trackers using discriminative correlation based filters [14][15][16] have shown great success. However, these trackers tend to provide DoF motion information where only translation is computed though some also estimate isotropic scaling [9][15]. However, for manipulation tasks, knowledge of the *orientation* of the object is necessary to be able to guide the robot motion precisely. The CMT tracker [17] incorporates rotation but it relies heavily on the detected key points and their descriptors and thus faces difficulties with less textured objects that are common in industrial scenarios. Recently, a deep regression network [18] was used to track by matching the query template within a candidate region. Although it was shown to run at 100 frames per second with offline training on a large dataset, it does not generalize well enough to variations in the sequences as will be demonstrated for manipulation scenarios.

In terms of tracking evaluation, the general object tracking category has two recent benchmarks - VOT [19] and OTB [20] - that overlap in some of the sequences. However, these benchmarks are not suitable for robotics applications as they predominantly feature surveillance type videos and their ground truth is also not very precise. A recent tracking benchmark for manipulation tasks [6] provided a public dataset for robotics scenarios. It had several challenges including partial occlusions, out-of-plane rotation and illumination changes. Nonetheless, it lacked sequences that encompass complete tasks and also did not have any that were captured from EIH configuration that can cause great variability in scale. Finally, the motion - both by human and robotic arm - in all of its sequences was executed too smoothly to accurately represent realistic tasks where the motion is often jerky.

III. Rotation and Scale Space Tracker (RSST)

The proposed approach is closely related to [14][15][16]. These trackers are based on learning a discriminative correlation filter to localize the object of interest. Some of the above mentioned trackers support only 2D translation [14][16] while others were extended to include isotropic scaling [15]. In this section, these correlation based trackers are extended further to include rotation. Similar to [15], HOG features of the search region patch are used and denoted as x. Assuming that this feature map is of dimension d, feature map $l \in 1, ..., d$ is denoted as x^l. A correlation filter h^l is then learned for each feature dimension by optimizing the following objective function:

$$C = \|\sum_{l=1}^{d} h^l * x^l - f\|^2 + \lambda \sum_{l=1}^{d} \|h^l\|^2 \tag{1}$$

where f is the desired correlation output and λ is a factor to control the regularization term. The desired correlation output is a Gaussian centered at the optimum translation, scale or rotation. Solving the above equation in the frequency domain yields:

$$H^l = \frac{F\bar{X}^l}{\sum_{k=1}^{d} X^k \bar{X}^k + \lambda} \tag{2}$$

where H, F, X denote the discrete Fourier transforms of their corresponding signals h, f ,x, and $\bar{X}$ is the complex conjugate. The power of correlation based tracking is its usage of the simple convolution theorem to formulate the problem in the Fourier domain. This approach makes it possible to learn a linear classifier for different shifts of the original patch without rigorously going through all of them as in other tracking by detection approaches.

In order to adapt to appearance changes of the object, the correlation filter is updated according to the following equations:

$$N_t^l = (1-\eta)N_{t-1}^l + \eta F_t \bar{X}_t^l \tag{3}$$

$$D_t = (1-\eta)D_{t-1} + \eta \sum_{k=1}^{d} X_t^k \bar{X}_t^k \tag{4}$$

where N_t^l and D_t respectively denote the numerator and denominator of the correlation filter for feature dimension l at time instant t while η is the learning rate. This mechanism ensures that the tracker does not drift with each update as it relies on previous history as well. Finally, the optimum parameter - whether it is for translation, scale or rotation - is computed from the peak response of the correlation between new feature maps and the correlation filter.

$$y = \mathscr{F}^{-1}\{\frac{\sum_{l=1}^{d} N^l Z^l}{D+\lambda}\} \tag{5}$$

where y is the new parameter and Z denotes the discrete Fourier transform of the feature maps of the new candidate region.

An ideal rotation and scale space tracker will search through the joint space of translation, rotation and scale. However, for the sake of computational efficiency, separate correlation filters for translation, scale and rotation are learned instead. This choice is based on the experiments in [15] that compared the joint computation of translation and scale against separate ones and showed that the latter provided a significant advantage in computational efficiency without degradation in accuracy. For computing the orientation, rotation samples are extracted within a range of potential rotations $[-20, 20]$ with 2 degrees increment. A patch is extracted for each candidate rotation and HOG features are used to encode it. This constructs the rotation feature matrix that is used within the correlation equations. This is used afterwards to update the components of a 1D correlation filter. The steps used for updating the tracker at each time step t are shown in Algorithm 1.

Algorithm 1 Rotation and Scale Space Tracker

```
Input:
  Image $I_t$
  Target position $p_{t-1}$, scale $s_{t-1}$ and rotation $r_{t-1}$.
  Translation model $N_{t-1}^{trans}$, $D_{t-1}^{trans}$
  Scale model $N_{t-1}^{scale}$, $D_{t-1}^{scale}$
  Rotation model $N_{t-1}^{rot}$, $D_{t-1}^{rot}$

Output:
  New target position $p_t$, scale $s_t$ and rotation $r_t$.
  Models $N_t^{trans}$, $D_t^{trans}$,$N_t^{scale}$, $D_t^{scale}$,$N_t^{rot}$, $D_t^{rot}$

Translation
Extract features for translation sample $z_{trans}$ at $p_{t-1}$, $s_{t-1}$, $r_{t-1}$.
Compute translation response $y_{trans}$ using $z_{trans}$, $N_{t-1}^{trans}$, $D_{t-1}^{trans}$ as in equation 5.
Set $p_t$ to maximum location of 2D response $y_{trans}$.

Scale
Extract features for scale sample $z_{scale}$ at $p_t$, $s_{t-1}$, $r_{t-1}$.
Compute scale response $y_{scale}$ using $z_{scale}$, $N_{t-1}^{scale}$, $D_{t-1}^{scale}$ similar to translation.
Set $s_t$ to maximum location of 1D response $y_{scale}$.

Rotation
Extract multiple patches at different sampled rotations in the range [-20,20] around the previous accumulated rotation.
Extract HOG features for rotation sample $z_{rot}$ at $p_t$, $s_t$, $r_{t-1}$.
Compute rotation response $y_{rot}$ using $z_{rot}$, $N_{t-1}^{rot}$, $D_{t-1}^{rot}$ similar to translation .
Set $r_t$ to maximum location of 1D response $y_{rot}$.

Update
Extract samples $x_{trans}$, $x_{scale}$, $x_{rot}$ at the new parameters.
Compute $N_t^{trans}$, $D_t^{trans}$,$N_t^{scale}$, $D_t^{scale}$,$N_t^{rot}$, $D_t^{rot}$ with equations 3 and 4.
```

IV. RANSAC BASED TRACKER (RKLT)

This is a state of the art registration based tracker [7] that is also used in the experiments to demonstrate the benefits of

4 DoF tracking and contrast the two trackers from different paradigms that provide solutions to the same problems in the fine manipulation context. RKLT is a two layer tracker. In the first layer, evenly sampled points are tracked between consecutive images using the pyramidal KLT tracker[21] and the corresponding point pairs are used as input to a RANSAC based method that estimates that similarity transform that can best explain the warping between them. The points that could not be tracked by the KLT tracker, due to partial occlusions or other appearance changes, are rejected as outliers and not used for the RANSAC estimation.

The output of the first layer is used as input to an inverse compositional (IC) Lucas Kanade tracker [22] that refines it further by aligning it with the original template. Only inliers are considered in the IC algorithm to achieve better convergence. Finally, in order to provide robustness to illumination changes, normalized cross correlation (NCC) [23] is used as the similarity metric in the second layer rather than the conventional SSD [22].

V. Experimental Analysis

This section details the experimental setup and data collection procedure. It also presents quantitative analysis of the two trackers compared again eight state of the art trackers from literature. The results are first presented for general manipulation tasks using the recent tracking benchmark for manipulation tasks (TMT) [24]. This is followed by evaluation on the specific fine manipulation aspect on the TFMT dataset this is presented in this work. Finally an evaluation for the general 2D object tracking using VOT[19] benchmark is also provided to highlight the differences in the two domains of tracking.

A. Experimental Setup

TFMT includes sequences with three fine manipulation tasks performed through tele-manipulation: 1) Opening a lock with a key (denoted as Key Task in our experiments). 2) Inserting a thread through a fishing lure (denoted as Fish Lure). 3) Inserting a rivet in an industrial part (denoted as Hexagon Task). Our data collection was performed through tele-manipulation using a 4-DoF arm with a hand gimbal as master and a 7-DoF WAM arm with a barrett hand as slave. This gives a different type of motion compared to the ones in TMT sequences that were executed by smooth human and robot motions.

The experimental setup is shown in Figure 2. We used two raspberry pi cameras located in the barrett hand for the eye-in-hand configuration and two point gray grass hopper cameras with 3mm lenses for the eye to hand configuration. The resolution of the images recorded from eye-in-hand configuration is 640x480. Since two cameras were used, TFMT has 12 sequences in all with a total of 1841 frames. In order to better utilize the frames that follow a tracker's first failure in any sequence, subsequences were used during evaluation where the tracker was initialized at 10 different frames. This increases the effective number of frames to 10,360. The sequences are made publicly available [6].

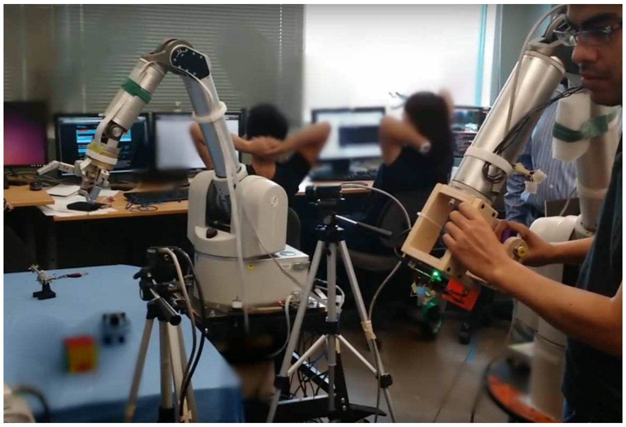


Figure 2: Tele-manipulation setup used for experimental data collection. Left: 7-DoF WAM arm with barrett hand and two eye-inhand-cameras. Right: 4-DoF WAM arm with hand gimbal.

The challenges posed in these sequences include partial occlusions, and objects partially going out of the field of view. Motion blur was a significant challenge as well, especially in the fast version of these sequences. Some of the objects were texture less like the industrial aluminum part. It is worth noting that one of the main differences to the TMT dataset [24] is that these sequences are captured with eye-in-hand configuration. The images in TMT had to be resized to half the size for RSST in order to obtain real-time performance. However, in fine manipulation tasks, since the eye-in-hand cameras have low resolutions, the original size is maintained.

We compare against 8 state of the art trackers: TLD [9], DSST [15], CMT [17], KCF [14], Struck [8], Fragtrack [25] and Goturn [18] and RCT [10]. The error metric used for evaluating the tracking performance in the manipulation tasks experiments is the alignment error E_{al}:

$$E_{al} = \sqrt{\frac{\sum_{i=1}^{4}(X_{T_i} - X_{GT_i})^2}{4}} \tag{6}$$

where X_T denotes the tracking output corners, and X_{GT} is the corresponding ground truth. The reason for using this metric is that manipulation tasks rely heavily on the accuracy of the measurements. For VOT evaluation, however, Jaccard error E_{jac} (eq 7) is used following the exact procedure in [19] since it is sufficient for the scenarios in that dataset:

$$E_{jac} = 1 - \frac{A_T \cap A_{GT}}{A_T \cup A_{GT}} \tag{7}$$

where A_T is the tracking output bounding box, and A_{GT} is the ground truth.

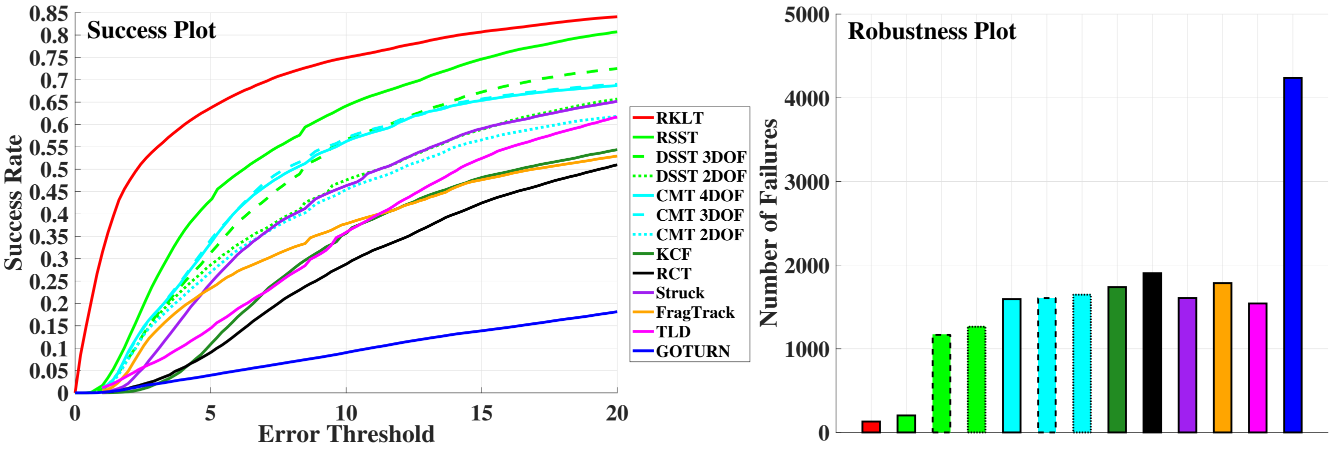
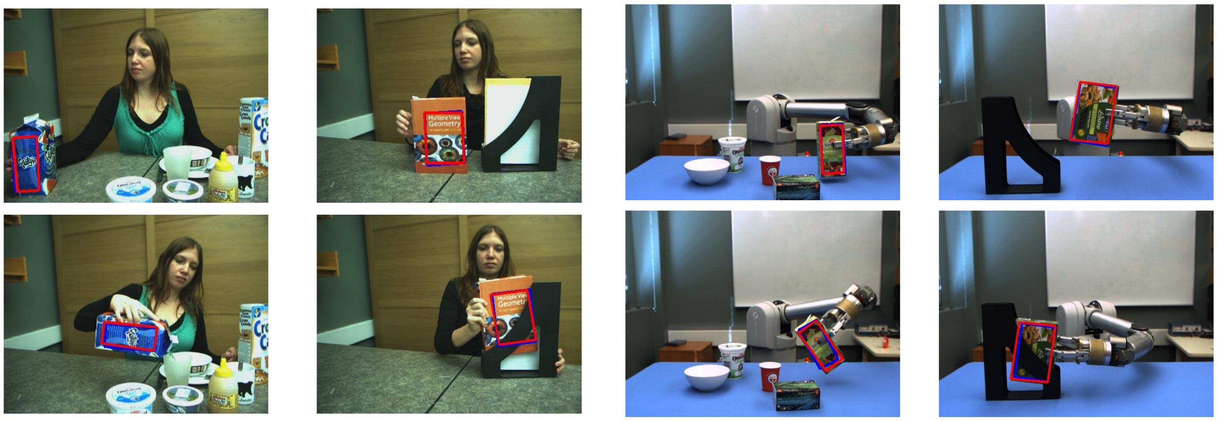


Figure 3: Comparing different trackers on TMT using alignment error(MCD error), where RKLT and RSST outperform the state of the art.



(a) (b) (c) (d)

Figure 4: Tracking Results for RSST in blue and RKLT in red on TMT. a: Juice Sequence. b: Book III Sequence. c: Robot Juice Sequence. d: Robot BookIII Sequence

B. Tracking for Manipulation Tasks

This section presents the quantitative and qualitative evaluation on the manipulation tasks benchmark in [24]. Success and robustness plots are shown in Figure 3. It can be seen that RKLT and RSST both significantly outperform all other trackers with respect to both robustness and accuracy. It can be seen too that RKLT which is a registration based tracker outperforms RSST in terms of accuracy since gradient based methods that are used in registration based trackers tend to provide higher precision.

Qualitative results for these two trackers in the general manipulation tasks are shown in Figure 4. The first and third columns show scenarios where estimating rotation of the object tracked is necessary to track it accurately. RSST and RKLZT are both capable of tracking the rotated object with RKLT providing more precise results. The second and last columns show scenarios of partial occlusions. Again RSST and RKLT are both able to track the occluded object robustly.

C. Tracking for Fine Manipulation (TFMT)

Figure 6 shows success and robustness results on TFMT where RKLT and RSST can again be seen to outperform all other trackers. However, since these are combined plots on all sequences, a more detailed comparison of RKLT and RSST is also performed on individual tasks. Figure 7 shows the average misalignment error on the four corners of the bounding box in pixels. The figure shows both left and right sequence alignment error for two speeds of performing the tasks. Images of the tracking results of both are shown in Figure 8.

It is interesting to note that both trackers have an advantage in certain aspects. The RKLT tracker performs very

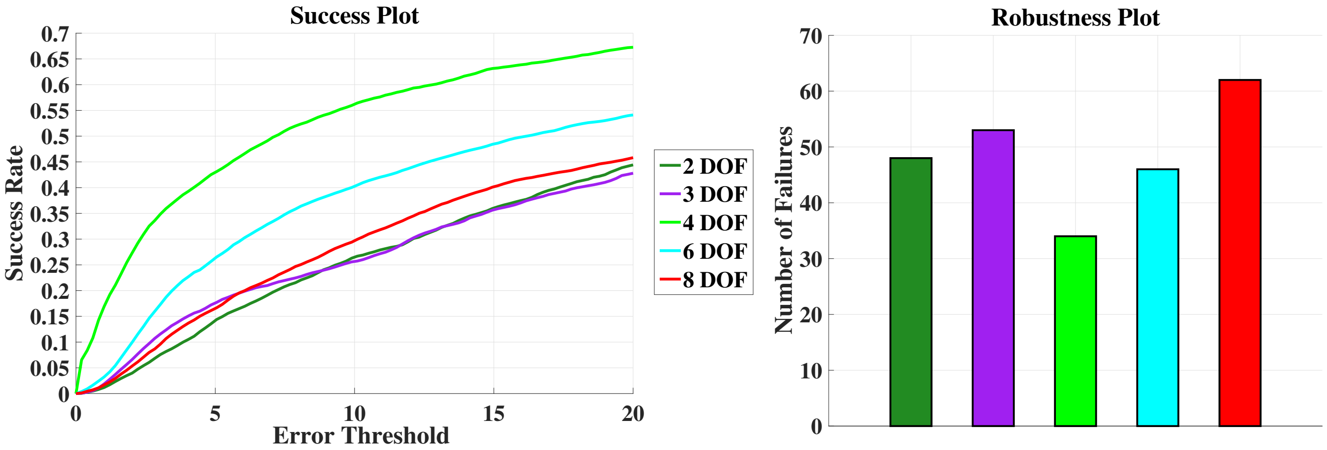


Figure 5: Comparing different DoFs with RKLT on TFMT using alignment error, where 4 DoF seems to be the best compromise

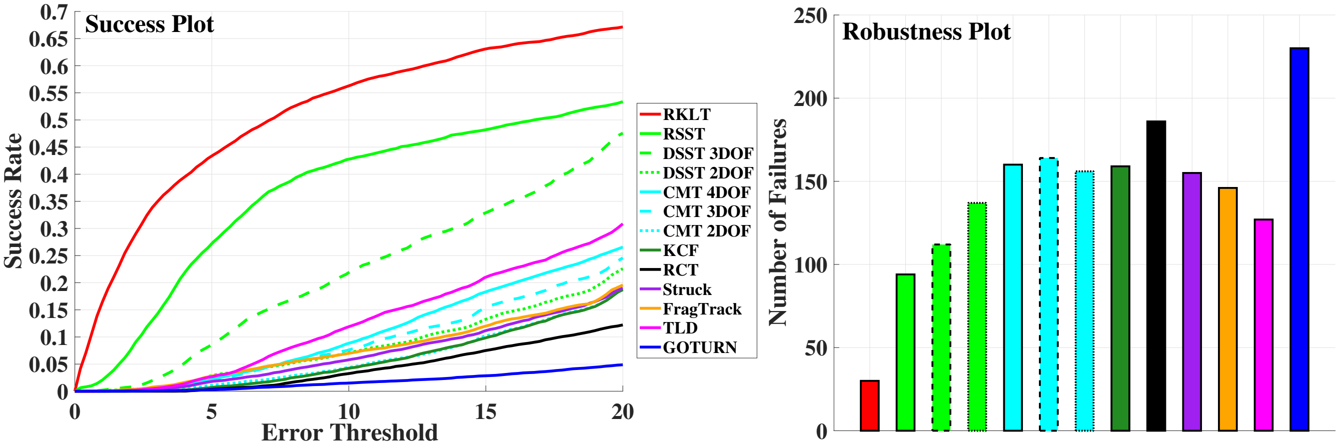


Figure 6: Comparing different trackers on TFMT using alignment error

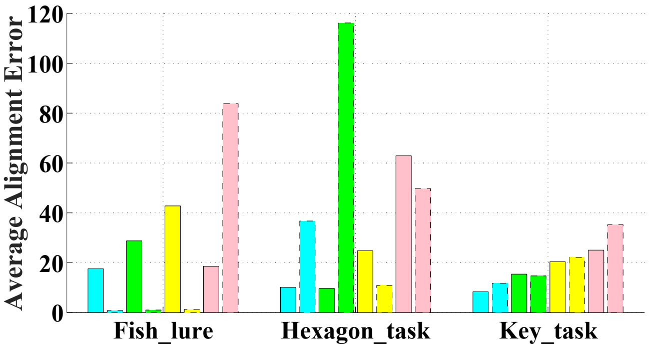
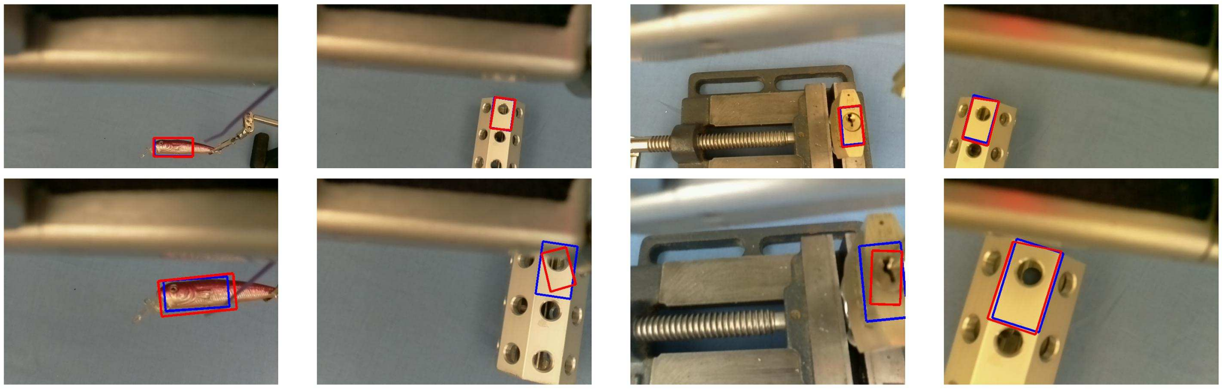


Figure 7: Average Alignment Errors for RSST (solid edge) and RKLT (dotted edge) for different sequences in TFMT. Slow sequences from left and right camera are in cyan and green respectively while the corresponding fast sequences are in yellow and pink.

well with high precision on fish lure which does not suffer as much from partial occlusions as the rest of the sequences. This is expected since one of the strengths of registration based trackers is their accuracy. On the other hand, hexagon task with normal speed and key task suffer a lot from partial occlusions and also have an object that is almost texture less. In this case, RSST is generally more robust than RKLT as shown by both the alignment error and the second column of the qualitative results. That shows these two trackers complement each other and can be used for validating one another.

Another finding that led us to use four DoF trackers is that low DoF trackers generally tend to be more stable as their search space is limited. However, their overall tracking performance is only better when the actual object motion to be tracked does not significant exceed their capabilities. This is shown in Figure 5, where RKLT outperforms the 2, 3, 6 and 8 DoF versions where two DoF is simple 2D



(a) (b) (c) (d)

Figure 8: Tracking Results for RSST in blue and RKLT in red on TFMT. a: Fish Lure Left, b: Hexagon Task Left, c: Key Task Left, d: Hexagon Task Right.

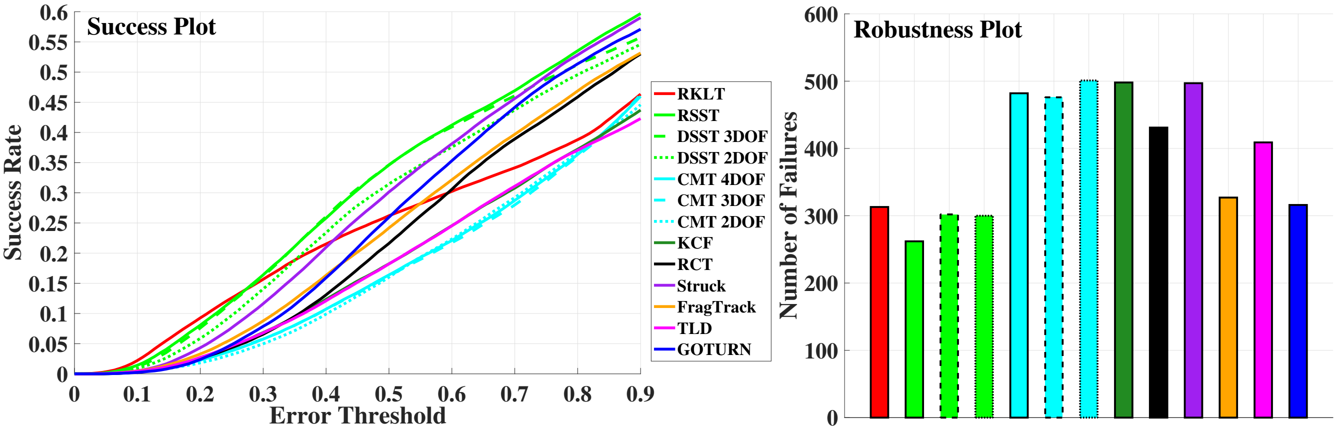


Figure 9: Comparing different trackers on VOT 2016 using Jaccard error

translation, three DoF includes isotropic scaling while four DoF adds on rotation. Six DoF uses affine transformation and eight degrees of freedom stands for using a homography. The superiority of 4 DoF is apparent in both the robustness and success plots. This can be explained because four DoF is the minimum to capture most motions that objects undergo. At the same time it's low enough to not have the gradient based methods get stuck in local minimas.

D. General 2D Object Tracking

Finally, an evaluation of these trackers in comparison to the state of the art on the VOT [19] benchmark is presented in this section. The reason for this is two fold. The first reason is to show that RSST is still able to perform at par with the best tracker for the general 2D object tracking problem. The second and more important reason is to demonstrate the shortcomings of VOT sequences for evaluating general 2D object tracking problem. Figure 9 shows both the success and robustness plot. RSST is the best in terms of robustness while being slightly better than DSST in terms of success rate and outperforming the rest. On the other hand, RKLT does not perform well on this benchmark, except when using small error thresholds.

It is very interesting to see that one of the state of the art trackers, GOTURN [18] that is based on deep regression networks, achieves very good results on VOT. However, it was the worst tracker on TMT. This is due to the fact that the tracker is trained offline with videos that are more similar to VOT sequences. These sequences are significantly different from the manipulation tasks scenarios. It seems fair then to conclude that manipulation tasks and robotic scenarios in general offer different challenges than those present in VOT like benchmarks that are so popular in literature.

VI. Conclusions

This paper introduced a new 4-DoF correlation based tracker (RSST) that can be used in the robotic fine manipulation context. A detailed analysis on manipulation tasks benchmark along with fine manipulation sequences and VOT benchmark was presented. This analysis showed that the strength of RSST lies in its ability to handle partial occlusions and objects going partially out of the field of view while still providing sufficiently precise results. This is the reason that it is the only tracker that is able to perform well on both the general 2D object tracking and manipulation tasks benchmarks. A registration based tracker based on RANSAC and IC for four degrees of freedom was also presented and shown to perform competitively with RSST on the manipulation tasks. It was also shown that four DoF tracking provides a good compromise between accuracy and robustness.

A new fully annotated dataset called tracking for fine manipulation tasks (TFMT) was presented with eye-in-hand camera configuration sequences. This complements the manipulation tasks (TMT) dataset with eye in hand sequences and different motion patterns. Finally, the differences between the general object tracking benchmarks and manipulation tasks benchmark were shown too. This motivates the work for gathering sequences from real robot and human manipulation scenarios for testing such trackers.

References

[1] S. Hutchinson, G. D. Hager, and P. I. Corke, "A tutorial on visual servo control," *IEEE transactions on robotics and automation*, vol. 12, no. 5, pp. 651–670, 1996. 1

[2] M. Gridseth, O. Ramirez, C. P. Quintero, and M. Jagersand, "Vita: Visual task specification interface for manipulation with uncalibrated visual servoing," in *ICRA*. IEEE, 2016, pp. 3434–3440. 1

[3] C. P. Quintero, O. Ramirez, M. Gridseth, and M. Jägersand, "Small object manipulation in 3d perception robotic systems using visual servoing." IROS, 2014. 1

[4] H. Jiang, J. P. Wachs, and B. S. Duerstock, "Integrated vision-based robotic arm interface for operators with upper limb mobility impairments," in *Rehabilitation Robotics (ICORR), 2013 IEEE International Conference on*. IEEE, 2013, pp. 1–6. 1

[5] A. Singh and M. Jagersand, "Modular tracking framework: A unified approach to registration based tracking," *arXiv preprint arXiv:1602.09130*, 2016. 1

[6] "TMT:Tracking Manipulation Tasks," http://webdocs.cs.ualberta.ca/~vis/trackDB/, 2016. 2, 4

[7] X. Zhang, A. Singh, and M. Jagersand, "Rklt: 8 dof real-time robust video tracking combing coarse ransac features and accurate fast template registration," in *CRV*. IEEE, 2015, pp. 70–77. 2, 3

[8] S. Hare, A. Saffari, and P. H. Torr, "Struck: Structured output tracking with kernels," in *ICCV*. IEEE, 2011, pp. 263–270. 2, 4

[9] Z. Kalal, K. Mikolajczyk, and J. Matas, "Tracking-learning-detection," *TPAMI*, vol. 34, no. 7, pp. 1409–1422, 2012. 2, 4

[10] K. Zhang, L. Zhang, and M.-H. Yang, "Real-time compressive tracking," in *ECCV*. Springer, 2012, pp. 864–877. 2, 4

[11] D. A. Ross, J. Lim, R.-S. Lin, and M.-H. Yang, "Incremental learning for robust visual tracking," *International Journal of Computer Vision*, vol. 77, no. 1-3, pp. 125–141, 2008. 2

[12] S. Baker and I. Matthews, "Lucas-kanade 20 years on: A unifying framework," *IJCV*, vol. 56, no. 3, pp. 221–255, 2004. 2

[13] S. Benhimane and E. Malis, "Real-time image-based tracking of planes using efficient second-order minimization," in *IROS*, vol. 1. IEEE, 2004, pp. 943–948. 2

[14] J. F. Henriques, R. Caseiro, P. Martins, and J. Batista, "High-speed tracking with kernelized correlation filters," *TPAMI*, vol. 37, no. 3, pp. 583–596, 2015. 2, 4

[15] M. Danelljan, G. Häger, F. Khan, and M. Felsberg, "Accurate scale estimation for robust visual tracking," in *BMVC*. BMVA Press, 2014. 2, 3, 4

[16] D. S. Bolme, J. R. Beveridge, B. A. Draper, and Y. M. Lui, "Visual object tracking using adaptive correlation filters," in *CVPR*. IEEE, 2010, pp. 2544–2550. 2

[17] G. Nebehay and R. Pflugfelder, "Clustering of static-adaptive correspondences for deformable object tracking," in *CVPR*, 2015, pp. 2784–2791. 2, 4

[18] D. Held, S. Thrun, and S. Savarese, *Learning to Track at 100 FPS with Deep Regression Networks*. Cham: Springer International Publishing, 2016, pp. 749–765. 2, 4, 7

[19] "VOT 2016 Challenge," http://www.votchallenge.net/vot2016/, 2016. 2, 4, 7

[20] Y. Wu, J. Lim, and M.-H. Yang, "Online object tracking: A benchmark," in *CVPR*, 2013, pp. 2411–2418. 2

[21] J.-Y. Bouguet, "Pyramidal implementation of the affine lucas kanade feature tracker description of the algorithm," *Intel Corporation*, vol. 5, no. 1-10, p. 4, 2001. 4

[22] S. Baker and I. Matthews, "Equivalence and efficiency of image alignment algorithms," in *CVPR*, vol. 1. IEEE, 2001, pp. I–1090. 4

[23] G. G. Scandaroli, M. Meilland, and R. Richa, "Improving NCC-based Direct Visual Tracking," in *ECCV*. Springer, 2012, pp. 442–455. 4

[24] A. Roy, X. Zhang, N. Wolleb, C. Perez, Quenterio, and M. Jagersand, "Tracking benchmark and evaluation for manipulation tasks," in *ICRA*. IEEE, 2015. 4, 5

[25] A. Adam, E. Rivlin, and I. Shimshoni, "Robust fragments-based tracking using the integral histogram," in *CVPR*, vol. 1. IEEE, 2006, pp. 798–805. 4