Coupled 3D Tracking and Pose Optimization of Rigid Objects Using Particle Filter

Heng Yang[1], Yueqiang Zhang[2], Xiaolin Liu[2], Ioannis Patras[1]
[1]*Queen Mary University of London* [2]*National University of Defense Technology*
{*heng.yang,i.patras*}@*eecs.qmul.ac.uk* {*zyoung,lxl06*}@*yahoo.com.cn*

Abstract

In order to track and estimate the pose of known rigid objects with high accuracy in unconstrained environment with light disturbance, scale changes and occlusion, we propose to combine 3D particle filter (PF) framework with algebraic pose optimization in a closed loop. A new PF observation model based on line similarity in 3D space is devised and the output of 3D PF tracking, namely line correspondences (model edges and image line segments), are provided for algebraic line-based pose optimization. As a feedback, the optimized pose serves as a particle with high weight during re-sampling. To speed up the algorithm, a dynamic ROI is used for reducing the line detection and search space. Experiments show our proposed algorithm can effectively track and accurately estimate the pose of freely moving 3D rigid objects in complex environment.

1. Introduction

Many applications require tracking and pose estimation of a known object in real-time with high accuracy based on visual sensors. These include Augmented Reality systems, visual servoing of robotic arms for capturing object, etc. The aim is to continuously recover all six degrees of freedom that define the camera position and orientation relative to the object.

Conventional methods often assume that certain feature correspondences are available between the 2D image and 3D model of the object. These correspondences are assumed to be easily established in each frame, for example, point correspondence [4, 7], line correspondence [3, 6], where fiducials, or markers are required. In our work, we focus on marker-less, low-texture, freely tumbling objects. The 3D model of the object is known or can be reconstructed on-line using methods such as structure from motion. Since textures or fiducials are absent, we depend on object edges for tracking because edges are rich on man-made equipments and easy to detect in images. They offer a large degree of invariance to pose and illumination changes and have some resilience to difficult imaging conditions (noise and blur). The primary disadvantage of edges as features to track is that one image line looks much like another. Therefore we employ Particle Filter tracking to identify edge/line pairs and further optimize the pose algebraically based on these correspondences.

In related work, Pupilli *et al.* [8] firstly developed a particle filter framework for 3D object tracking based on edges, which was further improved in [9] by grouping the edge segments into junctions. [5] firstly exploited capacity of GPU to perform likelihoods evaluation. A more general GPU-based particle filter framework was proposed in [2]. There are two main drawback of particle filter, namely, low accuracy and high computational complexity. [2] tried to improve the accuracy by only increasing the number of particles and achieve real-time implementation based on GPU-acceleration. A similar work is [1] which improved the performance by a new observation model which measures particle weights in object space instead of on 2D image plane.

In this paper, we firstly propose a new observation model in object space in order to better measure the likelihood of observation and each particle. We then combine particle filtering with algebraic pose optimization in a closed loop to improve the 3D pose accuracy using only a small number of particles. To further speed up, we propose a dynamic image ROI to reduce line detection and search space. Finally, experiments are conducted to show the performance of the proposed method in tracking low texture freely moving target under light changes in clutter background and occlusion.

2. Method

The framework of our method is shown in Fig. 1. Unlike the previous methods, we combine the alge-

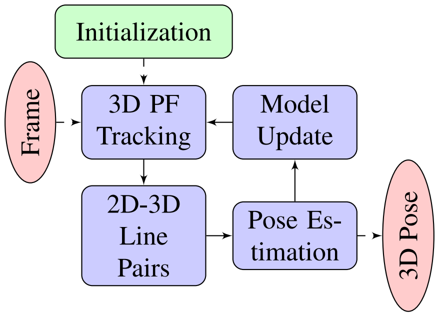

Figure 1. Algorithm flowchart

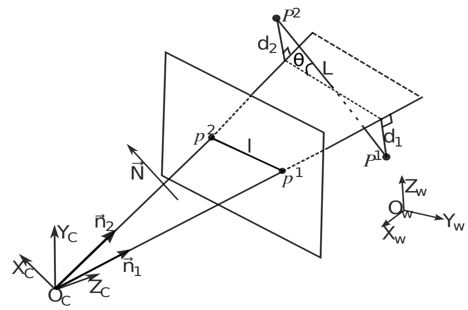

Figure 2. Perspective projection of 3D line

braic pose estimation and 3D particle filter tracking in a closed loop. Firstly, a rough pose is obtained through 3D particle filter tracking. Based on the pose and model information, all possible 3D edges are projected onto the image to identify the extracted line segments. By this end, the correspondences between the 3D model edges and 2D selected line segments are built. Then, pose is optimized using Mirzaei's method [6]. Further more, this optimized pose information serves as a highly confident re-sampling center during model updating in order to guarantee the convergence of the particle filter.

2.1 Problem Formulization

The relative pose state between the camera and the object is represented by position and orientation, representing by Rodrigues' vector. Thus the pose state is given by: $X = [r_1, r_2, r_3, t_x, t_y, t_z]^T$, and $[R, t]^T$ in short. At frame k, it is denoted by X_k. The 2D edge map in the current frame k forms the observation y_k. The sequence of observations up to the current frame then is denoted $y_{1:k}$. Tracking then involves recursively approximating the posterior density $p(X_k|y_{1:k})$. In particle filter, the posterior is represented by a set of weighted particles $X_k^1, X_k^2, ..., X_k^S$. The weight w_k^s is proportional to the likelihood of the observation given the pose state and model information, $p(y_k|X_k^s, Z)$, where Z denotes model prior and $\sum_{s=1}^{S} w_k^s = 1$. There two key components of PF, state dynamics and observation functions. The state dynamics describes the state evaluation probability between time steps. As we focus on freely tumbling objects, the random walk [1] model is adapted, i.e.

$$p(X_k|X_{k-1}) = U(X_{k-1} - v, X_{k-1} + v), \quad (1)$$

where U denotes uniform distribution and v is the uncertainty about the incremental movement.

2.2 Observation Model

Let the edge set extracted from the observation (frame) be $S_l = \{l_1, l_2, ..., l_i, ..., l_n\}$, where each edge line is represented by two 2D end-points (p_i^1, p_i^2). Let 3D model be a set of 3D edges, $Z = \{L_1, L_2, ..., L_j, ..., L_m\}$, where each edge is represented by two 3D end-points (P_j^1, P_j^2). Here, our method only requires the 3D information of some the salient edges instead of the full CAD model. These edges can be only from a part of a big target, such as an attached component. A perspective projection from a 3D point P to 2D image p can be given as $p \sim K \cdot [RP + t]$ with K the pre-calibrated camera intrinsic matrix. We further define the rigid transformation $\mathcal{F}$ which operates on the end points of a 3D line from object coordinate to camera coordinate, i.e.,$L^C = \mathcal{F}(L, X_k^s)$. To compute the likelihood, [1] used a function related to the number of the model 3D lines whose projection into the camera frame are within a given threshold of extracted projection planes, i.e.,

$$p(y_k|X_k^s, Z) = exp\{-\sum_{i=1}^{n}\sum_{j=1}^{m}\rho(\vec{N}_i \cdot \mathcal{F}(L_j, X_k^s))\}, \quad (2)$$

where $\vec{N} = \vec{n_1} \times \vec{n_2}$ is the normal of the projection plane deduced by a line on image as shown in Fig. 2 and $\rho(\vec{N}_i \cdot \mathcal{F}(L_j, X_k^s))$ indicates whether the 3D line L_j is an inlier or outlier w.r.t. the observation and state, i.e.,

$$\rho(\vec{N}_i, L_j, X_k^s) = \begin{cases} 1 & \text{if} \vec{N}_i \cdot \mathcal{F}(L_j, X_k^s) < \varepsilon \\ 0 & \text{otherwise} \end{cases} . \quad (3)$$

This observation model was proven more powerful than measuring the likelihood in 2D image but the function 1) failed to consider the relative localization of 3D model lines to optical rays deduced by image points; 2)

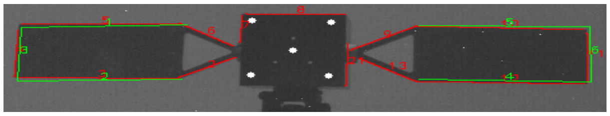

Figure 3. Identify line-correspondences

assigned each pair with only two scores, 0 and 1, which limited the diversity of particles resulting in heavy degeneracy phenomenon when tracking fast moving objects; 3) the constant threshold in the function is sensitive to distance changes. Therefore, we have devised a new observation model in order to better measure the likelihood of 2D and 3D pairs

$$\rho(\vec{N}_i, L_j, X_k^s) = \begin{cases} e^{-(\lambda_1 \dfrac{\theta}{\varepsilon_\theta} + \lambda_2 \dfrac{d}{\varepsilon_d})} & \text{if}(\theta < \varepsilon_\theta)\&(d < \varepsilon_d) \\ 0 & \text{otherwise} \end{cases} \quad (4)$$

where θ is the angle between predicated 3D line and the projection plan. $d = (d_1 + d_2)/|(P^1 + P^2)/2|$ is the sum of the distance of the two end points to their nearer optical ray, divided by the the distance between the middle point and camera center, as shown in Fig. 2. λ_1 and λ_2 is designed to balance the relative importance of the angle and distance information. Finally, the weight of each particle is given by $w_k^s = \frac{p(y_t|X_k^s, Z)}{\sum_k^S p(y_t|X_k^s, Z)}$ and the tracked output is

$$\hat{X}_k = \sum_{s=1}^{S} w_k^s \cdot X_k^s. \quad (5)$$

2.3 Pose Estimation and Re-sampling

To this end, we have built a framework for 3D object tracking, however, in some application areas such as servoing control where high accuracy is required, we have to further improve the output. Based on the tracked output $\hat{X}$, according to perspective projection, the visible object edges, $Z^v \in Z$, are projected to the image frame as $S_l^P = \{l_1^P, ..., l_{m'}^P\}$, $m' \leq m$. Our purpose is to find a *real* edge segment from image line set S_l for each projection, in other word, to identify meaningful line segments and find 3D-edge/2D-line pairs. The output from tracking makes this step easy for implementation as the estimated projection is very close to its real image as shown in Fig. 3, in which red lines are line detection result and green lines are projection of meaningful model edges using tracked pose, and the closed line can be identified by searching in a small neighbour area. Therefore, the correspondences of this frame is $\{L_1 \leftrightarrow l_5, L_2 \leftrightarrow l_3, ..., L_6 \leftrightarrow l_1\}$. When at least 4 pairs of line correspondences are found, we employ the method proposed in [6] to compute an optimized pose $\bar{X}$. The key idea behind this method is to find a global minimum of the least-squares cost function for orientation error using algebraic geometry techniques firstly and then the position is determined. No initial estimate is required and the computational cost is only linear in the number of measurements. To make full use of the optimized pose, we treat it as a particle with state predication $\bar{X}$ and compute the weight as standard particles, mostly, the highest value, and then normalize all the particle weights. We adapt the standard Sequential Importance Re-sampling (SIR) procedure during which new particle is selected at a frequency proportional to its importance weight. In this way, we close the loop by adding an optimized pose as a feedback to the 3D tracking.

2.4 Dynamic ROI

From above description, the computational complexity is determined by the number of line segments in image because of ergodic searching. In order to speed up the algorithm, we propose a dynamic ROI to reduce the search space. Once the pose of the object is established, a bounding box b of the object on image can be detected according to the projection. We scale the bounding box to b^+ by adding perturbations on four sides considering object movement. In the following frame, instead of detecting lines in the whole image, we only favour the lines lie in the ROI, b^+. In this way, we greatly decrease the detection and search space particularly in clutter background environment.

3. Experiment

In this section, we built a simple satellite model as tracking target as shown in Fig.3. The main part is a cube with a rectangle board on left and right faces. On the front face, there are 5 fiducial points at different height. In experiment, we empirically set particle number as 400, $\varepsilon_d = 0.2$, $\varepsilon_\theta = \dfrac{\pi}{6}$, $\lambda_1 = \lambda_2 = 0.5$.

Firstly, we test the robustness of our algorithm on satellite model moving in complex environment with light disturbance, clutter background and occlusion. Image frame examples from experiment are shown in the first row in Fig. 4. The red lines are detected edges using Hough Transform. The magenta lines are projection result of six edges on model which fit the image edge well and the green dots in the right two images are projection deduced by predicated pose of all the particles. We also tried to track the fiducial points based on

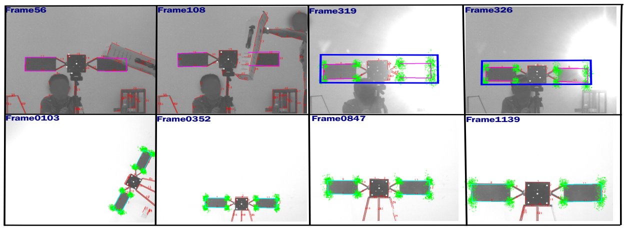

Figure 4. Image frame examples

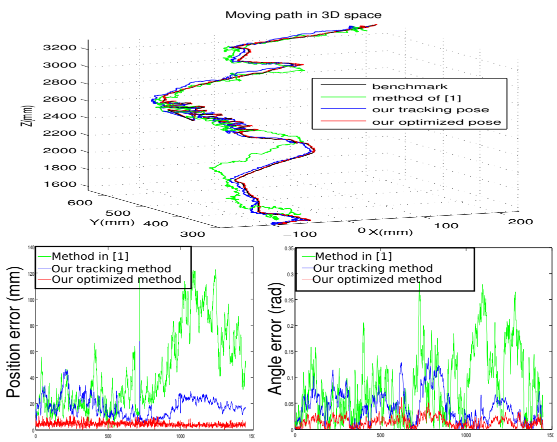

Figure 5. 3D pose and error comparison

appearance but failed when light disturbance and occlusion occurs. On the contrary, according to the result, our edge-based 3D PF tracker is much more robust to disturbance in this challenging environment.

To show to efficiency of our new observation model we compare with similar model in [1]. In order to track fiducial points and acquire benchmark pose, we carry out the experiments in simple background without heavy disturbance. Based on tracked fiducial points, we use the *PnP* method in [7] to compute benchmark pose. Example frames are shown in second row of Fig.4 of which the camera is approaching the target and adjusting the rotation from a distance around 4m to 1m at Z direction for all 1450 frames. Our proposed closed-loop algorithm is realized of which 6 corresponding edge/line pairs are identified and passed to algebraically pose optimization. As a feedback, the optimized pose is treated as a particle with high weight in re-sampling. The benchmark 3D pose, tracked result of [1], our proposed tracked result and optimized 3D pose along with the error (position and rotation) w.r.t benchmark are compared and shown in Fig. 5. According to the result, when the camera agent starts approaching the target at a fast speed, after around frame #700, [1] sharply loses performance while there is no significant change of our proposed tracking. In the whole process, our proposed observation model works better. Due to our closed-loop, we can achieve high accuracy in pose estimation ($<$ 5mm in position and 2°in rotation). The dynamic ROI predication for next frame, as the blue rectangle in right two frames of the first row in Fig. 4, guarantees our algorithm works in a fast speed. In this experiment, the average FPS with and without dynamic ROI is respectively 16.2 and 5.3. The improvement is more significant in clutter background.

4. Conclusion

In this paper, we have combined 3D particle filter tracking and algebraic pose optimization in a closed loop for pose estimation of known model rigid objects. Our proposed observation model which integrates distance and angle information to measure the line similarity in 3D space. Object pose is optimized algebraically based on tracking result and serves as a feedback for particle filter re-sampling. At last, our proposed dynamic ROI guarantees our algorithm effectively works in real-time. The performance of our algorithm is validated on tracking freely moving object in complex environments.

Acknowledgements
The author Heng Yang thanks QMUL-CSC scholarship and this work was mainly funded by National University of Defense Technology.

References

[1] F. Ababsa and M. Mallem. Robust camera pose tracking for augmented reality using particle filtering framework. *Machine Vision and Applications*, 2011.

[2] J. Brown and D. Capson. A framework for 3d model-based visual tracking using a gpu-accelerated particle filter. *T-V&CG*, 2011.

[3] P. David, D. DeMenthon, R. Duraiswami, and H. Samet. Simultaneous pose and correspondence determination using line features. In *CVPR 2003*.

[4] J. Hesch and S. Roumeliotis. A direct least-squares (dls) method for pnp. In *ICCV*, 2011.

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

[6] F. Mirzaei and S. Roumeliotis. Globally optimal pose estimation from line correspondences. In *ICRA2011*.

[7] F. Moreno-Noguer, V. Lepetit, and P. Fua. Accurate non-iterative o (n) solution to the pnp problem. In *ICCV*, 2007.

[8] M. Pupilli and A. Calway. Real-time camera tracking using a particle filter. In *BMVC*, 2005.

[9] M. Pupilli and A. Calway. Real-time camera tracking using known 3d models and a particle filter. In *ICPR*, 2006.