Image and Vision Computing 175 (2026) 106161



Contents lists available at ScienceDirect

Image and Vision Computing

journal homepage: www.elsevier.com/locate/imavis





Revisiting direct methods for monocular model-based 3D object tracking: A hierarchical optimization

Byung-Kuk Seo

Electronics and Telecommunications Research Institute, 218 Gajeong-ro, Yuseong-gu, Daejeon, 34129, Republic of Korea

ARTICLE INFO

Dataset link: https://www.mi.hs-rm.de/~schwan/research/RBOT/

Keywords:
Hierarchical optimization
Direct method
Model-based approach
3D object tracking
6DoF pose estimation

ABSTRACT

Monocular model-based 3D object tracking has steadily progressed and has continued to draw interest across diverse applications. In particular, region-based methods have recently become the predominant approach, consistently demonstrating state-of-the-art performance. In this work, we explore a new direction in monocular model-based 3D object tracking by revisiting direct methods and introducing a novel energy function and optimization strategy that achieve significant performance improvements. Specifically, the energy function is formulated as a multi-constraint cost function that effectively combines photometric and geometric constraints in a complementary manner. The optimization strategy is designed as a hierarchical framework with stage-tailored energy functions and is further enhanced by an efficient second-order approximation of the photometric cost function. To validate our method, we conduct comprehensive performance evaluations and analyses, including detailed comparative and ablation studies. Experimental results demonstrate that our method is highly competitive with current top-performing methods on challenging objects and scenes.

1. Introduction

3D object tracking — the task of estimating the six-degree-of-freedom (6DoF) pose of a target object, comprising its 3D position and orientation, from camera images — is a fundamental problem in 3D computer vision. It serves as a crucial component in applications ranging from augmented reality to robotics. Despite significant advances, achieving reliable 3D object tracking remains a considerable challenge due to factors such as diverse object types, background clutter, varying illumination, heavy occlusion, and dynamic motion. These difficulties are particularly pronounced when solely relying on monocular RGB images. Nevertheless, monocular 3D object tracking is in high demand and is often preferred to stereo- or depth-based tracking in practice, particularly for augmented and extended reality (AR/XR) applications [1–5].

For monocular 3D object tracking, model-based approaches have demonstrated promising results by leveraging a 3D model of a target object, which provides prior knowledge about the object [6,7] (hereafter referred to as monocular model-based 3D object tracking). From the early stages of these approaches, distinctive visual features (e.g., keypoints or edges) associated with a 3D object model have commonly been employed for 6DoF pose estimation. In particular, edge-based methods have proven advantageous for tracking less-textured or textureless objects, which are frequently encountered in man-made environments. This inherent strength has driven continuous advances in edge-based methods [8–18]. Alternatively, direct methods have gained prominence by integrating dense photometric features from the object region directly into 6DoF pose estimation. Although these methods rely on simple image alignment, which assumes brightness constancy between corresponding object regions across images, more comprehensive modeling of this alignment has shown substantial potential for monocular model-based 3D object tracking [19–24]. In recent years, region-based methods have emerged as a widely adopted baseline for monocular model-based 3D object tracking [25–34]. Through various follow-up studies and extensions, they have consistently achieved state-of-the-art (SOTA) results on challenging objects and scenes [35–39]. While outside the primary scope of this work, it is worth noting that deep learning-based methods have concurrently demonstrated remarkable performance in 3D object tracking [40,41]; however, they remain limited in scenarios restricted to monocular RGB images or requiring real-time performance without reliance on high-end hardware.

Inspired by the active research landscape in monocular model-based 3D object tracking and the lack of recent methodological progress beyond region-based methods, we revisit direct methods and introduce a novel energy function and optimization strategy. In this paper, we demonstrate substantial improvements over previous direct methods and strong competitive performance against current top-performing methods. Our achievements are attributed to the following key contributions:

E-mail address: byungkuk.seo@etri.re.kr.

https://doi.org/10.1016/j.imavis.2026.106161
Received 5 February 2026; Received in revised form 11 August 2026; Accepted 14 August 2026
Available online 21 August 2026
0262-8856/© 2026 The Author. Published by Elsevier B.V. This is an open access article under the CC BY license (http://creativecommons.org/licenses/by/4.0/).

- An energy function is formulated as a multi-constraint cost function that effectively combines photometric and geometric constraints in a complementary manner.
- A hierarchical optimization strategy is designed with stage-tailored energy functions.
- The optimization process is further enhanced by an efficient second-order approximation of the photometric cost function.

Furthermore, we present detailed mathematical derivations of the proposed energy function and optimization. We also provide extensive performance analyses with respect to object characteristics and scene conditions, which have not yet been thoroughly investigated.

In the remainder of this paper, we first briefly discuss related work (Section 2). Next, we describe our energy function and hierarchical optimization strategy with detailed mathematical derivations (Sections 3–5). We then present performance evaluations and analyses, including comparative and ablation studies (Section 6). Finally, we conclude with directions for future research (Section 7).

2. Related work

To clarify the contributions of our method, we categorize relevant work on monocular model-based 3D object tracking according to their energy functions and provide a brief discussion. From a methodological perspective, we particularly highlight several key methods identified by tracing citations of seminal works, rather than pursuing an exhaustive survey or review (see Table 1).

Table 1
Energy functions for monocular model-based 3D object tracking.

Energy function[a]	Method
E_g	Seo et al. [8], Wang et al. [10], Sun et al. [13], Tang et al. [16]
$- wE_g$	Wang et al. [11], Huang et al. [12], Chen et al. [17]
$- (T \rightarrow) wE_g$	Tian et al. [15], Liu et al. [18]
Others	Wang et al. [9], Dong et al. [14]
E_s	Tjaden et al. [25], Tjaden et al. [26], Tjaden et al. [27], Zhou et al. [33]
$- (wE_g \rightarrow) E_s$	Li et al. [31]
$- wE_s$	Zhong et al. [29], Huang et al. [34], Li et al. [39]
$- wE_s + w'E_g$	Li and Gao [28], Li et al. [32]
$- (E_p \rightarrow) wE_s + w'E_p$	Song et al. [36]
E'_s	Stoiber et al. [30], Stoiber et al. [35], Wang et al. [37]
$- wE'_s + w'E'_g$	Lv et al. [38]
E'_p	Crivellaro and Lepetit [19]
$- wE'_p$	Zhong et al. [22]
$- E_s + wE'_p$	Zhong and Zhang [23], Liu et al. [24]
Others	Caron et al. [20], Seo and Wüest [21]

[a] The notations E'_g, E'_s, and E'_p are energy functions, distinguishing them from E_g, E_s, and E_p based on Eqs. (1)–(3). The notations w and w' denote weighting functions. The notation T represents a non-local parameter search, as opposed to local optimization based on energy functions. The right arrow symbol indicates a two-stage optimization process.

2.1. Geometric constraint

Monocular model-based 3D object tracking has typically been formulated using geometric energy functions, commonly known as *edge-(or contour-) based methods*. These energy functions are based on the distances between edge points of a projected 3D object model and their corresponding image edge points:

$$E_g(\boldsymbol{\xi}) = \sum_{\boldsymbol{x} \in C} \|\mathcal{P}(\boldsymbol{X};\boldsymbol{\xi}) - \boldsymbol{x}\|^2, \tag{1}$$

where $\mathcal{P}(\boldsymbol{X};\boldsymbol{\xi})$ denotes the projection of a 3D object model $\boldsymbol{X}$ onto the 2D image plane given an object pose $\boldsymbol{\xi}$, and C is the set of image edge points $\boldsymbol{x}$ corresponding to $\boldsymbol{X}$.[1] Edge-based methods have a long research history and have demonstrated steady progress. In particular, the challenge of reliably establishing correspondences in complex scenes has been extensively addressed. For example, Seo et al. [8] proposed a local search scheme to identify correspondences along sampled search lines using the color appearance of object and background regions. Wang et al. [9] improved this search scheme by adopting a graph model with an additional smoothness term, which regularizes the geometric energy function by enforcing distance consistency between adjacent correspondences. Huang et al. [12] introduced an efficient selection mechanism based on the joint confidence scores of contour distances and color probabilities, which were also used to adaptively weight the geometric energy function. Rather than using sampled edge points, Sun et al. [13] constructed distinctive patches along contour edges and matched them using gradient orientations. Chen et al. [17] recently proposed two search schemes to establish point-to-distribution contour correspondences and sparse interior correspondences, respectively. They also formulated a unified geometric energy function that fuses both contour and interior information.

Other studies have focused on direct optimization rather than explicit correspondence search. For example, Wang et al. [10] optimized the geometric energy function using an edge distance field within a particle-filtering framework. They later extended this method by incorporating a weighting function based on the consistency of edge directions, derived from differences in edge orientations [11]. Similarly, Dong et al. [14] reshaped the geometric energy function using directional chamfer matching, which accounts for both edge distances and orientations. They also proposed a weighting function that considers differences between edge and gradient directions, modulated by gradient magnitudes. Tang et al. [16] reformulated the geometric energy function using the angle differences between the projection planes of 3D contour segments and the back-projection lines of corresponding 2D edge points.

In addition, to address the challenge associated with large inter-frame displacements, Tian et al. [15] proposed a non-local parameter search in the 2D out-of-plane rotation space, coupled with the local optimization of the geometric energy function. Liu et al. [18] recently improved this non-local parameter search by adopting an adaptive shift on a grid foreground probability map and performing a non-local search in rotation space via Bayesian optimization. They also introduced a weighting function that combines color probability with distance-based Gaussian weights.

2.2. Statistical constraint

Since statistical level-set segmentation has successfully been adopted for monocular model-based 3D object tracking [42], *region-based methods* have been characterized by statistical energy functions. These energy functions are based on the pixel-wise posterior probabilities of an object's foreground and background membership in an image, denoted by $P_f(\boldsymbol{x})$ and $P_b(\boldsymbol{x})$, respectively:

$$E_s(\boldsymbol{\xi}) = -\sum_{\boldsymbol{x} \in \Omega} \log(\mathcal{H}(\Phi(\boldsymbol{x};\boldsymbol{\xi}))P_f(\boldsymbol{x}) + (1 - \mathcal{H}(\Phi(\boldsymbol{x};\boldsymbol{\xi})))P_b(\boldsymbol{x})), \tag{2}$$

where $\Phi(\boldsymbol{x};\boldsymbol{\xi})$ denotes a level-set embedding function encoding the object shape given an object pose $\boldsymbol{\xi}$, $\mathcal{H}(\cdot)$ is a smoothed Heaviside step function, Ω is the image region partitioned into foreground and background regions by the object contour. Building on this energy function, substantial advances have been made in segmentation and optimization [43,44]. Notably, Tjaden et al. demonstrated impressive tracking robustness and speed by introducing a pixel-wise optimization strategy based on a Gauss–Newton-like algorithm [25] and a localized segmentation model using temporally consistent local color histograms [26]. They also achieved further improvements by reformulating the optimization problem as an iteratively reweighted least-squares estimation [27].

[1] Hereafter, unless otherwise specified, $\|\cdot\|^2$ denotes the squared L_2 norm.

After these seminal contributions by Tjaden et al. numerous follow-up studies and extensions have been proposed for monocular model-based 3D object tracking. For example, Zhong et al. [29] introduced a local region partitioning strategy based on polar coordinates, combined with a joint weighting function that leverages edge distances and color differences. Zhou et al. [33] designed local circular regions that adaptively adjust to changes in object scale. Instead of local regions, Stoiber et al. [30] employed local lines along the object contour, inspired by edge-based methods, in their segmentation model. In this work, they proposed an efficient discrete scale-space formulation of a probabilistic model, leveraging smoothed step functions to model uncertainty in the contour location. They later extended it through dual optimization strategies based on local and global uncertainties [35]. Huang et al. [34] also used local lines but initially performed an explicit search to identify optimal contour points based on edge distances and color probabilities. They then optimized the statistical energy function using a joint weighting function that integrates both distance and color constraints of the optimal contour points. Based on the method of [35], Wang et al. [37] introduced an efficient transitional view search strategy to effectively recover motion continuity, making region-based methods more robust against large pose shifts. Recently, Li et al. [39] improved the method of [29] by adopting a joint weighting function that leverages edge distances and color differences.

Other studies have explored combining the region-based statistical constraint with additional constraints. For example, Li and Gao [28] integrated statistical and geometric energy functions, adaptively weighting them according to the success rate of contour point matching. Li et al. [32] also followed a similar fusion strategy but introduced a weighting function based on the joint confidence scores of contour and region points, computed from orientation differences and color probabilities, respectively. To account for the inaccuracies in 3D object models, Song et al. [36] combined the statistical constraint with a color consistency constraint. Rather than performing direct fusion, Li et al. [31] proposed a two-step optimization strategy comprising coarse geometric optimization followed by statistical refinement. Lv et al. [38] employed multi-feature probabilities for the statistical energy function and directly fused geometric edge features into probability estimation.

2.3. Photometric constraint

Image alignment [45] has been widely used for tracking problems in computer vision, ranging from 2D planar tracking [46] to simultaneous localization and mapping [47]. In the context of monocular model-based 3D object tracking, image alignment is formulated using photometric energy functions. These energy functions are based on 3D warping between consecutive images, often referred to as *direct methods*:

$$E_p(\boldsymbol{\xi}) = \sum_{\boldsymbol{x}\in\mathcal{O}} \|I_{t+1}(\mathcal{W}(\boldsymbol{x};\boldsymbol{\xi})) - I_t(\boldsymbol{x})\|^2, \tag{3}$$

where $\mathcal{W}(\boldsymbol{x};\boldsymbol{\xi})$ denotes a 3D warping function that maps a point $\boldsymbol{x}$ in image I_t to its corresponding point in image I_{t+1} given an object pose $\boldsymbol{\xi}$, $I(\cdot)$ is the image intensity function, and $\mathcal{O}$ is the object region. In direct methods, the photometric energy function has been modeled in a variety of forms to improve tracking performance. For example, Caron et al. [20] formulated image alignment using mutual information rather than intensity differences. Crivellaro and Lepetit [19] proposed a local descriptor that applies a series of filters and operations to image intensities, demonstrating greater resilience and discriminative power compared to raw intensity values. Zhong et al. [22] designed dense feature fields that incorporate multiple pixel-level features. Seo and Wüest [21] reformulated the photometric energy function by modeling image intensity variations based on the surface normal information of an object model.

Beyond standalone use, direct methods have been fused with other methods. For example, the robust descriptor of [19] was combined with the region-based statistical constraint, resulting in better performance than either method alone [23,24]. The photometric constraint effectively served as a coarse alignment stage, providing an initial pose estimate that satisfies the small displacement assumption needed for a subsequent optimization step [36].

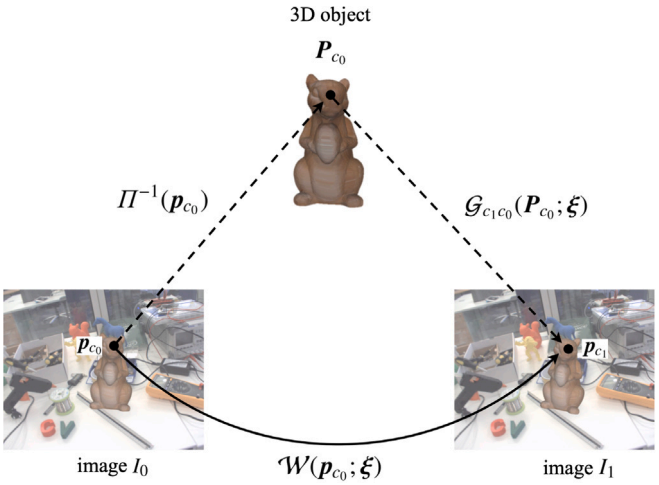


Fig. 1. Illustration of a 3D warping function $\mathcal{W}(\boldsymbol{p}_{c_0};\boldsymbol{\xi})$ that maps a point $\boldsymbol{p}_{c_0}$ in image I_0 to its corresponding point $\boldsymbol{p}_{c_1}$ in image I_1 given an object pose $\boldsymbol{\xi}$ (see Eqs. (4)–(6) for more details on the mathematical notation).

3. Preliminary

In direct methods, monocular model-based 3D object tracking is formulated as image alignment, in which a point $\boldsymbol{p}_{c_0}$ in image I_0 is aligned with its corresponding point $\boldsymbol{p}_{c_1}$ in image I_1 using a 3D warping function $\mathcal{W}(\cdot)$ parameterized by an object pose $\boldsymbol{\xi}$ (see Fig. 1). In this paper, we define the 3D warping function as

$$\begin{aligned}\mathcal{W}(\boldsymbol{p}_{c_0};\boldsymbol{\xi}) &= \Pi(\mathcal{G}_{c_1c_0}(\Pi^{-1}(\boldsymbol{p}_{c_0});\boldsymbol{\xi})) \\ &= \Pi(\mathcal{G}_{c_1c_0}(\boldsymbol{P}_{c_0};\boldsymbol{\xi})) = \Pi(\boldsymbol{P}_{c_1}) = \boldsymbol{p}_{c_1},\end{aligned} \tag{4}$$

where $\boldsymbol{P}_{c_0}$ and $\boldsymbol{P}_{c_1}$ are 3D surface points on the target object, represented in the camera coordinate systems associated with images I_0 and I_1, respectively. The function $\Pi(\cdot)$ is the perspective projection, and its inverse $\Pi^{-1}(\cdot)$ is given by

$$\Pi^{-1}(\boldsymbol{p}_{c_0}) = D(\boldsymbol{p}_{c_0})\boldsymbol{K}^{-1}[\boldsymbol{p}_{c_0}^\top \;\; 1]^\top = \boldsymbol{P}_{c_0}, \tag{5}$$

where $\boldsymbol{K}$ is the 3×3 intrinsic matrix of the camera, and $D(\boldsymbol{p}_{c_0})$ is the depth value at $\boldsymbol{p}_{c_0}$, obtained from the depth buffer generated by rendering the 3D object model. The function $\mathcal{G}_{c_1c_0}(\cdot)$ is the rigid-body transformation from the camera coordinate system associated with image I_0 to that associated with image I_1, parameterized by the 6D vector $\boldsymbol{\xi} = [\boldsymbol{\omega}^\top \;\; \boldsymbol{\upsilon}^\top]^\top \in \mathbb{R}^6$, where $\boldsymbol{\omega}$ is the 3D rotation vector and $\boldsymbol{\upsilon}$ is the 3D translation vector. Here, the rigid-body transformation is mapped to the matrix exponential of a twist $\hat{\boldsymbol{\xi}} \in \mathfrak{se}(3)$, and thus, $\mathcal{G}_{c_1c_0}(\cdot)$ can be described as

$$\begin{aligned}\mathcal{G}_{c_1c_0}(\tilde{\boldsymbol{P}}_{c_0};\boldsymbol{\xi}) &= \exp(\hat{\boldsymbol{\xi}})[\boldsymbol{P}_{c_0}^\top \;\; 1]^\top = \exp\left(\begin{bmatrix}[\boldsymbol{\omega}]_\times & \boldsymbol{\upsilon} \\ \boldsymbol{0}_{1\times 3} & 0\end{bmatrix}\right)[\boldsymbol{P}_{c_0}^\top \;\; 1]^\top \\ &= \tilde{\boldsymbol{T}}_{c_1c_0}[\boldsymbol{P}_{c_0}^\top \;\; 1]^\top = \begin{bmatrix}\boldsymbol{R}_{c_1c_0} & \boldsymbol{t}_{c_1c_0} \\ \boldsymbol{0}_{1\times 3} & 1\end{bmatrix}[\boldsymbol{P}_{c_0}^\top \;\; 1]^\top = \tilde{\boldsymbol{P}}_{c_1},\end{aligned} \tag{6}$$

where $\tilde{\boldsymbol{T}}_{c_1c_0}$ is the 4×4 homogeneous rigid-body transformation matrix, consisting of the 3D rotation matrix $\boldsymbol{R}_{c_1c_0} \in \mathrm{SO}(3)$ and the 3D translation vector $\boldsymbol{t}_{c_1c_0} \in \mathbb{R}^3$, the tilde symbol indicates a homogeneous-coordinate representation, and $[\cdot]_\times$ denotes the skew-symmetric matrix operator. Based on the image alignment defined by the 3D warping function, our goal is to estimate the rigid-body transformation between

consecutive images. This is achieved by estimating the 6D parameter vector $\boldsymbol{\xi}$ through the optimization of the following multi-constraint cost function.

4. Multi-constraint cost function

To formulate our energy function, we first derive a photometric cost function by modeling image intensity variations within the object region. We then incorporate a geometric cost function on the object contour, which complements the photometric cost function.

4.1. Photometric cost function

In general, image alignment is based on the brightness constancy assumption. However, in uncontrolled environments, the assumption can easily be violated due to motion or illumination changes. In early approaches, these intensity variations were modeled in a generalized form of the brightness constancy assumption between images over the time interval $[t, t+1]$, using a multiplicative scale factor M and an additive bias C [48]:

$$I_{t+1} = M I_t + C, \tag{7}$$

where I_t and I_{t+1} denote the image intensities at times t and $t+1$, respectively. In a more elaborate formulation, these variations were also modeled using the surface normal information from an object model [21]. In this work, we leverage such insights to formulate our photometric energy function.

Let us consider a rigid 3D object with a Lambertian surface observed by an RGB camera. The image intensity observed at a point in image I, associated with a 3D surface point of the object, is given by

$$I = \sigma \boldsymbol{n}^\top \boldsymbol{l}, \tag{8}$$

where σ is a surface albedo, $\boldsymbol{n}$ is a unit surface normal vector, and $\boldsymbol{l}$ is an unknown scaled light vector, representing its direction and intensity. By taking the derivative on both sides of Eq. (8) with respect to time t, we can obtain the differential change in image intensity:

$$\frac{dI}{dt} = \sigma \frac{d}{dt}(\boldsymbol{n}^\top \boldsymbol{l}) + (\boldsymbol{n}^\top \boldsymbol{l})\frac{d\sigma}{dt} = \sigma \frac{d}{dt}(\boldsymbol{n}^\top \boldsymbol{l}). \tag{9}$$

Here, σ is intrinsically defined on the surface, and its time derivative vanishes ($d\sigma/dt = 0$). Given the differential change, we can describe image intensity variations observed between consecutive images under camera motion relative to the 3D object:

$$I_{t+1} - I_t = \sigma(\boldsymbol{n}^\top \boldsymbol{l})_{t+1} - \sigma(\boldsymbol{n}^\top \boldsymbol{l})_t = \sigma(\boldsymbol{n}^\top \boldsymbol{l})_t \left(\frac{(\boldsymbol{n}^\top \boldsymbol{l})_{t+1}}{(\boldsymbol{n}^\top \boldsymbol{l})_t} - 1 \right) = I_t(\kappa - 1), \tag{10}$$

which is simplified to

$$I_{t+1} = \kappa I_t. \tag{11}$$

Note that this model can be considered a special case of the generalized form in Eq. (7) ($M = \kappa, C = 0$) where the conventional brightness constancy assumption holds when $\kappa = 1$. Building on this model, our photometric cost function e_p is defined as

$$e_p(\boldsymbol{p}; \boldsymbol{\xi}) = I_{t+1}(\mathcal{W}(\boldsymbol{p}; \boldsymbol{\xi})) - \kappa I_t(\boldsymbol{p}), \tag{12}$$

where κ denotes the compensation factor, computed using the surface normal information of a 3D object model:

$$\kappa = \frac{(\boldsymbol{n}^\top \boldsymbol{l})_{t+1}}{(\boldsymbol{n}^\top \boldsymbol{l})_t} \approx \frac{\mathcal{E}[I_{t+1}|\boldsymbol{n}_{t+1}]}{\mathcal{E}[I_t|\boldsymbol{n}_t]}, \tag{13}$$

where $\mathcal{E}[I|\boldsymbol{n}]$ denotes the conditional expectation of the image intensity I given the surface normal $\boldsymbol{n}$. Here, the conditional expectation can be computed by multivariate linear regression based on a first-order approximation of the linear radiance model [49]:

$$\mathcal{E}[I|\boldsymbol{n}] \approx I \approx \sigma(l_0 + n_x l_x + n_y l_y + n_z l_z), \tag{14}$$

where $\boldsymbol{n} = (n_x, n_y, n_z)^\top$ is a unit surface normal vector, $\boldsymbol{l} = (l_x, l_y, l_z)^\top$ is a light vector, l_0 is an additional offset, and σ is a surface albedo.

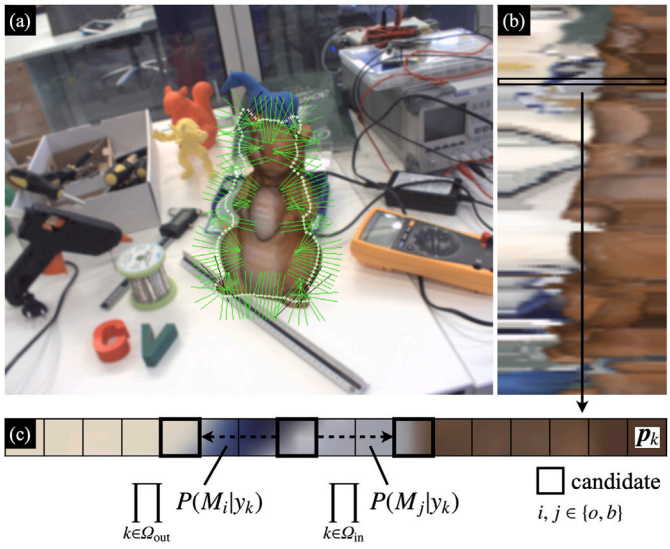


Fig. 2. Illustration of a local search scheme: (a) sampled edge points of the projected 3D object model (white dots) and search lines (green lines), (b) stack of search lines, (c) evaluation of correspondence candidates for suppression (see Eqs. (16)–(18) for more details on the mathematical notation). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

4.2. Geometric cost function

In direct methods, a 3D warping function is mathematically well-defined for all pixels within the object region. However, the function is likely to be discontinuous at the occluding contours of a 3D object under camera motion, making it non-differentiable. Moreover, the function is undefined when the 3D surface points of the object corresponding to the image points are not visible. To address these issues, we employ a geometric constraint on the object contour, where a photometric cost function may be weakly constrained.

Our geometric cost function e_g follows the conventional formulation shown in Eq. (1), which is based on the distance between an edge point of a projected 3D object model $\mathcal{P}(\boldsymbol{P}; \boldsymbol{\xi})$ and its corresponding image edge point $\boldsymbol{p}$:

$$e_g(\boldsymbol{p}; \boldsymbol{\xi}) = \mathcal{P}(\boldsymbol{P}; \boldsymbol{\xi}) - \boldsymbol{p}. \tag{15}$$

To establish the correspondences between $\mathcal{P}(\boldsymbol{P}; \boldsymbol{\xi})$ and $\boldsymbol{p}$, we adopt a local search scheme that identifies them along sampled search lines. In this scheme, however, we utilize region appearance statistics along the search lines to suppress undesired correspondence candidates, thereby reducing false matches in heavily cluttered scenes. Specifically, we first extract potential candidates by detecting local maxima of gradient responses along the search line. We then evaluate each candidate based on the overall posterior probabilities of its neighboring regions on both sides of the search line:

$$P_{i,j} = \prod_{k\in\Omega_{\text{out}}} P(M_i|y_k) \prod_{k\in\Omega_{\text{in}}} P(M_j|y_k), \quad i, j \in \{o, b\}, \tag{16}$$

where $P(M_i|y_k)$ denotes the pixel-wise posterior probability of model M_i (representing either the object model M_o or the background model M_b) given the image intensity $y_k = I(\boldsymbol{p}_k)$ at point $\boldsymbol{p}_k$, which is located in the inward region Ω_{in} or outward region Ω_{out} of the search line (see Fig. 2). Here, the pixel-wise posterior probability can be computed [50] as

$$P(M_i|y_k) = \frac{P(y_k|M_i)P(M_i)}{P(y_k|M_o)P(M_o) + P(y_k|M_b)P(M_b)}, \tag{17}$$

and assuming that the object and background region statistics are equally likely along the search line ($P(M_o) = P(M_b)$), it can be simplified to

$$P(M_i|y_k) = \frac{P(y_k|M_i)}{P(y_k|M_o) + P(y_k|M_b)}, \tag{18}$$

where $P(y_k|M_i)$ denotes the pixel-wise likelihood function of the object or background model. Using the overall posterior probabilities $P_{i,j}$, we finally filter out candidates as outliers if they are more likely to belong to either the object or the background region than to the transition region between them: if $P_{o,o}$ or $P_{b,b}$ is larger than $P_{b,o}$ and $P_{o,b}$.

5. Optimization

With the cost functions in Eqs. (12) and (15), the object pose ξ is estimated by optimizing the following joint energy function $E(\xi)$:

$$E(\xi) = E_p(\xi) + \eta E_g(\xi), \tag{19}$$

which is written as a reweighted nonlinear least-squares minimization:

$$\arg\min_{\xi} \frac{1}{2}\left(\sum_{p\in\mathcal{O}} w_1 \|e_p(\boldsymbol{p};\xi)\|^2 + \eta \sum_{p\in\mathcal{C}} w_2 \|e_g(\boldsymbol{p};\xi)\|^2\right), \tag{20}$$

where η denotes the balancing factor between the photometric energy function E_p and the geometric energy function E_g, $\mathcal{O}$ is the object region, $\mathcal{C}$ is the set of image edge points, and w_1 and w_2 are weighting functions. Meanwhile, the photometric energy function is defined by image alignment with a single scale factor, as shown in Eq. (12), which can lead to non-convex behavior, causing fluctuations or roughness during optimization. To alleviate this, we impose a regularization term e_{pr}, which is based on an additional photometric constraint derived from Eq. (13):

$$I_{t+1}(\mathcal{W}(\boldsymbol{p};\xi)) \approx \mathcal{E}[I_{t+1}|\boldsymbol{n}_{t+1}](\mathcal{W}(\boldsymbol{p};\xi)), \tag{21}$$

and is defined as the following cost function:

$$e_{pr}(\boldsymbol{p};\xi) = I_{t+1}(\mathcal{W}(\boldsymbol{p};\xi)) - \kappa\mathcal{E}[I_t|\boldsymbol{n}_t](\boldsymbol{p}). \tag{22}$$

Note that this function differs from that in [21], which does not account for the compensation factor κ. With the regularization term, the minimization in Eq. (20) is rewritten as

$$\begin{aligned}\arg\min_{\xi} \frac{1}{2}\Bigg(&\sum_{p\in\mathcal{O}} w_1\left(\|e_p(\boldsymbol{p};\xi)\|^2 + \lambda\|e_{pr}(\boldsymbol{p};\xi)\|^2\right)\\ &+ \eta\sum_{p\in\mathcal{C}} w_2\|e_g(\boldsymbol{p};\xi)\|^2\Bigg),\end{aligned} \tag{23}$$

where λ is the enforcing factor of the regularization term.

For optimization, we adopt the Gauss–Newton strategy, which iteratively minimizes the multi-constraint cost function in Eq. (23) with respect to an incremental motion $\Delta\xi$ at each iteration k. In this manner, the minimization in Eq. (23) can be rewritten using a first-order Taylor expansion, evaluated at $\xi = \mathbf{0}$ for the photometric cost functions and at $\xi = \xi^{(k)}$ for the geometric cost function, as

$$\begin{aligned}\arg\min_{\Delta\xi} \frac{1}{2}\Bigg(&\sum_{p\in\mathcal{O}} w_1\Big(\|e_p(\boldsymbol{p};\xi^{(k)}) + \boldsymbol{\mathcal{J}}_{I_{t+1}}(\mathcal{W}(\mathcal{W}(\boldsymbol{p};\mathbf{0});\xi^{(k)}))\Delta\xi\|^2\\ &+ \lambda\|e_{pr}(\boldsymbol{p};\xi^{(k)}) + \boldsymbol{\mathcal{J}}_{I_{t+1}}(\mathcal{W}(\mathcal{W}(\boldsymbol{p};\mathbf{0});\xi^{(k)}))\Delta\xi\|^2\Big)\\ &+ \eta\sum_{p\in\mathcal{C}} w_2\|e_g(\boldsymbol{p};\xi^{(k)}) + \boldsymbol{\mathcal{J}}_{\mathcal{P}}(\boldsymbol{p};\xi^{(k)})\Delta\xi\|^2\Bigg),\end{aligned} \tag{24}$$

where the Jacobians are computed as

$$\begin{aligned}\boldsymbol{\mathcal{J}}_{I_{t+1}}(\mathcal{W}(\mathcal{W}(\boldsymbol{p};\mathbf{0});\xi^{(k)})) &= \left.\frac{\partial I_{t+1}(\mathcal{W}(\mathcal{W}(\boldsymbol{p};\xi);\xi^{(k)}))}{\partial\xi}\right|_{\xi=\mathbf{0}}\\ \boldsymbol{\mathcal{J}}_{\mathcal{P}}(\boldsymbol{p};\xi^{(k)}) &= \left.\frac{\partial\mathcal{P}(\boldsymbol{p};\xi)}{\partial\xi}\right|_{\xi=\xi^{(k)}}.\end{aligned} \tag{25}$$

Here, the photometric cost functions are represented based on the forward compositional algorithm [45] for computational efficiency, where $\xi^{(k+1)}$ is defined such that $\mathcal{W}(\boldsymbol{p};\xi^{(k+1)}) = \mathcal{W}(\mathcal{W}(\boldsymbol{p};\Delta\xi);\xi^{(k)})$. Therefore, in this minimization, the object pose ξ is determined by iteratively computing $\Delta\xi$ and updating the current pose $\xi^{(k)}$ at each iteration until convergence.

5.1. Second-order approximation

In our method, we enhance the optimization process through an efficient second-order approximation of the photometric cost function. Start with a simple photometric cost function based on the forward compositional algorithm:

$$e_p(\boldsymbol{p};\xi^{(k+1)}) = I_{t+1}(\mathcal{W}(\mathcal{W}(\boldsymbol{p};\Delta\xi);\xi^{(k)})) - I_t(\boldsymbol{p}). \tag{26}$$

Assuming that $\mathcal{W}(\boldsymbol{p};\mathbf{0}) = \boldsymbol{p}$, its second-order Taylor expansion, evaluated at $\xi = \mathbf{0}$, can be written as

$$\begin{aligned}I_{t+1}(\mathcal{W}(\mathcal{W}(\boldsymbol{p};\Delta\xi);\xi^{(k)})) &\approx I_{t+1}(\mathcal{W}(\boldsymbol{p};\xi^{(k)}))\\ &+ \left.\frac{\partial I_{t+1}(\mathcal{W}(\mathcal{W}(\boldsymbol{p};\xi);\xi^{(k)}))}{\partial\xi}\right|_{\xi=\mathbf{0}}\Delta\xi\\ &+ \frac{1}{2}\Delta\xi^{\top}\left.\frac{\partial^2 I_{t+1}(\mathcal{W}(\mathcal{W}(\boldsymbol{p};\xi);\xi^{(k)}))}{\partial\xi^{\top}\partial\xi}\right|_{\xi=\mathbf{0}}\Delta\xi.\end{aligned} \tag{27}$$

By approximating the second-order derivative in Eq. (27), inspired by [51] (see Appendix A for detailed derivations):

$$\begin{aligned}&\left.\frac{\partial^2 I_{t+1}(\mathcal{W}(\mathcal{W}(\boldsymbol{p};\xi);\xi^{(k)}))}{\partial\xi^{\top}\partial\xi}\right|_{\xi=\mathbf{0}}\Delta\xi\\ &\approx \left.\frac{\partial I_t(\mathcal{W}(\boldsymbol{p};\xi))}{\partial\xi}\right|_{\xi=\mathbf{0}} - \left.\frac{\partial I_{t+1}(\mathcal{W}(\mathcal{W}(\boldsymbol{p};\xi);\xi^{(k)}))}{\partial\xi}\right|_{\xi=\mathbf{0}},\end{aligned} \tag{28}$$

the expansion in Eq. (27) can be simplified to

$$\begin{aligned}I_{t+1}(\mathcal{W}(\mathcal{W}(\boldsymbol{p};\Delta\xi);\xi^{(k)})) &\approx I_{t+1}(\mathcal{W}(\boldsymbol{p};\xi^{(k)}))\\ &+ \frac{1}{2}\left(\left.\frac{\partial I_t(\mathcal{W}(\boldsymbol{p};\xi))}{\partial\xi}\right|_{\xi=\mathbf{0}} + \left.\frac{\partial I_{t+1}(\mathcal{W}(\mathcal{W}(\boldsymbol{p};\xi);\xi^{(k)}))}{\partial\xi}\right|_{\xi=\mathbf{0}}\right)\Delta\xi,\end{aligned} \tag{29}$$

and the second-order approximation of Eq. (26) can be described as

$$\begin{aligned}e_p(\boldsymbol{p};\xi^{(k+1)}) &\approx e_p(\boldsymbol{p};\xi^{(k)})\\ &+ \frac{1}{2}\left(\left.\frac{\partial I_t(\mathcal{W}(\boldsymbol{p};\xi))}{\partial\xi}\right|_{\xi=\mathbf{0}} + \left.\frac{\partial I_{t+1}(\mathcal{W}(\mathcal{W}(\boldsymbol{p};\xi);\xi^{(k)}))}{\partial\xi}\right|_{\xi=\mathbf{0}}\right)\Delta\xi,\end{aligned} \tag{30}$$

where the Jacobians are computed as

$$\begin{aligned}&\left.\frac{\partial I_t(\mathcal{W}(\boldsymbol{p};\xi))}{\partial\xi}\right|_{\xi=\mathbf{0}} + \left.\frac{\partial I_{t+1}(\mathcal{W}(\mathcal{W}(\boldsymbol{p};\xi);\xi^{(k)}))}{\partial\xi}\right|_{\xi=\mathbf{0}}\\ &= \left.\frac{\partial I_t(\boldsymbol{p}')}{\partial\boldsymbol{p}'}\right|_{\boldsymbol{p}'=\mathcal{W}(\boldsymbol{p};\mathbf{0})=\boldsymbol{p}}\left.\frac{\partial\mathcal{W}(\boldsymbol{p};\xi)}{\partial\xi}\right|_{\xi=\mathbf{0}}\\ &\quad + \left.\frac{\partial I_{t+1}(\mathcal{W}(\boldsymbol{p}';\xi^{(k)}))}{\partial\boldsymbol{p}'}\right|_{\boldsymbol{p}'=\mathcal{W}(\boldsymbol{p};\mathbf{0})=\boldsymbol{p}}\left.\frac{\partial\mathcal{W}(\boldsymbol{p};\xi)}{\partial\xi}\right|_{\xi=\mathbf{0}}\\ &= \left(\left.\frac{\partial I_t(\boldsymbol{p}')}{\partial\boldsymbol{p}'}\right|_{\boldsymbol{p}'=\boldsymbol{p}} + \left.\frac{\partial I_{t+1}(\mathcal{W}(\boldsymbol{p}';\xi^{(k)}))}{\partial\boldsymbol{p}'}\right|_{\boldsymbol{p}'=\boldsymbol{p}}\right)\left.\frac{\partial\mathcal{W}(\boldsymbol{p};\xi)}{\partial\xi}\right|_{\xi=\mathbf{0}}\\ &\text{(using simplified notation)}\\ &= (\boldsymbol{\mathcal{J}}_{I_t}(\boldsymbol{p}) + \boldsymbol{\mathcal{J}}_{I_{t+1}}(\mathcal{W}(\boldsymbol{p};\xi^{(k)})))\boldsymbol{\mathcal{J}}_{\mathcal{W}}(\boldsymbol{p};\mathbf{0}).\end{aligned} \tag{31}$$

Using this second-order approximation, the minimization in Eq. (24) is finally rewritten as

$$\begin{aligned}\arg\min_{\Delta\xi} \frac{1}{2}\Bigg(&\sum_{p\in\mathcal{O}} w_1\left(\|e_p(\boldsymbol{p};\xi^{(k)}) + \boldsymbol{\mathcal{J}}_p\boldsymbol{\mathcal{J}}_{\mathcal{W}}(\boldsymbol{p};\mathbf{0})\Delta\xi\|^2\right.\\ &\left.+ \lambda\|e_{pr}(\boldsymbol{p};\xi^{(k)}) + \boldsymbol{\mathcal{J}}_{pr}\boldsymbol{\mathcal{J}}_{\mathcal{W}}(\boldsymbol{p};\mathbf{0})\Delta\xi\|^2\right)\\ &+ \eta\sum_{p\in\mathcal{C}} w_2\|e_g(\boldsymbol{p};\xi^{(k)}) + \boldsymbol{\mathcal{J}}_{\mathcal{P}}(\boldsymbol{p};\xi^{(k)})\Delta\xi\|^2\Bigg),\end{aligned} \tag{32}$$

where the Jacobians are computed as

$$\begin{aligned}\boldsymbol{\mathcal{J}}_p &= \frac{1}{2}\Big(\kappa\boldsymbol{\mathcal{J}}_{I_t}(\boldsymbol{p}) + \boldsymbol{\mathcal{J}}_{I_{t+1}}(\mathcal{W}(\boldsymbol{p};\boldsymbol{\xi}^{(k)}))\Big)\\ \boldsymbol{\mathcal{J}}_{pr} &= \frac{1}{2}\Big(\kappa\boldsymbol{\mathcal{J}}_{\mathcal{E}[I_t|\boldsymbol{n}_t]}(\boldsymbol{p}) + \boldsymbol{\mathcal{J}}_{I_{t+1}}(\mathcal{W}(\boldsymbol{p};\boldsymbol{\xi}^{(k)}))\Big).\end{aligned} \tag{33}$$

Here, in this minimization, we highlight the advantages of Jacobian computation during optimization: (1) the Jacobians $\boldsymbol{\mathcal{J}}_{I_t}(\boldsymbol{p})$, $\boldsymbol{\mathcal{J}}_{\mathcal{E}[I_t|\boldsymbol{n}_t]}(\boldsymbol{p})$, and $\boldsymbol{\mathcal{J}}_{\mathcal{W}}(\boldsymbol{p};\boldsymbol{0})$ can be precomputed before the iterative optimization, eliminating the need for recomputation; and (2) under incremental motion, the Jacobian $\boldsymbol{\mathcal{J}}_{\mathcal{E}[I_t|\boldsymbol{n}_t]}(\boldsymbol{p})$ can be further approximated by $\boldsymbol{\mathcal{J}}_{I_t}(\boldsymbol{p})$, allowing the simplified computation $\boldsymbol{\mathcal{J}}_p \approx \boldsymbol{\mathcal{J}}_{pr}$.

5.2. Hierarchical optimization

For robust estimation, we further employ a scale-space representation of the image [52], which allows motion information to propagate from coarse to fine scales, providing reliable initializations at each level and accelerating the convergence of the optimization. However, our photometric cost function, which is based on the intensity variation model, may be less effective at coarser image levels due to intensity smoothing. To capitalize on the benefits of the scale-space representation, we introduce a hierarchical optimization strategy with stage-tailored energy functions.

Given a multiscale image pyramid, we minimize the joint energy function in Eq. (19) at the finest image level (corresponding to the original image) using Gauss–Newton optimization based on Eqs. (32) and (33). At coarse image levels, on the other hand, we minimize the following energy function:

$$E(\boldsymbol{\xi}) = E_{(p;s)}(\boldsymbol{\xi}) + \Gamma(\boldsymbol{\xi}), \tag{34}$$

which is written as a nonlinear least-squares minimization:

$$\arg\min_{\boldsymbol{\xi}} \frac{1}{2}\left(\sum_{p\in\mathcal{O}} \|e_{(p;s)}(\boldsymbol{p};\boldsymbol{\xi})\|^2 + \|\Gamma(\boldsymbol{\xi})\|^2\right), \tag{35}$$

where $\Gamma(\boldsymbol{\xi})$ is a regularization term that stabilizes the pose update at coarse image levels. Here, the cost function $e_{(p;s)}$ is given by

$$e_{(p;s)}(\boldsymbol{p};\boldsymbol{\xi}) = G(I_{t+1}(\mathcal{W}(\boldsymbol{p};\boldsymbol{\xi})); s) - G(I_t(\boldsymbol{p}); s). \tag{36}$$

where $G(I;s)$ denotes the convolution of image I with an isotropic Gaussian kernel at scale s. By applying the Gauss–Newton strategy based on Eqs. (32) and (33), the minimization in Eq. (35) is rewritten as

$$\arg\min_{\Delta\boldsymbol{\xi}} \frac{1}{2}\left(\sum_{p\in\mathcal{O}} \|e_{(p;s)}(\boldsymbol{p};\boldsymbol{\xi}^{(s)}) + \boldsymbol{\mathcal{J}}_{(p;s)}\boldsymbol{\mathcal{J}}_{\mathcal{W}}(\boldsymbol{p};\boldsymbol{0})\Delta\boldsymbol{\xi}\|^2 + \|\boldsymbol{\mathcal{D}}\Delta\boldsymbol{\xi}\|^2\right), \tag{37}$$

where the Jacobian is computed as

$$\boldsymbol{\mathcal{J}}_{(p;s)} = \frac{1}{2}\Big(\boldsymbol{\mathcal{J}}_{G(I_t;s)}(\boldsymbol{p}) + \boldsymbol{\mathcal{J}}_{G(I_{t+1};s)}(\mathcal{W}(\boldsymbol{p};\boldsymbol{\xi}^{(s)}))\Big), \tag{38}$$

and $\boldsymbol{\xi}^{(s)}$ is the current pose at scale s. Here, $\boldsymbol{\mathcal{D}}$ is a diagonal weighting matrix that assigns separate weights to the two distinct components of $\boldsymbol{\xi}$, rather than using a single weight: $\boldsymbol{\mathcal{D}} = \mathrm{diag}(\sqrt{\boldsymbol{\alpha}})$ with $\boldsymbol{\alpha} = [\alpha_\omega \mathbf{1}_3^\top \ \alpha_v \mathbf{1}_3^\top]^\top \in \mathbb{R}^6$, where α_ω and α_v denote the weights corresponding to the rotation and translation vectors, respectively, and $\mathbf{1}_3$ is the 3D vector of ones. Note that although the isotropic Gaussian kernels can make the higher-order terms in the Taylor expansion negligible, we apply the second-order approximation, as we observed that it led to some performance improvement. During the hierarchical optimization, the incremental motion computed at each coarse image level is propagated to the next image level. The motion estimate obtained after processing all coarse image levels serves as the initial estimate for optimization at the finest image level.

6. Experiment

6.1. Implementation

For the scale-space representation of the image in the photometric cost function, we built a four-level multiscale image pyramid based on the image resolution of the RBOT dataset (see Appendix C for additional analyses). Each level was downsampled by a factor of two using 5×5 Gaussian smoothing and bilinear interpolation. The pixel-wise depth data corresponding to each pyramid level was not downsampled to avoid interpolation across boundaries; instead, they were directly sampled from the depth data of the finest image level.

To identify potential correspondence candidates in the geometric cost function, we calculated the maximum responses of the first-order finite differences for each color channel along sampled search lines, followed by non-maximum suppression. The appearance models for the object and background regions were constructed using an RGB color histograms with 32 bins per channel (see Appendix C for additional analyses). Following the update scheme of [50], which incorporates previous observations, both appearance models were updated only when tracking was successful.

In the joint energy function, we determined the modality-balancing factor based on the contribution of each modality: $\eta = (N_p^{\text{inlier}}/N_g^{\text{inlier}})\eta_0$ with $\eta_0 = 10.0$, where N_p^{inlier} and N_g^{inlier} denote the numbers of inliers in the photometric and geometric cost functions, respectively. The enforcing factor was formulated as an exponential decay function of κ, rather than as a constant value: $\lambda(\kappa) = \exp(\beta(\kappa-1)^2)$ with $\beta = -2.0$. For the weighting functions with respect to the error e, we used the Charbonnier penalty function for the photometric cost function: $w_1(e) = (e^2+\epsilon^2)^{-0.5}$ with $\epsilon = 0.001$, and Tukey's biweight function for the geometric energy function: $w_2(e) = (\max(0, 1-(e/c)^2))^2$ with $c = 4.6851$.

In the coarse-level optimization, the weights for the regularization term were determined based on the diagonal blocks of the Hessian matrix $\boldsymbol{H}$, derived from the first term (the photometric cost function) in Eq. (37), corresponding to the translational and rotational components: $\alpha_\omega = (\max_{i\in\Omega_\omega}|\boldsymbol{H}_{ii}|/\max_{j\in\Omega_v}|\boldsymbol{H}_{jj}|)\alpha_0$ and $\alpha_v = \alpha_0$ with $\alpha_0 = 5.0$. For the iterative optimization, we set the maximum number of iterations to three at each coarse image level and 50 at the finest image level. The optimization at each coarse image level was performed three times before propagating the estimated pose to the next image level.

For computational efficiency, we adopted adaptive sampling strategies for the photometric and geometric energy functions. In the photometric energy function, we evaluated points within the object region using a regular grid. The grid interval was adjusted based on the extent of the object region, which varies with camera motion. We excluded points with low intensity gradients and those located on occluding contours with high depth gradients. In the geometric energy function, we uniformly sampled the edges of the projected 3D object model and constructed search lines of a certain length at each sampled edge point. The sampling interval and search-line length were adjusted at each iteration. In our experiments, we observed that these adaptive schemes not only reduced computational costs but also improved tracking performance.

In addition, pixel-wise depth, surface normal, and object silhouette data were obtained from the 3D object model via the GPU rendering pipeline for each frame. Apart from this, all other processing was implemented solely on the CPU using multithreaded operations, without relying on GPU-specialized acceleration.

6.2. Dataset

Based on the related work discussed in Table 1, we observed that the majority of the key methods (17 of the 32 methods, including current top-performing ones [12–15,17,18,24,27,29–32,34–38]) reported performance results on the RBOT dataset [27]. To ensure fair and

direct comparisons with SOTA methods, we therefore conducted all performance evaluations using the RBOT dataset.

The RBOT dataset consists of 18 object models with varying textures and shape complexities, including 12 object models from the LINEMOD dataset [53], five object models from the Rigid Pose dataset [54], and one custom [27] (see Table 7 for the characteristics of each object, classified by [13,27]). For image sequences, each textured 3D object model was rendered and composited onto real image sequences captured by a hand-held camera moving arbitrarily in a highly cluttered real scene. Four semi-synthetic RGB image sequences, each comprising 1001 frames (at a resolution of 640 × 512), were generated for each object model across four scene conditions: Regular, Dynamic Light, Noisy (with Dynamic Light), and Occlusion (with Dynamic Light). The ground-truth 6DoF poses follow the same predefined trajectories across all frames.

6.3. Performance metrics

Tracking performance on the RBOT dataset was evaluated by measuring the tracking success rate (SR) [27]:

$$\mathrm{SR} = \frac{N_{\mathrm{success}}}{N_{\mathrm{total}}} \times 100\%, \tag{39}$$

where N_{success} is the number of image frames when tracking was successful, and N_{total} is the total number of image frames. In our evaluations, we considered tracking successful when both the translational error $e(\boldsymbol{t})$ and the rotational error $e(\boldsymbol{R})$ satisfied the following conditions:

$$\begin{aligned} e(\boldsymbol{t}) &= \|\boldsymbol{t} - \boldsymbol{t}_{\mathrm{gt}}\|_2 < 5\,\mathrm{cm}, \\ e(\boldsymbol{R}) &= \cos^{-1}\left(\frac{\mathrm{tr}(\boldsymbol{R}^{\top}\boldsymbol{R}_{\mathrm{gt}}) - 1}{2}\right) < 5^{\circ}, \end{aligned} \tag{40}$$

where gt denotes the ground truth data, provided by the RBOT dataset. In addition, we initialized tracking using the ground-truth pose of the first frame in each sequence. When tracking failed under the conditions above, we reinitialized it with the ground-truth pose during evaluations.

6.4. SOTA methods

For our comparative study, we first collected performance results on the RBOT dataset from the key methods discussed in Section 2. We then selected six top-performing methods that achieve average SR scores exceeding 90% across all objects and scenes [15,18,35–38]. In this selection, we excluded the method of [17] because its offline preprocessing stage can provide prior information unavailable to our method and the other baselines, which are primarily tailored for online tracking, thereby making a direct comparison not entirely fair. For additional perspective, we also included two representative baselines: the method of [23], which serves as an earlier direct method, and the method of [27], which has significantly influenced recent methods. Note that due to space constraints, this paper focuses on a comparative study of our method against these six top-performing methods and two representative baselines.[2]

[2] As supplementary material, we provide a complete list of performance results on the RBOT dataset for all other collected methods, available at https://byungkukseo.github.io/mbt3d/.

Table 2
Overall performance (average SR scores (%) across all objects and scenes). **The best** results are highlighted in bold, and the second-best results are underlined.

Method	Constraint[a]	Avg. SR
Zhong and Zhang [23]	P, S	73.9
Tjaden et al. [27]	S	72.8
Stoiber et al. [35]	S	90.9
Tian et al. [15]	G	92.2
Wang et al. [37]	S	91.6
Song et al. [36]	S, P	92.4
Lv et al. [38]	S, G	92.5
Liu et al. [18]	G	92.0
Ours	P, G	**93.9**

[a] The constraints are denoted by G (Geometric), S (Statistical), and P (Photometric).

6.5. Results and discussion

We evaluated our method on the RBOT dataset — comprising 18 objects, four scenes, with each object–scene sequence containing 1000 frames (the first frame was used solely for initialization) — using a standard desktop equipped with an Intel Core i9 CPU running at 3.50 GHz. Performance results for the selected methods were taken from their original papers, except for [23], whose results were sourced from [12]. For [15,18], we used the results reported under the regular tracking condition, which uses a frame interval of $S = 1$, to ensure consistency with the other results. Note that our results on the RBOT dataset and those of all key methods compared in this paper are based on evaluations conducted under the same protocol introduced by [27], including its initialization and failure recovery, to ensure a fair comparison.

In the following subsections, the performance results and analyses are based on the detailed overall performance presented in Table 7. Note that the results of [18,36,37] were excluded from the object-wise performance analyses, including the corresponding tables and figures, because their original papers do not provide complete per-object results on the RBOT dataset.

6.5.1. Overall performance

Table 2 summarizes the average SR scores across all objects and scenes. The results demonstrated that our method significantly outperformed the methods of [23] (20.0%) and [27] (21.1%); achieved improvements over the methods of [35] (3.0%) and [37] (2.3%); and was highly competitive with four top-performing methods [15,18,36,38] (differences within 2.0%). In addition, we observed that the geometric constraint remained beneficial for improving tracking performance (all four methods using this constraint achieved average SR scores of at least 92%), while the statistical constraint remained the most widely adopted (six of the nine methods).

6.5.2. Object-wise performance

As shown in Fig. 3 and Table 3, our method achieved performance improvements for Baking Soda, Bench Vise, Camera, Can, Cube, Egg Box, Glue, Koala Candy, Lamp, and Phone in terms of the average SR scores for each object across all scenes. For Clown and Squirrel, the average SR scores were comparable to those of the first-ranked method for each object, with only marginal score differences (less than 1%). In particular, our method significantly outperformed the other methods, achieving improvements of 12.3% for Baking Soda and 7.9% for Cube over the second-ranked method for each object (see Fig. 3).

Considering the object characteristics presented in Table 7, we observed no clear dependence of our method's performance on these characteristics. However, our method performed relatively well on weakly-textured and less-distinct objects (Camera, Can, Egg Box, Lamp, and Phone). Meanwhile, our method exhibited relatively large performance gaps for highly-distinctive objects (Ape, Broccoli Soup, and Duck), compared with the highest-ranked method for each object. Although our method achieved the highest scores for Baking Soda and Glue, their average SR scores did not reach 90% (see Section 6.5.4 for additional analyses).

Table 3
Object-wise performance (average SR scores (%) for each object across all scenes). **The best** results are highlighted in bold, and the second-best results are underlined.

Method/Object[a]	Ape	Soda	Vise	Soup	Came.	Can	Cat	Clow.	Cube	Dril.	Duck	Egg	Glue	Iron	Cand.	Lamp	Phon.	Squi.
Zhong and Zhang [23]	80.7	38.0	88.0	83.6	74.6	77.2	96.7	83.5	68.1	81.9	81.9	61.4	43.4	72.7	48.9	81.4	74.0	93.9
Tjaden et al. [27]	81.9	42.1	95.3	80.0	73.0	74.2	94.1	84.4	62.2	84.7	85.6	57.5	35.3	67.4	42.8	85.1	68.8	95.4
Stoiber et al. [35]	97.6	64.8	98.3	95.8	93.8	91.1	**100**	96.2	85.7	94.7	97.1	87.2	72.0	92.6	86.9	90.9	92.2	99.8
Tian et al. [15]	**99.5**	65.5	98.9	**97.7**	96.0	87.3	**100**	**97.4**	85.9	97.4	**99.2**	89.1	77.3	95.3	87.8	91.2	93.6	**100**
Lv et al. [38]	97.7	52.0	98.5	97.1	97.4	94.7	99.9	96.9	88.4	**97.6**	96.8	93.2	78.7	**95.4**	92.2	92.3	**95.4**	99.9
Ours	92.1	**77.8**	**99.2**	92.7	**97.9**	**96.2**	98.8	97.3	**96.3**	95.7	90.2	**97.0**	**79.0**	94.1	**94.8**	**95.6**	**95.4**	99.6

[a] Several object names are shortened for brevity: Soda (Baking Soda), Vise (Bench Vise), Soup (Broccoli Soup), Came. (Camera), Clow. (Clown), Dril. (Driller), Egg (Egg Box), Cand. (Koala Candy), Phon. (Phone), and Squi. (Squirrel).

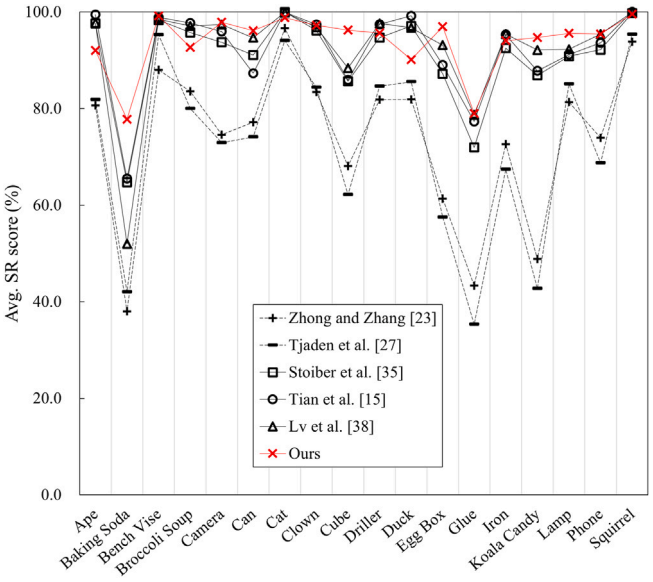


Fig. 3. Average SR scores (%) for each object across all scenes (see Table 3 for detailed SR scores).

6.5.3. Scene-wise performance

Regarding the average SR scores for each scene across all objects (see Table 4), our method achieved average SR scores exceeding 90% in all scenes and attained the highest scores in the Regular, Dynamic Light, and Noisy (with Dynamic Light) scenes. In particular, as shown in Fig. 4, our method demonstrated substantial improvements in the Noisy (with Dynamic Light) scene, which was severely challenging for the other methods (see Appendix B for additional figures covering the other scenes).

On the other hand, we observed that tracking performance gradually degraded as scene complexity increased (96.7% > 95.5% > 91.4% ≈ 91.8%), likely due to the increased difficulty in constraining the energy function. Since our method does not employ specialized occlusion-handling strategies, such as those presented in [27,35,38], its average SR score in the Occlusion (with Dynamic Light) scene was relatively low. Nevertheless, our method demonstrated robustness to a certain degree of partial occlusion.

6.5.4. Failure analysis

We further analyzed the pose errors for Baking Soda and Glue, whose average SR scores across all scenes were below 90%, by examining the number of image frames and the associated translational and rotational errors when tracking failed. As shown in Table 5, we observed that tracking failures for both objects were primarily driven by rotational rather than translational errors. We attribute this behavior to their symmetric shapes, which introduce significant rotational ambiguities and fail to fully constrain the rotation parameters during optimization (as also discussed in [15,35]). We also found a higher rate of translational errors for Glue because of its complex characteristic (symmetric shape combined with weak texture and low distinctiveness) that make gradient-based optimization particularly challenging.

Meanwhile, regarding the cumulative percentages of rotational errors (see Fig. 5), we noticed that the majority of rotational errors for Baking Soda (51.81%) and Glue (40.65%) were less than 7°, suggesting that significant performance improvements could be achieved by incorporating dedicated optimization in the rotation parameter space [15].

Table 4
Scene-wise performance (average SR scores (%) for each scene across all objects). **The best** results are highlighted in bold, and the second-best results are underlined.

Method/Scene[a]	R	D	N+D	O+D
Zhong and Zhang [23]	80.8	79.0	61.6	74.1
Tjaden et al. [27]	79.9	81.2	56.6	73.3
Stoiber et al. [35]	94.2	94.6	81.7	93.2
Tian et al. [15]	95.2	95.4	83.2	**94.9**
Wang et al. [37]	94.6	95.2	83.1	93.6
Song et al. [36]	95.4	94.9	86.2	93.2
Lv et al. [38]	94.7	95.0	86.0	94.1
Liu et al. [18]	94.7	95.2	83.4	94.5
Ours	**96.7**	**95.5**	**91.4**	91.8

[a] The scenes are denoted by R (Regular), D (Dynamic Light), N+D (Noisy with Dynamic Light), and O+D (Occlusion with Dynamic Light).

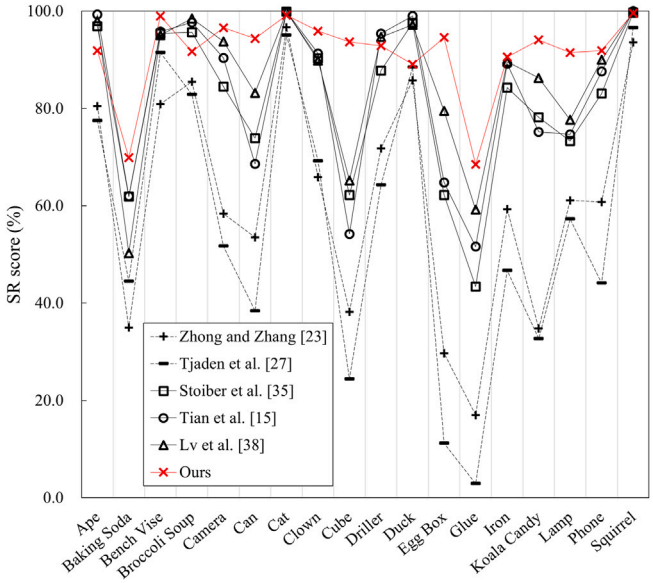


Fig. 4. SR scores (%) for each object in the Noisy (with Dynamic Light) scene (see Table 7 for detailed SR scores).

6.5.5. Runtime

We evaluated the average runtimes across all objects and scenes. Overall, the results reflected an inherent trade-off between estimation robustness and computational efficiency. Specifically, the average

Table 5
Number of frames in which tracking failed (N_{fail}) for Baking Soda and Glue in each scene (fail_t: $e(t) \geq 5$ cm, fail_R: $e(\boldsymbol{R}) \geq 5°$, $\text{fail}_{t,R}$: both).

Object	Scene[a]	N_{fail_t}	N_{fail_R}	$N_{\text{fail}_{t,R}}$
Baking Soda	R	0	126	6
	D	1	210	3
	N+D	0	294	7
	O+D	1	214	26
Glue	R	5	85	22
	D	7	125	44
	N+D	18	206	91
	O+D	8	163	66

[a] The scenes are denoted by R (Regular), D (Dynamic Light), N+D (Noisy with Dynamic Light), and O+D (Occlusion with Dynamic Light).

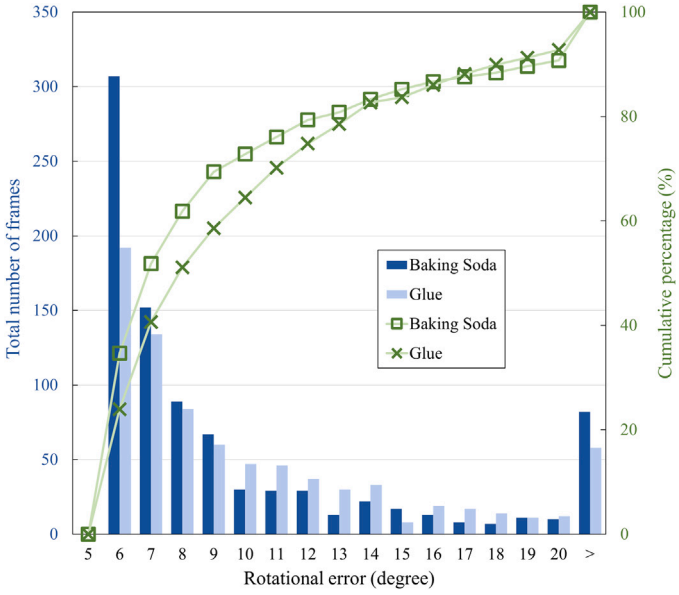


Fig. 5. Cumulative percentages of rotational errors with respect to the total number of frames in which tracking failed for Baking Soda and Glue across all scenes (4 × 1000 frames).

runtimes ranged from 38.6 to 63.1 ms per frame for each object across all scenes, and from 46.4 to 58.3 ms per frame for each scene across all objects. Our method also exhibited computational bottlenecks for relatively large objects (Driller, Iron, and Lamp), requiring approximately 60 ms per frame, likely due to increased sampling density and more expensive energy evaluations associated with the larger object extents. However, we believe that the runtime could be further improved through efficient sampling strategies [17] or by taking advantage of precomputation stages [15,35], without relying on GPU-specialized acceleration. This approach is highly advantageous — and often strictly required — for real-time tracking, as it avoids occupying GPU resources for computation and incurring frequent CPU–GPU data transfer overhead, particularly on resource-constrained AR/XR devices.

6.5.6. Ablation study

To validate the contribution of each component in our framework, we conducted a comprehensive ablation study, as shown in Table 6. Our hierarchical optimization had a significant impact on overall performance, yielding an improvement of nearly 15%. The geometric constraint effectively complemented the photometric constraint within this hierarchical optimization, demonstrating that relying solely on the photometric cost is insufficient for robust tracking. Although the second-order approximation yielded a comparatively modest performance gain, it provided the final contribution needed to achieve the best performance. Even without this component, our method remained highly competitive, ranking third overall with a marginal score deficit

Table 6
Ablation study (average SR scores (%) across all objects and scenes).

Method	Avg. SR
w/o Hierarchical optimization	79.0
w/o Geometric constraint	87.2
w/o Second-order approximation	91.9
w/o Hierarchical optimization + *w/o* Geometric constraint	48.2
w/o Hierarchical optimization + *w/o* Second-order approximation	67.9
w/o Geometric constraint + *w/o* Second-order approximation	83.4
Ours	93.9

of less than 1% relative to the two highest-ranked methods (see Table 2). Overall, the results demonstrated that our hierarchical optimization served as the critical backbone, while the geometric constraint and second-order approximation provided complementary regularization and fine-grained optimization, ultimately resulting in an overall score of 93.9%.

6.5.7. Limitation

In the evaluations, our method achieved performance improvements over current top-performing methods, but it still faces several limitations. The photometric cost function, derived from an intensity variation model, assumes Lambertian surfaces, which limits its effectiveness for objects with specular or transparent surfaces. Under frequent or severe illumination changes, this function may also struggle to maintain the brightness constancy assumption. While the hierarchical optimization strategy can handle moderate inter-frame displacements, it tends to fail under rapid or drastic camera motion. More broadly, like most existing monocular methods, our method may be limited when handling symmetric 3D objects, which introduce rotational ambiguities and hinder accurate 6DoF pose estimation, as well as under heavy occlusion, where insufficient visible information can further degrade tracking performance. Although addressing these challenges typically requires problem-specific strategies, we believe that our method can serve as a foundation for monocular model-based 3D object tracking and can be readily integrated with such strategies to mitigate these limitations.

7. Conclusion

In this paper, we present a novel energy function for direct methods in monocular model-based 3D object tracking, in which photometric and geometric constraints are complementarily combined and hierarchically optimized using stage-tailored energy functions. We also introduce an efficient second-order approximation of the photometric cost function, enhancing the optimization process. Experimental results and comparisons on the RBOT dataset demonstrate the strong competitiveness of our method compared to current top-performing methods. An ablation study further validates the effectiveness of our contributions.

In this work, our findings underscore the promise of direct methods for monocular model-based 3D tracking, establishing a new foundation for follow-up studies and extensions. Looking ahead, our future research will explore the tight coupling of statistical and photometric constraints, moving beyond conventional fusion strategies. A more compelling direction involves leveraging vision foundation models to provide rich structural priors, such as object detection and segmentation, as well as reliable temporal priors, including feature matching and tracking. We expect that this strategy could be more effective and practical than fully end-to-end 6DoF pose estimation pipelines, particularly in scenarios that demand both high accuracy and real-time performance.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Table 7
Detailed overall performance (SR score (%) for each object and scene). **The best** results are highlighted in bold, and the second-best results are underlined.



Characteristic / Object[a]	Ape	Soda	Vise	Soup	Came.	Can	Cat	Clow.	Cube	Dril.	Duck	Egg	Glue	Iron	Cand.	Lamp	Phon.	Squi.
well-textured [27]		★		★				★	★						★			
weakly-textured [27]	★		★		★	★	★			★	★	★	★	★		★	★	★
very-distinct [13]	★	★	★	★			★	★			★			★	★			★
less-distinct [13]					★	★			★	★		★	★			★	★	

[a] Several object names are shortened for brevity: Soda (Baking Soda), Vise (Bench Vise), Soup (Broccoli Soup), Came. (Camera), Clow. (Clown), Dril. (Driller), Egg (Egg Box), Cand. (Koala Candy), Phon. (Phone), and Squi. (Squirrel).
[b] The figure was adapted from [27].

Method / Scene																		
								Regular										
Zhong and Zhang [23]	82.6	40.1	92.6	85.0	82.8	87.2	98.0	92.9	81.3	84.5	83.3	76.2	56.1	84.6	57.6	90.5	82.6	95.6
Tjaden et al. [27]	85.0	39.0	98.9	82.4	79.7	87.6	95.9	93.3	78.1	93.0	86.8	74.6	38.9	81.0	46.8	97.5	80.7	99.4
Stoiber et al. [35]	98.8	65.1	99.6	96.0	98.0	96.5	**100**	98.4	94.1	96.9	98.0	95.3	79.3	96.0	90.3	97.4	96.2	99.8
Tian et al. [15]	**99.8**	67.1	**100**	**97.8**	97.3	93.7	**100**	99.4	**97.4**	97.6	**99.3**	96.9	84.7	**97.7**	93.4	96.7	95.4	**100**
Lv et al. [38]	97.7	54.2	99.6	96.9	99.1	**98.7**	**100**	**99.5**	95.6	**98.9**	96.9	96.9	83.0	97.1	95.3	97.6	97.6	99.8
Ours	94.0	**86.8**	99.8	95.9	**99.6**	98.4	99.3	99.0	**98.3**	97.4	96.2	**98.8**	**88.8**	97.5	**95.5**	**98.3**	**98.1**	99.6
Method / Scene								Dynamic Light										
Zhong and Zhang [23]	81.8	39.7	91.5	85.1	82.6	87.1	98.1	90.7	79.7	87.4	81.6	73.1	51.7	75.9	53.4	88.8	78.6	95.6
Tjaden et al. [27]	84.9	42.0	99.0	81.3	84.3	88.9	95.6	92.5	77.5	94.6	86.4	77.3	52.9	77.9	47.9	96.9	81.7	99.3
Stoiber et al. [35]	98.2	65.2	99.2	95.6	97.5	98.1	**100**	98.5	94.2	97.5	97.9	96.9	86.1	95.2	89.3	97.0	95.9	99.9
Tian et al. [15]	**100**	64.5	**99.8**	**97.9**	97.9	94.0	**100**	**99.5**	97.0	98.8	**99.3**	97.6	87.5	97.4	92.4	97.1	96.4	**100**
Lv et al. [38]	98.5	49.7	**99.8**	97.1	98.6	**98.8**	**100**	99.2	97.1	**98.9**	97.9	98.4	**87.8**	**97.9**	94.4	**97.9**	**97.8**	**100**
Ours	94.0	**78.6**	**99.8**	94.3	**99.5**	97.5	**100**	99.3	**97.7**	98.1	90.7	**99.1**	82.4	95.9	**96.7**	97.6	**97.8**	99.9
Method / Scene								Noisy (with Dynamic Light)										
Zhong and Zhang [23]	80.5	35.0	80.9	85.5	58.4	53.5	96.7	65.9	38.2	71.8	85.8	29.7	17.0	59.3	34.8	61.1	60.8	93.6
Tjaden et al. [27]	77.5	44.5	91.5	82.9	51.7	38.4	95.1	69.2	24.4	64.3	88.5	11.2	2.90	46.7	32.7	57.3	44.1	96.6
Stoiber et al. [35]	96.9	61.9	95.4	95.7	84.5	73.9	**99.9**	90.3	62.2	87.8	97.6	62.2	43.4	84.3	78.2	73.3	83.1	99.7
Tian et al. [15]	**99.3**	62.0	95.8	97.7	90.4	68.6	**99.9**	91.3	54.2	**95.4**	**99.0**	64.8	51.6	89.2	75.2	74.7	87.6	**100**
Lv et al. [38]	98.0	50.3	95.1	**98.5**	93.8	83.2	**99.9**	89.9	65.2	94.7	97.2	79.5	59.2	89.6	86.3	77.7	90.0	99.9
Ours	91.9	**69.9**	**99.0**	91.7	**96.6**	**94.4**	99.2	**95.9**	**93.7**	92.9	89.1	**94.6**	**68.5**	**90.6**	**94.1**	**91.5**	**91.9**	99.6
Method / Scene								Occlusion (with Dynamic Light)										
Zhong and Zhang [23]	77.7	37.3	87.1	78.7	74.6	81.0	93.8	84.3	73.2	83.7	77.0	66.4	48.6	70.8	49.6	85.0	73.8	90.6
Tjaden et al. [27]	80.0	42.7	91.8	73.5	76.1	81.7	89.8	82.6	68.7	86.7	80.5	67.0	46.6	64.0	43.6	88.8	68.6	86.2
Stoiber et al. [35]	96.5	66.8	99.0	95.8	95.0	95.9	**100**	97.6	92.2	96.6	95.0	94.4	79.0	94.7	89.8	95.7	93.6	99.6
Tian et al. [15]	**98.7**	68.4	**99.9**	**97.5**	**98.3**	93.0	99.9	**99.4**	95.1	97.9	**99.1**	96.9	**85.5**	97.0	90.3	**96.3**	95.1	**100**
Lv et al. [38]	96.7	53.8	99.4	96.0	98.1	**98.0**	99.7	98.8	**95.7**	**98.0**	95.0	**97.8**	84.8	**97.1**	92.6	96.0	**96.3**	99.9
Ours	88.4	**75.9**	98.2	89.0	95.8	94.3	96.7	94.8	95.3	94.3	84.6	95.4	76.3	92.3	**92.7**	95.0	93.9	99.1

Acknowledgments

This work was supported by the Institute for Information & Communications Technology Planning & Evaluation (IITP) grant funded by the Korean government, Ministry of Science and ICT (MSIT) (No. 2021-0-00320, Technology Development of Creation, Manipulation and Augmentation for XR in the Real-World Environment). It was also supported by the Culture, Sports and Tourism R&D Program through the Korea Creative Content Agency (KOCCA) grant funded by the Korean government, Ministry of Culture, Sports and Tourism (MCST) (No. RS-2024-00341886, Development of Video Synthesis and Editing Technology for Linking Neural Assets-Based Video Production Pipeline).

Appendix A. Approximation of the second-order derivative

To approximate the second-order derivative, we start by recalling Eq. (27):

$$I_{t+1}(\mathcal{W}(\mathcal{W}(\boldsymbol{p};\Delta\boldsymbol{\xi});\boldsymbol{\xi}^{(k)})) \approx I_{t+1}(\mathcal{W}(\boldsymbol{p};\boldsymbol{\xi}^{(k)})) + \frac{\partial I_{t+1}(\mathcal{W}(\mathcal{W}(\boldsymbol{p};\boldsymbol{\xi});\boldsymbol{\xi}^{(k)}))}{\partial\boldsymbol{\xi}}\bigg|_{\boldsymbol{\xi}=\mathbf{0}}\Delta\boldsymbol{\xi} + \frac{1}{2}\Delta\boldsymbol{\xi}^{\top}\frac{\partial^2 I_{t+1}(\mathcal{W}(\mathcal{W}(\boldsymbol{p};\boldsymbol{\xi});\boldsymbol{\xi}^{(k)}))}{\partial\boldsymbol{\xi}^{\top}\partial\boldsymbol{\xi}}\bigg|_{\boldsymbol{\xi}=\mathbf{0}}\Delta\boldsymbol{\xi}. \tag{A.1}$$

Assuming that $\Delta\boldsymbol{\xi}$ is the increment such that $I_{t+1}(\mathcal{W}(\mathcal{W}(\boldsymbol{p};\Delta\boldsymbol{\xi});\boldsymbol{\xi}^{(k)})) = I_t(\boldsymbol{p})$, we obtain

$$I_t(\boldsymbol{p}) \approx I_{t+1}(\mathcal{W}(\boldsymbol{p};\boldsymbol{\xi}^{(k)})) + \frac{\partial I_{t+1}(\mathcal{W}(\mathcal{W}(\boldsymbol{p};\boldsymbol{\xi});\boldsymbol{\xi}^{(k)}))}{\partial\boldsymbol{\xi}}\bigg|_{\boldsymbol{\xi}=\mathbf{0}}\Delta\boldsymbol{\xi} + \frac{1}{2}\Delta\boldsymbol{\xi}^{\top}\frac{\partial^2 I_{t+1}(\mathcal{W}(\mathcal{W}(\boldsymbol{p};\boldsymbol{\xi});\boldsymbol{\xi}^{(k)}))}{\partial\boldsymbol{\xi}^{\top}\partial\boldsymbol{\xi}}\bigg|_{\boldsymbol{\xi}=\mathbf{0}}\Delta\boldsymbol{\xi}, \tag{A.2}$$

and since $\mathcal{W}(\boldsymbol{p};\mathbf{0}) = \boldsymbol{p}$, it can be rewritten as

$$I_t(\mathcal{W}(\boldsymbol{p};\mathbf{0})) \approx I_{t+1}(\mathcal{W}(\mathcal{W}(\boldsymbol{p};\mathbf{0});\boldsymbol{\xi}^{(k)})) + \frac{\partial I_{t+1}(\mathcal{W}(\mathcal{W}(\boldsymbol{p};\boldsymbol{\xi});\boldsymbol{\xi}^{(k)}))}{\partial\boldsymbol{\xi}}\bigg|_{\boldsymbol{\xi}=\mathbf{0}}\Delta\boldsymbol{\xi} + \frac{1}{2}\Delta\boldsymbol{\xi}^{\top}\frac{\partial^2 I_{t+1}(\mathcal{W}(\mathcal{W}(\boldsymbol{p};\boldsymbol{\xi});\boldsymbol{\xi}^{(k)}))}{\partial\boldsymbol{\xi}^{\top}\partial\boldsymbol{\xi}}\bigg|_{\boldsymbol{\xi}=\mathbf{0}}\Delta\boldsymbol{\xi}. \tag{A.3}$$

By taking derivatives at $\boldsymbol{\xi} = \mathbf{0}$ on both sides and neglecting the higher-order terms, we can therefore obtain the approximation of the second-order derivative, presented in Eq. (28) as follows:

$$\frac{\partial I_{t+1}(\mathcal{W}(\mathcal{W}(\boldsymbol{p};\boldsymbol{\xi});\boldsymbol{\xi}^{(k)}))}{\partial\boldsymbol{\xi}}\bigg|_{\boldsymbol{\xi}=\mathbf{0}} + \frac{\partial^2 I_{t+1}(\mathcal{W}(\mathcal{W}(\boldsymbol{p};\boldsymbol{\xi});\boldsymbol{\xi}^{(k)}))}{\partial\boldsymbol{\xi}^{\top}\partial\boldsymbol{\xi}}\bigg|_{\boldsymbol{\xi}=\mathbf{0}}\Delta\boldsymbol{\xi} \approx \frac{\partial I_t(\mathcal{W}(\boldsymbol{p};\boldsymbol{\xi}))}{\partial\boldsymbol{\xi}}\bigg|_{\boldsymbol{\xi}=\mathbf{0}}, \tag{A.4}$$

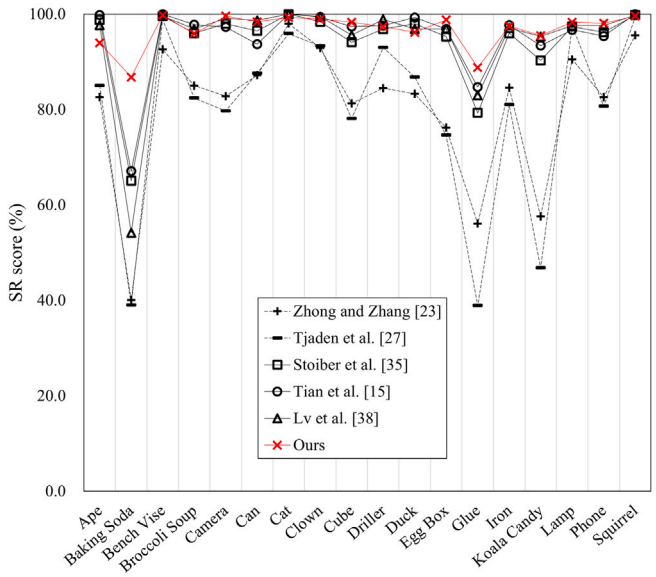


Fig. B.6. SR scores (%) for each object in the Regular scene (see Table 7 for detailed SR scores).

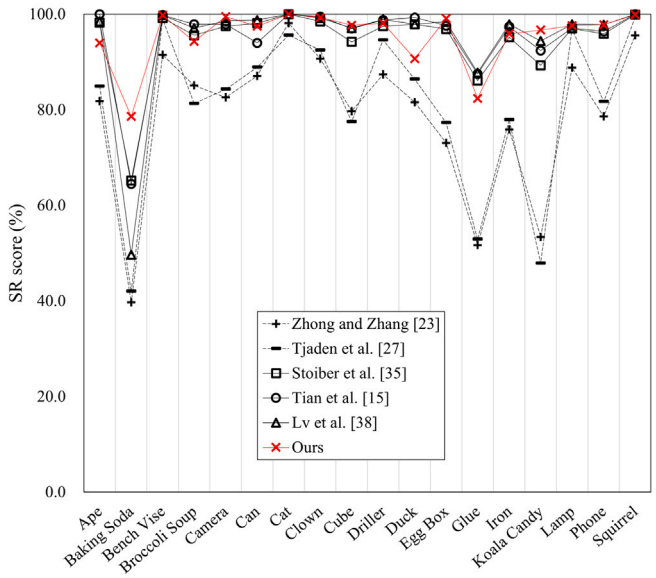


Fig. B.7. SR scores (%) for each object in the Dynamic Light scene (see Table 7 for detailed SR scores).

Appendix B. Additional scene-wise performance

We provide additional performance figures for all objects in the Regular, Dynamic Light, and Occlusion (with Dynamic Light) scenes (see Figs. B.6–B.8).

Appendix C. Parameter sensitivity

We provide experimental results on parameter sensitivity, specifically focusing on the number of bins in the color histogram for each appearance model and the number of pyramid levels (see Fig. C.9).

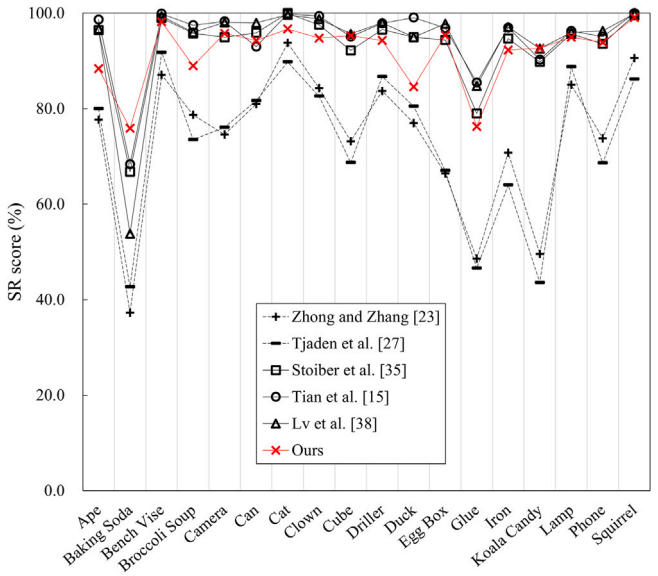


Fig. B.8. SR scores (%) for each object in the Occlusion (with Dynamic Light) scene (see Table 7 for detailed SR scores).

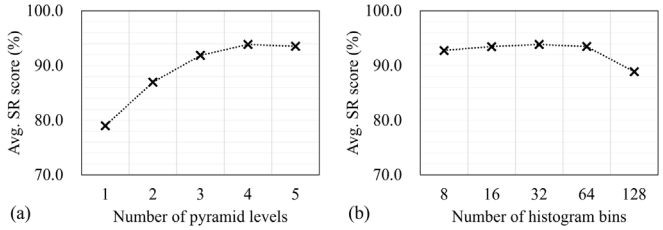


Fig. C.9. Average SR scores (%) across all objects and scenes for (a) different numbers of pyramid levels and (b) different numbers of histogram bins.

Data availability

The data supporting the findings of this study are derived from the publicly available dataset (RBOT dataset) that can be found at: https://www.mi.hs-rm.de/~schwan/research/RBOT/.

References

[1] P. Panteleris, D. Michel, A. Argyros, Toward augmented reality in museums: Evaluation of design choices for 3D object pose estimation, Front. Virtual Real. 2 (2021) 1–10, http://dx.doi.org/10.3389/frvir.2021.649784.

[2] X. Yang, J. Cai, K. Li, X. Fan, H. Cao, A monocular-based tracking framework for industrial augmented reality applications, Int. J. Adv. Manuf. Technol. 128 (5) (2023) 2571–2588, http://dx.doi.org/10.1007/s00170-023-12082-w.

[3] S. Li, H. Schieber, N. Corell, B. Egger, J. Kreimeier, D. Roth, GBOT: Graph-based 3D object tracking for augmented reality-assisted assembly guidance, in: Proceedings of IEEE Conference Virtual Reality and 3D User Interfaces, 2024, pp. 513–523, http://dx.doi.org/10.1109/VR58804.2024.00072.

[4] B.-K. Seo, H.S. Park, Y.S. Kim, AlterVerse: A framework for interactive virtual altering of physical objects in extended reality environments, IEEE Comput. Graph. Appl. 44 (5) (2024) 76–84, http://dx.doi.org/10.1109/MCG.2024.3427188.

[5] N. Iwai, F. Matulic, Bringing everyday objects to life in augmented reality with AI-powered talking characters, in: Proceedings of the Extended Abstracts of the CHI Conference on Human Factors in Computing Systems, 2025, pp. 1–7, http://dx.doi.org/10.1145/3706599.3719978.

[6] V. Lepetit, P. Fua, Monocular model-based 3D tracking of rigid objects: A survey, Foundations and Trends in Computer Graphics and Vision 1 (1) (2005) 1–89, http://dx.doi.org/10.1561/0600000001.

[7] E. Marchand, H. Uchiyama, F. Spindler, Pose estimation for augmented reality: A hands-on survey, IEEE Trans. Vis. Comput. Graphics 22 (12) (2016) 2633–2651, http://dx.doi.org/10.1109/TVCG.2015.2513408.

[8] B.-K. Seo, H. Park, J.-I. Park, S. Hinterstoisser, S. Ilic, Optimal local searching for fast and robust textureless 3D object tracking in highly cluttered backgrounds, IEEE Trans. Vis. Comput. Graphics 20 (1) (2014) 99–110, http://dx.doi.org/10.1109/TVCG.2013.94.

[9] G. Wang, B. Wang, F. Zhong, X. Qin, B. Chen, Global optimal searching for textureless 3D object tracking, Vis. Comput. 31 (2015) 979–988, http://dx.doi.org/10.1007/s00371-015-1098-7.

[10] B. Wang, F. Zhong, X. Qin, Pose optimization in edge distance field for textureless 3D object tracking, in: Proceedings of Computer Graphics International Conference, 2017, pp. 1–6, http://dx.doi.org/10.1145/3095140.3095172.

[11] B. Wang, F. Zhong, X. Qin, Robust edge-based 3D object tracking with direction-based pose validation, Multimedia Tools Appl. 78 (2019) 12307–12331, http://dx.doi.org/10.1007/s11042-018-6727-5.

[12] H. Huang, F. Zhong, Y. Sun, X. Qin, An occlusion-aware edge-based method for monocular 3D object tracking using edge confidence, Comput. Graph. Forum 39 (7) (2020) 399–409, http://dx.doi.org/10.1111/cgf.14154.

[13] X. Sun, J. Zhou, W. Zhang, Z. Wang, Q. Yu, Robust monocular pose tracking of less-distinct objects based on contour-part model, IEEE Trans. Circuits Syst. Video Technol. 31 (11) (2021) 4409–4421, http://dx.doi.org/10.1109/TCSVT.2021.3053696.

[14] Y. Dong, L. Ji, S. Wang, P. Gong, J. Yue, R. Shen, C. Chen, Y. Zhang, Accurate 6DOF pose tracking for texture-less objects, IEEE Trans. Circuits Syst. Video Technol. 31 (5) (2021) 1834–1848, http://dx.doi.org/10.1109/TCSVT.2020.3011737.

[15] X. Tian, X. Lin, F. Zhong, X. Qin, Large-displacement 3D object tracking with hybrid non-local optimization, in: Proceedings of European Conference on Computer Vision, 2022, pp. 627–643, http://dx.doi.org/10.1007/978-3-031-20047-2_36.

[16] H. Tang, C. Liu, Y. Su, Q. Wang, W. Hu, Model-based monocular 6-degree-of-freedom pose tracking for asteroid, Front. Space Technol. 5 (2024) 1–13, http://dx.doi.org/10.3389/frspt.2024.1337262.

[17] J. Chen, J. Chen, K. Liu, H. Chang, S. Fu, J. Yang, Robust 6DoF pose tracking considering contour and interior correspondence uncertainty for AR assembly guidance, IEEE Trans. Instrum. Meas. 74 (2025) 1–16, http://dx.doi.org/10.1109/TIM.2025.3571175.

[18] K. Liu, J. Chen, J. Chen, Y. Li, Y. Wang, Efficient contour-based monocular 3D object tracking with two-stage pose initialization for large shifts, J. Electron. Imaging 34 (4) (2025) 1–23, http://dx.doi.org/10.1117/1.JEI.34.4.043028.

[19] A. Crivellaro, V. Lepetit, Robust 3D tracking with descriptor fields, in: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, 2014, pp. 3414–3421, http://dx.doi.org/10.1109/CVPR.2014.436.

[20] G. Caron, A. Dame, E. Marchand, Direct model based visual tracking and pose estimation using mutual information, Image Vis. Comput. 32 (1) (2014) 54–63, http://dx.doi.org/10.1016/j.imavis.2013.10.007.

[21] B.-K. Seo, H. Wüest, A direct method for robust model-based 3D object tracking from a monocular RGB image, in: Proceedings of European Conference on Computer Vision Workshops, 2016, pp. 551–562, http://dx.doi.org/10.1007/978-3-319-49409-8_48.

[22] L. Zhong, M. Lu, L. Zhang, A direct 3D object tracking method based on dynamic textured model rendering and extended dense feature fields, IEEE Trans. Circuits Syst. Video Technol. 28 (9) (2018) 2302–2315, http://dx.doi.org/10.1109/TCSVT.2017.2731519.

[23] L. Zhong, L. Zhang, A robust monocular 3D object tracking method combining statistical and photometric constraints, Int. J. Comput. Vis. 127 (2019) 973–992, http://dx.doi.org/10.1007/s11263-018-1119-x.

[24] F. Liu, Z. Wei, G. Zhang, An off-board vision system for relative attitude measurement of aircraft, IEEE Trans. Ind. Electron. 69 (4) (2022) 4225–4233, http://dx.doi.org/10.1109/TIE.2021.3075889.

[25] H. Tjaden, U. Schwanecke, E. Schömer, Real-time monocular segmentation and pose tracking of multiple objects, in: Proceedings of European Conference on Computer Vision, 2016, pp. 423–438, http://dx.doi.org/10.1007/978-3-319-46493-0_26.

[26] H. Tjaden, U. Schwanecke, E. Schömer, Real-time monocular pose estimation of 3D objects using temporally consistent local color histograms, in: Proceedings of IEEE International Conference on Computer Vision, 2017, pp. 124–132, http://dx.doi.org/10.1109/ICCV.2017.23.

[27] H. Tjaden, U. Schwanecke, E. Schömer, D. Cremers, A region-based gauss-newton approach to real-time monocular multiple object tracking, IEEE Trans. Pattern Anal. Mach. Intell. 41 (8) (2019) 1797–1812, http://dx.doi.org/10.1109/TPAMI.2018.2884990.

[28] C. Li, X. Gao, Adaptive contour feature and color feature fusion for monocular textureless 3D object tracking, IEEE Access 6 (2018) 30473–30482, http://dx.doi.org/10.1109/ACCESS.2018.2839761.

[29] L. Zhong, X. Zhao, Y. Zhang, S. Zhang, L. Zhang, Occlusion-aware region-based 3D pose tracking of objects with temporally consistent polar-based local partitioning, IEEE Trans. Image Process. 29 (2020) 5065–5078, http://dx.doi.org/10.1109/TIP.2020.2973512.

[30] M. Stoiber, M. Pfanne, K.H. Strobl, R. Triebel, A. Albu-Schäffer, A sparse gaussian approach to region-based 6DoF object tracking, in: Proceedings of Asian Conference on Computer Vision, 2020, pp. 1–16, http://dx.doi.org/10.1007/978-3-030-69532-3_40.

[31] J. Li, X. Song, F. Zhong, X. Qin, Fast 3D texture-less object tracking with geometric contour and local region, Comput. Graph. 97 (2021) 225–235, http://dx.doi.org/10.1016/j.cag.2021.04.012.

[32] J. Li, F. Zhong, S. Xu, X. Qin, 3D object tracking with adaptively weighted local bundles, J. Comput. Sci. Tech. 36 (3) (2021) 555–571, http://dx.doi.org/10.1007/s11390-021-1272-5.

[33] J. Zhou, Z. Wang, Y. Bao, Q. Wang, X. Sun, Q. Yu, Robust monocular 3D object pose tracking for large visual range variation in robotic manipulation via scale-adaptive region-based method, Int. J. Adv. Robotic Syst. 19 (1) (2022) 1–19, http://dx.doi.org/10.1177/17298806221076978.

[34] H. Huang, F. Zhong, X. Qin, Pixel-wise weighted region-based 3D object tracking using contour constraints, IEEE Trans. Vis. Comput. Graphics 28 (12) (2022) 4319–4331, http://dx.doi.org/10.1109/TVCG.2021.3085197.

[35] M. Stoiber, M. Pfanne, K.H. Strobl, R. Triebel, A. Albu-Schäffer, SRT3D: A sparse region-based 3D object tracking approach for the real world, Int. J. Comput. Vis. 130 (4) (2022) 1008–1030, http://dx.doi.org/10.1007/s11263-022-01579-8.

[36] X. Song, W. Xie, J. Li, N. Wang, F. Zhong, G. Zhang, X. Qin, 3D object tracking for rough models, Comput. Graph. Forum 42 (7) (2023) 1–11, http://dx.doi.org/10.1111/cgf.14976.

[37] Q. Wang, J. Zhou, Z. Li, X. Sun, Q. Yu, Robust and accurate monocular pose tracking for large pose shift, IEEE Trans. Ind. Electron. 70 (8) (2023) 8163–8173, http://dx.doi.org/10.1109/TIE.2022.3217598.

[38] N. Lv, D. Zhao, F. Kong, Z. Xu, F. Du, A multi-feature fusion-based pose tracking method for industrial object with visual ambiguities, Adv. Eng. Inform. 62 (2024) 1–16, http://dx.doi.org/10.1016/j.aei.2024.102788.

[39] D. Li, Y. Hu, S. Wang, M. Zhu, F. Yuan, J. Yu, A lightweight integrated positioning system with occlusion-aware region-based pose tracking for oral and maxillofacial surgery, IEEE Trans. Autom. Sci. Eng. 22 (2025) 6839–6850, http://dx.doi.org/10.1109/TASE.2024.3455435.

[40] C. Sahin, G. Garcia-Hernando, J. Sock, T.-K. Kim, A review on object pose recovery: From 3D bounding box detectors to full 6D pose estimators, Image Vis. Comput. 96 (2020) 1–23, http://dx.doi.org/10.1016/j.imavis.2020.103898.

[41] Z. Fan, Y. Zhu, Y. He, Q. Sun, H. Liu, J. He, Deep learning on monocular object pose detection and tracking: A comprehensive overview, ACM Comput. Surv. 55 (4) (2022) 1–40, http://dx.doi.org/10.1145/3524496.

[42] V.A. Prisacariu, I.D. Reid, PWP3D: Real-time segmentation and tracking of 3D objects, Int. J. Comput. Vis. 98 (2012) 335–354, http://dx.doi.org/10.1007/s11263-011-0514-3.

[43] S. Zhao, L. Wang, W. Sui, H.-y. Wu, C. Pan, 3D object tracking via boundary constrained region-based model, in: Proceedings of IEEE International Conference on Image Processing, 2014, pp. 486–490, http://dx.doi.org/10.1109/ICIP.2014.7025097.

[44] J. Hexner, R.R. Hagege, 2D-3D pose estimation of heterogeneous objects using a region based approach, Int. J. Comput. Vis. 118 (2016) 95–112, http://dx.doi.org/10.1007/s11263-015-0873-2.

[45] S. Baker, I. Matthews, Lucas-Kanade 20 years on: A unifying framework, Int. J. Comput. Vis. 56 (2004) 221–255, http://dx.doi.org/10.1023/B:VISI.0000011205.11775.fd.

[46] S. Benhimane, E. Malis, Real-time image-based tracking of planes using efficient second-order minimization, in: Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems, 2004, pp. 943–948, http://dx.doi.org/10.1109/IROS.2004.1389474.

[47] J. Engel, T. Schops, D. Cremers, LSD-SLAM: Large-scale direct monocular SLAM, in: Proceedings of European Conference on Computer Vision, 2014, pp. 834–849, http://dx.doi.org/10.1007/978-3-319-10605-2_54.

[48] S. Negahdaripour, Revised definition of optical flow: Integration of radiometric and geometric cues for dynamic scene analysis, IEEE Trans. Pattern Anal. Mach. Intell. 20 (9) (1998) 961–979, http://dx.doi.org/10.1109/34.713362.

[49] R. Basri, D.W. Jacobs, Lambertian reflectance and linear subspaces, IEEE Trans. Pattern Anal. Mach. Intell. 25 (2) (2003) 218–233, http://dx.doi.org/10.1109/TPAMI.2003.1177153.

[50] C. Bibby, I. Reid, Robust real-time visual tracking using pixel-wise posteriors, in: Proceedings of European Conference on Computer Vision, 2008, pp. 831–844, http://dx.doi.org/10.1007/978-3-540-88688-4_61.

[51] E. Malis, Improving vision-based control using efficient second-order minimization techniques, in: Proceedings of IEEE International Conference on Robotics and Automation, Vol. 2, 2004, pp. 1843–1848, http://dx.doi.org/10.1109/ROBOT.2004.1308092.

[52] Y. Ahmine, G. Caron, E.M. Mouaddib, F. Chouireb, Adaptive Lucas-Kanade tracking, Image Vis. Comput. 88 (2019) 1–8, http://dx.doi.org/10.1016/j.imavis.2019.04.004.

[53] S. Hinterstoisser, V. Lepetit, S. Ilic, S. Holzer, G. Bradski, K. Konolige, N. Navab, Model based training, detection and pose estimation of texture-less 3D objects in heavily cluttered scenes, in: Proceedings of Asian Conference on Computer Vision, 2012, pp. 548–562, http://dx.doi.org/10.1007/978-3-642-37331-2_42.

[54] K. Pauwels, L. Rubio, J. Díaz, E. Ros, Real-time model-based rigid object pose estimation and tracking combining dense and sparse visual cues, in: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, 2013, pp. 2347–2354, http://dx.doi.org/10.1109/CVPR.2013.304.