Pose Optimization in Edge Distance Field for Textureless 3D Object Tracking

Bin Wang  
School of Computer Science and Technology, Shandong University  
binwangsdu@gmail.com

Fan Zhong  
School of Computer Science and Technology, Shandong University  
zhongfan@sdu.edu.cn

Xueying Qin  
School of Computer Science and Technology, Shandong University  
quxy@sdu.edu.cn

ABSTRACT

This paper presents a monocular model-based 3D tracking approach for textureless objects. Instead of explicitly searching for 3D-2D correspondences as previous methods, which unavoidably generates individual outlier matches, we aim to minimize the holistic distance between the predicted object contour and the query image edges. We propose a method that can directly solve 3D pose parameters in unsegmented edge distance field. We derive the differentials of edge matching distance with respect to the pose parameters, and search the optimal 3D pose parameters using standard gradient-based non-linear optimization techniques. To avoid being trapped in local minima and to deal with potential large inter-frame motions, a particle filtering process with a first order autoregressive state dynamics is exploited. Occlusions are handled by a robust estimator. The effectiveness of our approach is demonstrated using comparative experiments on real image sequences with occlusions, large motions and cluttered backgrounds.

KEYWORDS

• Computing methodologies → Mixed / augmented reality; Tracking;

CCS CONCEPTS

3D tracking, Pose optimization, Distance field, Particle filter

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

ACM Reference format:


https://doi.org/10.1145/3095140.3095172

1 Introduction

3D object tracking is a fundamental computer vision task with a variety of applications in augmented reality and robotics.

3D tracking systems are expected to estimate the six degrees of freedom (6DoF) pose parameters of an object relative to the camera in unknown and dynamic environments.

Thanks to robust keypoint extractors and descriptors [1, 14, 19], keypoint-based 3D tracking methods [13, 17, 23] have been proposed in last decades. Although these methods achieve impressive performance for textured objects, they are not applicable for textureless objects due to the lack of reliable feature matches.

For textureless objects, edges or contours are the vital visual cue that can be detected in most situations. Therefore, edges or contours are exploited in edge-based 3D tracking methods. RAPID [7] is the first edge-based 3D tracker by projecting the sampled 3D model edge points to a 2D image and aligning the projected edge points with the image edge points. A 1D search for the image edge point is performed at the projected edge point along the direction perpendicular to the projected model edge, and the 2D pixel position with maximum gradient is considered as the correspondence of the sampled 3D model point. Several improvements [5, 15, 20, 22, 24, 25] have been proposed for better 3D-2D correspondences afterwards. These methods are shown to be effective in some situations. However, in order to obtain 3D-2D correspondences, all of these methods perform 1D local search perpendicular to object contours within a limited extent. Since individual contour points are usually indistinctive, incorrect 3D-2D correspondences are unavoidable, especially in more complicated cases of occlusions, large inter-frame motions and background clutters. On the other hand, the search extent is difficult to be determined, large extent may result in more incorrect correspondences, while small extent will lead to sensitivity with large inter-frame motions.

In this paper, we propose an edge-based 3D tracking approach without explicit 3D-2D correspondences. We formulate the 3D object tracking as a contour matching problem by fitting the 3D object contour to the image edge distance field. The distance between the predicted object contour and the query image edge in distance field is minimized by direct optimisation of the 3D pose parameters. The differentials of this energy with respect to the pose parameters are derived, and the pose parameters are optimized iteratively with the Levenberg-Marquardt (L-M) algorithm. The image edges are extracted by edge detector, without requirements to do object segmentation and edge filtering. Cluttered backgrounds can be handled because of the holistic matching energy function. For better tracking performance, a particle filtering process
with a first order autoregressive state dynamics is exploited to deal with potential large inter-frame motions, and a robust estimator is adopted to handle occlusions. Comparative experiments demonstrate that the proposed method is effective on real image sequences with occlusions, large motions and cluttered backgrounds.

2 RELATED WORK

The literature of 3D object tracking is particularly massive. Given a 3D model of the target, 3D-2D correspondences between 3D features of the model and 2D measurements in the image are exploited for 3D tracking. According to the type of 2D measurements, 3D-2D correspondences based methods are classified as fiducial-based [10], keypoint-based [13, 17, 23] and edge-based [5, 7, 11, 15, 25]. We refer the reader to [12, 16] for more details. Here we restrict ourselves to monocular edge-based methods with a 3D model of the target available.

As the vital visual cue of textureless objects, edges or contours are employed by edge-based trackers. To construct 3D-2D correspondences, [5, 15] adopted a precomputed convolution kernel function of the contour orientation to find the image edge point only with an orientation similar to the projected contour orientation, not the edge point with maximum gradient in the scanline. [25] proposed multiple edge hypotheses that it attributed all the local extrema of the gradient along the scanline as potential correspondences. Multiple hypotheses prevent a wrong gradient maximum from being assigned as a correspondence, but increase the computation cost. [20, 24] exploited the local region knowledge of foreground and background, and the affinity of adjacent image edge points to search the optimal correspondences. These improvements are impressive, however the robustness decreases when ambiguities between different edges occur in the scene. All of these methods assume that edge correspondences are determined by local search in a limit extent based on a prior pose. If the prior pose is sufficiently incorrect, tracking probably fails especially when the object moves fast.

To ensure a good prior pose, multiple pose hypotheses are proposed to propagate the pose using particle filters. Since Isard et al. [9] applied particle filters to 2D edge tracking, various edge-based 3D tracking methods have been implemented using a particle filter framework. [11] tracked complex 3D objects by utilizing the GPU to calculate edges and evaluate pose likelihoods. [4] employed keypoint correspondences for particle initialization, and then refined the estimated pose by aligning the projected model edges and the image edges using 3D-2D correspondences explicitly. Our approach is similar to [3] by both using the edge distance field. [3] is a tracking-by-detection framework that uses distance field for chamfer matching between offline 2D edge templates and the scene image, and the coarse pose of the matched template is used for initializing particles. It employed a standard edge-based tracking to establish the correspondences and predict the final pose, while our approach directly optimizes the pose in distance filed. These methods can achieve impressive results especially for large inter-frame motions.

3 POSE PARAMETERIZATION

3D tracking aims to estimate the 6DoF pose of an object relative to the camera given the camera intrinsic matrix \( K \in \mathbb{R}^{3 \times 3} \), the image \( I \), the 3D model \( M \). A 3D model point \( X \in \mathbb{R}^3 \) is projected to an image pixel \( x \in \mathbb{R}^2 \) using the standard pinhole camera model by

\[
\bar{x} = K \cdot [R(r)|t] \cdot \bar{X}
\]

where \( t \) and \( R(r) \) are respectively the translation vector and rotation matrix parameterized by the Rodrigues rotation vector \( r \). \( \bar{x} \) and \( \bar{X} \) are respectively the homogenous representation of \( x \) and \( X \). The 6DoF pose is parameterized by \( p = (r, t) \) in this paper.

4 3D TRACKING IN EDGE DISTANCE FIELD

This section describes our 3D tracking approach in edge distance field in detail. We begin with an energy function that defines the contour matching in distance field. Then in order to overcome the local minima in 3D pose optimization, we introduce a particle filtering process with a first order autoregressive state dynamics. Finally, the optimization can be made more immune to occlusions by employing a robust estimator.

4.1 Pose optimization as 3D contour matching

We formulate the pose optimization as a contour matching process by fitting the 3D contour points \( \Phi \) to the edge distance field \( D \) of the image \( I \).

To generate \( D \), we use canny edge detector [2] to extract the edge map, then apply a fast distance transform [6] to the edge map. For each image pixel \( x \), \( D(x) \) indicates the distance to its nearest image edge point. Figure 1(a), 1(b), 1(c) illustrate the procedure of generating edge distance field.

Given a prior pose \( p_i \) and the 3D model \( M \), we can render the depth map and extract the 2D contour points on it. Then \( \Phi \) can be easily obtained by back-projecting the 2D contour points to the the 3D model \( M \).

For a 3D contour point \( X_i \in \Phi \), the matching cost \( e_i \) is denoted as follows:

\[
e_i = D(\pi(K \cdot [R(r)|t]) \cdot \bar{X}_i))
\]
where $\pi$ transforms the homogenous coordinates into its non-homogenous representation. Therefore, the whole matching cost $E$ between $\Phi$ and $D$ is defined by following objective energy function:

$$E(r, t) = \sum_{X_i \in \Phi} e_i.$$  

Starting from $p_0$, the optimal pose $p_o$ is calculated by iteratively minimizing Equation 3 using L-M algorithm:

$$p_o = \arg \min_{p} E(r, t).$$  

Figure 1(c) shows the evolution of projected 3D contour points from $p_t$ to $p_o$ by Equation 4. Figure 1(d) shows the optimal pose $p_o$ by overlying green wireframe on the target.

We can differentiate Equation 3 with respect to the pose $R$, $T$, $v$, and $\omega$ to get the Jacobian required by L-M:

$$J = \sum_{i} \frac{\partial e_i}{\partial x_i} \cdot \frac{\partial x_i}{\partial p}.$$  

where $x_i$ is the projection of $X_i$. The differential $\frac{\partial x_i}{\partial x_i} \in \mathbb{R}^{1 \times 2}$ can be computed using centered finite difference in $D$, and $\frac{\partial x_i}{\partial p} \in \mathbb{R}^{2 \times 6}$ can be derived from Equation 1 analytically.

### 4.2 Particle filtering

Generally 3D tracking starts from a prior pose $p_t$ at frame $t$. Many edge-based tracking methods [5, 15, 20, 24] initialize $p_t$ using the estimated pose $p_{t-1}$ of frame $t-1$ under small inter-frame motion assumption. If large inter-frame motions occur, these methods with single pose hypothesis fail inevitably as the initial pose is not close to the global minimum. In this paper we exploit a particle filtering framework with a first order autoregressive dynamical model to deal with large inter-frame motions. Figure 2(a), 2(b) give 2 consecutive frames with a relative large inter-frame motion. Figure 2(c) shows the optimized pose with our particle filtering method in contrast to the estimated pose only using a single prior pose from previous frame as in Figure 2(d).

In our particle filtering framework, the posterior distribution (denoted $+$) at $t-1$ is represented as a set of $N$ particles $S_{t-1}^+$ associated with normalized weights $\Theta_{t-1}^+$ by

$$\{ S_{t-1}^+ = \{ p_{t-1}^{(0)}, \ldots, p_{t-1}^{(N-1)} \}, \Theta_{t-1}^+ = \{ \theta_{t-1}^{(0)}, \ldots, \theta_{t-1}^{(N-1)} \} \}$$

where the particle $p_{t-1}^{(k)}$ is the $k$th sample in the 6DoF pose space with an associated weight $\theta_{t-1}^{(k)}$. For the next frame $t$, particles $S_t^-$ are resampled according to the weights $\Theta_{t-1}^+$ and transited by a motion model to form the prior distribution (denoted $-$) of frame $t$:

$$\{ S_t^- = \{ p_{t-1}^{(0)}, \ldots, p_{t-1}^{(N-1)} \}, \Theta_t = \{ 1/N, \ldots, 1/N \} \}$$

where $\Theta_t$ indicates each particle with an uniform weight. $S_t^-$ is updated to $S_t^+$ using 3D contour matching as described in section 4.1, and $\Theta_t^+$ is evaluated according to the contour matching cost.

For each particle $p_{t}^{(k)} \in S_t^-$ at frame $t$, the transition is processed as:

$$p_{t}^{(k)} = p_{t-1}^{(k)} + \lambda_v v_{t-1}^{(k)} + \lambda_n n_{t-1}^{(k)}, \quad v_{t-1}^{(k)} = p_{t-1}^{(k)} - p_{t-2}^{(k)}$$

where $v_{t-1}^{(k)}$ denotes the velocity of the $k$th particle between $p_{t-1}^{(k)} \in S_{t-1}^+$ and $p_{t-2}^{(k)} \in S_{t-2}^+$, $n_{t}^{(k)} \in \mathbb{R}^6$ is a Gaussian noise from $\mathcal{N}(0, \Sigma)$ with a zero mean and a covariance $\Sigma \in \mathbb{R}^{6 \times 6}$. $\lambda_v$ and $\lambda_n$ are the weights for balancing the autoregressive motion and random motion respectively.

Once each particle is transited, it is employed in Equation 3 as the initial pose. The updating from $p_{t}^{(k)} \in S_t^-$ to $p_{t}^{(k)} \in S_t^+$ is accomplished by optimizing the Equation 3. Figure 2(e), 2(f) illustrate the particles updated from $S_t^-$ to $S_t^+$, and Figure 2(g) shows the evolution of the best particle in contrast to the evolution using a single prior pose as given in Figure 2(h). We denote the matching cost $e_{t}^{(k)}$ of $p_{t}^{(k)}$ as the residual after optimization, then the corresponding weight $\theta_t^{(k)}$ of $p_{t}^{(k)} \in S_t^+$ is evaluated using the residual $e_{t}^{(k)}$ as follows:

$$\theta_t^{(k)} = \exp(-\frac{e_{t}^{(k)}}{\lambda_e})$$

where the positive $\lambda_e$ is a control parameter for scaling the residual. After updating all the particles, the weight $\theta_t^{(k)} \in \Theta_t^+$ of each particle $p_{t}^{(k)} \in S_t^+$ is normalized by

$$\theta_t^{(k)} = \frac{\theta_t^{(k)}}{\sum_{i=1}^{N} \theta_t^{(i)}}.$$  

We consider the particle $p_{t}^{(k)} \in S_t^+$ with the highest weight as the optimal pose at frame $t$. 

Figure 2: Particle filtering between 2 consecutive frames of the target (a CAT). We take $N$ as 10 for example. (a) $I_{t-1}$ with pose $p_{t-1}$. (b) $I_t$ with a relative large motion from $I_{t-1}$. (c) The estimated optimal pose using our method is visualized by green wireframe overlaid on the target. (d) The estimated pose with $p_{t-1}$ as the single prior pose is visualized, obviously it converged to a local minimum. (e) The prior particles $S_{t-1}^-$ sampled from $S_{t-1}^+$ are visualized by red projected contours. (f) The updated particles $S_t^+$ by Equation 4 are visualized by green projected contours. (g) Projected 3D contour of the best particle is evolved using our method from a prior pose (in red) to the optimal pose (in green), it got stuck in a local minimum.
When updating is done, we obtain the posterior distribution at frame $t$, and it is used to generate the prior distribution at next frame $t + 1$ by importance resampling. Each particle $p_{i}^{(k)}$ in the prior particles $S_{t+1}$ are randomly drawn from $S_{t}^{k}$ according to the weights $\Theta_{i}^{k}$. After resampling is done, we can start the next particle filtering process.

4.3 Occlusion handling

We assume that occluded 3D contour points tend to have a large distance to the nearest image edge. A simple quadratic error in Equation 3 is sensitive to occluded contour points. Therefore, an alternative weight function $w$ can be incorporated by generalizing Equation 3:

$$E(r, t) = \sum_{i} w(e_{i})e_{i}^{2}.$$  \hspace{1cm} (11)

We can still apply the L-M algorithm to solve the iterated re-weighted least-squares (IRLS). In order to suppress the occluded 3D contour points strongly by assigning them zero weights, in this paper we chose the Tukey estimator:

$$w(c) = \begin{cases} 0 & \text{if } |e| < c \\ 1 - \left( \frac{c}{|e|} \right)^{2} & \text{otherwise} \end{cases}$$ \hspace{1cm} (12)

where $c$ is the maximum valid distance for a projected 3D contour point to the nearest image edge.

4.4 Implementation details

This section elaborates the complete framework of our approach and the details of parameter settings. Given an image sequence $I$, the camera intrinsic matrix $K$ and an initial pose $p_{0}$, our 3D tracking approach estimates the pose $p_{t}$ at each frame $t$ with the 3D model $M$. The framework of our approach is summarized into Algorithm 1. To minimize Equation 11, the L-M algorithm is employed and terminated until an maximum iteration steps(100). The number of particles $N$ is set as a different value from $\{1, 10, 100\}$ so as to evaluate the efficiency and effectiveness of particle filtering. The covariance matrix $\Sigma \in \mathbb{R}^{6 \times 6}$ in Equation 8 is diagonal, and $\text{diag}(\Sigma) = (0.1, 0.1, 0.1, 0.1, 0.1, 0.1)$. In Equation 8, $\lambda_{c}$ is 0.1, and $\lambda_{v}$ in Equation 9 is set as the number of 3D contour points(i.e. $|\Phi|$) to normalize the matching residual. The threshold $c$ of Equation 12 is set as 20.

5 EXPERIMENTS

We validate our method using different comparative experiments. Firstly, we demonstrate the effectiveness of particle filtering and occlusion handling. Then we compare our method with a pixel-wise tracker PWP3D [18] and a state-of-the-art tracker GOS [24] with a single pose hypothesis. Finally, we adopt marker-based tracker [10] as the baseline, and compare our method with it to accomplish quantitative evaluations.

Our system is implemented in C++, and runs on an Intel i5 CPU with 8GB RAM. The test sequences are captured by a camera with $640 \times 480$ resolution, and both the object and camera are movable.

Many edge-based methods assume that the motion of camera or object is small and smooth, thus the prior pose is close to the global minimum. If large inter-frame motions occur, the tracking fails and the estimated pose converges to a local minimum. In order to deal with large inter-frame motions, we respectively employ 1, 10, 100 particles to track the object under fast camera or object motion. Figure 3 gives the comparison results of 1, 10, 100 particles in case of fast camera motion, we use 100 particles to track the BUNNY even with a bit motion blur.

Most edge-based methods construct 3D-2D edge correspondences explicitly. When the target object is occluded, they find wrong edge correspondences, or even cannot find edge correspondences. As we described in section 4.3, we employ the Tukey estimator to suppress the importance of occluded contour points. Our method can work if the occlusion is not
very severe. Figure 4 shows that the DUCK is tracked successfully using 10 particles with occlusion handling when it is occluded by a card or a hand.

We compare our method with a pixel-wised tracker PWP3D [18]. PWP3D proposed a probabilistic framework for simultaneous 2D image segmentation and 3D object tracking without building 3D-2D correspondences explicitly. It employed the statistic color information of foreground and background based on a prior pose, thus the tracking drifts when the target object has similar color statistics with the environment. Figure 5 compares the tracking results obtained by PWP3D (the second row) with our method \( (N = 100, \) the first row). PWP3D drifted due to the fast camera motion and white background.

As a representative edge-based tracker, GOS [24] constructs 3D-2D edge correspondences explicitly. The image edge correspondences are determined by a 1D local search with a limited extent based on a prior pose. Although GOS exploits the region knowledge around the edge point and the affinity of adjacent image edge points, it still suffers from the error correspondences raised from the similar edge of background. Figure 5 compares the tracking results between GOS (the third row) and our method \( (N = 100, \) the first row). For the GOS tracker, the edge correspondences of white BUNNY are disturbed by the white edges from background.

In order to evaluate the tracking accuracy and time performance of our method, we adopt the marker-based tracking method [10] as the baseline. The coordinate system of the object is predefined and fixed relative to the marker, thus the ground truth pose of the object can be transformed from the marker. We captured 4 sequences respectively for the BUNNY, CAT, DUCK and LEGO with a hand-hold camera as Figure 6 shows. Table 1 gives the accuracy and time performance of our method, we use two criteria to evaluate the accuracy: rotation error, \( T \) for translation error, and \( AD \) for average distance.

\[
\begin{array}{c|c|c|c|c|c|c}
\hline
\text{Seq[#]} & \text{Method} & \text{Time (ms)} & \text{Accuracy} \\
\hline
\text{BUNNY} & \text{PWP3D} & 79.8 & 7.9 & 10.2 & 11.3 \\
\text{GOS} & 158.8 & 3.4 & 13.0 & 10.9 \\
\text{Our method} (N = 1) & \textcolor{red}{16.0} & 8.0 & 1.9 & 2.0 \\
\text{Our method} (N = 10) & 90.7 & 2.9 & 1.6 & 1.5 \\
\text{Our method} (N = 100) & 934.8 & \textcolor{red}{2.4} & 1.6 & 1.3 \\
\hline
\text{CAT} & \text{PWP3D} & 90.0 & 208.8 & 6.4 & 13.9 \\
\text{GOS} & 194.7 & 7.8 & 4.8 & 1.3 \\
\text{Our method} (N = 1) & \textcolor{red}{16.7} & 3.6 & 1.9 & 1.9 \\
\text{Our method} (N = 10) & 110.6 & 1.5 & 2.0 & 2.0 \\
\text{Our method} (N = 100) & 1015.8 & \textcolor{red}{1.3} & 1.9 & 1.9 \\
\hline
\text{DUCK} & \text{PWP3D} & 7.5 & 189.2 & 7.8 & 9.1 \\
\text{GOS} & 149.8 & 158.8 & 6.2 & 3.5 \\
\text{Our method} (N = 1) & \textcolor{red}{15.9} & 6.5 & 1.9 & 2.0 \\
\text{Our method} (N = 10) & 104.5 & 6.5 & \textcolor{red}{1.8} & 1.9 \\
\text{Our method} (N = 100) & 960.9 & \textcolor{red}{6.2} & 1.8 & 2.0 \\
\hline
\text{LEGO} & \text{PWP3D} & 52.3 & 235.8 & 10.9 & 22.6 \\
\text{GOS} & 170.8 & 169.6 & 5.3 & 3.3 \\
\text{Our method} (N = 1) & \textcolor{red}{15.9} & 9.3 & 1.5 & 1.9 \\
\text{Our method} (N = 10) & 87.4 & 4.6 & 1.5 & 1.7 \\
\text{Our method} (N = 100) & 830.7 & \textcolor{red}{4.4} & 1.3 & 1.7 \\
\hline
\text{AVG} & \text{PWP3D} & 73.9 & 208.9 & 10.2 & 12.9 \\
\text{GOS} & 148.7 & 243.9 & 2.9 & 2.3 \\
\text{Our method} (N = 1) & \textcolor{red}{235.8} & 6.9 & 1.8 & 2.0 \\
\text{Our method} (N = 10) & 98.3 & 3.9 & \textcolor{red}{1.7} & 1.8 \\
\text{Our method} (N = 100) & 936.5 & \textcolor{red}{3.6} & \textcolor{red}{1.7} & \textcolor{red}{1.7} \\
\hline
\end{array}
\]

As Table 1 shows, the accuracy and time performance of our method are better than that of the other methods. Our method achieves a lower rotation error, a lower translation error, and a smaller average distance.

6 LIMITATIONS AND FUTURE WORK

Our method directly optimizes the pose parameters in edge distance field, thus it depends on the edge map of the query image. When the color of background is similar to the object, or severe motion blur occurs in the scene, we cannot get adequate contours of the target. Our method will fail because it merely exploits the contour information. In future work, we will consider the inner pixel information of the object to enhance the tracking robustness.

For symmetrical objects, multiple poses may result into the same contour, which is ambiguous for our method to estimate the pose correctly. Moreover, the computation cost...
is also a critical problem for fast object tracking using massive particles. We will speed up the particle filtering using GPU technique.

7 CONCLUSIONS

This paper proposes a monocular model-based 3D tracking approach for textureless objects. We minimize the holistic distance between the predicted object contour and the query image edge in distance field via direct optimisation of the 3D pose parameters. We derive the differentials of this energy with respect to the pose parameters, and search the optimal pose parameters using L-M algorithm. We employ a particle filtering framework to avoid being trapped in local minima. Occlusions are handled by a robust estimator. We demonstrated the effectiveness of our method using comparative experiments on real image sequences with occlusions, large motions and cluttered backgrounds.

ACKNOWLEDGMENTS

The authors gratefully acknowledge the anonymous reviewers for their comments to help us to improve our paper. This work is supported by the National Key Research and Development Program of China (No. 2016YFB1001501).

REFERENCES


