Vis Comput (2015) 31:979–988
DOI 10.1007/s00371-015-1098-7

ORIGINAL ARTICLE

Global optimal searching for textureless 3D object tracking

Guofeng Wang[1] · Bin Wang[1] · Fan Zhong[1] · Xueying Qin[1] · Baoquan Chen[1]

Published online: 7 May 2015

Abstract Textureless 3D object tracking of the object's position and orientation is a considerably challenging problem, for which a 3D model is commonly used. The 3D–2D correspondence between a known 3D object model and 2D scene edges in an image is standardly used to locate the 3D object, one of the most important problems in model-based 3D object tracking. State-of-the-art methods solve this problem by searching correspondences independently. However, this often fails in highly cluttered backgrounds, owing to the presence of numerous local minima. To overcome this problem, we propose a new method based on global optimization for searching these correspondences. With our search mechanism, a graph model based on an energy function is used to establish the relationship of the candidate correspondences. Then, the optimal correspondences can be efficiently searched with dynamic programming. Qualitative and quantitative experimental results demonstrate that the proposed method performs favorably compared to the state-of-the-art methods in highly cluttered backgrounds.

Keywords 3D tracking · 3D–2D correspondence · Global optimization · Dynamic programming

✉ Xueying Qin
qxy@sdu.edu.cn

Guofeng Wang
wangguofeng525@gmail.com

Bin Wang
binwangsdu@gmail.com

Fan Zhong
zhongfan@sdu.edu.cn

Baoquan Chen
baoquan.chen@gmail.com

[1] School of Computer Science and Technology, Shandong University, Jinan, China

1 Introduction

3D object tracking is a fundamental computer vision problem and has been widely used in augmented reality, visual servoing, etc. The aim of 3D object tracking is to estimate camera poses (i.e., positions and orientations) with six degrees of freedom (6DOF) relative to the rigid object [13].

Many algorithms have been developed in this area, including feature-based tracking [7,11,22]. However, for monocular-based textureless 3D object tracking, little information can be used, owing the lack of texture. In most situations, the edge information is the only cue that can be used. Therefore, model-based tracking has been widely used for textureless 3D object tracking [13]. With a given 3D object model—which can be obtained with Kinect Fusion [9] or 3D Max, for instance—the camera poses can be estimated by projecting the 3D object model onto an image plane. It is then matched with its corresponding 2D scene edges in the image. In this paper, we focus on the monocular camera, which is challenging when the object is textureless.

Rapid Tracker [6] is the first model-based tracker and is based exclusively on edge information. However, additional methods have since become well-established and have steadily been improving [3,5,20,23]. Though edge-based tracking is fast and plausible, errors are common in images with cluttered backgrounds, as shown in Fig. 1. In this paper, we explore a critical problem with monocular cameras regarding textureless objects in heavily cluttered background environments. Several methods have been developed to manage this situation with either multiple-edge hypotheses [21,23] or multiple-pose hypotheses [2,12]. Recently, a new method [20] based on region knowledge was adopted, and it is more robust than previous methods. However, its correspondences are searched independently, and drift is a

Fig. 1 Critical problem with edge-based tracking in a highly cluttered background. *Left* 3D object model projected on a target object (*red line*) at frame t with a previous camera pose. *Right* edge image of a target object in a highly cluttered background that may cause false matches (local minima)

problem when the object has a complex structure in a highly cluttered background.

In this paper, we propose a new correspondence method based on global optimization, in which all of the correspondences are searched interdependently by employing contour coherence. This way, the optimal correspondences can be searched efficiently with fewer errors. Like [20], our method uses region knowledge to infer these correspondences, but unlike [20], we build a graph model to describe the relationship between these correspondences, rather than searching them independently. In our searching scheme, candidates for correspondences are first evaluated by gradient response and non-maximum suppression along the normal lines. Then, a graph model with source and terminate nodes based on an energy function is adopted to establish the relationship between all these candidate correspondences. Finally, dynamic programming is adopted to search the optimal correspondences efficiently. Experimental results demonstrate that the proposed method is efficient in highly cluttered backgrounds with arbitrary complex models.

2 Related work

Several different algorithms have been developed for 3D object tracking, whether exploiting a single visual cue [5,7,20], multiple visual cues [18,21], or additional sensors [14]. We refer the reader to [13] for more details. In this section, we shall discuss only the research pertinent to our proposal.

In monocular camera-based tracking, the algorithm first finds the correspondences between 3D object model points and 2D image points. Then, the algorithm estimates the 3D pose of the object with these correspondences. Drummond et al. [5] proposed a very simple strategy for searching the nearest correspondence points by the intensity discontinuity above a certain threshold in the normal direction of the sample points. Marchand et al. [15] used pre-computed filter masks to extract the corresponding points in the normal direction of the contour. However, these searching strategies are less effective in heavily cluttered backgrounds and easily trapped in local minima, despite the use of robust estimators. To avoid the local minima, Vacchetti et al. [21] and Wuest et al. [23] proposed using multiple-edge hypotheses rather than single-edge hypotheses for searching the correspondences, such that more than one correspondence point is used for one sample point when computing the pose. This improves the robustness of the algorithm. However, these approaches face difficulties when outliers are close to the correct correspondence, and they require considerable processing time.

Multiple-pose hypothesis methods [2,12] have also been proposed to improve the robustness of the algorithm, and a particle-filter strategy is usually adopted by these methods. Such methods are effective at avoiding undesirable errors resulting from background clutter. However, the computational cost is usually too high for real-time object tracking, because larger state spaces are needed for more complex scenes.

Whereas many methods merely use the edge information of the object model, compromising them in heavily cluttered backgrounds, other methods use region knowledge to improve the robustness of the algorithm. These methods are based on level-set region segmentation and are robust for 3D object tracking [4,16,17,19]. Typically, such approaches formulate a unique energy functional framework for simultaneous 2D segmentation and 3D object tracking. The 2D segmentation results can be used for pose estimation, and the 3D object model, in turn, improves the robustness of 2D segmentation. However, such region-based methods are dependent on the segmentation results. Consequently, it is difficult to derive the accurate pose, because segmentation cannot guarantee precise results in a cluttered environment. As an alternative to region segmentation, Seo et al. [20] exploited local region knowledge for reliably establishing 3D–2D correspondences with an edge-based approach, and their method is much more robust than previous methods. However, its correspondences are searched independently, meaning that drift is a problem in heavily cluttered backgrounds with complex object models, owing to the local minima.

3 Problem statement and notation

For monocular model-based 3D object tracking, when provided with a 3D model $\mathbf{M}$, the aim is to estimate the camera pose $\mathbf{E}_t = \mathbf{E}_{t-1}\Delta_{t-1}^t$ in the current frame t when provided with pose $\mathbf{E}_{t-1}$ from the previous frame $t-1$. The infinitesimal camera motions Δ_{t-1}^t can be computed by minimizing the errors between the 3D model projected with the previous camera pose and its corresponding 2D scene edges m_i in the image, such that

$$\hat{\Delta}_{t-1}^{t} = \arg\min_{\Delta_{t-1}^{t}} \sum_{i=1}^{N} \|m_i - \mathrm{Proj}(\mathbf{M}; \mathbf{E}_{t-1}, \Delta_{t-1}^{t}, \mathbf{K})_i\|^2,$$
$$= \arg\min_{\Delta_{t-1}^{t}} \sum_{i=1}^{N} \|m_i - \mathbf{K}\mathbf{E}_{t-1}\Delta_{t-1}^{t} \cdot \mathbf{M}_i\|^2,$$
$$= \arg\min_{\Delta_{t-1}^{t}} \sum_{i=1}^{N} \|m_i - s_i\|^2, \quad (1)$$

where $\mathbf{K}$ is the camera intrinsic parameter, assumed to be known a priori, $\mathbf{E}$ is the camera extrinsic parameter, $\mathbf{M}_i$ denotes a sample point from the 3D model $\mathbf{M}$, s_i is a sample point from projected 3D model in the image, and N is the number of s_i.

With this minimization scheme, we can derive the accurate pose of the camera in the current frame. The key problem with this minimization scheme is that given the 3D sample points $\mathbf{M}_i$, the corresponding 2D sample points m_i must be accurately determined in the image. This is difficult when the object is in a highly cluttered background with complex structure. In the next section, we explain how this correspondence problem can be solved efficiently with global optimization.

Like [20], we do not consider all of the data from the 3D object model; rather, we merely use the visible model contour data, because the model data are usually complex and its valuable interior data are especially difficult to extract.

4 Correspondence with global optimal searching

In this section, we describe our global optimal searching algorithm for the correspondences between 3D object model points and 2D image points.

4.1 Overview

To search the optimal correspondences in the image, we first discover the candidate correspondences $\mathbf{c}_i$ for each correct correspondence m_i. Each $\mathbf{c}_i$ may include some candidates, denoted by $c_{i1}, c_{i2}, \ldots, c_{ij}, \ldots$, where $\ldots$ indicates that the number of candidate correspondences $\mathbf{c}_i$ may be different when they belong to a different correspondence m_i, as shown in Fig. 2.

Let $\mathbf{A} = (\boldsymbol{\alpha}_1, \boldsymbol{\alpha}_2, \ldots, \boldsymbol{\alpha}_N)$, where $\boldsymbol{\alpha}_i = (\alpha_{i1}, \alpha_{i2}, \ldots, \alpha_{ij}, \ldots) = (0, 0, \ldots, 1, \ldots)$ indicating whether the candidate correspondence c_{ij} belongs to the correct correspondence m_i. Note that $\alpha_{ij} \in \{0, 1\}$ and $\sum_j \alpha_{ij} = 1$. Then, $\mathbf{A}$ can be obtained by minimizing the following energy function:

$$E(\mathbf{A}) = \sum_{i,j} E_d(\alpha_{ij}) + \lambda \sum_{i=1}^{N-1} \sum_{j,k} E_s(\alpha_{ij}, \alpha_{i+1,k}), \quad (2)$$

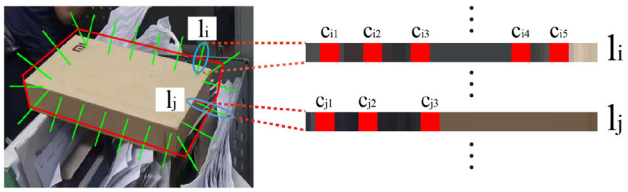


Fig. 2 Candidate correspondences. *Left* each *green line* denotes the 1-D scan line along the normal vector of the projecting point s_i. *Right* candidate correspondences $\mathbf{c}_i$ in each searching line (i.e., *red squares*). The number of candidates may be different in each searching line

where $E_d(\alpha_{ij})$ is the data term measuring the cost, under the assumption that the candidate correspondence c_{ij} belongs to the correct correspondence m_i. $E_s(\alpha_{ij}, \alpha_{i+1,k})$ is the smoothness term encoding prior knowledge about the candidate correspondences, and λ is a free parameter used as a trade-off between the data and smooth terms.

Typically, the data term $E_d(\alpha_{ij})$ can be computed as the negative log of the probability that c_{ij} belongs to candidate m_i, such that $E_d(\alpha_{ij}) = -\log\, p(c_{ij}|m_i)$; $p(c_{ij}|m_i)$ can be computed using a color histogram with nearby foreground and background information. The smooth term $E_s(\alpha_{ij}, \alpha_{i+1,k})$ is not dependent on the color histogram. It is only dependent on the consistency between the two candidates c_{ij} and $c_{i+1,k}$ and whether they are neighboring. If two candidates c_{ij} and $c_{i+1,k}$ belong to the correspondences m_i and m_{i+1}, respectively, they are neighboring, and the term $E_s(\alpha_{ij}, \alpha_{i+1,k})$ is small. Typically, $E_s(\alpha_{ij}, \alpha_{i+1,k})$ can be defined as $E_s(\alpha_{ij}, \alpha_{i+1,k}) = \alpha_{ij} \cdot \alpha_{i+1,k} \cdot \exp(\|x_{ij} - x_{i+1,k}\|^2/\sigma^2)$, where x_{ij} and $x_{i+1,k}$ are the locations of the candidate correspondences c_{ij} and $c_{i+1,k}$ in the image.

Equation 2 can be transformed into a graph model with source and terminate nodes that can be solved efficiently with dynamic programming. In the following subsections, we discuss these candidate correspondences (Sect. 4.2), we show how the probability $p(c_{ij}|m_i)$ can be computed (Sect. 4.3) and explain how to efficiently solve Eq. 2 (Sect. 4.4).

4.2 Candidate correspondence

To efficiently search the correspondences between a projected 3D object model and the 2D scene edges, the following must be calculated. At each sample point M_i, we search its corresponding 2D scene edge on a 1D scan line along the normal vector of its projecting point s_i. We assume that the correct correspondences m_i exist in the intensity change along the searching line of each sample point s_i. The 1D searching lines $\mathbf{l}_i$ are defined using Bresenham's line-drawing algorithm [1], which covers the interior, contour, and exterior of the object in the image. The candidates for correspondences $\mathbf{c}_i$ are computed by 1D convolution of a 1×3 filter mask ($[-1\ 0\ 1]$) and 1D non-maximum (3-neighbor)

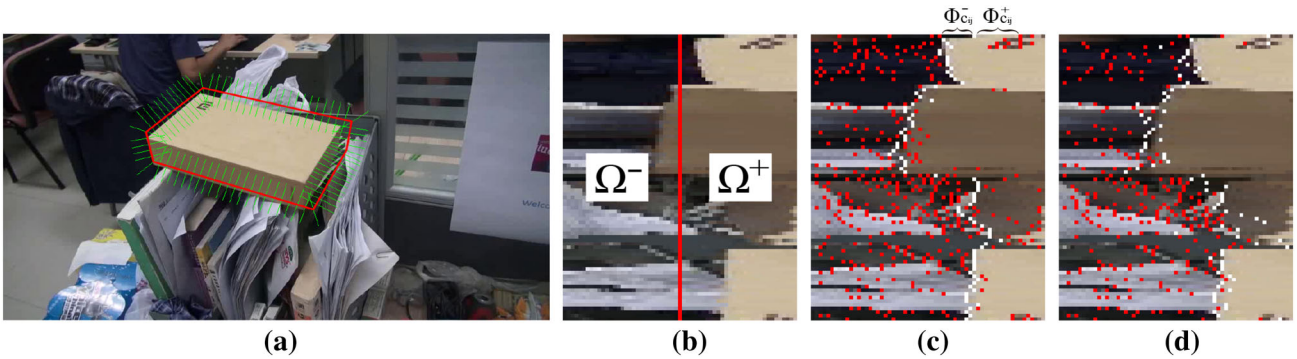


Fig. 3 Global optimal searching for correct correspondences m_i. **a** The search lines (*green lines*) in the image, where the *red line* is the contour of projected 3D object model in the previous camera pose $\mathbf{E}_{t-1}$. **b** The new image $\mathbf{L}$ built by stacking the searching lines, where Ω^- is the background regions and Ω^+ is the foreground regions. The size of new image $\mathbf{L}$ is 61×150, which is much smaller than the origin image (960×540). **c** The correct correspondences searched by our global optimal algorithm (the *white dots* denote the correct correspondences and the *red dots* denote the candidate correspondences $\mathbf{c}_i$). **d** The correct correspondences (*white dots*) searched by local optimal algorithm in [20]

suppression along the lines. To increase robustness, we separately compute the gradient responses for each color channel of our input image, and for each image location using the gradient response of the channel whose magnitude is largest, such that

$$c_{ij} = \underset{\mathcal{C}\in\{R,G,B\}}{\arg\max} \parallel \bigtriangledown I_{\mathcal{C}}(l_{ij})\parallel, \tag{3}$$

where $\mathcal{C}$ is the color channel belonging to one of the R, G, B channels, l_{ij} is the location in the searching line $\mathbf{l}_i$, and $I_{\mathcal{C}}(.)$ is the intensity in each channel. The candidate correspondences $\mathbf{c}_i$ can then be extracted by the maximum magnitude norm larger than a threshold. In our experiments, this threshold is set to 40, and the range of each search line $\mathbf{l}_i$ is set to 61 pixels (30 for the interior, 1 for the contour, and 30 for the exterior), as illustrated in Fig. 3.

4.3 Color probability model

Before describing the probability $p(c_{ij}|m_i)$ of each candidate c_{ij}—as [20] proceeds to do—we first introduce a new image $\mathbf{L}$, which is a set of searching lines $\mathbf{l}_i$. $\mathbf{L}$ can be built simply by stacking each searching line and arranging it symmetrically, where the left area of $\mathbf{L}$ indicates the exterior, the right area of $\mathbf{L}$ is the interior, and center of $\mathbf{L}$ is the contour of the 3D object projecting into the image, as shown in Fig. 3(b). The size of $\mathbf{L}$ is much smaller than the input image, where the resolution is the length of $\mathbf{l}_i$ multiplied by the number of s_i. Modeling the probability in this image is much faster and efficient than calculating the probability in the original input image, because we do not need to compute unnecessary information about the object in the input image.

After deriving the new image $\mathbf{L}$, we can build the foreground probability for the object based on the right side of the columns in the image $\mathbf{L}$ and the background probability of the object based on the left side. Therefore, the probability $p(c_{ij}|m_i)$ can be determined from the foreground probability and the background probability. Specifically, if the candidate c_{ij} belongs to the correct correspondence m_i, its left area belongs to the background of the object and right area belongs to the foreground of the object.

To model the probability of the foreground and background, we adopt a nonparametric density function based on hue-saturation-value (HSV) histograms. The HSV color space decouples the intensity from colors that are less sensitive to illumination changes. Thus, it is more suitable than the RGB color space. In our experiments, we take H and S components from the HSV color space. Of these, V components are sensitive to illumination changes. An HS histogram has N bins that are composed of bins from the H and S histogram ($N = N_H N_s$), and this is represented by the kernel density $H(\Omega) = \{h_n(\Omega)\}_{n=1,\ldots,N}$, where $h_n(\Omega)$ is the probability of a bin n within a region Ω. After defining the nonparametric density function $H(\Omega)$, we can compute the probability of the foreground and background. Supposing that $P^f(\Omega^+)$ and $P^b(\Omega^-)$ indicate the probability of the foreground and background, respectively, then $P^f(\Omega^+) = \{h_n(\Omega^+)\}_{n=1,\ldots,N}$ and $P^b(\Omega^-) = \{h_n(\Omega^-)\}_{n=1,\ldots,N}$. Here, Ω^+ is the right side and Ω^- is the left side of the columns in the image $\mathbf{L}$ in Fig. 3b.

To compute the probability of each candidate correspondence c_{ij}, we also define two relative probabilities $P^f(\Phi^+_{c_{ij}})$, $P^b(\Phi^-_{c_{ij}})$ for c_{ij}, where $\Phi^+_{c_{ij}}$ and $\Phi^-_{c_{ij}}$ are the foreground area and background area of candidate c_{ij}, respectively. Because the left-side regions of the candidate correspondences in the searching line l_i are probable, the background region and the right-side regions of the candidate correspondences in the searching line l_i may be the object regions. Thus, we define the foreground area of candidate c_{ij} in line l_i as the region from the jth candidate to the $(j+1)$th candidate, $c_{i,j} < \Phi^+_{c_{ij}} < c_{i,j+1}$. Likewise, we define the background

area of candidate c_{ij} in line l_i as the region from the jth candidate to the $(j-1)$th candidate, $c_{i,j-1} < \Phi^{-}_{c_{ij}} < c_{i,j}$, as illustrated in Fig. 3c.

After deriving these histograms, we measure their similarity by computing the Bhattacharyya similarity coefficient between two probabilities in an HS space, such that $\mathcal{D}[\Omega, \Phi] = \sum_{n=1}^{N} \sqrt{h_n(\Omega) h_n(\Phi)}$. Following this, we find the score for the foreground and background of each candidate c_{ij} as follows:

$$\begin{aligned} \mathcal{D}^{f}_{ij}[\Omega^{+}, \Phi^{+}] &= \sum_{n=1}^{N} \sqrt{h_n(\Omega^{+}) h_n(\Phi^{+}_{c_{ij}})}, \\ \mathcal{D}^{b}_{ij}[\Omega^{-}, \Phi^{-}] &= \sum_{n=1}^{N} \sqrt{h_n(\Omega^{-}) h_n(\Phi^{-}_{c_{ij}})}. \end{aligned} \tag{4}$$

If candidate c_{ij} belongs to the correct correspondence m_i, then its corresponding $\mathcal{D}^{f}_{ij}$ and $\mathcal{D}^{b}_{ij}$ are both large. If candidate c_{ij} does not belong to the correct correspondence m_i, then at least one corresponding of $\mathcal{D}^{f}_{ij}$, $\mathcal{D}^{b}_{ij}$ is small. According to this property, we can define the probability $p(c_{ij}|m_i)$ of candidate c_{ij} simply as $p(c_{ij}|m_i) = \frac{1}{\mathcal{Z}} \cdot \mathcal{D}^{f}_{ij} \cdot \mathcal{D}^{b}_{ij}$, where $\mathcal{Z}$ is the normalizing constant that ensures $\sum_j p(c_{ij}|m_i) = 1$.

In some situations, the interior of the object can be influenced by text or figures on the object's surface (so-called "object clutter") even though the target object has no or little texture. Thus, it is insufficient to merely use the object foreground or background to respectively measure $\mathcal{D}^{f}_{ij}$ and $\mathcal{D}^{b}_{ij}$ of candidate c_{ij}. In such situations, when computing the score $\mathcal{D}^{f}_{ij}$, we acquire not only the foreground histogram $P^{f}(\Omega^{+})$, but also the background histogram $P^{b}(\Omega^{-})$. This is because when the candidate c_{ij} turns out to be object clutter, the appearance of it can be relatively far from the background region even though it is not very close to the appearance of the object region. A similar property is found when computing the score $\mathcal{D}^{b}_{ij}$. Thus, like [20], we can evaluate the foreground and background scores in multiple phases as follows:

$$\begin{aligned} \mathcal{S}^{f}_{ij} &= \Gamma(\theta) \mathrm{D}^{f}_{ij}[\Omega^{+}, \Phi^{+}] + (1 - \Gamma(\theta))(1 - \mathrm{D}^{b}_{ij}[\Omega^{-}, \Phi^{+}]), \\ \mathcal{S}^{b}_{ij} &= \Gamma(\theta) \mathrm{D}^{b}_{ij}[\Omega^{-}, \Phi^{-}] + (1 - \Gamma(\theta))(1 - \mathrm{D}^{f}_{ij}[\Omega^{+}, \Phi^{-}]), \end{aligned} \tag{5}$$

where $\Gamma(\theta)$ is the phase function defined as $\Gamma(\theta) = 1$, if $\mathrm{D}^{f}_{ij}[\Omega^{+}, \Phi^{+}] > \tau$; otherwise, $\Gamma(\theta) = 0$. Finally the probability for candidate c_{ij} can be defined as $p(c_{ij}|m_i) = \frac{1}{\mathcal{Z}} \cdot \mathcal{S}^{f}_{ij} \cdot \mathcal{S}^{b}_{ij}$.

4.4 Graph model

We now transform Eq. 2 into a graph model to solve it efficiently with dynamic programming. Typically, a directed graph $\mathcal{G} = \langle \mathcal{V}, \mathcal{E} \rangle$ is defined as a set of nodes (with vertices $\mathcal{G}$) and a set of directed edges ($\mathcal{E}$) that connect these nodes. The graph related to Eq. 2 is shown in Fig. 4. Each node v in the graph denotes the candidate correspondence c_{ij}, and its value is the cost $E_d(\alpha_{ij})$. Each edge e in the graph connects node $c_{i,j}$ and node $c_{i+1,k}$, and assigns a non-negative weight $E_s(\alpha_{i,j}, \alpha_{i+1,k})$. There are also two special nodes, separately called source $\mathcal{S}$ and terminal $\mathcal{T}$. The source node connects to all the candidate correspondences $c_{1,j}$, and the terminal node connects to all the candidate correspondences $c_{N,j}$. The weight of the source and terminal connection edges is equal to 1 in both cases. By defining this weighted-directed graph model, Eq. 2 can then be seen as an equation for finding a minimal-cost path from the source node $\mathcal{S}$ to the terminal node $\mathcal{T}$.

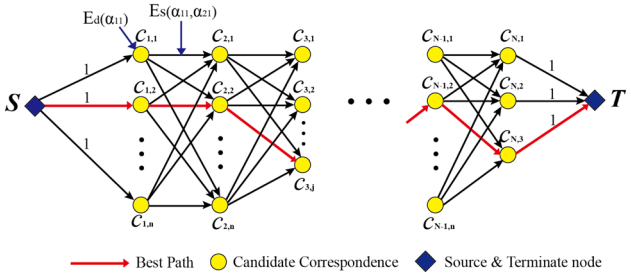


Fig. 4 Graph model: the nodes denote the candidate correspondences c_{ij}; the edges denote the connection between two near candidate correspondences c_{ij} and $c_{i+1,k}$

In the case of minimal-cost path, a heuristic method for solving this problem is to enumerate all the paths, to find the one with the minimal cost. However, this is unnecessary, because for each node, the path from its source only depends on its previous nodes. Suppose the minimal cost from source $\mathcal{S}$ to node c_{ij} is denoted as $\mathrm{cost}(c_{ij})$. Then, for node $c_{i+1,k}$, its minimal cost is $\mathrm{cost}(c_{i+1,k}) = \min\{\mathrm{cost}(c_{ij}) + E_d(\alpha_{i+1,k}) + E_s(\alpha_{ij}, \alpha_{i+1,k})\}$. Here, j is from $1, 2, \ldots$ the index of its previous nodes c_{ij}. The minimal cost for the first nodes c_{1j} is denoted as $\mathrm{cost}(c_{1j}) = 1 + E_d(\alpha_{1j})$, $j = 1, 2, \ldots$, and the minimal cost for the terminal node is $\mathrm{cost}(\mathcal{T}) = \min\{\mathrm{cost}(c_{Nj}) + 1\}$, $j = 1, 2, \ldots$. After finding the minimal-cost path (i.e., the red line in Fig. 4), we trace back the path to find all the correct correspondences m_i, (see the white dots in Fig. 3c.

4.5 Discussion

Compared with local optimal searching (LOS) [20], the global optimal searching (GOS) strategy has inherent advantages, because the correspondence points for the contour of the object are essentially continuous in the video. Thus, insofar as our global optimal searching strategy directly captures this inherent property, it is more effective for the complex 3D object models and highly cluttered backgrounds.

By connecting all the correspondence points m_i, these points construct the contour of the object in the image. From this point of view, the correspondence problem is similar to the segmentation problem, insofar as they both must discover the contour of the object in the image. However, unlike the segmentation problem, we do not need to segment out every pixel in the contour of the object. Rather, we merely need to discover the correspondence points. Moreover, the accuracy and speed are bottlenecked with the state-of-the-art segmentation methods. Consequently, they cannot be used for real-time 3D object tracking.

5 Experiments

To examine the effectiveness of the proposed method, we tested it on seven different challenge sequences and compared it with two state-of-the-art methods including LOS method [20] and PWP3D method [16]. We implemented the proposed method in C++, running the algorithm on a 3.2-GHz Intel i5-3470 CPU, with 8 GB of RAM, achieving approximately 15–20 fps with unoptimized code. The initial camera pose and camera calibration parameters were provided in advance. All of the 3D object models were represented by wireframes with vertexes and lines, and we visualized the results with the wireframes directly on the model in the video.

5.1 Implementation

To implement our proposed method, we used the Levenberg–Marquardt method to efficiently solve Eq. 1. Specifically, the 6DOF camera poses can be represented by x, y, z, rx, ry, rz, where x, y, z denote the position of the camera in relation to the object, and rx, ry, rz denote the Euler angles, representing rotations around the x, y and z axes. The framework of our 3D object tracking method can be found in Algorithm 1. In each frame, the iteration is terminated when the re-projection error is small (< 1.5 pixel) or when the number of iterations is more than a pre-defined number (> 10). Usually, only 4–5 iterations are required for each frame.

Algorithm 1 Tracking framework

Input: 3D object model $\mathbf{M}$, camera pose $\mathbf{E}_1$ in the first frame, and camera intrinsic parameters $\mathbf{K}$
Output: camera pose $\mathbf{E}_t$ in frame t

1: For each frame t = 1 : T, T is the total number of frames.
2: **repeat**
3: Get sample points $\mathbf{M}_i$ in 3D object model contour;
4: For each projected sample point s_i of $\mathbf{M}_i$, computing some candidate 2D image points $\mathbf{c}_i$ in its normal direction by Equation 3;
5: For each projected sample point s_i, computing its correct correspondence m_i from these candidates $\mathbf{c}_i$ by Equation 2 through Dynamic Programming;
6: Using Equation 1 to solve the 6DOF pose Δ_{t-1}^{t} by Levenberg-Marquardt method;
7. Updating the pose $\mathbf{E}_{t-1} = \mathbf{E}_{t-1}\Delta_{t-1}^{t}$;
8: **until** reprojection error $<$ min or iteration $>$ max
9: Get the current pose $\mathbf{E}_t = \mathbf{E}_{t-1}$.
10: end for

For the visible model contour data, to deal with arbitrary complex 3D object models, we first project the lines of the 3D object model into the image, before finding its 2D contour in the image. This 2D contour corresponds to the contour of the 3D object model. For the points from the contour of the 3D object model, its corresponding projected 2D points must exist in the contour of the image. In this way, we can filter out the lines of the arbitrary complex 3D object model leaving only the contour data. Then, the filtered 3D object model is regularly sampled, generating the 3D object model points $\mathbf{M}_i$. The number of sampled points N is dependent on the 3D object model, and we set $N = 150$ in all our experiments.

Furthermore, we set N_H and N_S to 4 in the HS histogram, and λ in Eq. 2 is set to be 1.

5.2 Quantitative comparison

For a quantitative evaluation, we compared the estimated 6DOF camera poses from the well-known ARToolKit [10] as the ground truth using the Cube model. The coordinates of the markers and the 3D object were registered in advance. We also compared the proposed method with the LOS method. As shown in Fig. 5, the trajectories estimated with the proposed method were comparable to the ones from the ARToolKit in all tests. With the LOS method, however, the trajectories began to drift at Frame 250, because it is easy for the LOS method to find the correspondence points at the inner edge of the model. More results can be found in the qualitative comparison, discussed in Sect. 5.3. The average angle difference with our algorithm was approximately 2°, and the average distance difference was approximately 2.5 mm.

5.3 Qualitative comparison

For a qualitative comparison, we tested the proposed method with seven different sequences with complex 3D object models in highly cluttered backgrounds.

We first compared our GOS method with LOS method by hollow-structured Cube model. The hollow-structured Cube model (10,000 faces) was challenging for the algorithm, owing to the inner hollow of the model, as shown in Fig. 6. Nevertheless, whereas LOS method easily finds the correspondence points at the inner hollow edges and drifts quickly in this sequence, our GOS method passed the entire sequence perfectly.

We then compared our GOS method with PWP3D method by Bunny model (5000 faces). As shown in Fig. 7, owing to

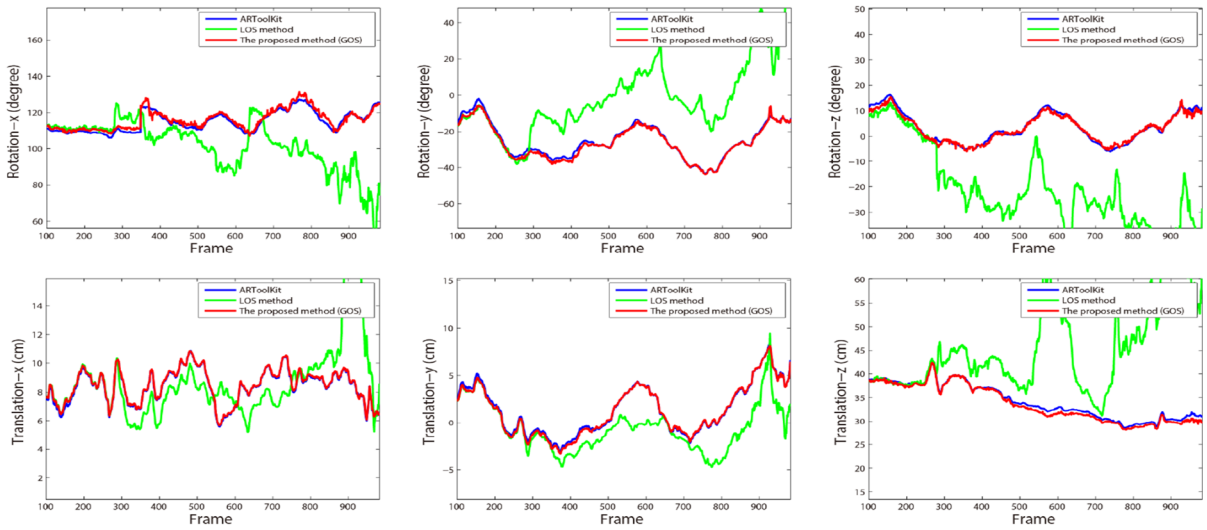


Fig. 5 Quantitative comparison of the proposed method, LOS method, and ARToolKit. The *red curves* denote our proposed GOS method; the *green curves* denote LOS method; and the *blue curves* denote the ground truth, which is captured by ARToolKit

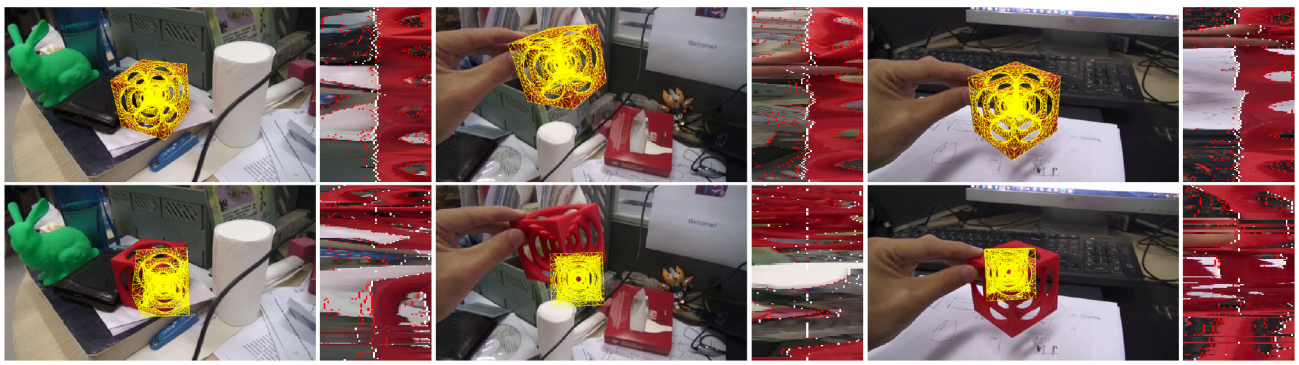


Fig. 6 Comparison of the Cube model between proposed GOS method and LOS method. *First row* result of our GOS method; *second row* result of LOS method. Second, fourth, and sixth columns are searching lines **L** where the white *dots* denote the correct correspondences m_i and *red dots* denote the candidate correspondences $\mathbf{c}_i$

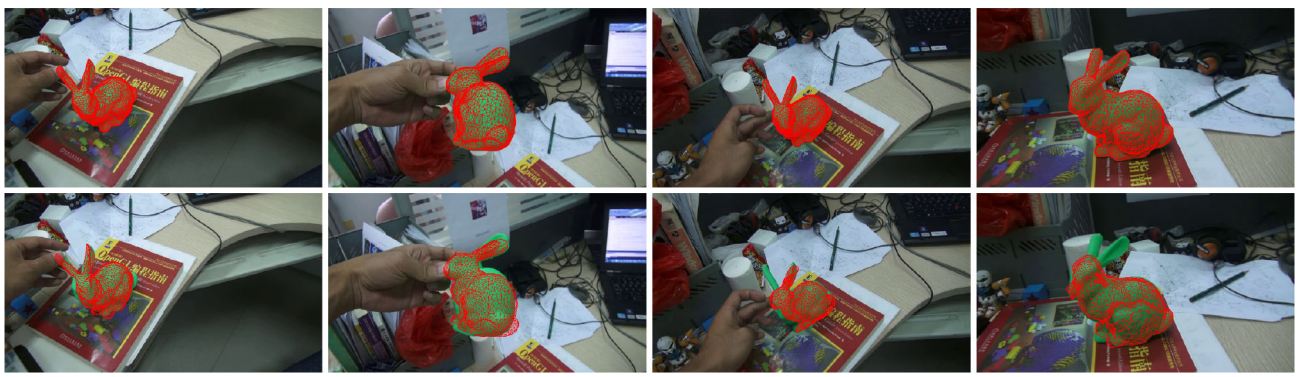


Fig. 7 Comparison of the Bunny model between proposed GOS method and PWP3D method. *First row* result of our GOS method; *second row* result of PWP3D method

the cluttered background, PWP3D method drifts in many frames, revealing a deficiency in the algorithm, whereas our GOS method can pass the entire sequence perfectly.

More experimental results are shown in Figs. 8 and 9. In Fig. 8, the Simple Box model, Shrine model and Simple Lego model are presented. For the Simple Box model,

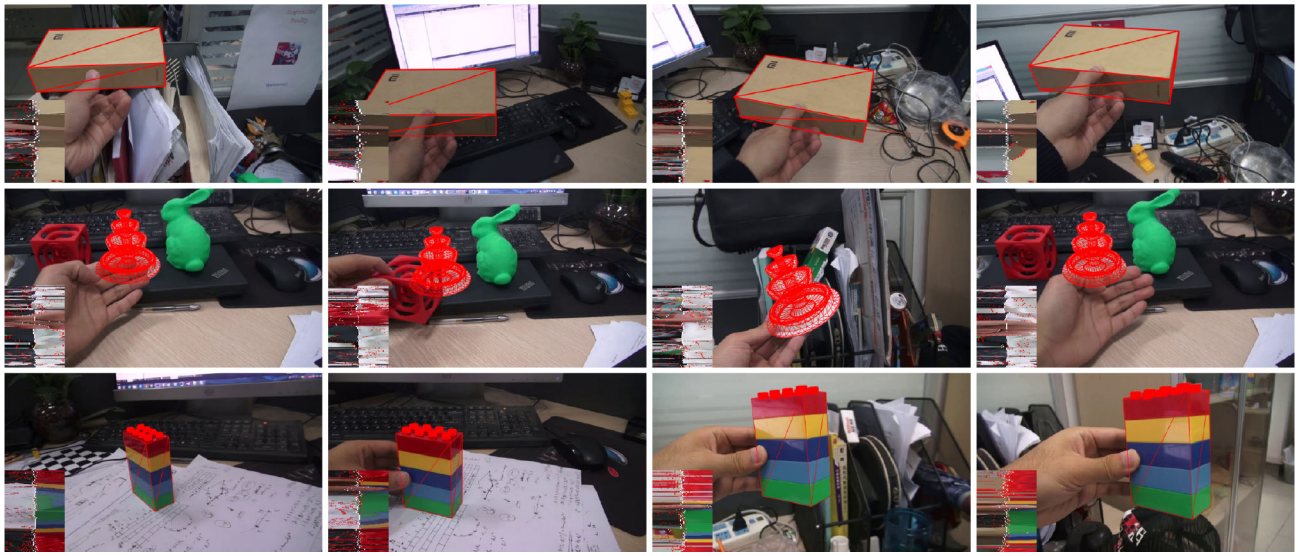


Fig. 8 Results of simple Box model, Shrine model and Simple Lego model, visualized by wireframes. *First row* denotes the result of Simple Box model; *second row* denotes the result of Shrine model; *third row* denotes the result of Simple Lego model

Fig. 9 Results of Vase model and Complex Lego model, visualized by wireframes. *First column* denotes the object in the scene and its scene edges. *Other columns* denote the results of Vase model (*first and second rows*) and Complex Lego model (*third and fourth rows*) by our GOS method. The *left down corner* of the image denotes the searching line

it is easy for the algorithm to drift, owing to the similarity in the appearance of the Simple Box model to the background and skin color of the hand. The Shrine model has a symmetrical structure, and this is ambiguous for the algorithm. However, the tracking performance of our method was not considerably degraded even with heavy occlusions. The Simple Lego model has multiple color; thus, our tracker can perform regardless of whether the object had a single color or multiple color.

In Fig. 9, the Vase and Complex Lego models are presented, with 10,000 and 25,000 faces, respectively. The structure of the Vase model is complex and has ambiguous

Fig. 10 The influence of different initial camera pose for the Simple Box model, where the *white lines* denote the initial camera poses, and the *red lines* denote the tracking results

aspects as well. When the user moves the 3D object model, it is easy to drift, owing to the complexity of the model and background. Nonetheless, our method performed well. The Complex Lego model is constructed with basic Lego parts, and the model has multiple colors, which leads to erroneous correspondence points, owing to the little color information for some parts of the model (e.g., the green baseplate of the model). However, because of the global constraint on the correspondence points, our method worked perfectly in this sequence.

Next, we examined the influence of the initial camera pose for the object tracking. We set different initial camera poses; however, the tracker can get the accurate results, as shown in Fig. 10, which means that the different initial camera poses do not influence much about the robust of the tracking.

5.4 Limitations

Though our method is robust in most situations, it still has some limitations. For example, our method only depends on the contour of the object, resulting in ambiguity to the pose when the object has a symmetrical structure. This presents a difficulty when we exclusively rely on contour information for tracking, because the different poses of the object may have the same 2D projected contours. In future work, we will consider the inner structure of the object, rather than merely the contour information. This may improve the performance of the tracking.

Furthermore, our method drifts when the object moves quickly and has a color similar to that of the background, or even in the heavy occlusion. However, with the state-of-the-art 3D object detection [8], we can easily reinitialize the method and repeat the process of tracking the object.

6 Conclusion

In this paper, we presented a new correspondence searching algorithm based on global optimization for textureless 3D object tracking in highly cluttered backgrounds. A graph model based on an energy function was adopted to formulate the problem, and dynamic programming was exploited to solve the problem efficiently. In directly capturing the inherent properties of the correspondence points, our proposed method is more robust than local optimal searching methods, which search the correspondence points independently. Moreover, our method is more suitable for situations with complex 3D object models in highly cluttered backgrounds.

In future work, we will consider exploring the inner structure to solve the ambiguity of the pose when the object has a symmetrical structure, and will also consider combining our proposal with a 3D object detection method to solve the initialization and re-initialization of the 3D object tracking. We believe that by combining our proposal with detection, the performance of the tracking will be further improved, and that we can demonstrate its feasibility for practical application.

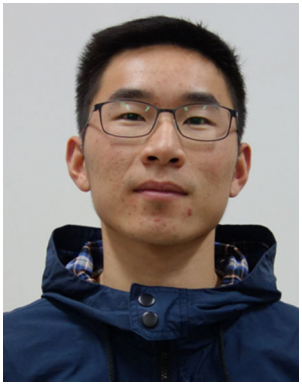
Acknowledgments The authors gratefully acknowledge the anonymous reviewers for their comments to help us to improve our paper, and also thank for their enormous help in revising this paper. This work is supported by 973 program of China (No. 2015CB352500), 863 program of China (No. 2015AA016405), and NSF of China (Nos. 61173070, 61202149).

References

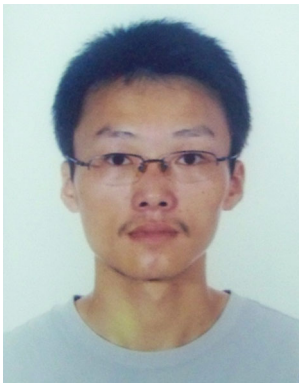
1. Bresenham, J.E.: Algorithm for computer control of a digital plotter. IBM Syst. J. **4**(1), 25–30 (1965)
2. Brown, J., Capson, D.: A framework for 3d model-based visual tracking using a gpu-accelerated particle filter. IEEE Trans. Vis. Comput. Graph. **18**(1), 68–80 (2012)
3. Comport, A., Marchand, E., Pressigout, M., Chaumette, F.: Real-time markerless tracking for augmented reality: the virtual visual servoing framework. IEEE Trans. Vis. Comput. Graph. **12**(4), 615–628 (2006)
4. Dambreville, S., Sandhu, R., Yezzi, A., Tannenbaum, A.: Robust 3d pose estimation and efficient 2d region-based segmentation from a 3d shape prior. In: Proceedings of European Conference on Computer Vision, pp 169–182 (2008)
5. Drummond, T., Cipolla, R.: Real-time visual tracking of complex structures. IEEE Trans. Pattern Anal. Mach. Intell. **24**(7), 932–946 (2002)
6. Harris, C., Stennett, C.: Rapid: a video-rate object tracker. In: Proceedings of British Machine Vision Conference, pp. 73–77 (1990)
7. Hinterstoisser, S., Benhimane, S., Navab, N.: N3m: natural 3d markers for real-time object detection and pose estimation. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 1–7 (2007)
8. Hinterstoisser, S., Cagniart, C., Ilic, S., Sturm, P., Navab, N., Fua, P., Lepetit, V.: Gradient response maps for real-time detection of texture-less objects. IEEE Trans. Pattern Anal. Mach. Intell. **34**(5), 876–888 (2012)

9. Izadi, S., Kim, D., Hilliges, O., Molyneaux, D., Newcombe, R., Kohli, P., Shotton, J., Hodges, S., Freeman, D., Davison, A., Fitzgibbon, A.: Kinectfusion: real-time 3d reconstruction and interaction using a moving depth camera. ACM Symposium on User Interface Software and Technology (2011)
10. Kato, H., Billinghurst, M.: Marker tracking and hmd calibration for a video-based augmented reality conferencing system. In: IEEE and ACM International Symposium on Mixed and Augmented Reality, pp. 85–94 (1999)
11. Kim, K., Lepetit, V., Woo, W.: Scalable real-time planar targets tracking for digilog books. Vis. Comput. **26**(6–8), 1145–1154 (2010)
12. Klein, G., Murray, D.: Full-3d edge tracking with a particle filter. In: Proceedings of British Machine Vision Conference, pp. 114.1-114.10 (2006)
13. Lepetit, V., Fua, P.: Monocular model-based 3d tracking of rigid objects: a survey. Found. Trends Comput. Graph. Vis. **1**(1), 1–89 (2005)
14. Ma, Z., Wu, E.: Real-time and robust hand tracking with a single depth camera. Vis. Comput. **30**(10), 1133–1144 (2014)
15. Marchand, E., Bouthemy, P., Chaumette, F.: A 2d–3d model-based approach to real-time visual tracking. Image Vis. Comput. **19**(13), 941–955 (2001)
16. Prisacariu, V.A., Reid, I.D.: Pwp3d: real-time segmentation and tracking of 3d objects. Int. J. Comput. Vis. **98**(3), 335–354 (2012)
17. Rosenhahn, B., Brox, T., Weickert, J.: Three-dimensional shape knowledge for joint image segmentation and pose tracking. Int. J. Comput. Vis. **73**(3), 243–262 (2007)
18. Rosten, E., Drummond, T.: Fusing points and lines for high performance tracking. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 1508–1515 (2005)
19. Schmaltz, C., Rosenhahn, B., Brox, T., Weickert, J.: Region-based pose tracking with occlusions using 3d models. Mach. Vis. Appl. **23**(3), 557–577 (2012)
20. Seo, B., Park, H., Park, J., Hinterstoisser, S., Llic, S.: Optimal local searching for fast and robust textureless 3d object tracking in highly cluttered backgrounds. IEEE Trans. Vis. Comput. Graph. **20**(1), 99–110 (2014)
21. Vacchetti, L., Lepetit, V., Fua, P.: Combining edge and texture information for real-time accurate 3d camera tracking. In: IEEE and ACM International Symposium on Mixed and Augmented Reality, pp. 48–56 (2004)
22. Vacchetti, L., Lepetit, V., Fua, P.: Stable real-time 3d tracking using online and offline information. IEEE Trans. Pattern Anal. Mach. Intell. **26**(10), 1385–1391 (2004)
23. Wuest, H., Vial, F., Stricker, D.: Adaptive line tracking with multiple hypotheses for augmented reality. In: IEEE and ACM International Symposium on Mixed and Augmented Reality, pp. 62–69 (2005)

Guofeng Wang is now a Ph.D. Candidate of School of Computer Science and Technology in Shandong University. He received his B.Sc. from Central South University in 2009. His research interests include object tracking and computer vision, etc.



Bin Wang is now a Ph.D. Candidate of School of Computer Science and Technology in Shandong University. He received his B.Sc. from Shandong University in 2012. His research interests include object detection and computer vision.



Fan Zhong is a lecturer of School of Computer Science and Technology, Shandong University. He received his Ph.D. from Zhejiang University in 2010. His research interests include image and video editing, and augmented reality, etc.

Xueying Qin is currently a professor of School of Computer Science and Technology, Shandong University. She received her Ph.D. from Hiroshima University of Japan in 2001, and M.S. and B.S. from Zhejiang University and Peking University in 1991 and 1998, respectively. Her main research interests are augmented reality, video-based analyzing, rendering, and photorealistic rendering.

Baoquan Chen is a professor of School of Computer Science and Technology, Shandong University. He received his Ph.D. from the State University of New York at Stony Brook in 1999, and M.S. from Tsinghua University in 1994. His research interests generally lie in computer graphics, visualization, and human–computer interaction, focusing specifically on large-scale city modeling, simulation and visualization.