Real-Time Monocular Pose Estimation of 3D Objects using Temporally Consistent Local Color Histograms

Henning Tjaden, Ulrich Schwanecke
RheinMain University of Applied Sciences
Wiesbaden, Germany
{henning.tjaden, ulrich.schwanecke}@hs-rm.de

Elmar Schömer
Johannes Gutenberg University Mainz
Mainz, Germany
schoemer@uni-mainz.de

Abstract

We present a novel approach to 6DOF pose estimation and segmentation of rigid 3D objects using a single monocular RGB camera based on temporally consistent, local color histograms. We show that this approach outperforms previous methods in cases of cluttered backgrounds, heterogenous objects, and occlusions. The proposed histograms can be used as statistical object descriptors within a template matching strategy for pose recovery after temporary tracking loss e.g. *caused by massive occlusion or if the object leaves the camera's field of view. The descriptors can be trained online within a couple of seconds moving a handheld object in front of a camera. During the training stage, our approach is already capable to recover from accidental tracking loss. We demonstrate the performance of our method in comparison to the state of the art in different challenging experiments including a popular public data set.*

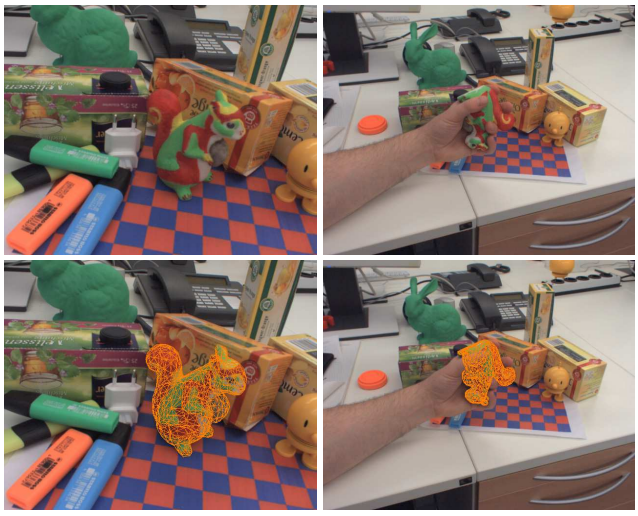

Figure 1. Estimating the pose of a heterogenous object in a cluttered scene under strong occlusions and viewpoint changes. Top: RGB input frames. Bottom: Tracking result (within ∼16 ms).

1. Introduction

Visually estimating the pose, meaning 3D orientation and translation, of rigid objects is an essential and challenging task in many computer vision based systems. The fields of application include robotics, medical navigation, sports therapy, augmented reality and human computer interaction (see *e.g.* [16] for a detailed survey). Thereby, for many practical scenarios it is important that the underlying pose estimation algorithms are real-time capable. Furthermore, they should be robust to cluttered backgrounds, different lighting conditions and surface properties such as texture, reflectance and color. In particular for handheld objects it is crucial that occlusions can be handled appropriately (see Fig. 1). In practice, it is often desirable to use only one ordinary camera instead of a multi-camera setup as this keeps the hardware and calibration requirements at a minimum and suffers least from visibility issues.

In human environments, objects of interest such as tools, components or vehicles are often weakly textured or even texturless, which is why methods based on point features [17] cannot be used in general. Additionally, intensity gradient-based descriptors of the object's surface are prone to local minima in case of cluttered background, motion blur or defocus [16, 27]. Since the appearance of such objects is characterized by their silhouette, so-called region-based approaches have been introduced. Based on prior shape knowledge usually in form of a 3D model, these methods try to estimate the object's pose by minimizing the discrepancy between a suitable representation of both the camera image and the projected silhouette of the model parametrized by the sought pose.

In general, the problem of pose estimation can be separated into *pose tracking* and *pose detection*. In case of tracking, the object is assumed to be seen in a sequence of consecutive images such as a video footage. Thereby, the mo-

tion of the object is assumed to be relatively small between two consecutive frames. Thus, only the pose difference from one frame to the next has to be determined and therefore tracking can be performed quite efficiently. The main downside of pure pose tracking algorithms is the need for manual initialization at the beginning and re-initialization after tracking loss to get a coarse starting pose.

This leads to the problem of pose detection, where the object's pose has to be estimated from a single image without any prior pose knowledge. This lack of information makes pose detection generally more complex and computationally demanding than pose tracking. To obtain robust real-time applications tracking has to be combined with detection, which provides a starting solution whenever tracking is lost, *e.g*. in cases of strong occlusion, rapid movement or when the object leaves the camera's field of view.

1.1. Related Work

In recent years, research on tracking [5, 20, 19, 14] and detection [11, 9, 2, 12] has mainly focused on RGB-D sensor data. Although these methods outperform those only based on monocular RGB image data, they are limited in distance to the camera and struggle with sunlight. Thus, we do not include them for comparison in this work.

For pose tracking using RGB images, region-based methods relying on statistical level-set segmentation [6] have shown to produce state of the art results. Thereby the object's pose is determined in an interleaved process, comprising a pixel-wise segmentation of the object's silhouette based on a statistical foreground/background model and its alignment with a level-set representation of the rendered silhouette of a model of the object.

Early region-based methods were too computationally demanding for real-time applications [22, 4, 23]. The first real-time capable algorithm called PWP3D was presented by Prisacariu and Reid in [18]. It was recently improved by Tjaden *et al*. in [26] which enhanced the pose-optimization strategy to better handle fast rotations and scale changes and further reduced its overall runtime. In parallel to this, Hexer and Hagege in [7] proposed a localized segmentation model to PWP3D, that improves its performance with cluttered backgrounds and heterogeneous object surfaces.

A segmentation strategy similar to the local color histograms used in [7] was presented within a contour edge-based approach by Seo *et al*. in [24] and further improved by Wang *et al*. in [28]. Although it performs well in cluttered scenes, the approach struggles with motion blur and defocus and is limited to slow movement for real-time use.

All of the aforementioned methods are strictly designed for tracking and do not provide a solution for pose detection. For real-time applications, pose detection approaches based on 2D template matching are currently yielding the best results [10, 9, 11, 8, 21, 13]. Thereby, the templates are projections of the model at varying perspectives. Probably the most popular and still the most generic template-based method for real-time use is LINE-2D, introduced by Hinterstoisser *et al*. in [8] and improved by Rios-Cabrera and Tuytelaars in [21]. Here, both the input image and the templates are transformed into so-called gradient response maps, by computing the dominant orientations of RGB intensity gradients. LINE-2D and similar approaches are usually demonstrated in scenarios where the objects are assumed to be standing or lying on a surface. This allows to only include the upper hemisphere for outer image plane rotations and a small range of inner image plane rotation during template generation, instead of the full hemisphere that is needed in case of *e.g*. handheld objects which we are targeting in this paper. In addition, the pose accuracy of these methods is constrained to the resolution of the templates and they do not incorporate a solution for pose refinement or tracking.

Latest results on pose detection using RGB images were based on learning of so-called object coordinates using random forests presented by Brachmann *et al*. in [3] as an improvement of [2]. To our best knowledge this approach currently yields state of the art results, but its runtime performance is far from real-time capable.

1.2. Contribution

We present a novel approach to real-time pose tracking and pose detection of rigid objects. Our region-based approach incorporates the improved optimization procedure presented in [26] and combines it with the localized segmentation idea presented in [7]. The core novelty of our method is to attach local color histograms to the object's surface. This allows to enforce temporal consistency within each of them which improves the robustness of pose tracking in case of dynamic occlusion, motion of both the camera as well as the object and light changes in cluttered scenes superior to the current state of the art. We also show that the resulting temporally consistent, local color histograms (*tclc-histograms*) form a novel object descriptor that can be used for pose detection. This has not been previously addressed by other level-set-based pose estimation approaches. Thereby, a unique similarity measure is used for both template matching and pose optimization. We also introduce a corresponding novel image representation called *posterior response maps* in order to speed up our pose detection approach.

The rest of the paper is structured as follows. In section 2 we give a detailed mathematical and algorithmic description of the proposed method. An overview of the key technical details of our implementation is presented in section 3, followed by an experimental evaluation in section 4. In section 5 we conclude with a final discussion of the proposed system and potential future work.

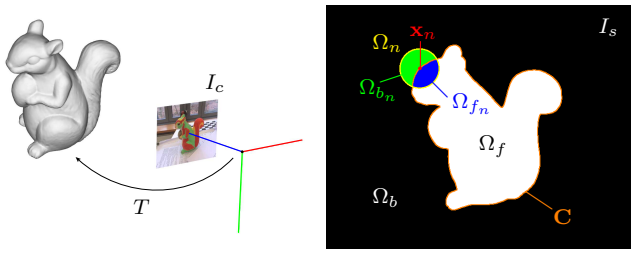

Figure 2. Overview of our pose estimation setting. Left: The object pose T relative to a camera based on color image I_c and a 3D model of the object. Right: Silhouette I_s generated by projecting the surface model into the image plane using an estimated pose T.

2. Method

We represent an object by a dense surface model consisting of vertices $\mathbf{X}_n := (X_n, Y_n, Z_n)^\top \in \mathbb{R}^3$, $n = 1 \dots N$ building a triangular mesh. A camera color image is denoted by $I_c : \Omega \to \mathbb{R}^3$, with $\Omega \subset \mathbb{R}^2$ being the image domain (see Fig. 2). Accordingly, a synthetic silhouette projection of the model is given by $I_s : \Omega \to \{0, 1\}$ that yields a contour $\mathbf{C}$ splitting the image into a foreground region $\Omega_f \subset \Omega$ and a background region $\Omega_b = \Omega \setminus \Omega_f$.

The pre-calibrated and fixed intrinsic matrix of the camera is denoted by K, the pose of an object relative to the camera by $T \in \mathbb{SE}(3)$. All camera images I_c are remapped removing non-linear distortion such that the perspective projection of a surface point to an image point is given by $\mathbf{x} = \pi(K(T\tilde{\mathbf{X}})_{3\times1}) \in \mathbb{R}^2$, with $\pi(\mathbf{X}) = (X/Z, Y/Z)^\top$, being $\tilde{\mathbf{X}} = (X, Y, Z, 1)^\top$ the homogenous extension of $\mathbf{X}$. The color of a pixel at $\mathbf{x}$ is denoted by $\mathbf{y} = I_c(\mathbf{x})$.

For pose optimization we model the rigid body motion using twists $\hat{\xi} \in \mathfrak{se}(3)$ parametrized by $\xi \in \mathbb{R}^6$. A twist can be mapped to its corresponding rigid body transform via $\exp(\hat{\xi}) \in \mathbb{SE}(3)$.

2.1. Segmentation

Our image segmentation strategy is based on local color histograms with extension to pose tracking. The core idea is to build a segmentation model from multiple overlapping color histograms and update it for each frame after pose optimization. Each of these histograms correspond to a circular image region $\Omega_n := \{\mathbf{x} \text{ with } |\mathbf{x} - \mathbf{x}_n| < r\}$ with radius r, centered at pixel $\mathbf{x}_n \in \mathbf{C}$ as proposed by Lankton and Tannenbaum in [15]. Thus, each Ω_n is split into a foreground region $\Omega_{f_n} \subset \Omega_f$ and background region $\Omega_{b_n} \subset \Omega_b$ determined by I_s (see Fig. 2). This allows to compute foreground and background color histograms for each region.

In our case we are using the RGB color model with a quantization of 32 values per channel and $r = 40$ px regardless of the object's distance to the camera. As presented by Bibby and Reid in [1], pixel-wise local foreground P_{f_n} and background P_{b_n} posteriors can be calculated using Bayes rule, with prior probabilities η_{f_n} and η_{b_n} as well as color likelihoods $P^t(\mathbf{y}|M_{f_n})$ and $P^t(\mathbf{y}|M_{f_n})$ at time t as

$$P_{i_n}(\mathbf{y}) = \frac{P^t(\mathbf{y}|M_{i_n})}{\eta_{i_n} P^t(\mathbf{y}|M_{i_n}) + \eta_{j_n} P^t(\mathbf{y}|M_{j_n})} \tag{1}$$

where $i \neq j \in \{f, b\}$.

We associate each local histogram with a 3D surface point in order to memorize and identify them and thereby enable temporal consistency (see Fig. 3 a)). In contrast to [7] where the 2D histogram centers are computed as an arbitrary subset of $\mathbf{C}$ for each individual frame, we use projected 3D mesh vertices, *i.e.* $\mathbf{x}_n = \pi(K(T\tilde{\mathbf{X}}_n)_{3\times1})$. This correspondence between surface points and histograms enables to enforce temporal consistency of them as

$$P^t(\mathbf{y}|M_{i_n}) = (1-\alpha_i)P^{t-1}(\mathbf{y}|M_{i_n}) + \alpha_i P^t(\mathbf{y}|M_{i_n}) \tag{2}$$

where $i \in \{f, b\}$. Here the current color likelihoods are computed from a silhouette projection resulting from pose optimization based on the previous segmentation model. This strategy was originally used in PWP3D for the global segmentation model, using learning rates of $\alpha_f = 0.05$ and $\alpha_b = 0.02$. Since the localized model captures spatial variations a lot more precisely our experiments showed that higher learning rates of $\alpha_f = 0.1$ and $\alpha_b = 0.2$ can be used, enabling faster adaptation to dynamic changes.

2.2. Pose Tracking

For pose optimization, based on a rough pose estimate either from the previous frame or pose detection, [18, 26] suggest to measure the discrepancy between the posterior segmentation of the current image I_c and a synthetic object silhouette projection by

$$E_{global} = -\sum_{\mathbf{x}\in\Omega} \log\big(H_e(\Phi(\mathbf{x}))P_f(\mathbf{y}) + (1 - H_e(\Phi(\mathbf{x})))P_b(\mathbf{y})\big). \tag{3}$$

Thereby, Φ is a level-set embedding of the pose given by a 2D Euclidian signed distance transform of the silhouette

$$\Phi(\mathbf{x}) = \begin{cases} -d(\mathbf{x}, \mathbf{C}) & \forall \mathbf{x} \in \Omega_f \\ d(\mathbf{x}, \mathbf{C}) & \forall \mathbf{x} \in \Omega_b \end{cases}, \tag{4}$$

with $d(\mathbf{x}, \mathbf{C}) = \min_{\mathbf{c}\in\mathbf{C}} |\mathbf{c}-\mathbf{x}|$, $H_e(x)$ is a smoothed Heaviside function and P_f, P_b are pixel-wise posteriors using a single global foreground and background color histogram.

In [7] this formulation was adapted to

$$E_{localized} = \frac{1}{N}\sum_{n=1}^{N} E_n, \tag{5}$$

where

$$E_n = -\sum_{\mathbf{x}\in\Omega} \log\big(H_e(\Phi(\mathbf{x}))P_{f_n}(\mathbf{y}) + (1 - H_e(\Phi(\mathbf{x})))P_{b_n}(\mathbf{y})\big)\mathbf{B}_n(\mathbf{x}) \tag{6}$$

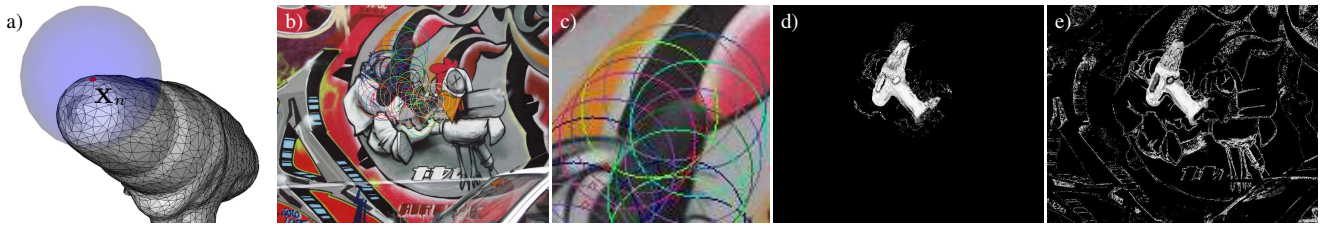

Figure 3. Temporally consistent, local color histogram segmentation. a) Schematic 3D visualization of a tclc-histogram attached to a mesh vertex of 3D driller model. b) Example image from a synthetic sequence where the textured model is rendered and animated on a static but cluttered background. The local histogram regions are depicted by colored circles corresponding to their vertex index around small red circles at their centers along the object's contour estimated in the previous frame. b) Detailed view of the dark tip of the driller in front of a dark background region. d) Average posterior probabilities $\bar{P}_f(\mathbf{x}) - \bar{P}_b(\mathbf{x})$. e) $P_f(\mathbf{x}) - P_b(\mathbf{x})$ from global color histograms.

uses local histograms and a corresponding masking function

$$\mathbf{B}_n(\mathbf{x}) = \begin{cases} 1 & \forall \mathbf{x} \in \Omega_n \\ 0 & \forall \mathbf{x} \notin \Omega_n \end{cases}, \tag{7}$$

indicating whether a pixel $\mathbf{x}_c$ lies within that histogram or not. Here, each individual local energy E_n (6) again is only influenced by the posteriors P_{f_n} and P_{b_n} computed from a single local foreground and background histogram. Thus, locally they suffer from the same segmentation problems as the global energy (3). This becomes a problem when the local background color is similar to the local object surface color (see *e.g.* Fig. 3 c)) resulting in many misclassified pixels (outliers) that have a negative impact on the overall energy term (see Fig. 4 a) - c)). Thus, in order to improve

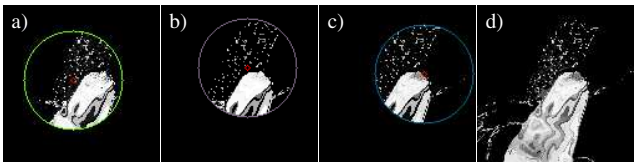

Figure 4. A comparison in the image region of Fig. 3 c) between pixel-wise posteriors computed from a single Ω_n and those averaged over all Ω_n. a) - c) Segmentation $P_{f_n}(\mathbf{y}) - P_{b_n}(\mathbf{y})$ for different local regions. d) The averaged posterior probabilities $\bar{P}_f(\mathbf{x}, \mathbf{y}) - \bar{P}_b(\mathbf{x}, \mathbf{y})$. Here especially the foreground segmentation is significantly more reliable for the average posteriors.

the quality of the energy term per pixel we suggest to first compute the average posteriors from all local histograms instead of computing the average energy over all local regions Ω_n, (see Fig. 4 d)). This leads to a slightly different energy formulation, changing (3) into

$$\begin{aligned} E = -\sum_{\mathbf{x}\in\Omega} \log\big(&H_e(\Phi(\mathbf{x}))\bar{P}_f(\mathbf{x},\mathbf{y}) \\ &+ (1 - H_e(\Phi(\mathbf{x})))\bar{P}_b(\mathbf{x},\mathbf{y})\big) \end{aligned} \tag{8}$$

with

$$\bar{P}_i(\mathbf{x},\mathbf{y}) = \frac{1}{\sum_{n=1}^{N} \mathbf{B}_n(\mathbf{x})} \sum_{n=1}^{N} P_{i_n}(\mathbf{y}) \mathbf{B}_n(\mathbf{x}), \tag{9}$$

where $i \in \{f, b\}$, being the posterior probabilities per pixel averaged over all corresponding histograms. Although this may seem like a minor change, our experiments show that it leads to a significant increase in robustness and accuracy for both pose tracking and detection in cluttered scenes.

Since $\mathbf{x} = \pi(K(\exp(\hat{\xi})T\tilde{\mathbf{X}})_{3\times1})$ pose optimization can be performed by minimizing E with respect to the pose parameters given as twist coordinates. The gradient of (8) is then given by

$$\frac{\partial E}{\partial \xi} = -\sum_{\mathbf{x}\in\Omega} \frac{\bar{P}_f - \bar{P}_b}{H_e(\Phi)\bar{P}_f + (1 - H_e(\Phi))\bar{P}_b} \frac{\partial H_e(\Phi)}{\partial \xi} \tag{10}$$

where $\bar{P}_f = \bar{P}_f(\mathbf{x},\mathbf{y})$, $\bar{P}_b = \bar{P}_b(\mathbf{x},\mathbf{y})$ and $\Phi = \Phi(\mathbf{x})$. Based on the iterative optimization scheme presented in [26] pose update is given by

$$T \leftarrow \exp(\hat{\Delta\xi})T. \tag{11}$$

with the update step

$$\Delta\xi = -\left(\sum_{\mathbf{x}\in\Omega} J^\top J\right)^{-1} \sum_{\mathbf{x}\in\Omega} J^\top, \tag{12}$$

where $J = J(\mathbf{x}, \xi_0) = \partial E(\mathbf{x}, \xi_0)/\partial\xi$ is the 1×6 per pixel Jacobi vector at $\xi_0 = \mathbf{0}^\top$.

2.3. Pose Detection

For each frame we evaluate (8) after pose optimization. If $E/|\Omega| > t$, we consider the tracking to be lost. Especially for handheld objects no assumptions can me made about the pose, when it becomes visible again, for example when it is moved outside the field of view and back in from a different side or held upside down. Once the tracking has been lost, we thus perform a full search for the object and its current pose in each subsequent frame until pose recovery.

Our strategy for re-localization is generally related to current state of the art template matching-based pose detection methods [11, 8, 21, 13]. A template view consists of a level-set Φ from that pose and an associated set of

tclc-histograms along $\mathbf{C}$. Thereby, the orientation of the so-called base templates is given by one of the 12 corner vertices of an icosahedron, defining the outer image plane rotation (see Fig. 5). The base templates are augmented with four different rotations of $0°, 90°, 180°$ and $270°$ within the image plane. In order to cover different scales, each of these 48 base orientations is used to generate a template at a close, an intermediate and a far distance to the camera, resulting in overall 144 base templates. As soon as all histograms corresponding to a template have been filled, either during a dedicated training stage or regular tracking, the template can be used for re-localization.

2D template matching is started at the 4th level of an image pyramid (80×64 px resolution), where we perform an exhaustive search for all base templates by evaluating (8). This first matching step is split into two stages in order to reduce the number of function evaluations and thereby improve the runtime. The relatively wide basin of convergence for in-plane translation of (5) and (8) (see [7] for a detailed analysis) allows us to use a stride of 4 pixels in the first stage to get a rough estimate of the 2D location of each template. In the second stage, this location is refined considering all pixels in a 5×5 neighborhood around the coarse estimate. Inspired by the gradient response maps of LINE-2D [8], we introduce *posterior response maps*, a novel image representation based on tclc-histograms that allows us to skip image regions by a rough statistical pixel-wise foreground or background membership decision. We define a posterior response map I_p as a binary representation of the current camera image I_c by

$$I_p(\mathbf{x}) = \begin{cases} 1 & \text{if} \quad \bar{P}_f(I_c(\mathbf{x})) > \bar{P}_b(I_c(\mathbf{x})) \\ 0 & \text{else} \end{cases} \quad \forall \mathbf{x} \in \Omega, \quad (13)$$

with

$$\bar{P}_i(\mathbf{y}) = \frac{1}{N} \sum_{n=1}^{N} P_{i_n}(\mathbf{y}), \quad (14)$$

where $i \in \{f, b\}$, being the average posterior probabilities over all histograms regardless of the pixels location. Given this representation, we can compute the binary overlap between the silhouette mask I_s of a template at each potential 2D matching location and I_p. If this intersection is less than 50% of the area of Ω_f, we skip this location without evaluating the energy function, which reduces the number of necessary computations. To further refine the orientation, we continue the search in the next higher resolution of the image pyramid (160×128 px). Thereby, we discard two out of three distances per base templates and only keep that with the best matching score. For the remaining base templates the matching score of the so-called neighboring orientation templates is computed at the previously estimated 2D location of its corresponding base template. These templates were generated at the corner vertices resulting from

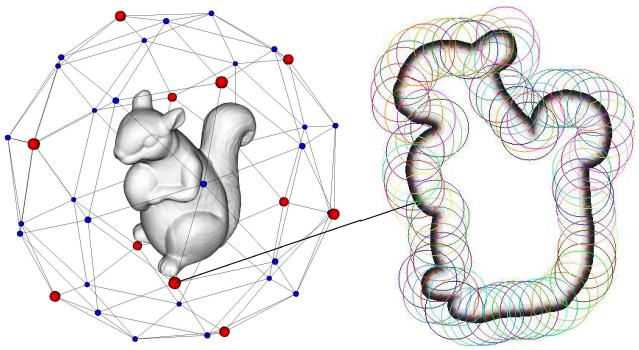

Figure 5. The template views used for pose detection. Left: A subdivided icosahedron generates the outer plane rotation of the template views. Red dots indicate vertices corresponding to the base templates, blue dots indicate those provided by subdivision used for the neighboring templates. Right: An example base template visualized by the corresponding $H_e(\Phi)$ (grey pixels) with the local histogram regions depicted (colored circles) along the contour.

sub-dividing the icosahedron (see Fig. 5) in order to finer sample the outer plane rotation. They are augmented with 12 in-plane rotations of $0°, 30°, \ldots, 330°$. Each base template is associated with 18 neighboring templates in the next pyramid level. Those include itself and that corresponding to the 5 closest vertices of the subdivided icosahedron, each with the same in-plane rotation as the base template as well as those with $\pm30°$. The poses of the 4 best matches of all neighboring templates are finally optimized as described in section 2.2 with three times the number of iterations compared to frame-to-frame optimization.

If $E/|\Omega|$ corresponding to one of these optimized poses is smaller than the threshold t, we consider the re-localization as successful and switch back to frame-to-frame tracking. Increasing the value of t thus results in both the tracking and re-localization to be more tolerant to slight misalignment and false positive detections. This can help to improve the overall performance in case of heterogenous objects and background clutter if continuity is more important than precision. In our experiments we chose $t \in [0.5, 0.6]$.

3. Implementation

Our C++ implementation is oriented towards the multi-resolution image pyramid approach presented in [26] with an additional 4th level only used within template matching for pose detection. We are using OpenGL exclusively for rendering the object silhouette I_s and the corresponding depth buffer I_d while the major image processing steps are performed in parallel per image row on the CPU during tracking. Template matching within pose detection is performed in parallel per template for each step and the posterior response maps are speeded up with an LUT. The key idea to efficiently build and update the localized segmentation model is to process each histogram region in parallel

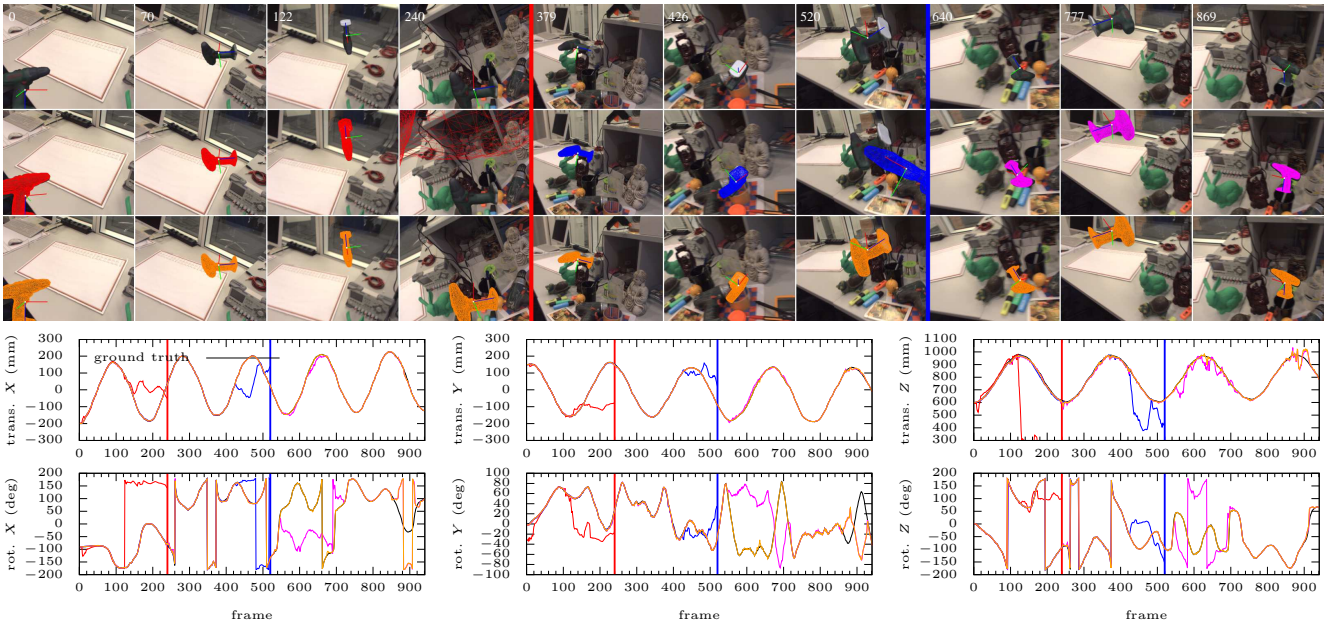

Figure 6. Top: Results of our ground truth pose tracking experiment (1st row: virtually augmented input images with coordinates axes. 2nd row: Result of [26] (red), tracking without temporal consistency (blue) and tclc-histograms using (5) (magenta). 3rd row: Proposed method (orange).). Bottom: Determined pose parameters. Plot color corresponds to mesh color drawn in the example frames with respect to the algorithm used. The red and blue vertical lines between the frames and in the plots mark the tracking losses of the respective approaches.

using a Bresenham circle to scan the corresponding pixels.

In practice, when computing the 2D histogram centers by projecting all mesh vertices $\mathbf{X}_n$ onto pixels $\mathbf{x}_n$ in the image plane, we consider those with $\mathbf{x}_n \in \mathbf{C}$ as well as $d(\mathbf{x}_n, \mathbf{C}) \leq \lambda r$ (we use $\lambda = 0.1$), in order to ensure that they evenly sample the contour. For runtime reasons, since this can lead to a large number of histograms that have to be updated, we randomly only pick 100 centers per frame. This Monte Carlo approach requires the mesh vertices $\mathbf{X}_n$ to be uniformly distributed across the model in order to evenly cover all regions. They should also be limited in number to ensure that all histograms will get updated regularly. It is as well important for constraining the runtime of our algorithm, especially when computing (14) for the posterior response maps. We therefore use two different 3D mesh representations for the model. The original mesh is used to render exact silhouette views regardless of the mesh structure while a reduced (we use a maximum of 5000 vertices) and evenly sampled version of the model is used for computing the centers of the 2D histograms.

As suggested in [18], we designed H_e such that we only need to consider a band of ± 8 pixels around $\mathbf{C}$ for minimizing and evaluating (8), regardless of the pyramid level. We also use an inverse depth buffer I_d^{-1} corresponding to the back of the object for pose optimization in order to increase robustness. Although, using OpenGL this requires to render the model twice with different depth buffer settings, it does not significantly deteriorate the overall performance.

4. Evaluation

We evaluate the performance of our system on a laptop with an Intel Core i7 quad core CPU @ 2.8 GHz and an AMD Radeon R9 M370X GPU. The camera images used for our custom experiments (featured in our supplementary video) are of 640×512 px resolution.

4.1. Runtime Performance

Our implementation runs at approximately 30 – 80 Hz when tracking a single object depending on its distance to the camera, that impacts the number of processed pixels. These timings are only a few milliseconds higher than those presented in [26] due to the more complex segmentation model. Pose detection runs at 4 - 10Hz when including the full hemisphere for outer plane and 360° for in-plane rotation. In cases when the object is not present in the current image it can go up to 100Hz depending on how many regions can directly be skipped with help of the posterior response map. Note that the runtime of this approach decreases almost linearly depending on the number of templates in case of scenarios that are more constrained for viewpoints (*e.g.* only the upper hemisphere).

4.2. Experiments

In our first experiment we evaluate our method for tracking without loss detection in a semi-synthetic image sequence that allows us to compare with ground truth pose information. We used the textured driller model provided in

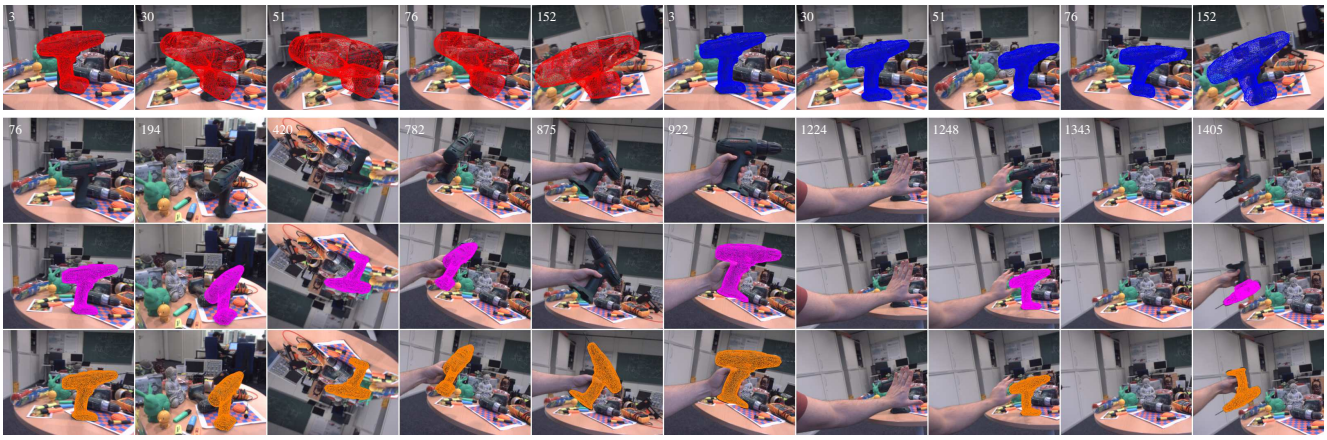

Figure 7. Results in the real data sequence (colors match Fig. 6). 1st row: Tracking failure of [26] (left) and local segmentation without temporal consistency (right) in the beginning. 2nd row: RGB input frames. 3rd row: Our approach using (5). 4th row: Proposed method.

the public single instance pose detection data set of Hinterstoisser *et al*. [11] and composed synthetic renderings of it with an image sequence captured by our camera (see Fig. 6). For the sequence all 6DOF of the driller pose were animated using harmonic motions with randomized step sizes in order to simulate slight hand jitter. The renderings include local lighting. For compositing we slightly blurred the contour of the renderings to smooth the transition between the object and the background. We compare our method to that of [26] using global histograms, localized histogram segmentation without temporal consistency as well as our approach using (5) instead of (8).

The global segmentation leads to a tracking loss as soon as the object moves close to a background region that is similarly dark as the tip of the driller (*e.g*. frame 122), even though the tip itself is not near it. Next, localized segmentation without temporal consistency gets stuck in another driller of similar color present in the background while moving across it (*e.g*. frame 426) eventually also leading to a tracking loss. The proposed method as well as the one using (5) are able to successfully track the object in the whole sequence. Here (5) suffers twice from silhouette pose ambiguity for rotation (*e.g*. frame 640 and 869) while this only happens once (*e.g*. frame 869) for the proposed method. Before this ambiguity occured the RMSE of our approach in (x, y, z) direction is $(1.2 \pm 0.9\,\text{mm}$, $1.3 \pm 1.1\,\text{mm}$, $7.5 \pm 5.7\,\text{mm})$ for translation and $(2.3 \pm 2.3°, 1.3 \pm 1.2°, 1.1 \pm 2.0°)$ for rotation around the (x, y, z) axes of the camera.

The second experiment demonstrates the performance of the whole system including tracking and detection in a real data scenario (see Fig. 7). We use the same 3D model as in the first experiment to estimate the pose of an identically constructed real drill held in hand. The sequence contains a cluttered background including another drill similar in shape and color in the background, complex motion of both the camera and the object, partial and total occlusion. Furthermore, the drill was equipped with a drill bit that is not part of the 3D model.

For evaluation we enabled loss detection and pose recovery for our method using both energy functions, which is not possible without temporal consistency and was not included by [26]. In this scene the global model does not provide a sufficient segmentation to initially estimate the pose due to the large amount of background clutter and gets lost immediately. The localized segmentation without temporal consistency again struggles with the background clutter and also gets lost at the beginning of the sequence. While rotating the drill 360° in hand our approach using (5) gets lost at one point (*e.g*. frame 782) and is not able to recover until the rotation is complete and the side of the drill that was previously tracked becomes visible again (*e.g*. frame 922). At the end of the sequence the drill is totally occluded leading to a tracking loss from which our method successfully recovers regardless of the energy used. A second loss occurs caused by moving the drill outside the field of view of the camera and moving it back in in an upside down pose. This time, pose detection using (8) successfully recovers while (5) leads to a misalignment which is characteristic for the results of the following pose detection experiment.

We evaluate the performance of our re-localization strategy in the full data set of [11], including 13 of the 15 provided models (for two of the models no proper triangle mesh was given) and the RGB color images. For this we simulate the training stage of the tclc-histograms for each model by projecting it with the given ground truth pose information in a randomly picked subset of the test images. After this we try to detect the pose of the object in the entire set of test images using our re-localization method. Fig. 8 shows the results for each individual model trained with 25, 50, 100, 200 images. For evaluation we are using four different error metrics. We consider pose detection com-

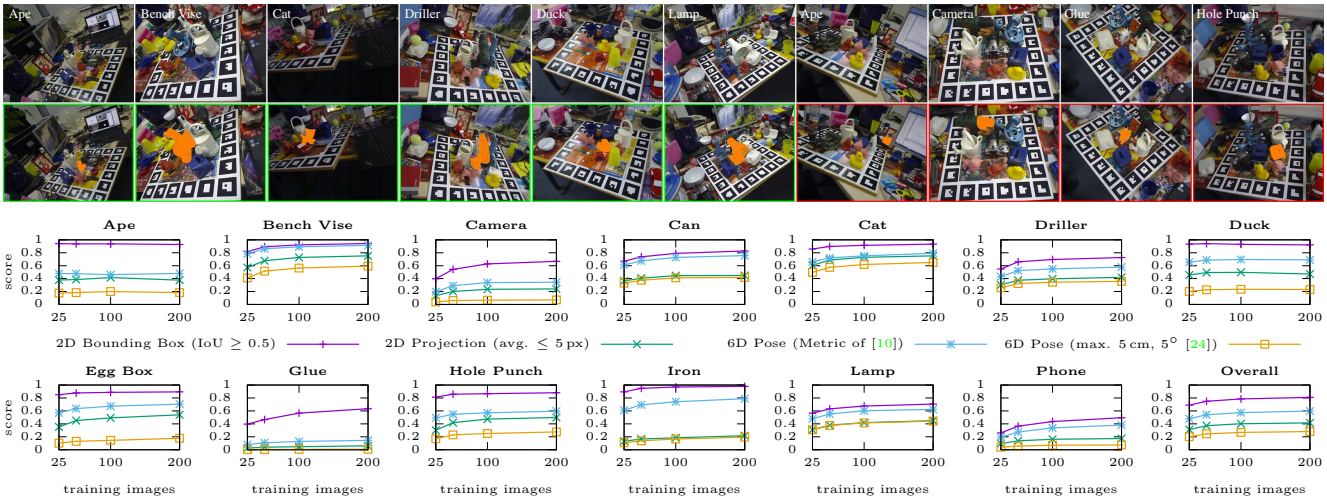

Figure 8. Top: Examples of our pose detection experiment (1st row: Original input images from the data set. 2nd row: Visual results of exact pose detection (green outline) and failure cases (red outline).). Bottom: Results separately visualized for each model as well as the overall average of all models of the ACCV data set with respect to the number of randomly picked training images averaged over 10 runs.

pared to ground truth successful if either the intersection over union (IoU) of the 2D bounding boxes is at least 50%, the average 2D projection error is below 5 px, the 6D pose metric used in [11] is below 10% of the model diameter or the maximum pose difference below 5 cm for translation and 5° for rotation [25]. Here the latter is most important for pose recovery. Even for very small amounts of training images our method performs well for most of the models and the improvement quickly converges when their number is increased. As expected the exact pose can be recovered especially well for models with a distinct color and silhouette such as the bench vise or the cat while it gets confused with regions of similar shapes and color. In Table 1 we compare

Table 1. Overall detection score comparison in the ACCV dataset.

Metric	Ours w. (8)	Ours w. (5)	[8] LINE-2D	[3] w. L_1 reg.
2D IoU	78.5%	68.5%	86.5%	**97.5%**
2D proj.	40.2%	28.8%	20.9%	**73.7%**
6D [11]	**57.3%**	49.8%	24.2%	50.2%
6D [25]	26.7%	18.6%	8.1%	**40.6%**

our overall results using 100 training images to those presented in [3] including their LINE-2D implementation. Our approach overall performs significantly better than LINE-2D, which runs a comparable speed (∼10 Hz). However, our method cannot be directly compared to it for two reasons. The first is, that our method was trained with images from the dataset including the backgrounds, which is required in order to fill our tclc-histograms while LINE-2D can be trained without scene knowledge. We consider this constraint acceptable for our approach, since it is mainly designed to recover from temporal tracking losses in known scenes other than detecting objects in arbitrary scenes. On the other hand, LINE-2D is only using templates of the upper hemisphere for outer plane rotation and ±80° in-plane rotation, while our approach is potentially able to detect the object at all possible rotations. Although our method performs worse in most cases compared to [3], we want to point out that this method also uses 15% (∼170 images) of the original images per model for training their random forests, while detection takes 1 – 2 seconds per image.

5. Conclusion and Future Work

In this paper we presented a novel combined solution to real-time pose tracking and detection based on so-called tclc-histograms. The method is able to recover from temporal tracking losses at arbitrary poses and thereby can even handle small occlusion in case of *e.g.* handheld objects. The robustness of our tracking is superior to the current state of the art as demonstrated in challenging experimental scenarios. Our approach also yields better results for pose detection than the current state of the art with comparable runtime while including all possible views and not only those of the upper hemisphere. The main downside of our pose detection approach is that training our templates requires scene or background knowledge. This prevents its application to the classical pose detection task, where a known object is to be detected in arbitrary scenes. For the scenarios we are targeting where either the camera or the object is static, we consider this constraint acceptable, especially because new objects or scenes can be trained within a couple of seconds. Both tracking and detection suffer from pose ambiguities that are inherent to methods only relying on silhouette information. We plan to resolve these in the near future by incorporating a photometric term that regards the inner structure of the objects.

References

[1] C. Bibby and I. D. Reid. Robust real-time visual tracking using pixel-wise posteriors. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2008. 3

[2] E. Brachmann, A. Krull, F. Michel, S. Gumhold, J. Shotton, and C. Rother. Learning 6d object pose estimation using 3d object coordinates. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2014. 2

[3] E. Brachmann, F. Michel, A. Krull, M. Y. Yang, S. Gumhold, and C. Rother. Uncertainty-driven 6d pose estimation of objects and scenes from a single RGB image. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 2, 8

[4] 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 (TPAMI)*, 32(3):402–415, 2010. 2

[5] C. Choi and H. I. Christensen. RGB-D object tracking: A particle filter approach on GPU. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2013. 2

[6] D. Cremers, M. Rousson, and R. Deriche. A review of statistical approaches to level set segmentation: Integrating color, texture, motion and shape. *International Journal of Computer Vision (IJCV)*, 72(2):195–215, 2007. 2

[7] J. Hexner and R. R. Hagege. 2d-3d pose estimation of heterogeneous objects using a region based approach. *International Journal of Computer Vision (IJCV)*, 118(1):95–112, 2016. 2, 3, 5

[8] S. Hinterstoisser, C. Cagniart, S. Ilic, P. F. Sturm, N. Navab, P. Fua, and V. Lepetit. Gradient response maps for real-time detection of textureless objects. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 34(5):876–888, 2012. 2, 4, 5, 8

[9] S. Hinterstoisser, S. Holzer, C. Cagniart, S. Ilic, K. Konolige, N. Navab, and V. Lepetit. Multimodal templates for real-time detection of texture-less objects in heavily cluttered scenes. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2011. 2

[10] S. Hinterstoisser, V. Lepetit, S. Ilic, P. Fua, and N. Navab. Dominant orientation templates for real-time detection of texture-less objects. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE Computer Society, 2010. 2

[11] S. Hinterstoisser, V. Lepetit, S. Ilic, S. Holzer, G. R. Bradski, K. Konolige, and N. Navab. Model based training, detection and pose estimation of texture-less 3d objects in heavily cluttered scenes. In *Proceedings of the Asian Conference on Computer Vision (ACCV)*, 2012. 2, 4, 7, 8

[12] W. Kehl, F. Milletari, F. Tombari, S. Ilic, and N. Navab. Deep learning of local RGB-D patches for 3d object detection and 6d pose estimation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2016. 2

[13] Y. Konishi, Y. Hanzawa, M. Kawade, and M. Hashimoto. Fast 6d pose estimation from a monocular image using hierarchical pose trees. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2016. 2, 4

[14] A. Krull, F. Michel, E. Brachmann, S. Gumhold, S. Ihrke, and C. Rother. 6-dof model based tracking via object coordinate regression. In *Proceedings of the Asian Conference on Computer Vision (ACCV)*, 2014. 2

[15] S. Lankton and A. Tannenbaum. Localizing region-based active contours. *IEEE Transactions on Image Processing*, 17(11):2029–2039, 2008. 3

[16] V. Lepetit and P. Fua. Monocular model-based 3d tracking of rigid objects: A survey. *Foundations and Trends in Computer Graphics and Vision*, 1(1), 2005. 1

[17] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision (IJCV)*, 60(2):91–110, 2004. 1

[18] V. A. Prisacariu and I. D. Reid. PWP3D: real-time segmentation and tracking of 3d objects. *International Journal of Computer Vision (IJCV)*, 98(3):335–354, 2012. 2, 3, 6

[19] C. Y. Ren, V. A. Prisacariu, O. Kähler, I. D. Reid, and D. W. Murray. 3d tracking of multiple objects with identical appearance using RGB-D input. In *Proceedings of the International Conference on 3D Vision (3DV)*, 2014. 2

[20] C. Y. Ren, V. A. Prisacariu, D. W. Murray, and I. D. Reid. STAR3D: simultaneous tracking and reconstruction of 3d objects using RGB-D data. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2013. 2

[21] R. Rios-Cabrera and T. Tuytelaars. Discriminatively trained templates for 3d object detection: A real time scalable approach. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2013. 2, 4

[22] B. Rosenhahn, T. Brox, and J. Weickert. Three-dimensional shape knowledge for joint image segmentation and pose tracking. *International Journal of Computer Vision (IJCV)*, 73(3):243–262, 2007. 2

[23] C. Schmaltz, B. Rosenhahn, T. Brox, and J. Weickert. Region-based pose tracking with occlusions using 3d models. *Machine Vision and Applications*, 23(3):557–577, 2012. 2

[24] B. Seo, H. Park, J. Park, S. Hinterstoisser, and S. Ilic. Optimal local searching for fast and robust textureless 3d object tracking in highly cluttered backgrounds. *IEEE Transactions on Visualization and Computer Graphics*, 20(1):99–110, 2014. 2

[25] J. Shotton, B. Glocker, C. Zach, S. Izadi, A. Criminisi, and A. W. Fitzgibbon. Scene coordinate regression forests for camera relocalization in RGB-D images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2013. 8

[26] H. Tjaden, U. Schwanecke, and E. Schömer. Real-time monocular segmentation and pose tracking of multiple objects. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2016. 2, 3, 4, 5, 6, 7

[27] L. Vacchetti, V. Lepetit, and P. Fua. Combining edge and texture information for real-time accurate 3d camera tracking. In *Proceedings of the IEEE and ACM International Symposium on Mixed and Augmented Reality (ISMAR)*, 2004. 1

[28] G. Wang, B. Wang, F. Zhong, X. Qin, and B. Chen. Global optimal searching for textureless 3d object tracking. *The Visual Computer*, 31(6-8):979–988, 2015. 2