In Proceedings of 6th European Conference on Mobile Robots (ECMR), Barcelona, Spain, 2013.

# Joint Detection and Pose Tracking of Multi-Resolution Surfel Models in RGB-D

Manus McElhone Jörg Stückler Sven Behnke

*Abstract*— **We propose a particle filter framework for the joint detection, pose estimation, and real-time tracking of objects in RGB-D video. We do not rely on the availability of CAD models, but employ multi-resolution surfel maps as a concise representation of object shape and texture that is acquired through SLAM. We propose to initialize the particle belief for tracking with pose votes cast from matching colored surfel-pair features at multiple resolutions. Multi-hypothesis tracking then finds the most consistent track over time. We utilize efficient registration of RGB-D images to the model to obtain improved proposals for particle filtering which greatly enhances tracking accuracy. We evaluate our approach on a publicly available RGB-D object tracking dataset, and show high rates of detection and good tracking performance with respect to various speeds of camera motion and occlusions.**

## I. INTRODUCTION

Object tracking is a common problem encountered in equipping mobile robots with autonomous capabilities. It is important to be able to detect and track object pose to interact with the objects or to understand actions on them. While many approaches consider initial object detection or tracking separately, we propose a coherent framework that both globally localizes objects and tracks them accurately and in real-time on a CPU.

Our approach keeps track of 3D object models using a robust particle filter. We use multi-resolution surfel maps (MRSMap, [1]) to provide a concise 3D description of objects which is obtained beforehand using SLAM. The scene as viewed in the current RGB-D image is also represented as a MRSMap which permits efficient registration between object model and the image. To facilitate efficient particle filtering with a small number of particles despite the six-dimensional pose space, we obtain improved proposals using the registration method. Tracking is initialized with pose hypotheses determined in a voting-based framework. To this end, we extend the MRSMap framework with point-pair features which describe geometry and texture locally at multiple resolutions. By matching and aligning such features between scene and object model we are able to determine object pose hypotheses. This way, the continuous search space over object poses is reduced to only a few hypotheses which are then verified by the particle filter during tracking.

In experiments, we evaluate the robustness and accuracy of our method in settings with varying degree of difficulty with regards to object motion, clutter, and occlusions. We compare our method with existing approaches and demonstrate state-of-the-art performance.

All authors are with Autonomous Intelligent Systems, Computer Science Institute VI, University of Bonn, 53113 Bonn, Germany mcelhone at cs.uni-bonn.de, stueckler at ais.uni-bonn.de, behnke at cs.uni-bonn.de

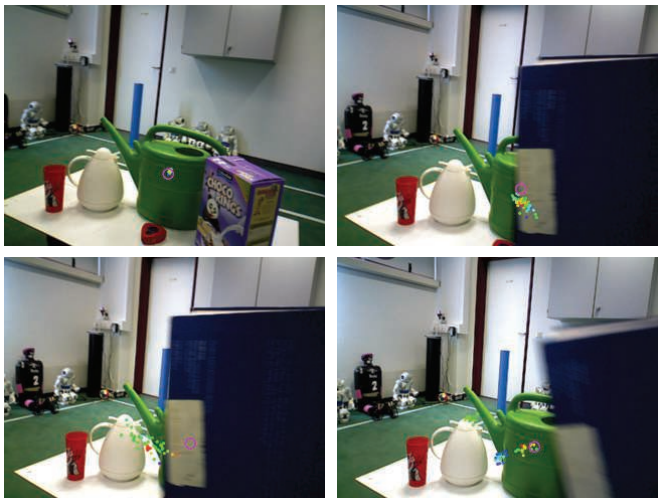


Fig. 1. Our model-based 6-DoF object tracking approach robustly handles strong occlusions by reinitializing the tracker with our detection method (bottom right). Particle poses are projected as object locations into the image (dots, color codes likelihood, blue: low, red: high; circle: ground truth).

## II. RELATED WORK

*1) Object Detection and Pose Estimation:* Local invariant 2D features such as SIFT [2] or SURF [3], which describe the local texture surrounding interest points in the image, have been widely and successfully applied for the detection and pose estimation of textured objects. Methods which rely solely on an object's texture, may however prove to be less robust when considering objects with limited texture. We utilize shape as well as texture in our detection approach.

With the advent of affordable, high-quality depth sensors, a considerable amount of attention has been given to incorporating geometric features for object detection. Approaches based on point-pair features (PPF) [4], [5] detect and localize 3D objects by finding locally consistent arrangements of point pairs through Hough voting or RANSAC. Several enhancements to the descriptive power of PPFs have been proposed using visibility context [6], contours [7] or color [8]. We extend PPF methods with multi-resolution processing and disambiguation over time in a particle filtering framework to reduce false positives.

*2) Real-time Object Tracking:* In [9] and [10] real-time model-based tracking is achieved by using iteratively re-weighted least squares (IRLS) to align model edges in the image in a tracking by optimization approach. Edges and

texture information are combined in the approach of [11] to facilitate tracking of textured and textureless objects. In [1], multi-resolution surfel maps (MRSMaps) are proposed as a compact and efficient representation for aggregated RGB-D images. In an efficient SLAM method, RGB-D data is registered incrementally in order to build indoor maps or 3D object models, while real-time tracking is achieved by recursively optimizing the current pose estimate. We improve the robustness of this method by embedding the registration method in a particle filtering framework using improved proposals and enhance it with initial pose estimation.

Particle filtering is particularly attractive for tracking due to its flexibility in noise characteristics and non-linear motion and observation models. Moreover the multi-hypothesis nature of particle filters has increased robustness by admitting multiple correspondences. Klein et al. [12] exploit the GPU to render visible edges which are tracked using an annealed particle filter. In [13], [14] edge-template-based tracking of textured and textureless objects is achieved while modelling the state evolution on the $SE(3)$ Group with autoregressive dynamics. By the improvement of the proposal distribution with an efficient optimization method, we obtain a highly accurate yet robust method that tracks 3D object models in real-time.

In contrast to the techniques described above, tracking-by-detection does not consider the temporal constraints between successive frames, but rather aims to estimate the object pose in each frame individually. In general, however, taking account of the previous estimate like in our approach yields a strong prior for determining the object's pose in the current frame, and yields better temporal coherence of the estimated trajectory.

### A. Combined Pose Estimation and Tracking

Few works have specifically addressed integrated solutions to detection and tracking. Prisacariu et al. [15] present an approach to real-time 3D object segmentation and tracking by aligning contours between model and scene, while high frame rates are achieved by leveraging parallel processing on GPUs. In [13] initial pose hypotheses are extracted by matching keypoint features, while tracking proceeds by matching points on the edges of a projected wireframe model with those in the image. Choi et al. [14] define edge-templates which are matched in the image in order to identify initial pose candidates, these are then refined using an annealed particle filter. Our approach tracks a single full-view 3D model of the object and is not restricted to a limited amount of discrete view points. It is efficient enough to perform real-time on a CPU.

## III. OBJECT DETECTION AND POSE INITIALIZATION

We use MRSMaps as a concise image representation that supports fast aggregation from RGB-D images and image registration [1]. Given a known object model $m_m$ and an input scene map $m_s$ we seek to detect the object and estimate the pose of the camera with respect to $m_m$. To this end, we extend the MRSMap representation with colored surfel-pair features ([4],[8]) and adapt the surfel-pair pose voting approach by Drost et al. [4] to multiple resolutions.

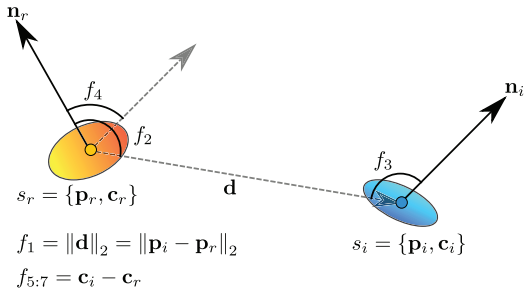



Fig. 2. Our color surfel-pair feature encodes the differences in geometry and color between two surfels by measuring angles between surface normals and differences between the means in chrominances and luminance.

### A. Colored Surfel-Pair Relations in MRSMaps

We define a 7-dimensional surfel-pair feature extending the surfel-pair feature of [4] with 3 extra components which describe the relative contrast in luminance and chrominance between the surfels (see Fig. 2). Given a surfel $s = (\mu_s, \Sigma_s, n_s)$ with mean $\mu_s$, covariance $\Sigma_s$, and normal $n_s$, we denote the spatial and color parts of the mean as $p_s = (p_x, p_y, p_z)^T$ and $c_s = (c_L, c_\alpha, c_\beta)^T$, respectively.

Each surfel-pair relates a *reference* surfel $s_r$ with a *referred* surfel $s_i$. A surfel-pair relation is then defined as $R(s_r, s_i) = (\|d_p\|_2, \angle(n_r, d), \angle(n_i, d), \angle(n_r, n_i), d_c)^T$ where $d_p := p_r - p_i$ and $d_c := c_r - c_i$ are differences in spatial and color means of the surfels.

### B. Pose Voting with Surfel-Pair Relations

Following the approach of [4], we efficiently match surfel-pairs between a scene map $m_s$ and a model map $m_m$ and accumulate votes in a pose Hough space. Each surfel-pair uniquely defines a reference frame which is used to cast votes for the object frame. We recover hypotheses for the object pose from the votes, which are refined by an efficient clustering step.

The surfel-pair-relations are hashed for efficient matching. To this end, distances, angles, and color contrasts are quantized into specific number of bins. In a preprocessing step, we compute surfel-pair relations from the model map $m_m$. On each resolution $r$ we compute the relations, determine their hashing index and store lists of relations, along with the pose of the object frame relative to the surfel-pair in a hash table $\mathcal{H}$. The voting procedure now proceeds similar to the approach in [4]. Since we have surfels at multiple resolutions available, we cast votes on all resolutions. We sequentially sample reference surfels $s_r$ in the scene, match its relations with relations in the model map, find the maximum in the voting space, and compute the corresponding pose from the responsible votes. We also add pose clusters with scores above a certain fraction of the maximum.

The pose hypotheses found by the above procedure will contain a number of false positives. In order to remove isolated poses and consolidate the more likely candidates, we

utilize a refinement process. We first sort the extracted poses by number of votes, and perform an agglomerative clustering with a fixed threshold on translation and orientation, until all poses have been processed. Finally we find the clusters with the highest scores (given by the sum of their pose votes) and return the mean pose for the top $n_p$ clusters.

## IV. POSE TRACKING

We track the 6-DoF pose of the camera with respect to a known object in a particle filter framework with improved proposal sampling. The choice of a particle filter for tracking is a good coupling for our detection approach which yields multiple pose hypotheses, since the distribution over states needs to be modelled in a non-parametric way. Convergence to a single hypothesis is then made possible by integrating successive observations.

### A. State Transition Model

By modelling the state transition on the $SE(3)$ group, and employing a simple autoregressive (AR) state dynamics model to predict the motion of the camera in each time step, we achieve an informed state prediction. We pose our problem as the estimation of the full 6-DoF configuration of the camera $x_t = (R_t, t_t) \in SE(3)$ at discrete time steps $t$. For convenience we refer to the pose as $x_t$, its homogeneous transformation matrix by $T(x_t)$, with rotation $R(x_t)$ and translation $t(x_t)$.

We employ a first-order, discrete-time AR state dynamics in order to propagate the particles in each time step:

$$\begin{aligned} g(x_{t-1}, \mathrm{d}W_t) &= T\left(x_{t-1}\right) \cdot \exp\left(A_{t-1}\Delta t + \mathrm{d}W_t\sqrt{\Delta t}\right) \\ A_{t-1} &= \lambda_{ar}\frac{1}{\Delta t}\log\left(T\left(x_{t-2}\right)^{-1} T\left(x_{t-1}\right)\right) \end{aligned} \tag{1}$$

where $x_t \in SE(3)$ is the state estimate at time $t$, $A_{t-1} \in \mathfrak{se}(3)$ is the velocity estimated in the previous time step (assumed to be part of the state), $\mathrm{d}\vec{W}_t$ is the Wiener process noise on $\mathfrak{se}(3)$, and exp and log are the exponential and logarithmic maps on $SE(3)$, while $\lambda_{ar}$ is the AR process parameter.

### B. Observation Model

At each time step $t$ the current observation $z_t$ is an RGB-D image, from which we build a scene map $m_{s,t}$. The observation model measures the alignment of associated scene and model surfels, $p\left(m_{s,t} \mid x, m_m\right) = \prod_{(i,j)\in\mathcal{A}} p\left(s_{s,i} \mid x, s_{m,j}\right)$ where $s_{s,i} = (\mu_{s,i}, \Sigma_{s,i})$, $s_{m,j} = (\mu_{m,j}, \Sigma_{m,j})$ are associated surfels. The observation likelihood of a surfel match is given by

$$\begin{aligned} p\left(s_{s,i} \mid x, s_{m,j}\right) &= \mathcal{N}\left(d\left(i,j;x\right); 0, \Sigma_{i,j}\left(x\right)\right) \\ d\left(i,j;x\right) &= \mu_{m,j} - T\left(x\right)\mu_{s,i} \\ \Sigma_{i,j}\left(x\right) &= \Sigma_{m,j} + R\left(x\right)\Sigma_{s,i}\,R\left(x\right)^T. \end{aligned} \tag{2}$$

We associate surfels between scene and model through efficient nearest neighbor look-ups in the octree representation of the MRSMap.

### C. Improved Proposal Distribution

For accurate 6-DoF tracking with a particle filter, the state transition model is not sufficient as the proposal distribution. When the likelihood is peaked then many particles are still required to sufficiently cover the 6-dimensional areas of high likelihood, and hence a lot of computation time is wasted on particles which are assigned very low weights.

Our aim is therefore to locally approximate the posterior $p(x_t \mid m_m, x_{t-1}^{(i)}, m_{s,t})$ for each particle,

$$p(x_t \mid m_m, x_{t-1}^{(i)}, m_{s,t}) = \eta^{(i)} p(m_{s,t} \mid x_t, m_m) p(x_t \mid x_{t-1}^{(i)}), \tag{3}$$

where $\eta^{(i)} := 1/\int p(m_{s,t} \mid x', m_m)\, p(x' \mid x_{t-1}^{(i)})\, \mathrm{d}x'$.

We recover this proposal distribution by propagating the particle with the state transition model to a new predicted mean pose. It serves as an initial guess in a registration step that aligns the scene map $m_{s,t}$ with the model map $m_m$. This yields a new mean pose estimate $\mu_t^{(i)}$ for the particle in the current time step. The covariance $\Sigma_t^{(i)}$ of the pose estimate is obtained from the uncertainty of the registration which is determined via a closed-form solution as in [1].

### D. Importance Weights

The importance weights $w^{(i)}$ in a particle filter account for the mismatch between target and proposal distribution $w^{(i)} := \text{target distribution}/\text{proposal distribution}$. Since we choose the target distribution as $p(m_{s,t} \mid x_t, m_m)\, p(x_t \mid x_{t-1}^{(i)})$, in our case the importance weights are $w^{(i)} = \eta^{(i)}$. These weights are the observation likelihood for the particle's predicted pose distribution $\overline{x}_t^{(i)} = \left(\overline{\mu}_t^{(i)}, \overline{\Sigma}_t^{(i)}\right)$ and depend on each particle individually.

Hence for particle $i$, the maximum likelihood data association $\mathcal{A}_t^{(i)} = \operatorname{argmax}_{\mathcal{A}} p(m_{s,t} \mid \mathcal{A}, \overline{x}_t^{(i)}, m_m)$ between scene and model map is established from the particle's predicted mean pose. We compute the weight $w^{(i)} = \exp\left(-\frac{1}{2} L\left(\overline{x}_t^{(i)}\right)\right)$ from

$$\begin{aligned} L(x) &= \sum_{a \in \mathcal{A}_t^{(i)}} \left[\log|\Sigma_a(x)| + d(a;\mu_x)^T \Sigma_a^{-1}(x)\, d(a;\mu_x)\right], \\ \Sigma_a(x) &= \Sigma_{m,j} + R\left(\mu_x\right)\Sigma_{s,i}\,R\left(\mu_x\right)^T + J_a\,\Sigma_x\,J_a^T \end{aligned} \tag{4}$$

for $x := (\mu_x, \Sigma_x)$ and associated surfels $a = (i,j)$ where $J_a = \frac{\partial T}{\partial x}(x)\,\mu_{s,a}$. The term $J_a\,\Sigma_x\,J_a^T$ accounts for the pose uncertainty of the particle's pose prediction through first-order error propagation to the observation model. After the importance weights are updated, we resample particles with probability proportional to the weights.

### E. Efficient Computation of the Proposal Distribution

In practice, in order to achieve tracking at high frame rates, an optimization is required to efficiently compute the proposal distribution. We propose a simple and robust method which circumvents the need to perform registration for each particle individually. Rather, we identify the modes of the density estimate $p(x_t \mid x_{t-1}) \propto \sum_i p(x_t \mid x_{t-1}^{(i)})$

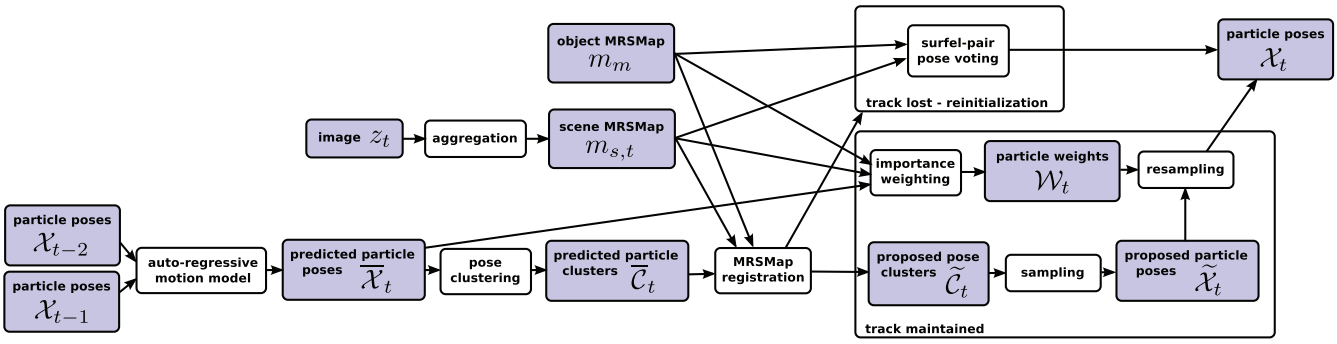



Fig. 3. Processing steps of our joint object detection and tracking framework.

and generate the proposal in the form of a Gaussian mixture distribution, hence for each particle a sample is drawn from the relevant mixture component. In order to identify the mixture components of the transition density we employ a clustering of the particles with a fixed threshold on translation and rotation. For efficient clustering, a kd-tree is constructed from the position estimates of the particles. Particles in a limited volume and with similar orientations are then clustered together until all particles have been assigned. Registration is then only performed starting from the means of these pose clusters. The resulting mean and covariance pose estimate is used to sample the particles in the same cluster. The importance weights of each particle are still evaluated separately for each particle by using individually predicted pose estimates $\overline{x}_t^{(i)}$ in Eq. (4). Surfel associations are shared between the particles within a cluster to further increase efficiency.

We further note that when the estimate of the tracker is good, the discrete distribution given by the particles typically has a single mode. However, after initialization or when the uncertainy increases, considering multiple modes aids robustness.

## V. INTEGRATED POSE ESTIMATION AND TRACKING

The pose estimation and tracking methods are integrated into a unified framework for tracking without prior knowledge about the initial pose (illustrated in Fig. 3).

*a) Initialization:* In the initialization step, $n_p$ pose hypotheses are found by constructing a scene map $m_s$ from the whole RGB-D image $z$. We extract surfel-pair relations from the scene map and apply our pose voting algorithm described in Sec. III. Particles are then sampled from the extracted pose estimates to initialize the particle filter.

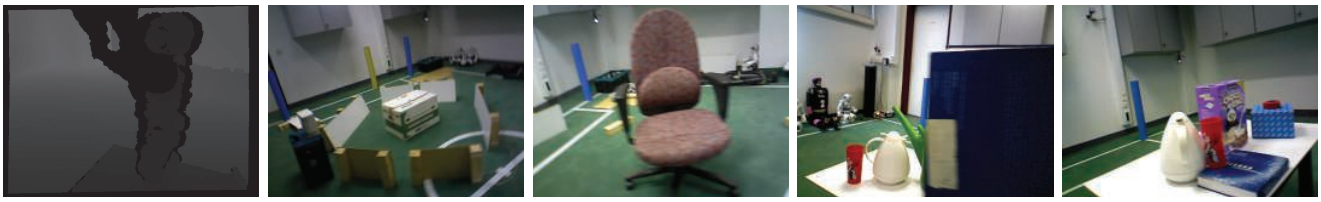
*b) Tracking:* Once initialized, tracking proceeds as described in the previous section. That is, the particles are propagated according to the state transition model, clustered into distinct hypotheses, and for each cluster the improved proposal distribution is computed. The improved proposal is then sampled and the sampled particles are assigned importance weights. We can process the observations more efficiently by focusing on a volume of interest close to the last pose estimate. Hence we segment away parts of the scene which are well beyond the spatial distribution of the model under the current particle estimates.

*c) Reinitialization:* Registration failures are handled by falling back to sampling from the state transition density. In some circumstances the object may leave the field of view or be highly occluded. In this case it is not possible to establish a sufficient number of surfel associations for robust registration. A number of techniques have been proposed for detecting degeneracy in particle filter methods. A simple scheme would be to detect drifting by a threshold on the displacement between frames. The average likelihood ratio [16] is a measure of how the average likelihood of the particles is evolving. Hence low values can point to dropping likelihood and degeneracy of the estimate. Similarly, the number of effective particles ($N_{\mathit{eff}}$), traditionally used for triggering resampling, has been proposed as a measure of the quality of the estimate [13]. When sampling from an improved proposal distribution, however, particles often get assigned similar weights. In this case the above schemes are not always applicable. Instead we propose a simple scheme for detecting loss of tracking and invoking reinitialization. We monitor the maximum number of associations established for all hypotheses. If this number drops below a threshold the estimate may be poor and we attempt to reinitialize.

## VI. EVALUATION

We evaluate our approach on a publicly available RGB-D object tracking dataset[1]. We compare our method to tracking-by-optimization using the MRSMap framework [1]. The dataset consists of 13 sequences each with around 1100 frames, featuring five objects of different characteristics and challenging aspects such as motion blur, clutter, partial and total occlusions. Ground truth for the camera pose has been captured by attaching reflective markers to the camera and tracking its configuration using a 12 camera OptiTrack motion capture system. All datasets were recorded using an Asus Xtion Pro Live camera in a resolution of 640x480 and a frame rate of 30Hz. Object models are learned using the MRSMap framework. This library provides tools for building object models by fusing views from a training sequence where the camera is moved around the object.

[1] http://www.ais.uni-bonn.de/download/objecttracking.html



(a) missing depth (b) distance to object (c) fast motions and blurring (d) occlusions (e) clutter

Fig. 4. Challenging situations: Example frames from sequences used in the experiments.

Figure 4 gives an impression of the sequences, and the challenges they pose. Missing depth results in fewer surfels in the scene map, while the depth precision of RGB-D is lower at larger distances to the object. Fast motions and blur are a challenge to tracking, while robustness against occlusions and clutter is required. We use these datasets to evaluate our object detection approach on individual frames, as well as the combined detection and tracking method on the full sequences. All tests with our approach were carried out on an Intel Core i7-940 (4 cores + 4 virtualized) with 12GB RAM.

### A. Detection and Pose Estimation

*1) Setup:* We assess the performance of the object pose estimation component on individual frames from the available sequences. Since the aim of detection is to estimate the initial pose for tracking, we also evaluate the extent to which the initialization allows the tracker to converge to the correct pose. To this end we plot the evolution of precision and recall over subsets of frames when the tracker is initialized by the detection method. In each frame, we record the number of true positives, false positives and false negatives, which are summed for corresponding frame indices $\{1, \ldots, 11\}$ over the whole sequence. We count a true positive if the pose of camera is among the hypotheses. We allow a small variation from the true camera pose of 10 cm in translation and 15° in rotation which we regard as close enough to resume robust tracking. Note, that poses which align the object well within the scene but do not meet this criterion do not count as true detections. We examine the detection performance in terms of precision and recall. All experiments were run 10 times and the results aggregated. Table I shows the parameters used for our experiments. The maximum resolution was set according to object size, hence for the Humanoid, Box, and Chair sequences, the slightly coarser resolution of 2.5 cm was used for performance reasons, while for the Watering Can and Cereal sequences, 1.25 cm was chosen to provide greater accuracy for the smaller objects.

*2) Results:* Precision and recall against frame indices are seen in Fig. 5 for all sequences. Average frame timings for detection and pose estimation on all sequences are given in Table II. As can be seen, our method achieves high detection rates of ca. 95-100% on average already in the first frame. However, the high detection rate is coupled with false detections, hence the precision for detection in the initial frame is relatively low. False detections arise from other parts of the scene or object itself which are locally similar in shape or texture and result in incorrect associations. Our choice of the particle filter is clearly justified since, after initialization with the pose hypotheses, we can see that typically the belief converges on the correct pose within just a few frames.

TABLE I

PARAMETERS USED FOR OUR EXPERIMENTS.

| parameter | value |
|---|---|
| Max. no. of detection hypotheses | 5 |
| Max. resolution | 1.25 cm / 2.5 cm |
| Angle quant. | 10° |
| Dist. quant. | 5 cm |
| Lum. quant. | 3 |
| Chrom. quant. | 3 |
| $\gamma$ | 0.7 |
| No. of particles | 25 |
| Max. alignment iterations | 10 (all frames) / 20 (real-time) |

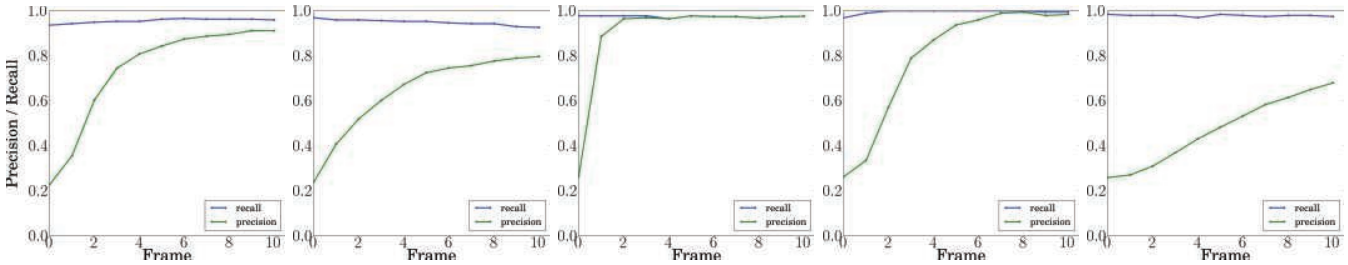
TABLE II

MEAN FRAME PROCESSING TIME $\pm$ STD. DEVIATION (IN SECONDS) FOR OBJECT DETECTION IN ALL SEQUENCES.

| dataset | timing |
|---|---|
| Humanoid | $0.750 \pm 0.226$ |
| Box | $3.43 \pm 0.764$ |
| Chair | $1.20 \pm 0.416$ |
| Watering Can | $1.05 \pm 0.35$ |
| Cereal | $0.68 \pm 0.22$ |

The rate of convergence can be seen to be highly dependent on the object characteristics. The Chair sequences show the most rapid rate of convergence, typically within 1 or 2 frames, likely owing to its distinctive shape and texture. The Humanoid and Watering Can sequences also see convergence to the correct pose within just a few frames. While the Box and Cereal sequences converge slowly (but at high precision) to the correct hypothesis. The rate of convergence seems also to be highly correlated with the object's degree of symmetry or self-similarity. Competing hypotheses with large overlap receive similar weights and are hard to differentiate.

The detection timing is also related to the object characteristics, despite the finer resolution, the cereal and watering can have fewer surfels and hence can be detected more quickly (typically within 1 second). As can be seen, the box has the highest detection time, owing to the fact that many surfel pairs lie on planar regions, resulting in many surfel-pair relations falling into the same bin, which slows down the voting phase. Meanwhile, the humanoid and chair can be detected in half the time or less, since their more distinctive




(a) Humanoid (b) Box (c) Chair (d) Watering Can (e) Cereal

Fig. 5. Evolution of precision and recall during tracking over sets of 11 frames for all sequences after detection with our method.

TABLE III

MEDIAN ABSOLUTE TRAJECTORY ERROR AND MEAN FRAME TIME ± STD. DEVIATION WHEN ALL FRAMES ARE USED.

| sequence | MRSMap [1] | ours | |
|---|---|---|---|
| | ATE (m) | ATE (m) | time (ms) |
| Humanoid slow | **0.0200** | 0.0226 | $37.5 \pm 4.0$ |
| Humanoid medium | **0.0261** | 0.0265 | $38.2 \pm 4.6$ |
| Humanoid fast | **0.0316** | 0.0334 | $39.4 \pm 4.6$ |
| Box slow | 0.0236 | **0.0234** | $73.7 \pm 9.0$ |
| Box medium | 0.0378 | **0.0247** | $68.2 \pm 18.3$ |
| Box fast | 0.0241 | **0.0214** | $66.3 \pm 12.5$ |
| Chair slow | **0.0229** | 0.0230 | $95.8 \pm 17.9$ |
| Chair medium | 0.0158 | 0.0158 | $103.9 \pm 13.2$ |
| Chair fast | **0.0277** | 0.0359 | $93.7 \pm 18.4$ |
| Watering Can 1 | 0.0650 | **0.0247** | $40.7 \pm 7.4$ |
| Watering Can 2 | **0.0147** | 0.0221 | $43.8 \pm 9.5$ |
| Cereal 1 | fails | **0.0225** | $36.2 \pm 10.6$ |
| Cereal 2 | 0.0445 | **0.0183** | $36.4 \pm 10.0$ |

TABLE IV

MEDIAN ABSOLUTE TRAJECTORY ERROR, MEAN FRAME TIME ± STD. DEVIATION, AND PERCENTAGE OF FRAMES USED FOR REAL-TIME MODE.

| sequence | MRSMap [1] | ours | |
|---|---|---|---|
| | ATE (m) | ATE (m) | frames used (%) |
| Humanoid slow | **0.0211** | 0.0236 | 59.5 |
| Humanoid medium | 0.0267 | **0.0264** | 62.9 |
| Humanoid fast | **0.0322** | 0.0334 | 60.5 |
| Box slow | 0.0243 | **0.0235** | 40.5 |
| Box medium | 0.0466 | **0.0293** | 48.3 |
| Box fast | 0.0286 | **0.0223** | 45.0 |
| Chair slow | **0.0228** | 0.0293 | 29.8 |
| Chair medium | **0.0161** | 0.0173 | 29.9 |
| Chair fast | **0.0291** | 0.0406 | 33.4 |
| Watering Can 1 | fails | **0.0263** | 56.8 |
| Watering Can 2 | **0.0149** | 0.0259 | 59.8 |
| Cereal 1 | fails | **0.0277** | 70.9 |
| Cereal 2 | 0.0454 | **0.0188** | 75.2 |

shape and texture result in fewer relations in the same bin.

### B. Object Tracking

*1) Setup:* We evaluate the performance of our tracker on the entire sequences in order to assess its robustness. We examine the absolute trajectory error (ATE) metric [17]. It measures the difference between points on the estimated and ground truth trajectories. All experiments were run 10 times and the results aggregated.

To initialize the tracker, the detection procedure is run in the first frame of each sequence. In the event that the track is lost, the tracker can reinitialize itself. We carried out experiments in two tracking "modes". For the first set of experiments, we process all frames in each dataset, while in a second set of tests, we enable real-time tracking whereby frames are skipped. Table I displays the parameters used for the experiments.

*2) Results:* In Tables III and IV we present average results w.r.t. accuracy and timing. Our method robustly tracks the pose of the camera with respect to different objects and under varying conditions. The experiments demonstrate typical median absolute trajectory error of approximately 15-35 mm, while for the real-time tracking experiments we observed ATEs of ca. 18-40 mm. Our approach yields accuracies and timings similar to the tracking-by-optimization approach in [1]. The results also demonstrate that our method is more robust in underdetermined situations (only planar surfaces visible) in which our method maintains pose uncertainty. Multi-hypothesis tracking also improves on cases with difficult occlusion settings. If tracking fails, our approach is capable of reinitializing itself.

For the box it can be seen in Fig. 6 that the vast majority of relative pose errors were less than 50 mm and 0.05 rad ($\approx 3°$), respectively. Experiments on the Humanoid sequences (Tables III, IV) also showed good performance, with only the fast sequence causing our tracker to drift slightly, however tracking was maintained throughout, while real-time tests saw barely any deterioration in performance. Fig. 7 shows the model rendered at the tracked pose in three frames from the medium sequence. The track is maintained accurately from different viewpoints and under fast camera motions.

Real-time performance suffered slightly on the fast Chairs dataset, possibly due to the relatively high percentage of dropped frames, and the fast motions also proved a challenge when tracking all frames. However, as can be seen from the histograms, accurate tracking was maintained for the vast majority of frames, and any misalignment could be resolved. Tracking performance on the Watering Can sequences was particularly good. In order to test reinitialization, in sequence 2, the object is briefly occluded with a book which is successfully handled with our approach (see Fig. 1). Although the median ATE is low in both Cereal sequences, we noticed

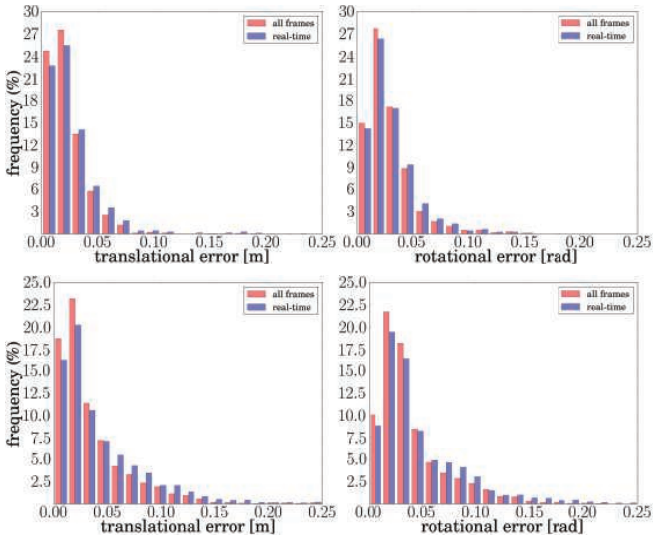



Fig. 6. Histograms of translational and rotational errors for the Box (top) and Chair (bottom) datasets.

Fig. 7. Robust tracking: Our tracker handles the varying appearance of the object (left and center), and copes well with motion blurring (right). The model is rendered as green overlay at the tracked pose.

a drift in orientation when only one side of the cereal is visible (see Fig. 8). However, when more of the object comes into view, accurate tracking is possible again. The increased degree of occlusion in these sequences in particular proved challenging for the single hypothesis tracker in [1], which cannot reinitialize if the track is lost.

## VII. CONCLUSIONS

In this paper we presented a novel approach for model-based pose estimation and tracking of objects in RGB-D image sequences. We propose a framework for joint detection and tracking using multi-resolution surfel maps (MRSMaps). Tracking combines efficient accurate registration of MRSMaps to object models and robust particle filtering using improved proposals. We extend MRSMaps with colored surfel-pair features and apply multi-resolution pose voting to detect the objects and estimate their 6-DoF pose initially. We integrated the pose estimation and tracking methods to allow tracking without prior knowledge of the initial pose, and reinitialization when the track is lost.

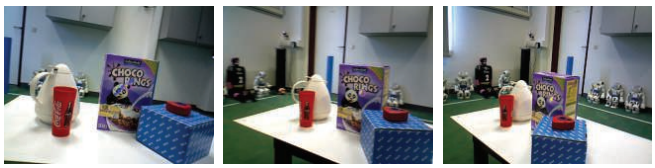


Fig. 8. Pose uncertainty: With only one side of the box visible, the uncertainty in camera pose grows (left). When more of the object comes into view, the estimate improves (center). The estimate remains good despite partial occlusion of the object (right).

We demonstrated high detection rates and accurate pose estimation of objects with a range of different visual and geometric characteristics. Furthermore, our method has been seen to be capable of robustly tracking the 6-DoF pose of the camera under a wide range of motions and occlusion.

There are a number of possible directions for future work. One such avenue would be to consider incorporating object boundary information into the detection and tracking pipeline, which could further improve performance on textureless objects. Another option for future work would be to explore an implementation which leverages the parallel processing capabilities of GPUs. This would also facilitate the parallel tracking of multiple objects.

## REFERENCES

[1] J. Stückler and S. Behnke, "Multi-resolution surfel maps for efficient dense 3D modeling and tracking," *Journal of Visual Communication and Image Representation*, 2013.
[2] D. G. Lowe, "Object recognition from local scale-invariant features," in *Proc. of the IEEE Int. Conf. on Computer Vision (ICCV)*, 1999.
[3] H. Bay, A. Ess, T. Tuytelaars, and L. Van Gool, "Speeded-up robust features (SURF)," *Computer Vision and Image Understanding*, vol. 110, no. 3, pp. 346–359, 2008.
[4] B. Drost, M. Ulrich, N. Navab, and S. Ilic, "Model globally, match locally: Efficient and robust 3D object recognition," in *IEEE Int. Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2010.
[5] C. Papazov and D. Burschka, "An efficient RANSAC for 3D object recognition in noisy and occluded scenes," in *ACCV 2010*, 2011.
[6] E. Kim and G. Medioni, "3D object recognition in range images using visibility context," in *Proc. of the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, pp. 3800–3807, 2011.
[7] C. Choi, Y. Taguchi, O. Tuzel, M.-Y. Liu, and S. Ramalingam, "Voting-based pose estimation for robotic assembly using a 3D sensor," in *Proc. of IEEE Int. Conf. on Robotics and Automation (ICRA)*, 2012.
[8] C. Choi and H. Christensen, "3D pose estimation of daily objects using an RGB-D camera," in *Proc. of the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, 2012.
[9] T. Drummond and R. Cipolla, "Real-time visual tracking of complex structures," *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, vol. 24, no. 7, pp. 932–946, 2002.
[10] A. I. Comport, É. Marchand, and F. Chaumette, "Robust model-based tracking for robot vision," in *Proc. of the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, 2004.
[11] L. Vacchetti, V. Lepetit, and P. Fua, "Combining edge and texture information for real-time accurate 3D camera tracking," in *Proc. of IEEE/ACM Int. Symp. on Mixed and Augm. Reality (ISMAR)*, 2004.
[12] G. Klein and D. Murray, "Full-3D edge tracking with a particle filter," in *British Machine Vision Conference*, pp. 1119–1128, 2006.
[13] C. Choi and H. I. Christensen, "Robust 3D visual tracking using particle filtering on the SE(3) group," in *Proc. of the IEEE Int. Conf. on Robotics and Automation (ICRA)*, 2011.
[14] C. Choi and H. I. Christensen, "3D textureless object detection and tracking: An edge-based approach," in *Proc. of the IEEE Int. Conf. on Intelligent Robots and Systems (IROS)*, 2012.
[15] V. A. Prisacariu and I. D. Reid, "PWP3D: Real-time segmentation and tracking of 3D objects," *Int. Journal of Computer Vision*, vol. 98, no. 3, pp. 335–354, 2012.
[16] S. Thrun, W. Burgard, and D. Fox, *Probabilistic Robotics*. 2005.
[17] J. Sturm, N. Engelhard, F. Endres, W. Burgard, and D. Cremers, "A benchmark for the evaluation of RGB-D SLAM systems," in *Proc. of the Int. Conf. on Intelligent Robot Systems (IROS)*, 2012.