Optimal Local Searching for Fast and Robust Textureless 3D Object Tracking in Highly Cluttered Backgrounds

Byung-Kuk Seo, *Student Member, IEEE,* Hanhoon Park, *Member, IEEE,* Jong-Il Park, *Member, IEEE,* Stefan Hinterstoisser, *Member, IEEE,* and Slobodan Ilic, *Member, IEEE*

Abstract—Edge-based tracking is a fast and plausible approach for textureless 3D object tracking, but its robustness is still very challenging in highly cluttered backgrounds due to numerous local minima. To overcome this problem, we propose a novel method for fast and robust textureless 3D object tracking in highly cluttered backgrounds. The proposed method is based on optimal local searching of 3D-2D correspondences between a known 3D object model and 2D scene edges in an image with heavy background clutter. In our searching scheme, searching regions are partitioned into three levels (interior, contour, and exterior) with respect to the previous object region, and confident searching directions are determined by evaluating candidates of correspondences on their region levels; thus, the correspondences are searched among likely candidates in only the confident directions instead of searching through all candidates. To ensure the confident searching direction, we also adopt the region appearance, which is efficiently modeled on a newly defined local space (called a searching bundle). Experimental results and performance evaluations demonstrate that our method fully supports fast and robust textureless 3D object tracking even in highly cluttered backgrounds.

Index Terms—Edge-based tracking, model-based tracking, background clutter, local searching, region knowledge

1 Introduction

Model-based tracking has been widely used for 3D visual tracking and servoing tasks in computer vision, robotics, and augmented reality. In model-based tracking, a 3D model of a target object is used for estimating six degrees of freedom (6DOF) camera poses (positions and orientations) relative to the object [1]. In general, a 3D model can be readily obtained by range scans or multi-view reconstructions online/offline. The camera poses are estimated using 3D-2D correspondences between the 3D model and its corresponding 2D scene observation in the image.

Similar to feature-based tracking, 3D objects with dense textures are advantageous for model-based tracking because the 3D-2D correspondences are explicitly established by feature points [2], [3], [4] or templates [5], [6], [7]. On the other hand, strong edges of a target object are great potential cues, particularly when texture information is not sufficient or available for the object. In edge-based tracking, a 3D object model is projected on an image and matched with its corresponding 2D scene edges in the image. Then 3D camera motions between consecutive frames are recovered from 2D displacements of the correspondences. Since the RAPID tracker was proposed [8], edge-based tracking has been well-established [9], [10] and has steadily been improved [11], [12], [13], [14]. Though edge-based tracking is fast and plausible, numerous errors are commonly caused by either background clutter or object clutter, as shown in Fig. 1. In this paper, we explore the critical problem of edge-based tracking when a textureless 3D object is in a highly cluttered background. In practice, fusion approaches using multiple visual cues [11], [13], [14], [15] or additional sensors [16], [17], [18] can be expected for robust tracking, but in many cases, they are confronted with expensive tasks for achieving real-time performance, particularly on low-power embedded platforms like mobile phones. Moreover, all the necessary information is not always readily available in common environments. Given the limited information such that scene edges are only available in a monocular RGB camera view, therefore, fast and robust tracking of textureless 3D objects even in highly cluttered backgrounds is of great importance.

To handle background clutter in edge-based tracking, many people have adopted robust estimators in a registration process [9], [10], [11], [12]. Multiple edge hypotheses can also be considered with mixture models in the estimators [11], [12]. However, false matches

- *B.-K. Seo and J.-I. Park are with the Department of Electronics and Computer Engineering, Hanyang University, Seoul 133791, R. Korea. E-mail: bkseo@mr.hanyang.ac.kr, jipark@hanyang.ac.kr*
- *H. Park is with the Department of Electronic Engineering, Pukyong National University, Busan 608737, R. Korea. E-mail: hanhoon_park@pknu.ac.kr*
- *S. Hinterstoisser and S. Ilic are with the Department of Computer Aided Medical Procedures (CAMP), Technische Universität München, Garching bei München, Germany, 85478. E-mail:{hinterst, Slobodan.Ilic}@in.tum.de*

Digital Object Indentifier 10.1109/TVCG.2013.94

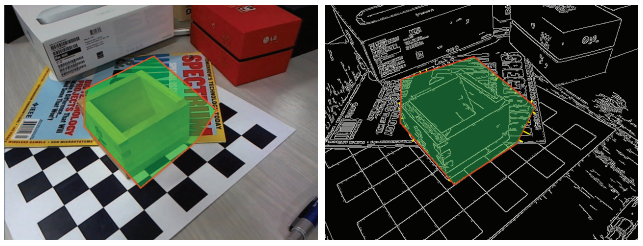


Fig. 1. Critical problem of edge-based tracking in a highly cluttered background. **Left**: 3D object model projected on a target object at a t frame with a previous camera pose ($t-1$ frame), **Right**: Incorrect camera pose due to false matches (local minima).

are unavoidable in highly cluttered scenes due to many candidates that have very similar residual errors with correct correspondences. Instead of finding a single edge hypothesis, multiple pose hypotheses can be employed in a Bayesian manner, but their computational costs are still very high in challenging scenes, despite some prominent improvements using a graphics processing unit (GPU) [19], [20], [21]. It is also difficult to accurately estimate camera poses when their distributions are multi-modal. To overcome these problems, we propose optimal local searching of 3D-2D correspondences where their searching directions are constrained by previous region knowledge. In our searching scheme, candidates of correspondences are evaluated on their region levels and region appearance, and it makes it possible to search for them among likely candidates in only confident searching directions. Moreover, this searching process is efficiently represented and handled in a searching bundle, which is a set of 1D searching regions. Our method is inspired by region-based approaches [22], but the main difference is that local region knowledge is exploited for reliably establishing 3D-2D correspondences in the edge-based approach, not for region segmentation.

Lots of methods have been proposed for dealing with false matches between 3D-2D correspondences in the literature. Our main challenge is to accomplish fast and robust tracking of textureless 3D objects in the presence of heavy background clutter using only a single cue (edge); thus, we highlight its relevant works.

The primary interest is how to establish 3D-2D correspondences and handle their false matches in edge-based tracking. In general, edge-based tracking searches for strong gradient responses on a 1D line along the normal of a projected 3D object model to find correspondences at sample points. Drummond and Cipolla [9] searched for the nearest intensity discontinuity above a certain threshold. Marchand et al. [23] computed the largest maximum gradient above a certain threshold within a certain search range using precomputed filter masks of contour orientations. Instead of precomputed filter masks, Wuest et al. [12] used a 1D mask along a searching line with a 2D anisotropic Gaussian mask perpendicular to the searching line. However, these searching schemes are very susceptible to heavy background clutter in scenes despite the use of robust estimators. In a pose estimation process, on the other hand, multiple edge hypotheses have been considered rather than attempting to find a single edge hypothesis. Vacchetti et al. [11] greatly improved the robustness of edge-based tracking using a multiple hypotheses robust estimator even though they combined edges with texture information. Similarly, Wuest et al. [12] used a multiple hypotheses robust estimator with a Gaussian mixture model while maintaining the visual properties of the previous edges. However, these approaches have difficulty when the outliers are close to the correct correspondences because they still maintain a single edge hypothesis on the camera pose.

As high-dimensional statistics, Bayesian approaches have been effective for avoiding undesirable errors due to background clutter. Since camera poses are predicted from probabilistic distributions without direct estimation using 3D-2D correspondences, in theses approaches, the overall tracking performance is less sensitive to individual false matches. Yoon et al. [24] presented a prediction-verification framework based on the extended Kalman filter, where the first predicted matches are verified by backtracking to avoid false matches. Pupilli and Calway [20] proposed a particle filter observation model based on minimal edge junctions for achieving real-time 3D tracking in dense cluttered scenes. Klein and Murray [19] demonstrated a full 3D edge tracker based on a particle filter, which is accelerated using a GPU. Teuliére et al. [25] presented a particle filtering framework that uses high potential particle sets constrained from low-level multiple edge hypotheses. Choi and Christensen [15] employed a first-order autoregressive state dynamics on the SE(3) group for improving the performance of the particle filter-based 3D tracker. In the Bayesian approaches, however, the computational cost is usually too high for reliable tracking because larger state spaces are needed in more complex scenes.

Region-based approaches can also be of interest in terms of 6DOF camera pose estimation using region knowledge. Several outstanding works based on level set region segmentation have been demonstrated for robust 3D object tracking [22], [26], [27], [28]. These approaches follow a general statistical representation of a level set function and evolve a contour of a 3D object model over the camera pose. In principle, such region segmentation is a very intensive task because the contour is evolved in an infinite-dimensional space, and it can also be difficult to guarantee good segmentation results according to scene complexity [26]. However, some approaches have substantially been improved using direct min-

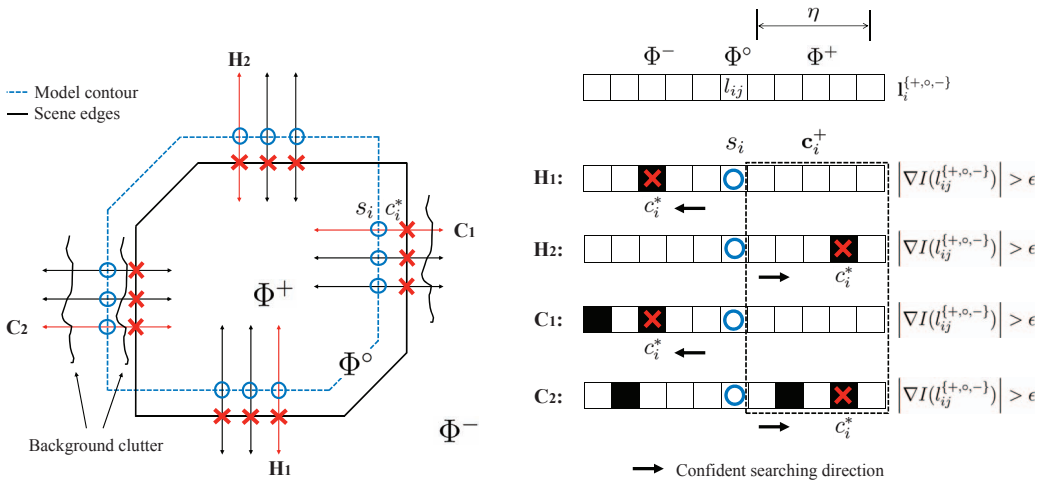


Fig. 2. Optimal local searching based on region levels. **Left**: To estimate camera poses, an infinitesimal camera motion $\mathbf{\Delta}$ is computed by minimizing distances between s_i (blue circle) and its correspondences c_i^* (red cross). **Right**: The c_i^* are searched to only confident searching directions (bold black arrows) without searching to both directions. Black rectangles (pixels) indicate candidates of correspondences within a certain range $|\eta|$ above a certain threshold ϵ. $\mathrm{H}_{1,2}$ and $\mathrm{C}_{1,2}$ are example cases in a homogeneous and cluttered background, respectively.

imization over camera poses. For instance, Schmaltz et al. [22] directly optimized 6DOF camera pose parameters by fitting image regions partitioned into an object and a background region of a projected 3D object model. Prisacariu and Reid [28] formulated a probabilistic framework that adopts the pixel-wise posterior background/foreground membership with GPU-assisted implementation for simultaneous segmentation and 3D object tracking. While region-based approaches would be beneficial for robust 3D object tracking, however, most of them are still difficult to fully support real-time performance even though it can be possible for speed-up on GPUs. Alternatively, Shahrokni et al. [29] showed fast 3D object tracking with background clutter using efficient texture boundary detection based on a Markov model instead of region segmentation, but this approach assumes uniformity of texture distributions and theoretically needs sufficient texture statistics for correct estimation.

In the remainder of the paper, we first clarify our problem with notation. We then explain optimal local searching based on region knowledge in detail. Finally, its feasibility is shown through experimental results and performance evaluations in challenging scenes.

2 Problem Statement and Notation

Given a 3D object model $\mathbf{M}$, edge-based tracking estimates the camera pose $\mathbf{E}^t$ by updating the previous camera pose $\mathbf{E}^{t-1}$ with infinitesimal camera motions between consecutive frames $\mathbf{\Delta}$, $\mathbf{E}^t = \mathbf{E}^{t-1}\mathbf{\Delta}$. The infinitesimal motions are computed by minimizing the errors between the 3D object model projected with the previous camera pose and its corresponding 2D scene edges m_i in the image such that

$$\widehat{\mathbf{\Delta}} = \arg\min_{\mathbf{\Delta}} \sum_{i=0}^{N-1} \left\| m_i - \mathrm{Proj}(\mathbf{M}; \mathbf{E}^{t-1}, \mathbf{\Delta}, \mathbf{K})_i \right\|^2 \quad (1)$$

$$= \arg\min_{\mathbf{\Delta}} \sum_{i=0}^{N-1} \left\| m_i - s_i \right\|^2 \quad (2)$$

where $\mathbf{K}$ is the camera intrinsic parameters, s_i are the sampled points of the projected object model, and N is the number of s_i. With this minimization scheme, we handle the local searching problem of the 3D-2D correspondences between the projected 3D object model and 2D scene edges in the image under the following tracking conditions:

- Single, rigid, and textureless 3D object where there is no or only little texture on the object
- Monocular RGB camera
- Scene edges are only the available visual cue.

Here, an initial camera pose, camera intrinsic parameters, and a 3D object model are given in advance. Note that we consider only the visible model contour instead of all data from the 3D object model because the model data is usually complex and its valuable interior data is very difficult to extract.

For region knowledge, searching regions $\Phi^{\{+,\circ,-\}}$ are partitioned into three levels (interior Φ^+, contour Φ°, and exterior Φ^-) with respect to a previous object region. The region appearance is modeled by the photometric property of the object region $\Psi(\Phi^+)$ or background region $\Psi(\Phi^-)$. For searching correspondences c_i^*, their candidates on each region level $\mathbf{c}_i^{\{+,\circ,-\}}$ are computed by local maximum gradient responses (above a certain threshold ϵ) along 1D searching lines

$\mathbf{l}_i^{\{+,\circ,-\}}$ through s_i toward normal directions (within a certain range $|\eta|$) (see Fig. 2).

3 Optimal Local Searching Based on Region Knowledge

3.1 Region Levels

First, we describe the relationship of the 3D-2D correspondences between the contour of the projected 3D object model and 2D scene edges in the image by reasoning it in object and background regions. If the camera motion is not fast and there are no drastic changes between consecutive frames, in general, the previous object region mostly overlaps with the corresponding current one. In our method, therefore, we partition the searching region into three levels, i.e., interior, contour, and exterior regions with respect to the previous object region, and delineate the local searching of the 3D-2D correspondences on their region levels as:

- Correspondences c_i^* always exist among candidates $\mathbf{c}_i^{\{+,\circ,-\}}$ that have intensity changes in an interior, contour, or exterior region of a 1D searching line $\mathbf{l}_i^{\{+,\circ,-\}}$ if and only if the search range covers correspondences and there are intensity discontinuities between an object and background region, $\exists c_i^* \in \mathbf{c}_i^{\{+,\circ,-\}} \left(\subset \mathbf{l}_i^{\{+,\circ,-\}}\right)$.

Since each correspondence occurs among the candidates in the 1D searching lines in one direction, not both directions, we consider likely candidates in only confident searching directions $\tilde{\mathbf{c}}_i^{\{+,\circ,-\}}$ to optimally search for c_i^* instead of all candidates. In practice, it is very advantageous to alleviate false matches due to background clutter because of greatly reducing nuisance searching. However, the question is how to determine the confident searching directions. In our method, the directions are determined by evaluating the candidates in the interior regions:

- A confident searching direction is definitely outward (or on contour) if there are no candidates of correspondences in an interior region $\mathbf{c}_i^+$.

Therefore, the correspondences are searched among likely candidates in the confident searching directions as

$$\begin{cases} \exists c_i^* \in \tilde{\mathbf{c}}_i^{\{\circ,-\}}, & \text{if } \mathrm{n}(\mathbf{c}_i^+) \text{ is null} \\ \exists c_i^* \in \tilde{\mathbf{c}}_i^{+}, & \text{otherwise} \end{cases} \tag{3}$$

where $\mathrm{n}(\cdot)$ is the cardinality of the finite set. Assuming that the target object is well extracted, the local searching based on the region levels is obvious without any uncertainty. If the target object is in the homogeneous background as shown in H_1 and H_2 cases of Fig. 2, for instance, the candidates in the confident searching directions are explicitly chosen as c_i^*. Indeed, this is a natural sense of matching correspondences, but it is straightforward in cluttered backgrounds. As shown in C_1 and C_2 cases of Fig. 2, c_i^* are chosen in the closest candidates in the confident directions if $\mathrm{n}(\mathbf{c}_i^+)$ is null and otherwise, the farthest ones. In our method, therefore, establishing 3D-2D correspondences is deterministic regardless of background clutter if and only if the target object is nicely extracted. Note that occlusion cases are not considered here because it is impossible to correctly establish the correspondences within occluded regions where the original scene edges are removed or altered. Instead, partial occlusions are alleviated by the M-estimator in the registration process of our tracking framework.

On the other hand, it is very difficult to perfectly extract an object contour in an image. In other words, many undesired candidates can be detected inside the object region by texts or figures on the object's surface (called object clutter) even though the target object has no or little texture, and they cause wrong searching directions. To ensure the confident searching directions, therefore, we explore efficient suppression of the object clutter by adopting the region appearance rather than precise extraction of the object contour from scene edges.

3.2 Region Appearance

Before describing the region appearance, we briefly present a searching bundle $\mathbf{L}$, which is a set of 1D searching lines $\mathbf{l}_i$. Simply, $\mathbf{L}$ is built by stacking each 1D searching line and arranging (shifting and flipping) it to be symmetric with the center of $\mathbf{L}$. The structure of $\mathbf{L}$ is as shown in Fig. 3(Middle-Left). In the local searching, there are great benefits of using the searching bundle. Basically, the resolution is the length of $\mathbf{l}_i$ (and padding) multiplied by the number of s_i, and it is much smaller than an input image resolution. In particular, unnecessary computations are reduced when information within the 1D searching lines has to be accessed multiple times because row vectors directly indicate $\mathbf{l}_i$, $\mathbf{c}_i$, $\tilde{\mathbf{c}}_i$, c_i^*, distances, and searching directions. Column vectors also include Φ^+ and $\Psi(\Phi^+)$ on the right side; Φ° and s_i on the center; and Φ^- and $\Psi(\Phi^-)$ on the left side. Furthermore, the region appearance can be modeled on its row and column spaces. For example, the right side of the columns is highly correlated with the object region appearance. Therefore, our local searching problem is more efficiently represented and handled in the searching bundle.

Now let's reconsider optimal local searching in the searching bundle. As shown in Fig. 3(Left), the target object has no textures and few dominant colors (mostly, wood), but the searching bundle has little object clutter due to texts on the object surface, inner boundaries of the object, and color changes under different light conditions. To efficiently suppress the object clutter, in our method, we exploit the region appearance, which is modeled by the photometric property of the object region or background region.

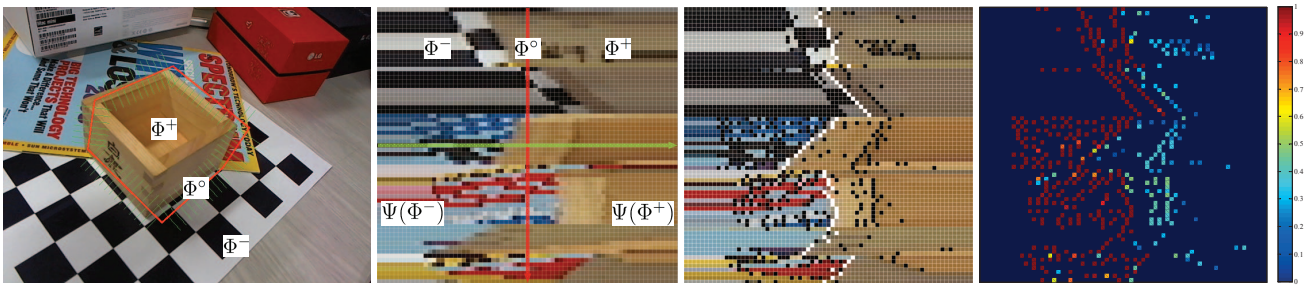


Fig. 3. Local searching on a searching bundle. **Left**: 3D object model projected on a target object with a previous camera pose (red line: previous model contour, green lines: 1D searching lines), **Middle-Left**: Searching bundle structure (71×65 local space vs. 640×480 input image resolution), **Middle-Right**: Candidates of correspondences including background clutter and object clutter (black dots) and final correspondences (white dots) by optimal local searching, **Right**: Similarity measure in uncertain regions of all candidates by computing distances between distributions using the Bhattacharyya similarity coefficient (distance range: 0 to 1, the zero value (blue) means that the candidate is on the object region).

The key idea is that the neighbor regions of the object clutter are highly correlated with the object region appearance.

For modeling the region appearance, we adopt a non-parametric density function based on Hue-Saturation-Value (HSV) histograms. Since HSV decouples the intensity from color, it is less sensitive to illumination changes. Following [30], [31], a HSV histogram has N bins which are composed of bins of the H, S, and V histogram ($N = N_H N_S + N_V$); and it is represented as the kernel density $\mathbf{H}(\Omega) = \{h(n;\Omega)\}_{n=1,\cdots,N}$. Here, $h(n;\Omega)$ is the probability of a bin n within a region Ω given by $h(n;\Omega) = \lambda \sum_{\mathbf{d}\in\Omega} \delta[\mathbf{b}(\mathbf{d}) - n]$ where δ is the delta function, λ is the normalizing constant that ensures $\sum_{n=1}^{N} h(n;\Omega) = 1$, $\mathbf{d}$ is any pixel location within a region Ω, and $\mathbf{b}(\cdot) \in \{1,\cdots,N\}$ is the bin index. If we denote $\Psi(\Phi^+) = \{h(n;\Phi^+)\}_{n=1,\cdots,N}$ as the object region appearance and $\Psi(\mathbf{U}^{\{+,\circ,-\}}) = \{h(n;\mathbf{U}^{\{+,\circ,-\}})\}_{n=1,\cdots,N}$ as the uncertain region appearance, which is modeled by the neighbor regions of the object clutter, we then measure the similarity between $\Psi(\Phi^+)$ and $\Psi(\mathbf{U}^{\{+,\circ,-\}})$ by computing distances of their distributions using the Bhattacharyya similarity coefficient on a HSV space [30] such that $\mathcal{D}^2[\Psi(\Phi^+), \Psi(\mathbf{U}^{\{+,\circ,-\}})] = 1 - \sum_{n=1}^{N} \sqrt{h(n;\Phi^+)h(n;\mathbf{U}^{\{+,\circ,-\}})}$.

Since the right side regions of both correspondences and object clutter in the searching bundle are object regions, the uncertain region of the kth candidate at the ith row of the searching bundle can be defined as the region of from the kth candidate to the $(k-1)$th candidate, $c_{ik-1} < \mathbf{U}_{ik} < c_{ik}$ and then the similarity measure is computed by

$$\mathcal{D}^2_{ik}[\Phi^+, \phi_k] = \mathcal{D}^2_{ik}\left[\Psi(\Phi^+), \Psi(\mathbf{U}^{\{+,\circ,-\}}_{ik}(\phi))\right] \quad (4)$$

where $\Psi(\mathbf{U}^{\{+,\circ,-\}}_{ik}(\phi)) = \sum_{c_{ik-1}<\phi<c_{ik}} \Psi(\phi)$. In some candidates, on the other hand, only the object region appearance is insufficient to model the uncertain region appearance. To handle such candidates, we incorporate the background region appearance because if the candidates are object clutter, the uncertain region appearance can be relatively far from the background region appearance even though it is not very close to the object region appearance. If we denote $\mathcal{D}^2_{ik}\left[\Psi(\Phi^-), \Psi(\mathbf{U}^{\{+,\circ,-\}}_{ik}(\phi))\right]$ as the similarity measure with the background region appearance, therefore, we evaluate the candidates through multiple phases as

$$\Gamma(\theta)\mathcal{D}^2_{ik}[\Phi^+, \phi_k] + (1 - \Gamma(\theta))\mathcal{D}^2_{ik}[\Phi^-, \phi_k] \quad (5)$$

where $\Gamma(\theta)$ is the phase function defined as $\Gamma(\theta) = 1$ if $\mathcal{D}^2_{ik}[\Phi^+, \phi_k] < \tau$ and otherwise, $\Gamma(\theta) = 0$; and $\mathcal{D}^2_{ik}[\Phi^-, \phi_k] = \left(1 - \mathcal{D}^2_{ik}\left[\Psi(\Phi^-), \Psi(\mathbf{U}^{\{+,\circ,-\}}_{ik}(\phi))\right]\right)$. In addition, the neighbor regions of the object clutter cannot be correlated with the object region appearance when they are occupied by small portions of the object clutter such as texts or figures. Assumed that most object clutter belongs to the interior region Φ^+, we can employ the interior region appearance prior to modeling the uncertain region appearance of $\mathbf{c}^+_i$, such as $\mathcal{D}^2_{ik}\left[\Psi(\Phi^+), \Psi(\mathbf{U}^+_{ik}(\omega))\right]$, where $\Psi(\mathbf{U}^+_{ik}(\omega)) = \sum_{s_i<\omega<c^+_{ik}} \Psi(\omega)$. Figure 3(Right) shows an example of our similarity measure in uncertain regions of all candidates (black dots in Fig. 3(Middle-Right)) in the searching bundle. Lower values (blue) indicate that they are much closer to the object region appearance. With this measure, finally, the correspondences c^*_i (white dots in Fig. 3(Middle-Right)) are searched by (3).

4 Experimental Results

This section first describes our underlying tracking framework along with implementation details and then shows its experimental results and performance evaluations.

4.1 Implementation

In our implementation, a tracking framework is based on minimizing distances between a contour of a projected 3D object model with a previous camera pose and its corresponding 2D scene edges in an image in an iterative manner. The infinitesimal motions are represented by a 3D rigid-body transformation group in $\Re^3$ [32], [33]. For robust estimation, an iterative reweighted least square (IRLS) approach is performed using a bisquare M-estimator. In this framework, the iteration is terminated when the reprojection error is small (< 1.5 pixel) or the number of iterations is larger than a defined one (> 10).

For the model contour, the visible boundary lines of the object model are filtered through a visibility and boundary test. In both tests, the hidden lines are sorted by computing the inner products among the camera viewpoint vector and the face normal vectors. The lines shared by the two faces are excluded among the visible lines. Here, the object model consists of wireframes with vertices and lines. The sampling interval of the model contour was properly determined according to target objects.

For the candidates of correspondences, the 1D searching lines $\mathbf{l}_i^{\{+,\circ,-\}}$ are defined using the Bresenham's line drawing algorithm (8-connectivity) [34]. The candidates of correspondences $\mathbf{c}_i^{\{+,\circ,-\}}$ are computed by 1D convolution of a 1×3 filter mask ($[-1 \quad 0 \quad 1]$) and 1D non-maximum suppression (3-neighbor) along the lines. To improve its robustness, we separately compute the gradient responses on each color channel of an input image and then take the one with the largest norm [35] such that

$$c_{ij}^{\{+,\circ,-\}} = \max_{\mathcal{C} \in \{R,G,B\}} \left\| \nabla I_{\mathcal{C}} \left(l_{ij}^{\{+,\circ,-\}} \right) \right\| \qquad (6)$$

where R, G, B are the RGB color channels and $I_{\mathcal{C}}(.)$ is the pixel intensity in the $\mathcal{C}$ channel image. The threshold for $\mathbf{c}_i$ was 10 ($\epsilon = 10$), and the search range for $\mathbf{l}_i$ was 30 pixels ($|\eta| = 30$).

For the object region appearance, each bin number was set as $N_H = N_S = N_V = 8$, and only the pixels with saturation and value larger than certain thresholds (> 0.1 and 0.2, respectively) were used for the HS histogram. The object region appearance $\Psi(\Phi^+)$ and the background region appearance $\Psi(\Phi^-)$ were updated when the tracking succeeded. The parameter for the phase function τ was 0.3. Overall procedures of our tracking framework are shown in Procedure 1.

Procedure 1 Tracking framework

Given: 3D object model $\mathbf{M}$, previous camera pose $\mathbf{E}^{t-1}$, camera intrinsic parameters $\mathbf{K}$

repeat
  Set $\mathbf{P}^{t-1} \leftarrow \mathbf{K}\,\mathbf{E}^{t-1}$.
  Set $\widetilde{\mathbf{M}}$ via a visibility and boundary test.
  Set $s_{i=0,\ldots,N-1}$ by sampling $\mathbf{P}^{t-1}\widetilde{\mathbf{M}}$ with equal distances.
  Set $\Phi^{\{+,\circ,-\}}$ and $\mathbf{L}_{N\times W}$ with $l_{i=0,\ldots,N-1;j=0,\ldots,W-1}^{\{+,\circ,-\}}$.
  For $i := 0, \ldots, N-1$, compute $\mathbf{c}_i^{\{+,\circ,-\}}$ by (6).
  For $i := 0, \ldots, N-1$, evaluate $\mathbf{c}_i^{\{+,\circ,-\}}$ via (5).
  For $i := 0, \ldots, N-1$, search for c_i^* by (3).
  Compute $\mathbf{\Delta}$ with correspondences $(s_i, c_i^*)_{i=0,\ldots,N-1}$ by (2).
  Update $\mathbf{E}^{t-1} \leftarrow \mathbf{E}^{t-1}\mathbf{\Delta}$.
until reprojection error $< min$ or iteration $> max$

Return: $\mathbf{E}^t \leftarrow \mathbf{E}^{t-1}$

4.2 Performance

For target objects, textureless 3D objects were chosen as shown in Fig. 4, Fig. 5, and Fig. 6. In our experiments, we mainly used rectangle-shaped objects for modeling simplicity, but complex-shaped objects whose contours are not formed of only several straight lines can also be considered if their 3D models are available, as shown in Fig. 5. The target objects were modeled as wireframe models offline. The backgrounds were arbitrarily prepared either with or without heavy clutter as shown in Fig. 4, Fig. 5, and Fig. 6. The experiments were performed on a standard laptop with 2.27 GHz of a CPU and 4 GB of a RAM. For capturing images, we used a standard web camera with 640×480 image resolution. An initial camera pose and camera calibration parameters were given in advance.

First, we tested our method with various camera motions and verified it by projecting a 3D object model on a target object with estimated camera poses. We also examined each 3D-2D correspondence established in searching bundles. In searching bundles, we can say that the 3D object model projected with estimated camera poses is perfectly matched on the target object if all the correspondences are laid on the center of the searching bundles. As shown in Fig. 4, our method was successfully performed with different textureless 3D objects in different highly cluttered backgrounds. In most cases, the searched correspondences were acceptable without interference from the background clutter. Our method also properly handled the object clutter from texts or figures on the object surface (this can easily be recognized by the non-uniform appearance in the interior regions of the searching bundles). On the other hand, we could see a few false matches when the object's color density was quite similar to the background's one (see Second Row-Three Column and Fourth Row-Fourth Column in Fig. 4) or the object was partially occluded (see Sixth Row-Third and Fourth Column in Fig. 4). However, these errors did not significantly affect the overall tracking results because they could be alleviated by the M-estimator during the registration

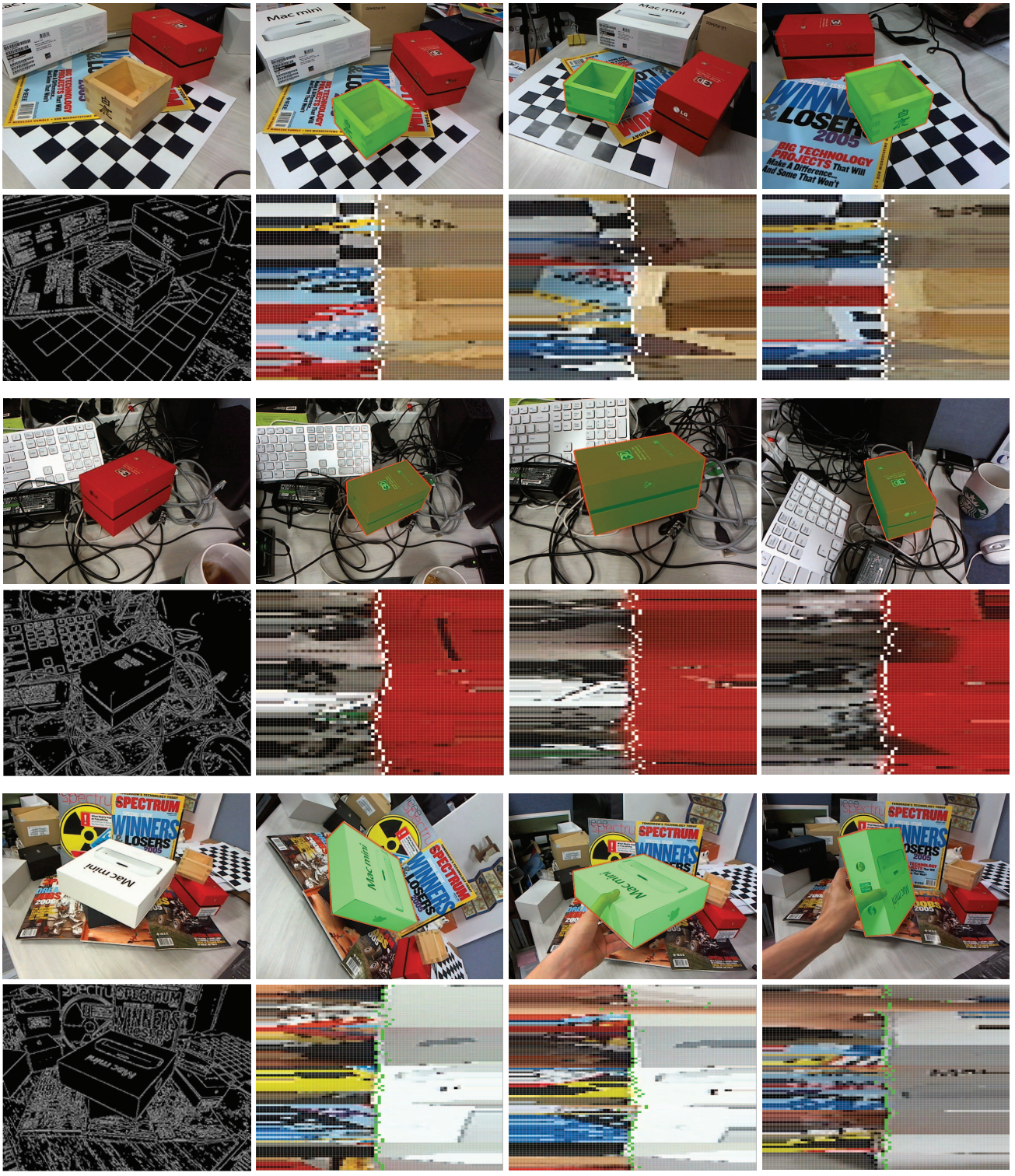


Fig. 4. Experimental results with different textureless 3D objects in different highly cluttered backgrounds. **Odd Rows-First Columns**: Target objects and backgrounds, **Even Rows-First Columns**: Scene edges, **Odd Rows-Second to Fourth Columns**: 3D object models (green rectangles) projected on target objects with estimated camera poses, **Even Rows-Second to Fourth Columns**: 3D-2D correspondences (white and green dots) established in searching bundles.

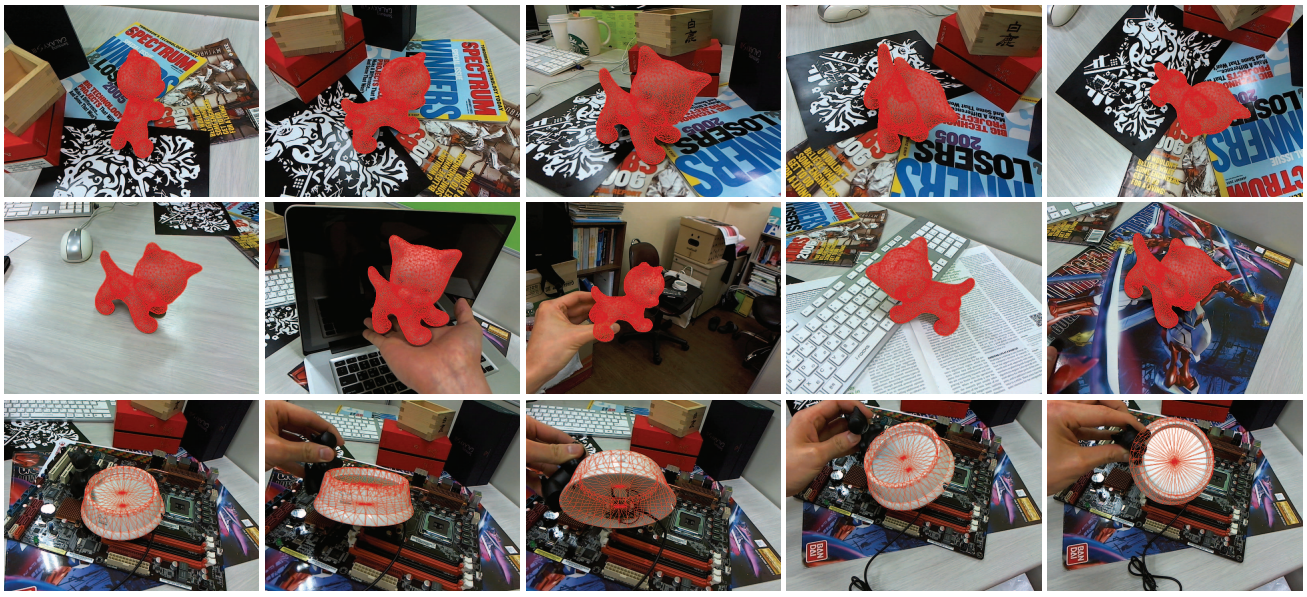


Fig. 5. Tracking results using textureless 3D objects with different shapes (**Top and Middle-Row**: Pink cat, **Bottom-Row**: White tray) in highly cluttered and homogeneous backgrounds. Red meshes indicate 3D object models projected on target objects with estimated camera poses.

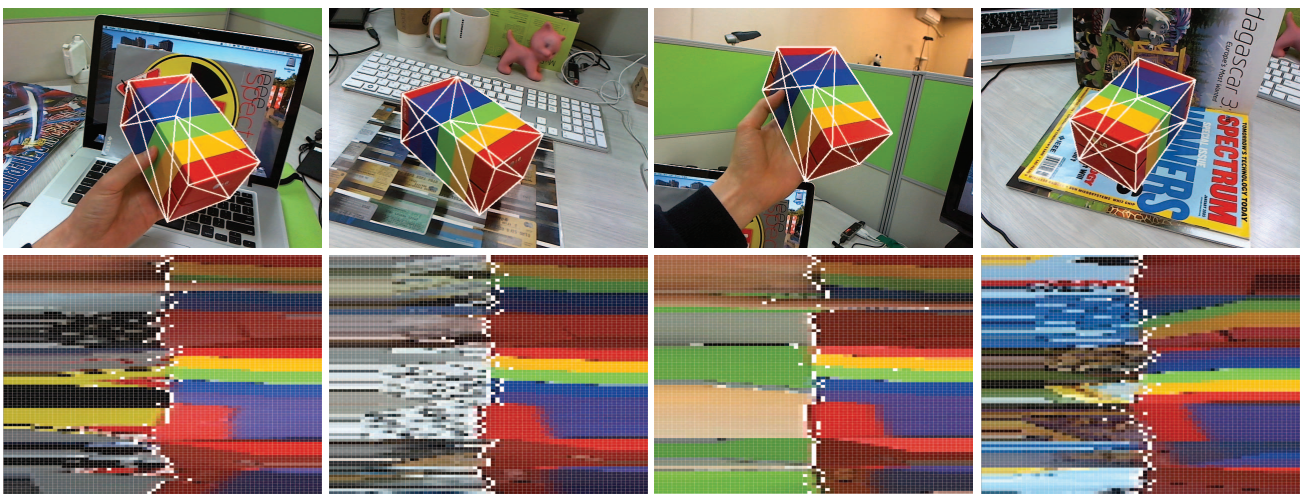


Fig. 6. Tracking results using a textureless 3D object with multiple colors in highly cluttered and homogeneous backgrounds. **Top-Row**: 3D object models (white mesh) projected on target objects with estimated camera poses, **Bottom-Row**: 3D-2D correspondences (white dots) established in searching bundles.

process.

More experimental results are shown in Fig. 5, Fig. 6, and Fig. 7. As stated above, our method allowed textureless 3D objects with complex shapes as well as simple rectangular shapes if their 3D models were available as shown in Fig. 5. In these experiments, we used two textureless 3D objects with different shapes (pink cat and white tray) and their models (5000 faces and 320 faces, respectively), which are visualized as red meshes in Fig. 5. In the tray case, however, there was some ambiguity about the rotation on one axis because its shape was symmetric about the axis. Since our method uses the region appearance, which is modeled by the color density, it could be restricted to the object's colors. Unless the majority of the object's color density was quite similar to the background's one, however, the tracking performance was not considerably degraded regardless of whether the object had a single dominant color (Fig. 4 and Fig. 5) or multiple colors (Fig. 6). When new surfaces (faces) of the 3D object appeared as shown in Fig. 7(Top-Row), additionally, the tracking could be susceptible to false matches because the visible and hidden boundary lines got closer. Since the previous visible boundary lines (black dashed line) were laid on the object region after switching, however, our method

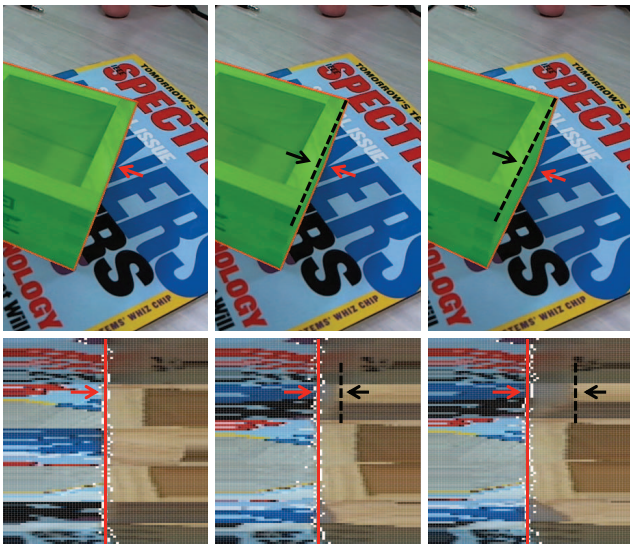


Fig. 7. Tracking result when a new surface (face) appears. **Top-Row**: The hidden boundary line is switched into the visible boundary line (black dashed line: previous visible boundary line, red solid line: current visible boundary line). **Bottom-Row**: The previous visible boundary line was handled as the object clutter in a searching bundle (white dots: searched correspondences).

well suppressed them as the object clutter as shown in Fig. 7(Bottom-Row). Our real-time demonstrations are shown in a supplementary video.

Next, we examined 3D-2D correspondences established in a searching bundle at each iteration and compared them with other approaches. Since our method handles a local searching problem of 3D-2D correspondences, we basically chose the iterative closest points (ICP)-like approach that searches for the closest intensity discontinuity above a certain threshold [9]. In the searching bundle, the local searching problem can also be considered as a local segmentation problem; thus, we compared our method with one of segmentation approaches (GrabCut [36]). As shown in Fig. 9, in the ICP-like approach, the majority of the correspondences were false matches due to background clutter and even object clutter, and the tracking was stuck in the local minima during the early iterations (we can also see the searching bundles were not changed). In the segmentation approach, the segmented boundaries were acceptable, but delicate user interactions were separately in need of the correct segmentation during the early iterations because automatic processing failed in the searching bundles. In our method, however, most correspondences were correctly matched and gradually merged to the center of the searching bundle on every iteration.

To evaluate the accuracy, we estimated 6DOF camera poses with ones by the well-known SIFT [37] because the SIFT could reliably estimate camera poses

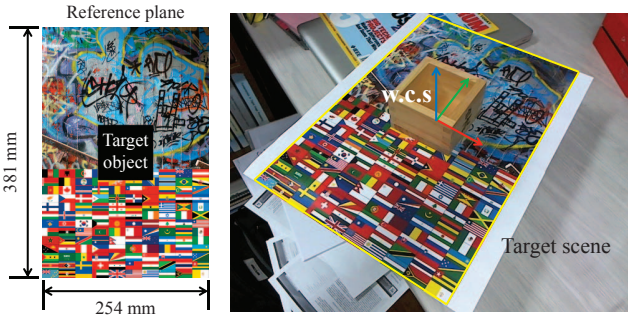


Fig. 8. Setup for the comparison of 6DOF camera poses. **Left**: Reference plane, **Right**: Target scene and 3D object.

as batch processing using dense feature points in the background of our setup as shown in Fig. 8. Note that the coordinates of the reference plane and the 3D object were registered in advance. As shown in Fig. 10, both trajectories were similar during all frames (765 frames). The average angle difference was about 1.35°, and the average distance difference was about 1.29 mm.

Since one of main concerns is real-time performance, we computed the overall processing time during 800 frames and the number of iterations at each frame with unconstrained camera motions. For the first textureless 3D object in Fig. 4, one iteration was performed within 6.3 ms (establishing correspondences: 4.2 ms, computing and updating camera motions: 0.5 ms). The overall processing time per frame linearly increased up to the defined maximum number of iterations. In the tracking framework, the maximum number of iterations was 10 for tolerating certain motions. During the test, finally, the average overall processing time was about 30 ms and the average number of iterations was less than 5. In the evaluation, the visibility and boundary test were done within a few milliseconds because the 3D object model was very simple. If the object models are complex, however, this process would not be trivial in the overall processing time. In the tracking framework, alternatively, the visible boundary lines were tested only once every frame because they were not much changed during iterations. Since only the model contour was used for tracking, moreover, it could maintain reasonable speed for real-time performance (20 fps with the cat object as shown in Fig. 5(Top and Middle-Row)). In addition, the search range could be properly set because the correspondences were not far from previous ones even in certain motions.

Finally, we compared the overall tracking performance with one of the region-based methods [28] that demonstrated the state-of-the-art results of fast and robust 3D object tracking. For evaluation, we used the same data (target image with the textureless 3D object and its model (Fig. 11(Top Row-First Column))) and

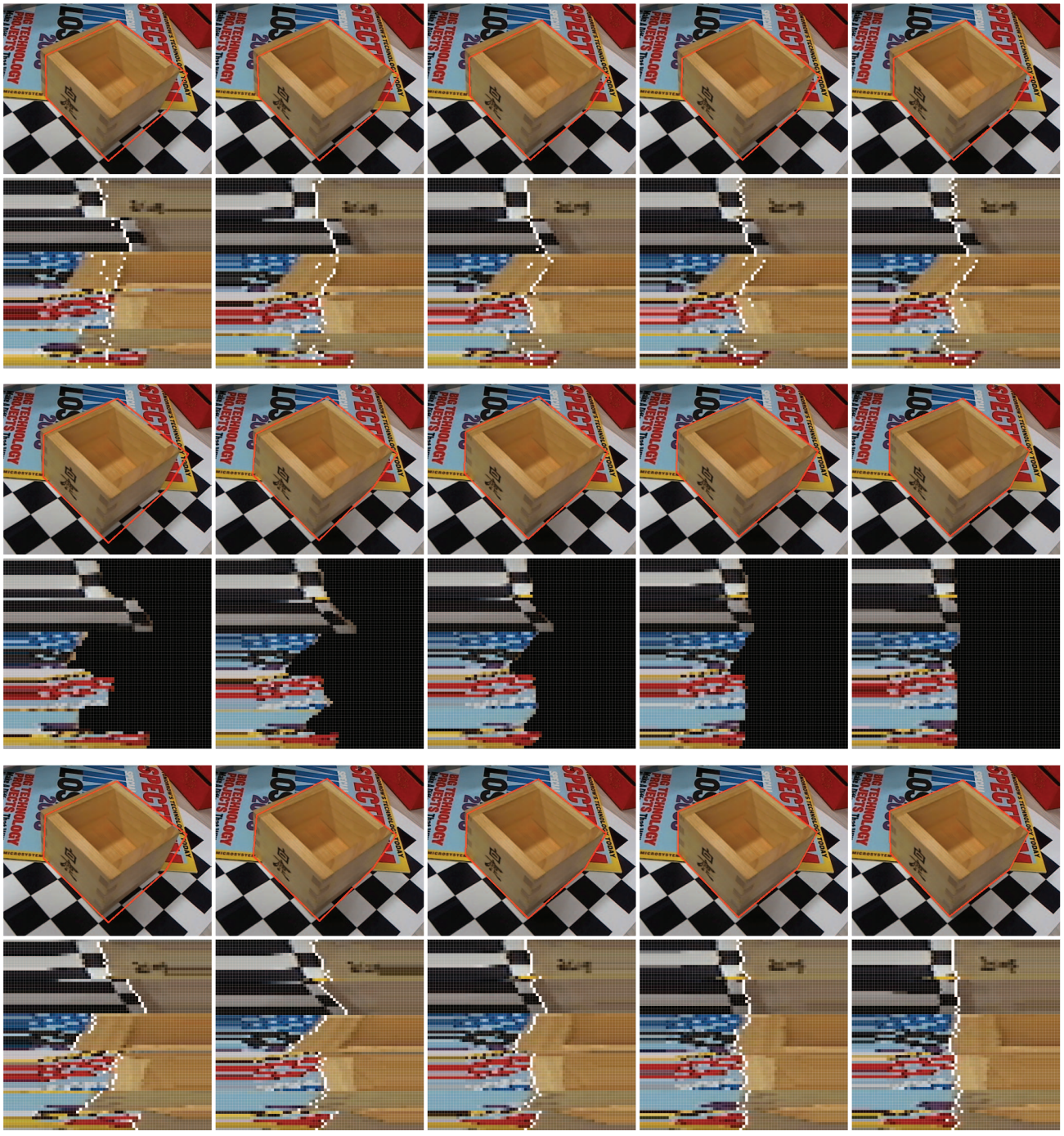


Fig. 9. Comparison of 3D object models projected on target scenes with estimated camera poses (**Odd-Rows**: red line) and 3D-2D correspondences (**Even-Rows**: white dots) established at the 1st, 3rd, 5th, 7th, and 9th iteration when using **Top-Rows**: ICP-like approach, **Middle-Rows**: segmentation approach (GrabCut [36]), and **Bottom-Rows**: our method.

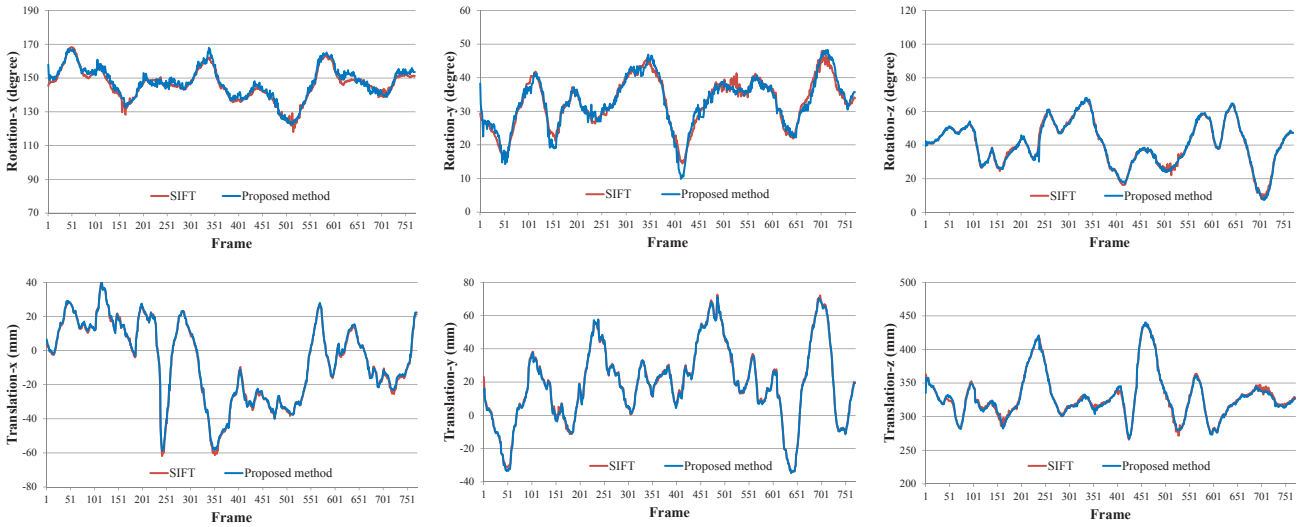


Fig. 10. Comparison of 6DOF camera poses estimated using the SIFT [37] (red line) and our method (blue line).

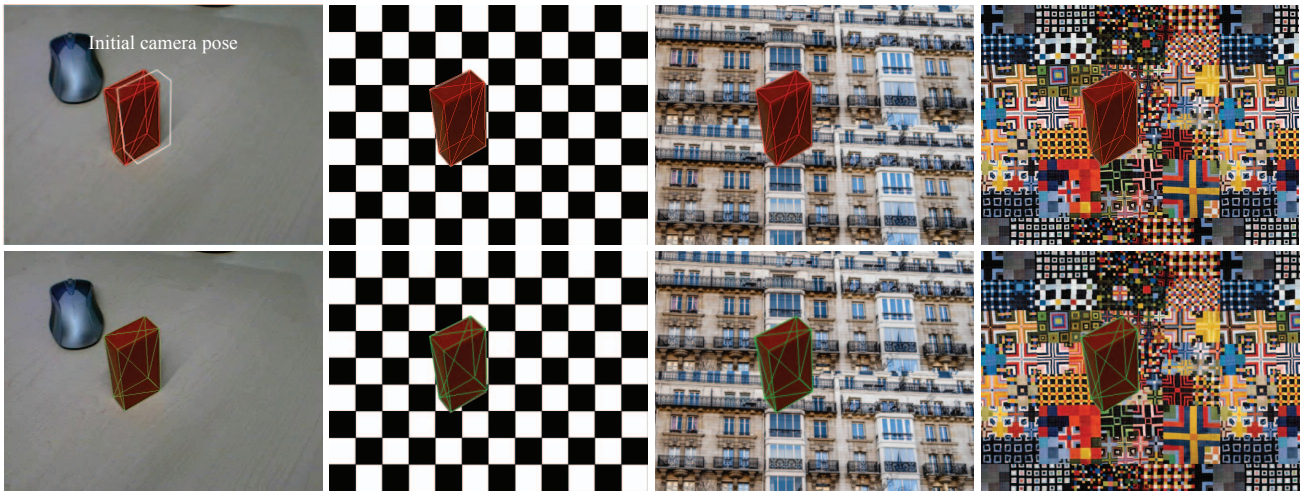


Fig. 11. Comparisons between **Top**: the region-based method [28] and **Bottom**: our method. Red and green meshes are 3D object models projected on target scenes with estimated camera poses. **Top Row-First Column**: The white contour is the model contour projected with the initial camera pose.

the same parameters (initial camera pose and camera calibration parameters) given by the available code in [28]. We also prepared additional target images by displacing the background of the original target image into different highly cluttered backgrounds as shown in Fig. 11. Since [28] uses the CUDA framework for GPU processing, we evaluated both methods on a laptop with an NVIDIA Geforce GTX 560M video card. As shown in Fig. 11, both methods correctly estimated camera poses for all target images regardless of the background clutter. In certain cases, our method was slightly faster, but we considered both runtimes to be comparable in most cases. However, it would be convinced that our method is more advantageous for real-time applications (even on mobile platforms) because it can sufficiently support fast tracking even without GPU processing.

4.3 Limitations

In our method, we assume that there always exist correspondences that have local maximum gradient responses above a certain threshold within a certain range. If camera motions are much faster or change drastically, however, the correspondences can be out of the search range or overlapped regions can be much smaller due to large displacement (Fig. 12(Left)). It can also be difficult to detect the correspondences under heavy occlusion (Fig. 12(Middle-Left)).

As demonstrated in the experiments, our method does not depend on specific shapes of textureless 3D objects. Since we use only the model contour to estimate 6DOF camera poses, however, the camera poses have some ambiguity when the objects have symmetric shapes. Though textureless 3D objects usually have few dominant colors, our method is not limited to

Fig. 12. Tracking failure cases. **Left**: Fast motion, **Middle-Left**: Heavy occlusion, **Middle-Right**: Similar object and background color, **Right**: Low contrast by poor illumination.

specific colors of objects. However, it can be difficult not only to detect the 3D-2D correspondences, but also to model the distinctive region appearance between the object and background region when the object's color density is quite similar to the background's one (Fig. 12(Middle-Right)) or the light conditions are poor (Fig. 12(Right)).

5 Conclusion

This paper presented optimal local searching for fast and robust textureless 3D object tracking in highly cluttered backgrounds. In the local searching of the 3D-2D correspondences, confident searching directions were determined by evaluating their candidates with region knowledge, and it led to sufficiently alleviate numerous false matches due to the background clutter. As the searching bundle was newly defined, moreover, the local searching was efficiently performed on the low-dimensional space. Through experiments and evaluations, finally, we showed that our method allowed robust textureless 3D object tracking even in highly cluttered backgrounds while retaining real-time performance.

Though we made substantial improvements in the edge-based approach, combining with other available cues would be necessary to handle more general cases including our limitations [38]. As another interest in our future works, tracking-by-detection schemes would be beneficial for improving tracking performance because the detection process could allow us good guesses for tracking by providing better prior knowledge such as approximated object regions or camera poses [17].

Acknowledgments

This research is supported by Ministry of Culture, Sports and Tourism (MCST) and Korea Creative Content Agency (KOCCA) in the Culture Technology (CT) Research & Development Program 2013. Corresponding author: J.-I. Park.

References

[1] V. Lepetit and P. Fua, "Monocular model-based 3D tracking of rigid objects: A survey," *Foundations and Trends in Computer Graphics and Vision*, vol. 1, no. 1, pp. 1–89, 2005.

[2] I. Skrypnyk and D. G. Lowe, "Scene modelling, recognition and tracking with invariant image features," in *IEEE and ACM International Symposium on Mixed and Augmented Reality*, 2004, pp. 110–119.

[3] L. Vacchetti, V. Lepetit, and P. Fua, "Stable real-time 3D tracking using online and offline information," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 26, no. 10, pp. 1385–1391, 2004.

[4] S. Hinterstoisser, S. Benhimane, and N. Navab, "N3M: Natural 3D markers for real-time object detection and pose estimation," in *IEEE International Conference on Computer Vision*, 2007, pp. 1–7.

[5] L. Masson, M. Dhome, and F. Jurie, "Robust real time tracking of 3D objects," in *International Conference on Pattern Recognition*, 2004, pp. 252–255.

[6] E. Ladikos, S. Benhimane, and N. Navab, "A realtime tracking system combining template-based and feature-based approaches," in *International Conference on Computer Vision Theory and Applications*, 2007, pp. 325–332.

[7] Y. Park, V. Lepetit, and W. Woo, "Handling motion-blur in 3D tracking and rendering for augmented reality," *IEEE Transactions on Visualization and Computer Graphics*, vol. 18, no. 9, pp. 1449–1459, 2012.

[8] C. Harris and C. Stennett, "RAPID: A video-rate object tracker," in *British Machine Vision Conference*, 1990, pp. 73–77.

[9] T. Drummond and R. Cipolla, "Real-time visual tracking of complex structures," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 7, pp. 932–946, 2002.

[10] A. I. Comport, E. Marchand, M. Pressigout, and F. Chaumette, "Real-time markerless tracking for augmented reality: The virtual visual servoing framework," *IEEE Transactions on Visualization and Computer Graphics*, vol. 12, no. 4, pp. 615–628, 2006.

[11] L. Vacchetti, V. Lepetit, and P. Fua, "Combining edge and texture information for real-time accurate 3D camera tracking," in *IEEE and ACM International Symposium on Mixed and Augmented Reality*, 2004, pp. 48–56.

[12] H. Wuest, F. Vial, and D. Stricker, "Adaptive line tracking with multiple hypotheses for augmented reality," in *IEEE and ACM International Symposium on Mixed and Augmented Reality*, 2005, pp. 62–69.

[13] E. Rosten and T. Drummond, "Fusing points and lines for high performance tracking," in *IEEE International Conference on Computer Vision*, 2005, pp. 1508–1515.

[14] M. Pressigout and E. Marchand, "Real-time hybrid tracking using edge and texture information," *International Journal of Robotics Research*, vol. 26, no. 7, pp. 689–713, 2007.

[15] C. Choi and H. I. Christensen, "Robust 3D visual tracking using particle filtering on the special euclidean group: A combined approach of keypoint and edge features," *International Journal of Robotics Research*, vol. 31, no. 4, pp. 498–519, 2012.

[16] Y. Park, V. Lepetit, and W. Woo, "Texture-less object tracking with online training using an RGB-D camera," in *IEEE International Symposium on Mixed and Augmented Reality*, 2011, pp. 121–126.

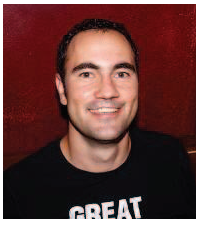
[17] S. Hinterstoisser, V. Lepetit, S. Ilic, S. Holzer, G. Bradski, K. Konolige, and N. Navab, "Model based training, detection and pose estimation of texture-less 3D objects in heavily cluttered scenes," in *Asian Conference on Computer Vision*, 2012, pp. 548–562.

[18] B. Drost and S. Ilic, "3D object detection and localization using multimodal point pair features," in *International Conference on 3D Imaging, Modeling, Processing, Visualization and Transmission*, 2012, pp. 9–16.
[19] G. Klein and D. Murray, "Full-3D edge tracking with a particle filter," in *British Machine Vision Conference*, 2006, pp. 114.1–114.10.
[20] M. Pupilli and A. Calway, "Real-time camera tracking using known 3D models and a particle filter," in *International Conference on Pattern Recognition*, 2006, pp. 199–203.
[21] J. Brown and D. Capson, "A framework for 3D model-based visual tracking using a GPU-accelerated particle filter," *IEEE Transactions on Visualization and Computer Graphics*, vol. 18, no. 1, pp. 68–80, 2012.
[22] C. Schmaltz, B. Rosenhahn, T. Brox, and J. Weickert, "Region-based pose tracking with occlusions using 3D models," *Machine Vision and Applications*, vol. 23, no. 3, pp. 557–577, 2012.
[23] E. Marchand, P. Bouthemy, and F. Chaumette, "A 2D-3D model-based approach to real-time visual tracking," *Image and Vision Computing*, vol. 19, no. 13, pp. 941–955, 2001.
[24] Y. Yoon, A. Kosaka, and A. C. Kak, "A new Kalman-filter-based framework for fast and accurate visual tracking of rigid objects," *IEEE Transactions on Robotics*, vol. 24, no. 5, pp. 1238–1251, 2008.
[25] C. Teuliére, E. Marchand, and L. Eck, "Using multiple hypothesis in model-based tracking," in *IEEE International Conference on Robotics and Automation*, 2010, pp. 4559–4565.
[26] B. Rosenhahn, T. Brox, and J. Weickert, "Three-dimensional shape knowledge for joint image segmentation and pose tracking," *International Journal of Computer Vision*, vol. 73, no. 3, pp. 243–262, 2007.
[27] S. Dambreville, R. Sandhu, A. Yezzi, and A. Tannenbaum, "Robust 3D pose estimation and efficient 2D region-based segmentation from a 3D shape prior," in *European Conference on Computer Vision*, 2008, pp. 169–182.
[28] V. A. Prisacariu and I. D. Reid, "PWP3D: Real-time segmentation and tracking of 3D objects," *International Journal of Computer Vision*, vol. 98, no. 3, pp. 335–354, 2012.
[29] A. Shahrokni, T. Drummond, and P. Fua, "Texture boundary detection for real-time tracking computer vision," in *European Conference on Computer Vision*, 2004, pp. 566–577.
[30] D. Comaniciu, V. Ramesh, and P. Meer, "Real-time tracking of non-rigid objects using mean shift," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2000, pp. 142–149.
[31] P. Pérez, C. Hue, J. Vermaak, and M. Gangnet, "Color-based probabilistic tracking," in *European Conference on Computer Vision*, 2002, pp. 661–675.
[32] C. Bregler and J. Malik, "Tracking people with twists and exponential maps," in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 1998, pp. 8–15.
[33] T. Drummond and R. Cipolla, "Application of lie algebras to visual servoing," *International Journal of Computer Vision*, vol. 37, no. 1, pp. 21–41, 2000.
[34] J. E. Bresenham, "Algorithm for computer control of a digital plotter," *IBM Systems Journal*, vol. 4, no. 1, pp. 25–30, 1965.
[35] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2005, pp. 886–893.
[36] C. Rother, V. Kolmogorov, and A. Blake, ""GrabCut": Interactive foreground extraction using iterated graph cuts," *ACM Transactions on Graphics*, vol. 23, no. 3, pp. 309–314, 2004.
[37] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International Journal of Computer Vision*, vol. 60, no. 2, pp. 91–110, 2004.
[38] T. Brox, B. Rosenhahn, J. Gall, and D. Cremers, "Combined region and motion-based 3D tracking of rigid and articulated objects," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, no. 3, pp. 402–415, 2010.

Byung-Kuk Seo received BS and MS degrees in electronics and computer engineering in 2006 and 2008, respectively, from Hanyang University, Seoul, Korea, where he is currently pursuing his PhD degree. His research interests include 3D computer vision, augmented reality, and human-computer interaction.

Hanhoon Park received BS, MS, and PhD degrees in electrical and computer engineering from Hanyang University, Seoul, Korea, in 2000, 2002, and 2007, respectively. From 2008 to 2011, he was a postdoctoral researcher at NHK Science & Technology Research Laboratories, Tokyo, Japan. He is currently an assistant professor at Pukyong National University. His research interests include augmented reality and human-computer interaction.

Jong-Il Park received BS, MS, and PhD degrees in electronics engineering from Seoul National University, Seoul, Korea, in 1987, 1989, and 1995, respectively. From 1996 to 1999, he was a researcher with the ATR Media Integration and Communication Research Laboratories, Kyoto, Japan. In 1999, he joined the Department of Electrical and Computer Engineering at Hanyang University, Seoul, Korea, where he is currently a professor. His research interests include computational imaging, augmented reality, 3D computer vision, and human-computer interaction.



Stefan Hinterstoisser received his PhD degree at the CAMP at theTechnische Universität München in Germany where he is part of the Computer Vision group. His current research interests include real-time object detection and pose estimation of generic 2D/3D objects. He is especially interested in improving the recognition and detection of almost texture-less objects as they are often found in human environments.

Slobodan Ilic is leading the Computer Vision Group of CAMP at TUM since 2009. From June 2006 he was a senior researcher at Deutsche Telekom Laboratories in Berlin. Before that he was a postdoctoral fellow at CVLab, EPFL, Switzerland, where he received his PhD in 2005. His research interests include: deformable surface modeling and tracking, 3D reconstruction, real-time object detection and tracking. He was recently an Area Chair for ICCV 2011 and is regularly a part of the Program Committee for all major computer vision conferences. Besides active academic involvement Slobodan has strong relations to industry and supervises a number of PhD students supported by industry.