Robust 3D Visual Tracking Using Particle Filtering on the SE(3) Group

Changhyun Choi and Henrik I. Christensen
Robotics & Intelligent Machines, College of Computing
Georgia Institute of Technology
Atlanta, GA 30332, USA
{cchoi,hic}@cc.gatech.edu

Abstract— **In this paper, we present a 3D model-based object tracking approach using edge and keypoint features in a particle filtering framework. Edge points provide 1D information for pose estimation and it is natural to consider multiple hypotheses. Recently, particle filtering based approaches have been proposed to integrate multiple hypotheses and have shown good performance, but most of the work has made an assumption that an initial pose is given. To remove this assumption, we employ keypoint features for initialization of the filter. Given 2D-3D keypoint correspondences, we choose a set of minimum correspondences to calculate a set of possible pose hypotheses. Based on the inlier ratio of correspondences, the set of poses are drawn to initialize particles. For better performance, we employ an autoregressive state dynamics and apply it to a coordinate-invariant particle filter on the *SE(3)* group. Based on the number of effective particles calculated during tracking, the proposed system re-initializes particles when the tracked object goes out of sight or is occluded. The robustness and accuracy of our approach is demonstrated via comparative experiments.**

I. Introduction

From robotic manipulation to augmented reality, estimating poses of objects is a key task. Since Harris [1] proposed his early system which tracks an object by projecting a 3D CAD model into a 2D image and aligning the projected model edges to image edges, there have been active efforts to enhance the early edge-based tracking system [2], [3]. Edges are employed because they are easy to compute and invariant to illumination and pose changes. However, a critical disadvantage of using edges in visual tracking is that they look similar to each other. In general, edge correspondences are determined by local search based on a prior pose estimate. So the tracking performance of an edge-based tracker directly depends on correct pose priors. To improve pose priors, there have been attempts to enhance the pose accuracy between frames by incorporating interest points [4], [5], [6] or employing additional sensors [7]. Since interest points and their rich descriptors [8], [9] can be extracted and matched well under illumination, scale, rotation changes, and reasonable projection transformation, keypoint features complement edges well.

Considering multiple edge correspondences was another interest in edge-based tracking. Since edges are ambiguous and false edge correspondences directly lead the tracker to false pose estimates, some approaches have considered multiple edge correspondences [5], [10]. However, their work was still limited because only one or two hypotheses were maintained from the multiple correspondences during tracking.

Multiple hypotheses tracking has been implemented using a particle filtering framework. Isard and Blake [11] applied a particle filter to 2D visual edge tracking and have shown great potential. Affine 2D visual trackers have also been proposed in a particle filter framework with incremental measurement learning [12], [13]. Among them, Kwon *et al.* [13] proposed a particle filter on a 2D affine Lie group, *Aff(2)*, in a coordinate-invariant way. For 3D visual tracking, Pupilli and Calway [14] have shown the possibility of applying a particle filter to 3D edge tracking. While [14] demonstrated the tracking of simple 3D objects, Klein and Murray [15] implemented a particle filtering approach which tracks complex full 3D objects in real-time by exploiting the GPU. Mörwald *et al.* [16] also exploited the parallel power of GPU to implement a fast model-based 3D visual tracker. With edges from 3D CAD, they also employed edges from texture which possibly contributes to avoid false edge correspondences as well as to enhance the accuracy of pose estimates. Teulière *et al.* [17] recently addressed a similar problem by maintaining multiple hypotheses from low-level edge correspondences.

With a few exceptions [18], most of the work has made an assumption in which trackers start from a given pose. Several efforts [15], [14] used annealed particle filters to find the true pose from scratch without performing an appropriate initialization, but the search space might be too large to converge to the true pose in reasonable time, and even it might not converge to the pose after enough time elapses. It is thus more desirable to employ other information for initialization. The BLORT [18] employed SIFT keypoints [8] to recognize objects and used them for particle initialization.

In this paper we utilize a particle filtering technique on the *SE(3)* group that is based on [19]. For robust 3D visual tracking, we employ keypoint features in initialization and edges in the calculation of measurement likelihoods. Like [15], our system can track complex objects by performing a self-occlusion test. By maintaining multiple hypotheses, our algorithm can reliably track an object on challenging image sequences that have complex background and heavy clutter. Our key contributions are as follows:

- We employ keypoint features as additional visual information. While [15], [14] have used annealing particle filter to find the initial pose, we initialize particles to

highly probable states based on pose estimates calculated from keypoint correspondences. The initialized particles tend to converge faster than the usual annealed particle filtering.
- While previous edge-based trackers [15], [17] have employed random walk models as a motion model, we apply a first-order autoregressive (AR) state dynamics on the *SE(3)* group to be more effective.
- To be fully automatic and reliable in practical settings, our approach monitors the number of effective particles and use the value to decide when the tracker requires re-initialization.

This paper is organized as follows. In Section II, we introduce a particle filtering framework with state and measurement equations. The AR state dynamics is then represented in Section II-B. After explaining how particles are initialized and their likelihoods are evaluated in Section II-C and II-D, respectively, the re-initialization scheme is represented in II-E. Experimental results on various image sequences are shown in Section III.

II. Particle Filter on the *SE(3)* Group

In 3D visual tracking, a state represents a 6-DOF pose of a tracked object, and tracking estimates time-varying change of coordinates. It is well known that the trajectory is not on general vector space, rather it is on Lie groups – in general, the Special Euclidean group *SE(3)* and the affine group *Aff(2)* in 3D and 2D visual tracking, respectively. Since the trajectory we want to estimate is on a Lie group, the particle filter should be applied on Lie groups. Monte Carlo filtering on Lie groups is explicitly addressed in [20], [19], [13]. As argued in [19] and [13], filtering performance and noise distribution of local coordinate-based particle filtering approaches are dependent on the choice of the local coordinates, while particle filtering on Lie groups is coordinate-invariant.

A. State and Measurement Equations

From the continuous general state equations on the *SE(3)* group, discrete system equations is acquired via the first-order exponential Euler discretization [19]:

$$X_t = X_{t-1} \cdot \exp(A(X,t)\Delta t + dW_t\sqrt{\Delta t}), \qquad (1)$$

$$dW_t = \sum_{i=1}^{6} \epsilon_{t,i} E_i,$$

$$\epsilon_t = (\epsilon_{t,1}, \ldots, \epsilon_{t,6})^{\mathsf{T}} \sim \mathcal{N}(\mathbf{0}_{6\times 1}, \Sigma_w)$$

where $X_t \in SE(3)$ is the state at time t, $A : SE(3) \to se(3)$ is a possibly nonlinear map, dW_t represents the Wiener process noise on $se(3)$ with a covariance $\Sigma_w \in \Re^{6\times 6}$, E_i are the i-th basis elements of $se(3)$. The corresponding measurement equation is then:

$$y_t = g(X_t) + n_t,\ n_t \sim \mathcal{N}(\mathbf{0}_{N_y\times 1}, \Sigma_n) \qquad (2)$$

where $g : X_t \to \Re^{N_y}$ is a nonlinear measurement function and n_t is a Gaussian noise with a covariance $\Sigma_n \in \Re^{N_y \times N_y}$.

B. AR State Dynamics

The dynamic model for state evolution is an essential part that has a significant impact on tracker performance. However, many particle filter-based trackers have been based on a random walk model because of its simplicity [15], [17]. AR state dynamics is a good alternative since it is flexible, yet simple to implement. In (1), the term $A(X,t)$ determines the state dynamics. A trivial case, $A(X,t) = 0$, is a random walk model. [13] modeled this via the first-order AR process on the *Aff(2)* as:

$$X_t = X_{t-1} \cdot \exp(A_{t-1} + dW_t\sqrt{\Delta t}), \qquad (3)$$

$$A_{t-1} = a \log(X_{t-2}^{-1} X_{t-1}) \qquad (4)$$

where a is the AR process parameter. Since the *SE(3)* is a compact connected Lie group, the AR process model also holds on the *SE(3)* group [21].

C. Particle Initialization using keypoint Correspondences

Most of the particle filter-based trackers assume that initial states are given. In practice, initial particles are crucial to ensure convergence to a true state. Several trackers [15], [14] search for the true state from scratch, but it is desirable to initialize particle states by using other information. Using keypoints allows for direct estimation of 3D pose, but due to the need for a significant number of correspondences it is either slow or inaccurate. As such, keypoint correspondences are well suited for the filter initialization.

For initialization, we use so-called keyframes which are composed of 2D images and keypoints coordinates (2D and 3D) that have been saved offline. An input image coming from a monocular camera is matched with the keyframes by extracting keypoints and comparing them. To find keypoint correspondences efficiently, we employ the Best-Bin-First (BBF) algorithm using kd-tree data structure [22] that allows execution of the search in $O(n \log n)$. As described in [8], the ratio test is then performed to find distinctive feature matches. While we used RANSAC [23] after determining putative correspondences in our previous work [24], we skip this procedure because in the particle filter framework we can initialize particles in an alternative way in which the basic idea is similar to RANSAC. Instead of explicitly performing RANSAC, we randomly select a set of correspondences from the given putative correspondences and estimate a possible set of poses calculated from them. Since we have 3D coordinates of keypoints in keyframes, we get 2D-3D correspondences from the matching process described above. So we can regard this problem as the Perspective-*n*-Point (P*n*P) problem, in which the pose of a calibrated monocular camera is estimated from n 2D-3D point correspondences, on each set of correspondences. To find a pose from the correspondences, we use the EP*n*P algorithm [25] that provide a $O(n)$ time non-iterative solution for the P*n*P problem. After all particle poses are initialized from randomly selected minimum correspondences, weights of particles are assigned from the number of remaining correspondences c_r and the number of inlier correspondences c_i which coincide with

Algorithm 1 Overall algorithm

```
Initialization
  1) Set $t := 0$.
  2) Set number of particles as $N$
  3) For $i := 1, \dots, N$, set $X_0^{*(i)}$ via EP$n$P, $\pi_0^{*(i)}$ via (5),
     and $A_0^{(i)} := \mathbf{0}_{4\times 4}$.
  4) For $i := 1, \dots, N$, normalize weights $\tilde{\pi}_0^{(i)}$ by (6).
  5) For $i := 1, \dots, N$, draw from $X_0^{*(i)}$ according to $\tilde{\pi}_0^{(i)}$
     to produce $X_0^{(i)}$.
Importance Sampling
  1) Set $t := t + 1$.
  2) For $i := 1, \dots, N$, draw $X_t^{*(i)} \sim P(X_t|X_{t-1}^{(i)}, \Sigma_w)$ by
     a) Generate the Gaussian $\epsilon_t \sim \mathcal{N}(0, \Sigma_w)$, and propagate
        $X_{t-1}^{(i)}$ to $X_t^{*(i)}$ with $A_{t-1}^{(i)}$ via (3).
     b) Compute $A_t^{*(i)}$ with (4).
  3) For $i := 1, \dots, N$, optimize $X_t^{*(i)}$ to $X_t^{'*(i)}$ with IRLS
     via (9) and (10).
  4) For $i := 1, \dots, N$, evaluate the importance weights
     $\pi_t^{*(i)}$ via (8) and (11).
  5) For $i := 1, \dots, N$, normalize the importance weights
     $\tilde{\pi}_t^{(i)}$ by (12).
  6) Evaluate $\widehat{N_{eff}}$ by (13)
Resampling
  1) If $\widehat{N_{eff}} < N_{thres}$
     a) Go to Initialization 3) and initialize $X_t^{(i)}, \tilde{\pi}_t^{(i)}$,
        and $A_t^{(i)}$ for $i := 1, \dots, N$.
  2) Otherwise
     a) For $i := 1, \dots, N$, resample from $X_t^{*(i)}$ and
        $A_t^{*(i)}$ with probability proportional to $\tilde{\pi}_t^{(i)}$ to
        produce i.i.d. random samples $X_t^{(i)}$ and $A_t^{(i)}$.
     b) For $i := 1, \dots, N$, set $\pi_t^{(i)} := \tilde{\pi}_t^{(i)} := \frac{1}{N}$.
     c) Go to Importance Sampling.
```

the pose calculated from the randomly selected set. For $i = 1, \dots, N$ where N is the number of particles, the initial weights of particles are assigned as:

$$\pi_0^{*(i)} \propto p(y_0|X_0^{*(i)}) \propto e^{(-\lambda_c \frac{c_r - c_i}{c_r})} \tag{5}$$

where λ_c is a parameter. Then the weights $\pi_0^{*(i)}$ are normalized by:

$$\tilde{\pi}_0^{(i)} = \frac{\pi_0^{*(i)}}{\sum_{j=1}^{N} \pi_0^{*(j)}} \tag{6}$$

After weights are normalized, particles are randomly drawn with probability proportional to these weights. By doing so, we can generate probable initial pose hypotheses. We initialize particles when the number of correspondences is bigger or equal to 9, and the number of randomly selected minimum correspondences is 7.

D. Measurement Likelihood and Optimization using IRLS

Once each particle is initialized and propagated according to AR process and Gaussian noise, it has to be evaluated based on its measurement likelihood. In edge-based tracking, a 3D wireframe model is projected to a 2D image according to a particle state $X_t^{*(i)}$. Then a set of points is sampled along edges in the wireframe model per a fixed distance. The sampled points are matched to nearest edge pixels from the image by 1D perpendicular search [2], [24]. Then the measurement likelihood can be calculated from the ratio between the number of matched sample points p_m and the number of visible sample points p_v which pass a self-occlusion test as:

$$p(y_t|X_t) \propto e^{(-\lambda_v \frac{(p_v - p_m)}{p_v})} \tag{7}$$

where λ_v is a parameter. This likelihood has been similarly used in [15]. Another choice is taking arithmetic average distances (error) $\bar{e}$ between the matched sample points and the edge pixels [17]:

$$p(y_t|X_t) \propto e^{(-\lambda_e \bar{e})}$$

where λ_e is also a parameter to be tuned. We noticed that both likelihoods are valid, and we empirically found that using both terms shows better results. Therefore, in our approach the measurement likelihood is evaluated as:

$$p(y_t|X_t) \propto e^{(-\lambda_v \frac{(p_v - p_m)}{p_v})} e^{(-\lambda_e \bar{e})} \tag{8}$$

One of the challenges in particle filtering for 3D visual tracking is the large state space, so a large number of particles is usually required for reliable tracking performance. To reduce the number of particles, [15] has used an annealed particle filter, and [26], [17] have selectively employed local optimizations in a subset of particles. For more accurate results, we optimize particles as well, in which Iterative Reweighted Least Squares (IRLS) is employed [24], [2]. From IRLS, the optimized particle $X_t^{'*(i)}$ is calculated as follows:

$$X_t^{'*(i)} = X_t^{*(i)} \cdot \exp(\sum_{i=1}^{6} \mu_i E_i) \tag{9}$$

$$\boldsymbol{\mu} = (J^{\mathsf{T}} W J)^{-1} J^{\mathsf{T}} W \boldsymbol{e} \tag{10}$$

where $\boldsymbol{\mu} \in \mathfrak{R}^6$ is the motion velocity that minimizes the error vector $\boldsymbol{e} \in \mathfrak{R}^{N_y}$, $J \in \mathfrak{R}^{N_y \times 6}$ is a Jacobian matrix of $\boldsymbol{e}$ with respect to $\boldsymbol{\mu}$ obtained by computing partial derivatives at the current pose, and $W \in \mathfrak{R}^{N_y \times N_y}$ is a weighted diagonal matrix. The diagonal element in W is $w_i = \frac{1}{c + e_i}$ where c is a constant and e_i is i-th element of $\boldsymbol{e}$. (For more detail information about J and IRLS, please refer to [24], [2]).

Note that the measurement likelihood in (8) is calculated before the IRLS optimization. To assign weights of particles, we have to evaluate the likelihood again with the optimized state $X_t^{'*(i)}$. However, computing the likelihood of every particle again is computationally expensive. Since every particle is optimized at the same time, we can approximate $p(y_t|X_t^{'*(i)})$ as:

$$\pi_t^{*(i)} \propto p(y_t|X_t^{'*(i)}) \approx p(y_t|X_t^{*(i)}) \tag{11}$$

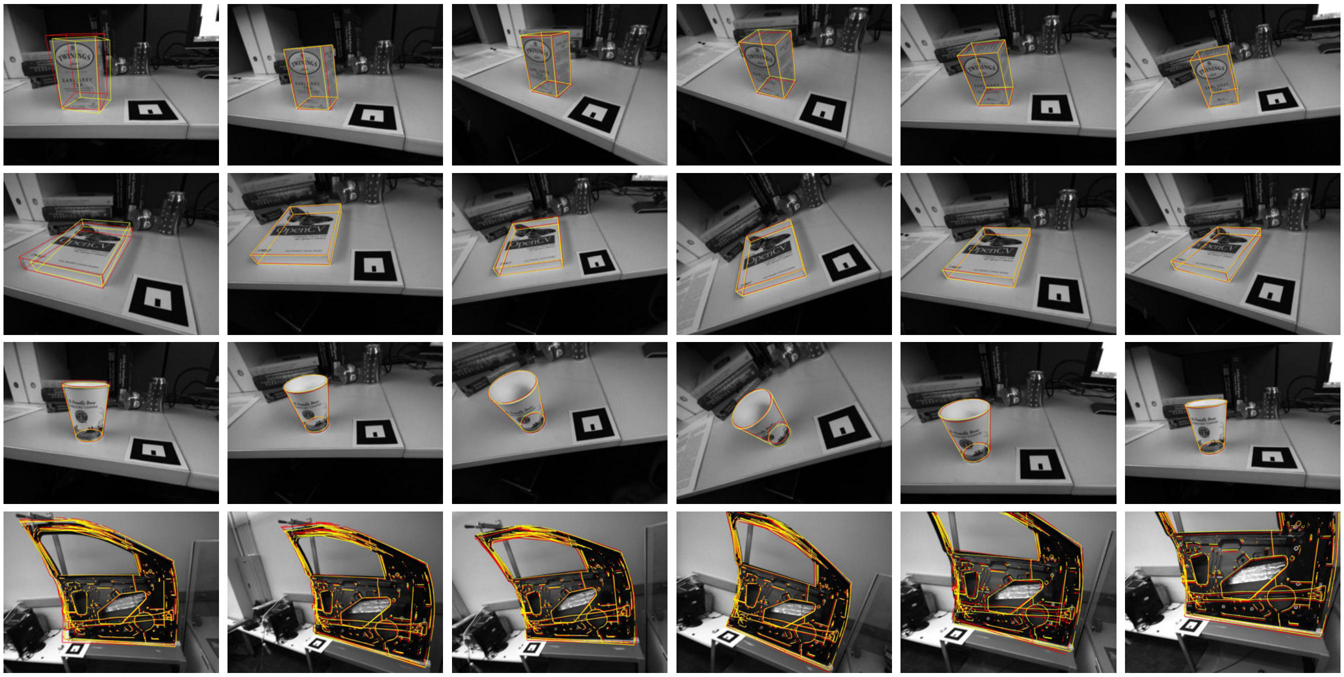

Fig. 1. Tracking results with (yellow wireframe) and without (red wireframe) our particle filter for the four targeted objects. From top to bottom, teabox, book, cup and car door. From left to right, $t < 10, t = 100, t = 200, t = 300, t = 400$ and $t = 500$ where t is the frame number. The very left images are results of the pose initialization. Note that yellow wireframes are well fitted to the tracking objects, while red wireframes are frequently mislocalized. 100 particles are used for the particle filter. (i.e. $N = 100$)

Then the weight $\pi_t^{*(i)}$ is normalized to $\tilde{\pi}_t^{(i)}$ by:

$$\tilde{\pi}_t^{(i)} = \frac{\pi_t^{*(i)}}{\sum_{j=1}^{N} \pi_t^{*(j)}} \tag{12}$$

E. Re-initialization based on $\widehat{N_{eff}}$

Ideally a tracked object should be visible during an entire tracking session. In reality, however, it is quite common that the object goes out of frame or is occluded by other objects. In these cases, the tracker is required to re-initialize the tracking. In general sequential Monte Carlo methods, the effective particle size N_{eff} has been introduced as a suitable measure of degeneracy [27]. Since it is hard to evaluate N_{eff} exactly, an alternative estimate $\widehat{N_{eff}}$ is defined [27]:

$$\widehat{N_{eff}} = \frac{1}{\sum_{i=1}^{N} (\tilde{\pi}^{(i)})^2} \tag{13}$$

Often it has been used as a measure to execute the resampling procedure. But, in our tracker we resample particles every frame, and hence we use $\widehat{N_{eff}}$ as a measure to do re-initialization. When the number of effective particles is below a fixed threshold N_{thres}, the re-initialization procedure is performed. The overall algorithm is shown in Algorithm 1.

III. Experimental Results

In this section, we validate our proposed particle filter-based tracker via various experiments. First, we compare the performance of our approach with the previous single hypothesis tracker [24] which was based on IRLS. For the comparison, we use new challenging image sequences as well as image sequences used in [24]. To verify the effectiveness of the AR state dynamics, we show the results of our proposed tracker with and without the AR state dynamics.

TABLE I

RMS errors in the general tracking

	RMS Errors*					
	x	y	z	roll	pitch	yaw
Teabox	0.0033† **0.0027**‡	**0.0018** 0.0020	0.0068 **0.0031**	3.27 **1.68**	4.32 **1.13**	3.95 **2.13**
Book	0.0026 **0.0016**	0.0021 **0.0012**	**0.0042** 0.0055	1.73 **0.87**	1.58 **0.82**	**0.95** 1.11
Cup	0.0083 **0.0078**	0.0092 **0.0084**	0.0272 **0.0216**	2.09 **1.20**	1.83 **1.00**	5.05 **3.57**
Car door	0.0211 **0.0104**	**0.0122** 0.0135	0.0411 **0.0352**	1.73 **0.89**	3.72 **3.16**	3.73 **1.96**

* The error units of translation and rotation are meter and degree, respectively.

† The upper rows are the results of the previous approach [24].

‡ The lower rows are the results of the proposed approach. In both upper and lower rows, better results are indicated in bold numbers.

A. Experiment 1: Image Sequences from Choi and Christensen [24]

In [24], the sequences of images were captured from a monocular camera in static object and moving camera setting. A set of image sequences used in Section III-A.1 are acquired to test tracking performance in general setting, i.e. no occlusion, relatively simple background, reasonable clutter, and smooth movements of the camera. To test re-initialization capability, a sequence of images is captured

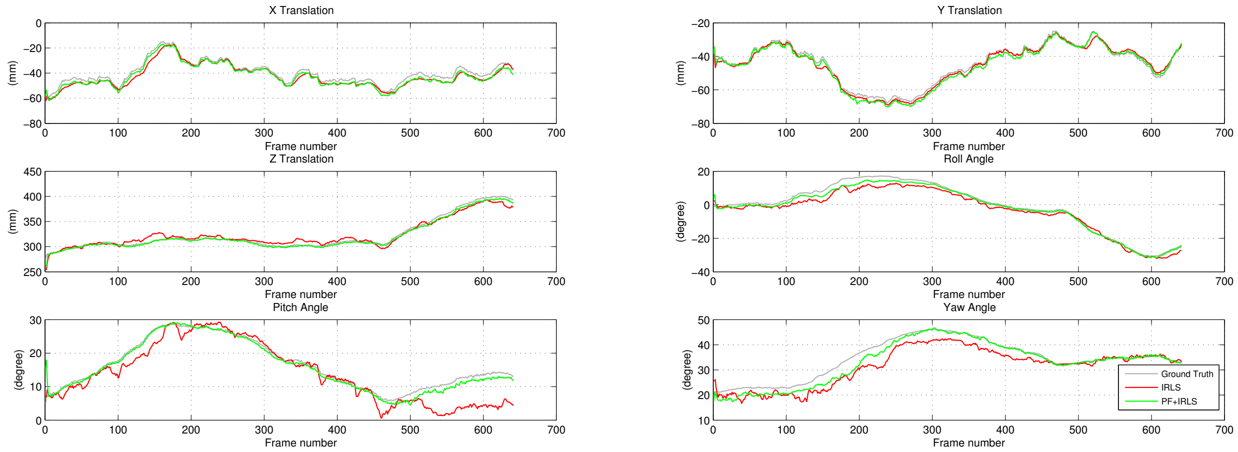

Fig. 2. Pose plots of the teabox object in the general tracking test. The proposed approach (PF+IRLS) shows superior accuracy than the previous approach (IRLS). Especially, using our particle filter significantly enhances rotational accuracy.

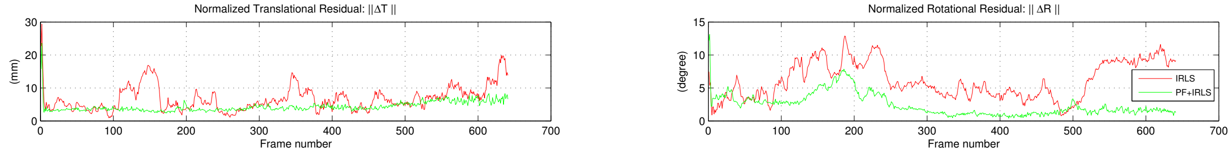

Fig. 3. Normalized residual plots of the teabox object in the general tracking test. In general, the proposed approach (PF+IRLS) shows lower residual and more consistent results than the previous one (IRLS).

with fast camera motion and occlusions. This sequence is tested in Section III-A.2.

1) General Tracking: The single hypothesis tracker in [24] has shown reasonable tracking results on the general tracking sequences. To show quantitative results, AR markers are employed to gather ground truth pose estimates. To compare our proposed approach with the previous one, we executed our new approach on the same image sequences. The tracking results are shown in Fig. 1. In the proposed approach, 100 particles are maintained, and the mean of particles is depicted in the figures. To calculate the mean of particles, we follow the mean rotation evaluation of the special orthogonal group *SO(3)* by Moakher [28] that is also considered in [19], [17]. To clearly show the difference of the two approaches, both of the pose results are depicted in the same image sequences. The yellow and red wireframes are projected to images with respect to the pose results estimated by the proposed approach and previous approach, respectively. We can easily verify that the proposed approach shows more accurate results. To decompose pose results, 6-DOF pose and residual plots of the teabox object are represented in Fig. 2 and 3, respectively. Based on the plots, we can easily see the difference where the proposed approach (PF+IRLS) shows much better results than the previous approach (IRLS). Note that the considerable differences in orientation estimates. This is due to some false edge correspondences that lead to errors in pose, mainly in orientation. Since our particle filter considers multiple hypotheses and resamples based on measurement likelihood, it is quite robust to false edge correspondences from which the single hypothesis tracker is often suffered. A quantitative analysis of these tests is represented in Table I which shows the root mean square (RMS) errors. For each object, the upper rows are the results of the previous approach and the lower rows are the results of the proposed approach. With a few exceptions, the proposed approach outperforms the previous one in terms of accuracy. As mentioned earlier, the proposed approach shows better orientation estimates. Although there are a few exceptions, the difference is about 1 mm in translation and 0.16 deg in rotation. Since the ground truth was measured via the AR marker and the displacement between the object and the AR marker was measured manually, that errors might come from these measures.

2) Re-initialization: In our previous system, we used a simple heuristic in which the difference in position of the object between frames and the number of valid sample points are monitored to trigger re-initialization [24]. While that heuristic works well when the pose hypothesis drifts fast, it might not always be the case when the hypothesis stuck in local minima. Here we propose another way for re-initialization by taking advantage of multiple hypotheses. As in Algorithm 1, our system re-initializes when the number of effective particles $\widehat{N_{eff}}$ is below a threshold. To verify this method, we run the proposed tracker on the re-initialization sequence. The tracking results are shown in Fig. 4 and the number of effective particles is plotted over the frame numbers in Fig. 5. The gray line represents the threshold value N_{thres}. When the tracked object goes out of frames, images are blurred because of camera shaking, or the object is occluded with a paper, the $\widehat{N_{eff}}$ decreases significantly, and that triggers the re-initialization. During re-initialization, the tracker matches keypoints until it has at least the minimum

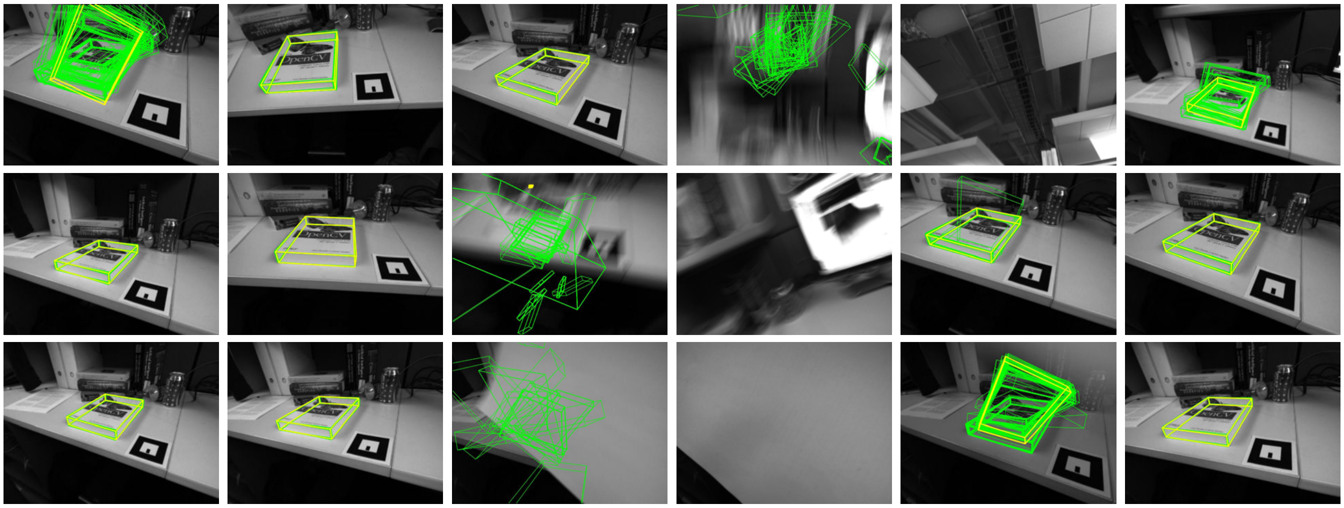

Fig. 4. Tracking results on the re-initialization sequence. From top-left to bottom-right, the frame numbers are $t = 0$, 100, 200, 275, 300, 328, 400, 500, 589, 600, 627, 700, 800, 900, 984, 1000, 1036, and 1100. The green wireframes represent particles and the yellow thick wireframe shows the mean of particles on each image. When the tracked object goes out of frames, images are blurred because of camera shaking, and the object is occluded with a paper, our tracker re-initialize. ($N = 100$)

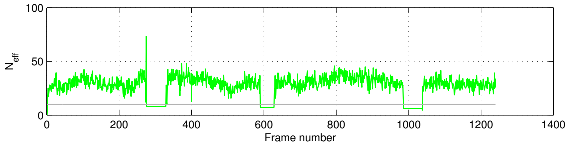

Fig. 5. The $\widehat{N_{eff}}$ plot for the re-initialization sequence. ($N = 100$)

number of keypoint correspondences. Once enough point correspondences are acquired, the proposed system initializes particles.

B. Experiment 2: More Challenging Image Sequences

Since the aforementioned image sequences were prepared for the previous approach, it is relatively easy to track objects in these sequences. Therefore, we need other datasets to compare our new approach. So we prepared more challenging image sequences in which a complex background is considered.

1) Effectiveness of Particle Filter: When the background is relatively simple, single hypothesis edge-based tracking works reasonably. But it is quite challenging to reliably track an object when the background is complex or there is an amount of clutter. These challenging situations often make erratic edge correspondences, hence single hypothesis tracking can be a fragile solution in these cases. To validate this argument, we compare our proposed tracker with the single hypothesis tracker. We captured two image sequences for the book and cup objects. To make the background complex, we put these objects on a camera calibration plate in which the grid pattern is likely to generate false edge correspondences. The comparative tracking results are shown in Fig. 6. The grid pattern and the background texture play a role as strong clutter, hence the previous approach suffers from the local minima. However, our approach dependably tracks objects in spite of the clutter.

2) Effectiveness of AR State Dynamics: To verify the effect of the AR state dynamics, we execute the proposed approach with and without the AR state dynamics. To disable the dynamics, we set the parameter a in (4) as 0 which is equivalent to a random walk model. For fair comparison, we use the same parameters except the AR parameter. We test on a sequence of the book object used in the previous experiment. The tracking results are represented in Fig. 7. Although both use the same number of particles, Gaussian noise, and measurement likelihood, the tracking performances are quite distinctive. This difference is mainly due to the AR state dynamics which propagates particles according to the camera motion.

IV. Conclusions

We have presented an approach to 3D visual object tracking based on a particle filtering algorithm on the *SE(3)* group. For fast particle convergence, we employed keypoint features and initialized particles by using a linear time non-iterative solution for the P*n*P problem. Particles are propagated by the state dynamics which is given by an AR process on the *SE(3)*, and the state dynamics distributed particles more effectively. Measurement likelihood was calculated from both the residual and the number of valid sample points of the edge correspondences. During the tracking, the proposed system appropriately re-initialized by itself when the number of effective particles is below a threshold. Our approach has been tested via various experiments in which our multiple hypotheses tracker has shown notable performance on challenging background and clutter.

One of the possibilities for future work is exploiting the parallel power of GPU for real-time performance [15], [16]. Another interest is in multiple object recognition and tracking [29] which is necessary for a realistic scenario.

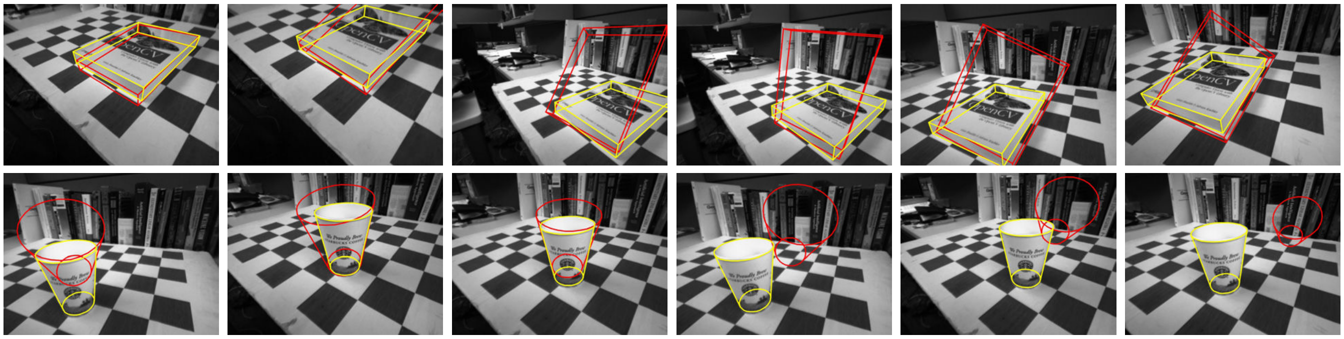

Fig. 6. Tracking results on more challenging image sequences. Top and bottom rows represent selected frames from book and cup sequences, respectively. For comparison, the poses estimated by the proposed (PF+IRLS) and the previous (IRLS) approaches are depicted in yellow and red wireframes, respectively. Because of background texture, the previous approach is often misled to local minima, while the proposed approach robustly estimates poses of objects under ambiguous background texture and fast camera motions. ($N = 100$)

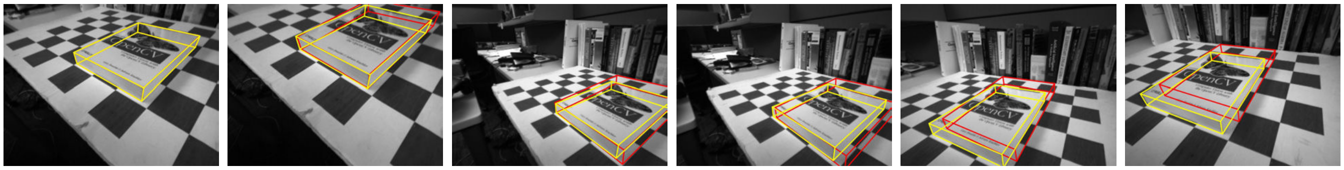

Fig. 7. Tracking results for the proposed approach with (yellow wireframe) and without (red wireframe) the AR state dynamics. It shows that the AR state dynamics contributes to propagate particles more effectively. ($N = 100$)

V. ACKNOWLEDGMENTS

This work was partially funded and developed under a Collaborative Research Project between Georgia Tech and the General Motors R&D, Manufacturing Systems Research Laboratory. The support from General Motors is gratefully acknowledged.

REFERENCES

[1] C. Harris, *Tracking with Rigid Objects*. MIT Press, 1992.

[2] T. Drummond and R. Cipolla, "Real-time visual tracking of complex structures," *PAMI*, vol. 24, no. 7, pp. 932–946, 2002.

[3] A. I. Comport, E. Marchand, and F. Chaumette, "Robust model-based tracking for robot vision," in *IROS*, vol. 1, 2004.

[4] E. Rosten and T. Drummond, "Fusing points and lines for high performance tracking," in *ICCV*, vol. 2, 2005.

[5] L. Vacchetti, V. Lepetit, and P. Fua, "Combining edge and texture information for real-time accurate 3d camera tracking," in *ISMAR*, 2004, pp. 48–56.

[6] M. Pressigout and E. Marchand, "Real-time 3d model-based tracking: Combining edge and texture information," in *ICRA*, 2006, pp. 2726–2731.

[7] G. Klein and T. Drummond, "Tightly integrated sensor fusion for robust visual tracking," *Image and Vision Computing*, vol. 22, no. 10, pp. 769–776, 2004.

[8] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *IJCV*, vol. 60, no. 2, pp. 91–110, 2004.

[9] H. Bay, A. Ess, T. Tuytelaars, and L. V. Gool, "Speeded-up robust features (SURF)," *Computer Vision and Image Understanding*, vol. 110, no. 3, pp. 346–359, 2008.

[10] C. Kemp and T. Drummond, "Dynamic measurement clustering to aid real time tracking," in *ICCV*, 2005, pp. 1500–1507.

[11] M. Isard and A. Blake, "Condensation–conditional density propagation for visual tracking," *IJCV*, vol. 29, no. 1, pp. 5–28, 1998.

[12] D. A. Ross, J. Lim, R. S. Lin, and M. H. Yang, "Incremental learning for robust visual tracking," *IJCV*, vol. 77, no. 1, pp. 125–141, 2008.

[13] J. Kwon and F. C. Park, "Visual tracking via particle filtering on the affine group," *IJRR*, vol. 29, no. 2-3, p. 198, 2010.

[14] M. Pupilli and A. Calway, "Real-time camera tracking using known 3D models and a particle filter," in *ICPR*, vol. 1, 2006.

[15] G. Klein and D. Murray, "Full-3d edge tracking with a particle filter," *BMVC*, 2006.

[16] T. Mörwald, M. Zillich, and M. Vincze, "Edge tracking of textured objects with a recursive particle filter," in *19th International Conference on Computer Graphics and Vision (Graphicon), Moscow*, 2009, pp. 96–103.

[17] C. Teulière, E. Marchand, and L. Eck, "Using multiple hypothesis in model-based tracking," in *ICRA*, 2010.

[18] T. Mörwald, J. Prankl, A. Richtsfeld, M. Zillich, and M. Vincze, "BLORT - the blocks world robotic vision toolbox," in *In Best Practice in 3D Perception and Modeling for Mobile Manipulation (in conjunction with ICRA 2010)*, 2010.

[19] J. Kwon, M. Choi, F. C. Park, and C. Chun, "Particle filtering on the Euclidean group: framework and applications," *Robotica*, vol. 25, no. 06, pp. 725–737, 2007.

[20] A. Chiuso and S. Soatto, "Monte Carlo filtering on Lie groups," in *IEEE Conference on Decision and Control*, vol. 1, 2000, pp. 304–309.

[21] J. Xavier and J. H. Manton, "On the generalization of AR processes to Riemannian manifolds," *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing.*, 2006.

[22] J. Beis and D. Lowe, "Shape indexing using approximate nearest-neighbour search in high-dimensional spaces," in *CVPR*, 1997, pp. 1000–1006.

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

[24] C. Choi and H. I. Christensen, "Real-time 3D model-based tracking using edge and keypoint features for robotic manipulation," in *ICRA*, 2010, pp. 4048–4055.

[25] V. Lepetit, F. Moreno-Noguer, and P. Fua, "EPnP: An accurate O(n) solution to the PnP problem," *IJCV*, vol. 81, no. 2, pp. 155–166, 2009.

[26] M. Bray, E. Koller-Meier, and L. V. Gool, "Smart particle filtering for 3D hand tracking," in *Sixth IEEE International Conference on Automatic Face and Gesture Recognition, 2004. Proceedings*, 2004, pp. 675–680.

[27] A. Doucet, S. Godsill, and C. Andrieu, "On sequential Monte Carlo sampling methods for Bayesian filtering," *Statistics and computing*, vol. 10, no. 3, pp. 197–208, 2000.

[28] M. Moakher, "Means and averaging in the group of rotations," *SIAM Journal on Matrix Analysis and Applications*, vol. 24, no. 1, p. 116, 2003.

[29] A. Collet, D. Berenson, S. S. Srinivasa, and D. Ferguson, "Object recognition and full pose registration from a single image for robotic manipulation," in *ICRA*, 2009, pp. 48–55.