Robust Surface Tracking in Range Image Sequences

Farzad Husain[a], Babette Dellen[b], Carme Torras[a]

[a]*Institut de Robòtica i Informàtica Industrial, CSIC-UPC, Llorens i Artigas 4-6, 08028, Barcelona, Spain*
[b]*RheinAhrCampus der Hochschule Koblenz, Joseph-Rovan-Allee 2, 53424 Remagen, Germany*

Abstract

A novel robust method for surface tracking in range-image sequences is presented which combines a clustering method based on surface models with a particle-filter-based 2-D affine-motion estimator. Segmented regions obtained at previous time steps are used to create seed areas by comparing measured depth values with those obtained from surface-model fitting. The seed areas are further refined using a motion-probability region estimated by the particle-filter-based tracker through prediction of future states. This helps resolving ambiguities that arise when surfaces belonging to different objects are in physical contact with each other, for example during hand-object manipulations. Region growing allows recovering the complete segment area. The obtained segmented regions are then used to improve the predictions of the tracker for the next frame. The algorithm runs in quasi real-time and uses on-line learning, eliminating the need to have *a priori* knowledge about the surface being tracked. We apply the method to in-house depth videos acquired with both time-of-flight and structured-light sensors, demonstrating object tracking in real-world scenarios, and we compare the results with those of an ICP-based tracker.

Keywords: Range video, Surface fitting, Tracking, Segmentation.

1. Introduction

Tracking the pose of objects in image sequences is one of the most fundamental tasks in computer vision [1], and many works in the past focused on tracking in grayscale and color images. Tracking in range images is less explored, but due to the availability of low-cost depth cameras and their increasing importance in science and industry, such tracking approaches are of growing interest to the machine-vision as well as the robotics community. For example, tracking of object surfaces based on range data can be used to monitor and control the actions of a robotic arm during object-manipulation tasks [2]. Using depth information as the primary information source has the advantage that objects can be directly described by their geometric form, which is not affected by changes in the object's appearance in terms of color and texture, lighting conditions, shadowing or reflections. Furthermore, geometric features required for grasping are immediately available. Disadvantages of using depth cameras are their limited resolution, accuracy, and operating range. This poses special demands regarding robustness and adaptability for the algorithms dealing with this type of data.

Surface tracking in the domain of range image sequences has two main components: (i) extract the surfaces and establish the correspondence of the surfaces over the frames in the sequence of range images, and (ii) compute the motion transformation using these surface correspondences [3]. Both tasks are intertwined, as the correct extraction of surface patches helps finding correct correspondences, and vice versa. As long as surfaces are spatially disconnected, problem (i) can be more or less easily solved by clustering the 3D points based on their spatial proximity [2, 4]. Problem (ii) can be solved by assuming 3D rigid-body motions between extracted point sets [3, 5, 6]. However, as soon as surfaces get in physical contact with each other, the problem becomes far more challenging, because in this case it is often impossible to distinguish between different objects based on depth differences alone. The situation becomes even more severe when both the manipulator and the manipulated surface undergo the same transformation at this time, e.g., during a hand-object manipulation. In this case, (i) and (ii) need to be solved conjointly, while taking the motion history of the objects into account.

Email addresses: shusain@iri.upc.edu (Farzad Husain), dellen@hs-koblenz.de (Babette Dellen), torras@iri.upc.edu (Carme Torras)

In this work, we offer a solution to this problem by combining a recent clustering approach based on surface-model fitting [2] with a particle-filter-based affine-motion-estimation approach [7] with some modifications. Because we use a split-and-merge procedure for region growing which automatically adapts to the dynamic range data, predictions from the particle filter can be incorporated in a straightforward manner by refining seeding and, in consequence, the input to the tracker.

2. Related work

Object tracking has previously been performed mostly for color/gray-scale image sequences [1, 7, 8, 9, 10]. However, depth images pose different challenges to the tracking algorithm than color/gray-scale images.

Most methods for tracking in range images use *a priori* knowledge of 3D point correspondences and find the affine or rigid-body transformation on this basis [11]. These methods mainly work for sparse data sets, but are less useful when working with dense range images. Other techniques match surface patches instead, eliminating the need for finding exact point correspondences [3]. The range data is segmented into surface patches, then correspondences are established between patches of adjacent frames, and the motion transformation is estimated. However, such an approach is only effective if the initial (presumably correct) segmentation can be maintained over time. This is however not a trivial task, as small variations in the data and motions can change the segmentation drastically. To overcome this problem, a seeding and region growing technique for range-image sequences was proposed in [2, 4]. Maintenance can be improved this way, but when two or more surfaces are in physical contact with each other, it remains difficult to determine the boundary between the surfaces in contact using depth differences alone [2].

To cope with the specific characteristics of range data, some existing approaches put limitations on the tracked surface by considering only articulated motion [12, 13, 14]. This simplifies the tracking problem but also limits the usability of the algorithm to particular scenarios. Robust tracking of human hands assuming articulated motion constrained by the 54-dimensional parameter space has been performed in [15]. In [16], object tracking using a depth camera was performed (for 3D object reconstruction), but here the robotic hand had to be separated from the range data before applying the algorithm.

Another option for 3D tracking is the Iterative Closest Point (ICP) algorithm. However, the basic ICP method [17] is a pairwise matching algorithm which does not take into account past measurements [18, 6], hence the error starts to propagate. The ICP algorithm has been previously combined with Kalman filtering for object reconstruction [19]. However, in this case, the background was removed, leaving only the target object. In cluttered scenes, this approach may thus not be applicable. Point-to-point matching in 3D space requires a high accuracy in the estimation of the transformation matrix. This makes these approaches less suitable for our scenario because of the limited accuracy of the depth camera. Several variants of ICP which achieve better point set registration have been proposed, such as the expectation-maximization ICP [20] and softassign [21]. However, these variants have a higher computational cost and require specialized hardware such as GPUs to achieve real-time performance [22].

In this work, we combine seeding and growing of surfaces with particle filtering to overcome the aforementioned limitations. Our main contribution is a robust mechanism for identifying a set of points belonging to a target object that is being manipulated in 3-D space, regardless of its physical contact with other objects.

3. Method

Our tracker requires a range image as input at each time step. The range image along with the camera's intrinsic parameters is used to construct a 3-D point cloud. The algorithm for surface tracking consists of the following steps. Initially a set of non-overlapping geometric surface patches are obtained by clustering the 3-D points. Each cluster is modeled by a quadratic function and the surface that we want to track is identified manually (see Section 3.1). Segmented surfaces at step t are used to create seed regions in the next frame $t + 1$ by comparing the predicted depth values (from quadratic surface models fitted to the segments) to the actual depth values (see Section 3.3). At the same time, a motion-probability region is found by the particle-filter-based tracker through the prediction of future states (see Section 3.2). The extracted motion-probability region is used to refine the seeding. Region growing allows reconstructing the segment at $t + 1$. Based on this segmented area, the translation parameters of resampled states are re-estimated (see Section 3.4), which, provided the segmentation is correct, improves the predictions of the tracker for the next frame. The basic idea behind our approach is illustrated in Fig. 1.

The adaptive surface fitting for clustering of 3-D points is used for both initial clustering, seeding and region growing during the tracking. At each time step

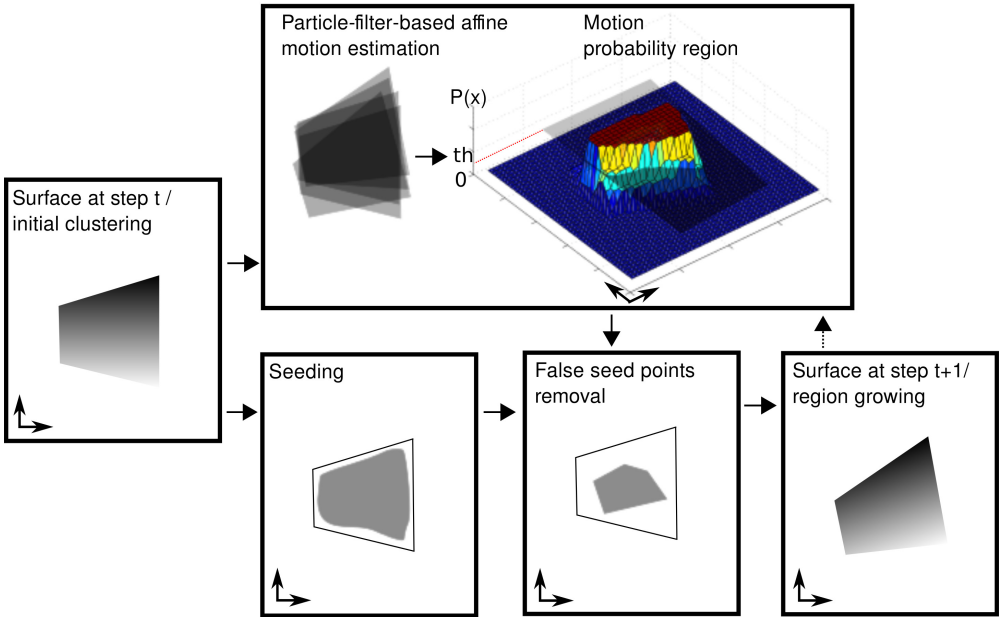

Figure 1: Schematic illustrating the basic idea of our approach (for further explanations, see main text).

t, a depth image $F_t(u, v)$ is acquired by the depth sensor, where (u, v) are the pixel positions in the image grid of size (image length $\times$ image width $\times$ 3), containing the 3D data $x(u, v)$, $y(u, v)$, and $z(u, v)$.

3.1. Initial Clustering

For the first frame $F_{t=0}(u, v)$, we achieve an initial clustering by a split-and-merge approach [23, 24]. Different from [24] where the depth image is segmented into planar surfaces only, our method uses a second-order surface model that is able to cluster 3D points belonging to curved surfaces. We first split the data points into two equally sized clusters $c_{j=1,2}$ with respective labels $l_{j=1,2}$. Then for each cluster we estimate the parameters $\{a_j, b_j, d_j, e_j, g_j\}$ of a quadratic surface $f_j(x, y)$ of the form

$$z = f_j(x, y) = a_j x^2 + b_j y^2 + d_j x + e_j y + g_j \qquad (1)$$

such that the difference $\sum_{k=1}^{n_j} [f_j(x^k, y^k) - z^k]^2$ is minimized, where n_j is the total number of points belonging to c_j. We use a Levenberg-Marquardt minimization to solve this problem.

For each cluster c_j, we first unlabel all points within the cluster for which the difference between the actual depth value z^k and the estimated depth value $f_j(x^k, y^k)$ is larger than a threshold $\psi_j = \sum_{(u,v)\in c_j} |f_j[x(u, v), y(u, v)] - z(u, v)|/(\rho n_j)$. Here, ρ is a constant which controls the number of unlabeled points. For all unlabeled points (u, v) in each cluster c_j, we find the new index label of the cluster that provides the smallest fitting error

$$\xi(u, v) \quad = \quad \arg[\min_j(\{\delta_{c_j}(u, v)\})] \qquad (2)$$

where

$$\delta_{c_j}(u, v) = |f_j[x(u, v), y(u, v)] - z(u, v)|. \qquad (3)$$

For every unlabeled point (u, v), we find the cluster c_ξ denoted by the index ξ, and assign the new respective label l_ξ to it. The resulting new clusters may be disconnected. In this case, we split all disconnected regions from the main (largest) region, assign new labels to them, and fit surface models to the data points (Eq. 1). Additionally, neighboring surfaces are merged if they can be described approximately by the same surface model. We perform this splitting and merging iteratively until a stable solution is reached. Figure 2 shows the pseudo-code of the splitting and merging procedure and Fig. 3 shows a typical segmentation result obtained with our clustering method. We manually identify the label of the surface that we want to track. We represent the points on the tracked surface as the 2-vector T which is a subset of the image grid.

3.2. Particle-Filter-Based Affine Motion Estimation

In order to determine the location of the projected tracked surface in the image at each time step t, we

Require: sampled 3-D points, $F_t(u, v)$
for $t = 0$ label the points (u, v) forming 2 clusters $\rightarrow c_j$
for all clusters c_j **do**
  Fit a surface f_j
end for
SPLITTING
for all clusters c_j **do**
  for all points $(u, v) \in c_j$ **do**
    if $|f_j(x(u, v), y(u, v)) - z(u, v)| > \psi_j$ **then**
      unlabel the point (u, v)
  end for
  update the surface parameters $\rightarrow f_j$,
end for
for all clusters c_i **do**
  for all clusters c_j **do**
    determine the model fitting error $\rightarrow \delta_{c_j}$, for unlabeled points in c_i
  end for
  for all unlabeled points $(u, v) \in c_i$ **do**
    $\xi(u, v) \leftarrow \arg[\min_j(\{\delta_{c_j}(u, v)\})]$
    $l(u, v) \leftarrow l_\xi$
  end for
end for
for all clusters **do**
  determine disconnected sub-clusters
  assign a new label to all disconnected sub-clusters
end for
MERGING
for all c_i **do**
  for all c_j **do**
    if $\sum_{(x,y,z)\in c_i} |f_j(x, y) - z|/n_i + \sum_{(x,y,z)\in c_j} |f_i(x, y) - z|/n_j < threshold$ **then**
      merge the two clusters
  end for
end for
go to 2

Figure 2: Pseudo code describing the iterative split-and-merge procedure.

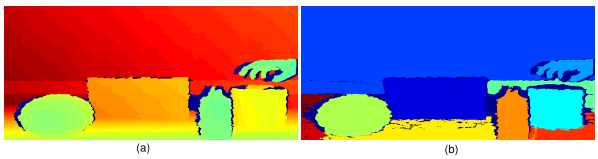

Figure 3: (a) Color-coded depth image (Kinect). (b) Segmentation result. Each segment has a unique color.

model its motion as a 2-D affine transformation matrix, i.e.,

$$X_t = \begin{bmatrix} A_t & k_t \\ 0 & 1 \end{bmatrix}, \tag{4}$$

where A_t is an invertible 2×2 matrix, k_t is the 2-D translation, and $X_{t=0} = I_3$. For projective motion estimation, we compute X_t using a particle filter, whose state dynamics is based on a constant velocity model as described in [7], i.e.,

$$X_t = X_{t-1} \cdot \exp(a \log(X_{t-2}^{-1} \cdot X_{t-1}) + V_t), \tag{5}$$

where a is the autoregressive process parameter and V is the Wiener process noise. For color/gray-scale images, the sum-of-squared differences (SSD) between the tracked image template $F_{t=0}^{color}(T)$ and the acquired image $F_t^{color}(K)$ in the predicted region K can be used as the measurement function. Here, T are the indexes defining the area of the template in the image grid and K is determined by transforming T with X_t.

Contrary to color/gray-scale data, the 3-D range data of the tracked surface depends on the object's pose relative to the camera pose. Hence, the SSD between the template and the tracked image region cannot be computed in the same way as for color images [7]. In our approach, we first compute the rigid transform between the 3-D range vectors $F_t(K)$ and $F_{t=0}(T)$ which minimizes the nearest neighbor distance in the least-squares sense [25], and then compute the distance as the l_1-norm of the difference between the corresponding points, i.e, the measurement function

$$h(K_i) = \left\| F_t(K_i)' - F_{t=0}(T) \right\|_1, \tag{6}$$

where $F_t(K_i)'$ is the 3-D feature vector obtained after applying the rigid transform to $F_t(K_i)$. The measurement function $h(K_i)$ has to be computed for every state $i = 0 \ldots m-1$, where m is the number of particles. Note that the affine transform handles the transformation of the projected tracked object in the image, while the 3D-rigid transform updates the 3D values of the points inside the tracked-object area. Since we know the actual shape of the template that we want to track (from the initial clustering), we can track the points that are inside the cluster silhouette instead of using a bounding box as in [7]. We generate a motion-probability-region $P_t(u, v)$ based on the re-sampled particles as $P_t(u, v) = 1$ if $\Omega_t(u, v) < \tau$, and 0 otherwise, with $\tau = m/3$ and $\Omega_t(k, l) = \sum_{i=0}^{m-1} W_t^i(k, l)$. W_t^i is a binary image which represents the tracked object-surface silhouette determined

from the affine state matrix X_t^i of the re-sampled particle i, i.e.,

$$W_t^i(k,l) = \begin{cases} 1 & \text{if } (k,l,1) \in \Pi \\ 0 & \text{otherwise,} \end{cases} \quad (7)$$

where $\Pi = \left\{ X_t^i \times (u,v,1) \mid \text{for all } (u,v) \in T \right\}$. The motion-probability-region is defined in order to determine the most probable location of the seed for the tracked surface. We achieve this by considering the output of all the resampled particles, i.e., summing over all the resampled particles W^i, and applying a threshold afterwards.

The particle filtering approach [26] consists of the following main steps:

1. Sample $X_t^{(i)} \sim p(X_t | X_{t-1}^{(i)}, y_t)$
2. Compute weights $w_t^{(i)} = w_{t-1}^{(i)} \frac{p\left(y_t | X_t^{(i)}\right) p\left(X_t^{(i)} | X_{t-1}^{(i)}\right)}{\pi(X_t^{(i)} | X_{0:t-1}^{(i)}, y_{0:t})}$
3. Resample $X_t^{(i)}$ according to $w_t^{(i)}$

The calculation of the measurement likelihood $p\left(y_t | X_t\right)$ and the importance function $\pi(X_t | X_{0:t-1}, y_{0:t})$ is described in detail in [7].

3.3. Seeding and Region-Growing Procedure

In order to generate seeds, we first estimate the surface models for each cluster. Based on these, we compute the difference between the predicted depth and the actual depth (see Eq. 3), providing a seed area for each cluster, i.e., for all $(u,v) \in c_j$,

$$s_t(u,v) = \begin{cases} j & \text{if } \delta_{c_j}(u,v) < \psi_j, \\ 0 & \text{otherwise,} \end{cases} \quad (8)$$

where the non-seed points are assigned a zero value. For the target cluster c_i (that is being tracked) we refine the seed area by removing all points that are not inside $P_t(u,v)$, and prohibit all other clusters $j \neq i$ to have seeds inside the probability region of the target, yielding

$$s'_t(u,v) = \begin{cases} 0 & \text{if } P_t(u,v) = 0 \text{ and } s_t(u,v) = i \\ 0 & \text{if } P_t(u,v) = 1 \text{ and } s_t(u,v) = j \\ s_t(u,v) & \text{otherwise.} \end{cases} \quad (9)$$

In Fig. 4, the basic idea behind the method is illustrated for a scene showing a hand manipulating a bottle. Figure 4(a) shows the seeding procedure without using the refinement step. It can be observed that with passage of time, seed points of the hand aggregate in the region of the bottle. This results in error propagation and after every time instant the segmentation/tracking result gets worse. The reason is that the region growing approach is a least-squares minimization, which does not take into account factors such as shape deformation. Because of the similarity of the adjacent regions of the bottle and the hand, and the limited resolution of the data, the hand and the target object get merged. Using the motion-probability-region in the seeding process prevents the bottle and the hand to be merged (see Fig. 4(b)).

Once we have obtained the seeds for all the surfaces, we determine the label for all the unlabeled points using the same procedure as during the initial clustering (see Section 3.1). Using Eq. 2 and Eq. 3, we determine the new labeling of the unlabeled points.

3.4. Refining Re-sampled Particles

In case of depth data, the employed quadratic surface fitting provides clusters of points whose spatial arrangement varies smoothly in 3-D space and is invariant to illumination changes. For the aforementioned reason, we did not find any advantage of periodically updating the target appearance model [27, 7] and hence omitted this step. A simple computation of the mean of the tracked target proved to be sufficient to filter quantization noise in depth data.

We refine the resampled particles of the tracker by averaging the translation component of the affine state matrices $X_{0:m-1}$ from the current time step according to

$$X'_{t,0:m-1} = \begin{bmatrix} A_{t,0:m-1} & (k_{t,0:m-1} + \mathfrak{c}_t)/2 \\ 0 & 1 \end{bmatrix}, \quad (10)$$

where $\mathfrak{c}_t = (\overline{x}_t, \overline{y}_t)$ is the centroid of the adaptive segment, obtained from the seeding and region growing procedure, which feeds back to the particle filter. The refined states $X'_{t,0:m-1}$ are used by the tracker for prediction at time step $t+1$.

4. Results

We tested the tracking algorithm on several in-house recorded depth videos of hand-object manipulations using a Microsoft Kinect and a PMD camera. The results for selected frames are shown in Figs. 5, 6 and 10. All the datasets along with the tracking results are made available for the readers [1]. In each of the examples, our goal is to track the manipulated object. To the best of our knowledge, to date there is no benchmark dataset for depth videos publicly available. Results obtained with our method are shown for selected movie frames and a video demonstrating the tracking results is provided as supplemental material.

[1] http://www.iri.upc.edu/people/shusain/tracking_data.html

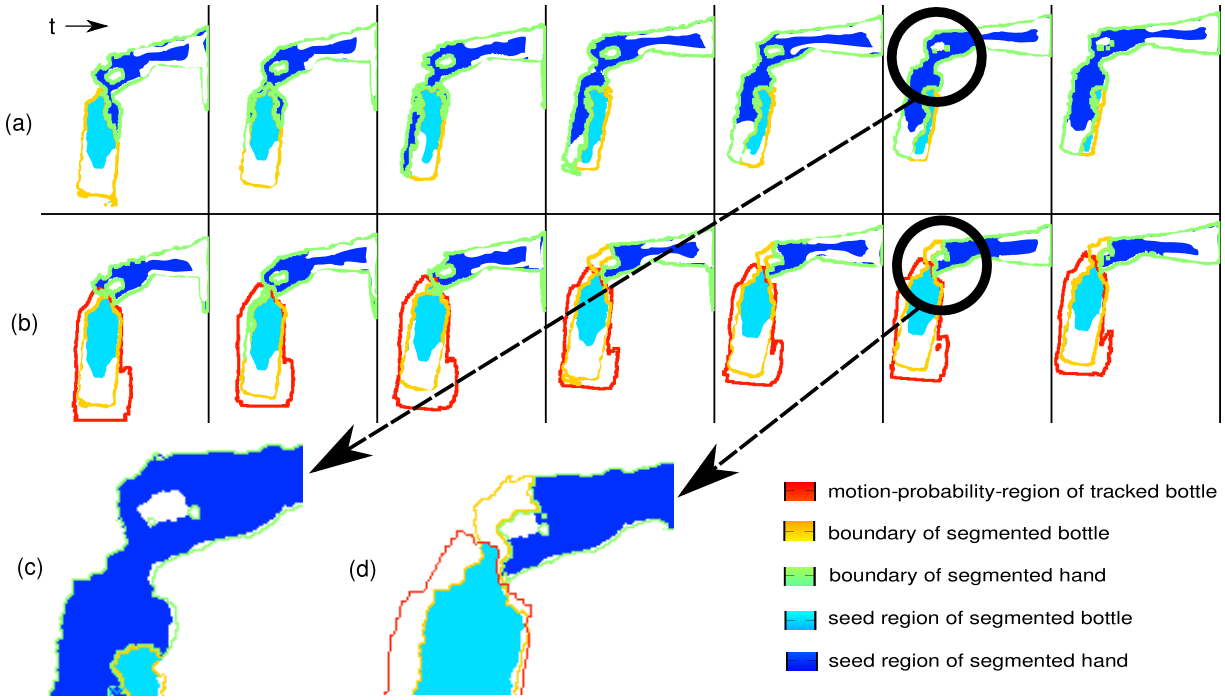

Figure 4: Illustration of the improved seeding procedure for a set of frames at regular intervals of time. (a) Seeds without refinement. (b) Seeds with refinement. (c) Enlarged inset of Fig. 4(a). (d) Enlarged inset of Fig. 4(b).

4.1. Tracking Under Large Translations and Rotations

Figure 5(a) shows a human hand moving a cup such that it undergoes large changes in orientation and position. From frame 104 to frame 188, both the hand and the cup go through the same transformation. Nevertheless, the method succeeded to track the cup correctly, even though the two surfaces got in smooth continuity. Figure 5(b) shows example of a human hand displacing a bottle from one spot to another. From frame 65 to frame 93, the bottle was tracked correctly. During this time, it was at the same depth and in physical contact with the surface on top of which it was placed. Figure 5(c) shows a human manipulating another bottle. This time we tested if our tracker is able to handle a full rotation of 180 degrees involving also translation and scaling. Despite these challenges, the shape of the bottle could be preserved while being tracked. Figure 6 shows tracking result for a cylindrical object with a low resolution PMD camera (200×200 pixels). The object was lifted up from the ground and then manipulated in different ways. It can be seen that the object was tracked correctly before, during, and after the human hand manipulated it.

4.2. Performance Evaluation and Comparisons

We calculate the RMS error e_{rms} (in number of pixels) between the ground truth centroid $(\bar{x}, \bar{y})$ and the estimated centroid $(\bar{x}', \bar{y}')$ location for quantitative analysis according to

$$e_{\text{rms}} = \sqrt{(\bar{x} - \bar{x}')^2 + (\bar{y} - \bar{y}')^2}. \tag{11}$$

Figure 7 shows a comparison of the RMS error at each time instant for the cup sequence (Fig. 5(a)). The range image sequence was used to track the cup with our approach, i.e., surface fitting along with refining seeds (see Eq. 9) and the approach of [2] (surface fitting only). After frame 80, the surface of the cup got in smooth continuity with the hand and henceforth the surface fitting procedure alone was unable to disambiguate the boundary between the two surfaces and lost tracking, whereas when it was combined with the particle filter, the cup was successfully tracked.

Figure 7 also shows the results using two different particle-filter-based trackers from [7] and [28]. We used the corresponding color images of the cup sequence as an input for these trackers. It can be observed that these trackers have a higher RMS error when compared to ours. This is because the accuracy of color-based track-

Figure 5: Tracking results (red color) for a hand (a) manipulating a cup, (b) moving a bottle and (c) manipulating another bottle using our proposed approach. Depth images are shown using grayscale gradients from black (near) to white (far). For illustration, color image has been overlaid in frame 006, frame 02 and frame 001 of Fig. 5(a), Fig. 5(b) and Fig. 5(c) respectively. The data were recorded using a Microsoft Kinect camera. More results are provided in the accompanying video.

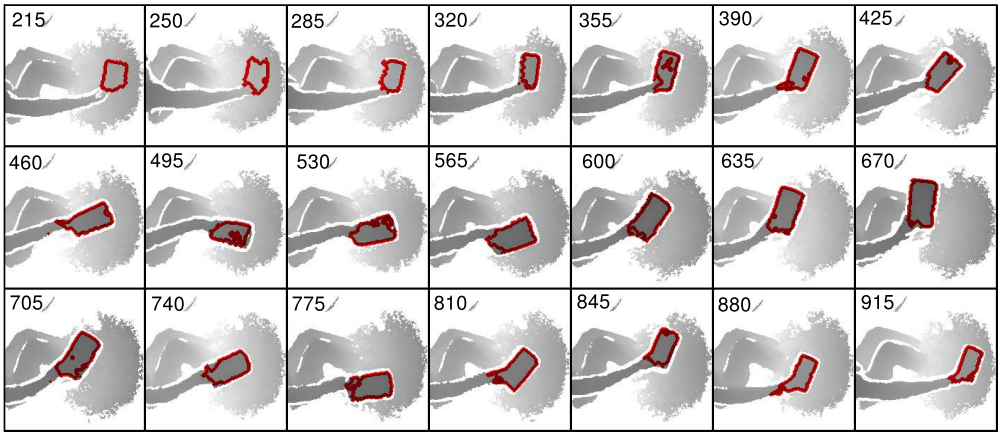

Figure 6: Tracking results (red color) for a cylindrical object manipulated by a human hand, using our proposed approach. The data was recorded using a PMD camera. Depth images are shown using grayscale gradients from black (near) to white (far). More results are provided in the accompanying video.

Table 1: Comparison of the average RMS and the region size error for the cup sequence.

	Our Approach	**Dellen et al.**	**Ross et al.**	**Kwon et al.**
$\bar{e}_{\text{rms}}$	5.69	116.86	13.20	76.52
$\bar{e}_{\text{size}}$	542.5	4368.1	2827.3	2983.6

ers depends on the richness of the texture in the color image.

We also compute the difference e_{size} between the size (in number of pixels) of the tracked surface in the image plane s' and the size of the surface in the ground truth s, i.e.,

$$e_{\text{size}} = |s - s'|. \tag{12}$$

Figure 8 shows a comparison of the region size using the four approaches described earlier. In Fig. 7 and Fig. 8 it can be seen that the method from [28] (green color), even though it was able to track the surface, could not maintain its size.

Table 1 shows the average of the RMS error and the region size error for the four approaches plotted in Fig. 7 and Fig. 8.

We also compare our approach to an ICP-based tracker [18]. We compute the mean of the nearest neighbor distances for each point on the tracked surface with respect to the ground truth. To find the nearest neighbors, we used the approach from [29]. Figures 9(a) and (b) show a comparison for the sequences shown in Fig. 5(a) and (c), respectively. The ICP algorithm was able to keep track of the surfaces but failed to correctly determine surface orientation, hence we see a greater average nearest neighbor distance with respect to the ground truth when the surface is rotating. Our approach

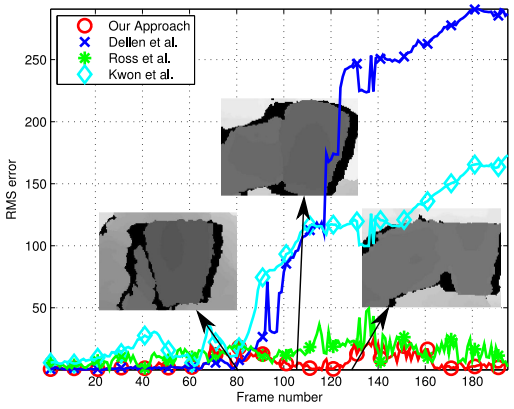

Figure 7: Comparison of the RMS error of the centroid with respect to the ground-truth for tracking with our approach (in red), surface fitting only [2] (in blue), the tracker from [28] (in green), and the tracker from [7] (in cyan) shown for the cup sequence (Fig. 5(a)). Enlarged insets of selected frames from the cup sequence are also shown.

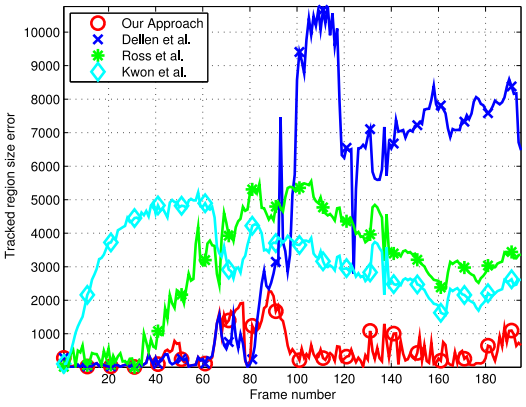

Figure 8: Comparison of error in the size of the tracked region with respect to the ground-truth for tracking with our approach (in red), surface fitting only [2] (in blue), the tracker from [28] (in green), and the tracker from [7] (in cyan) shown for the cup sequence (Fig. 5(a)).

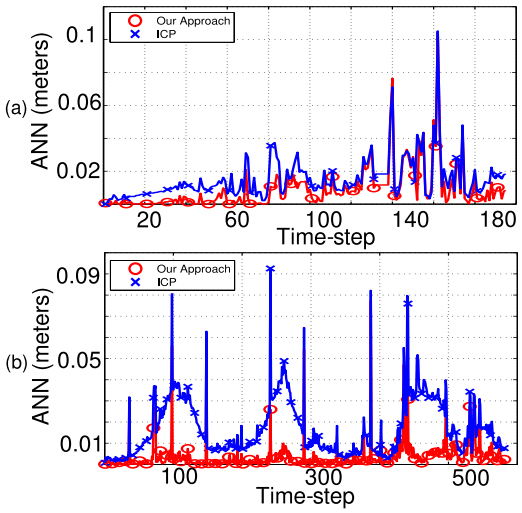

Figure 9: Plots of Average Nearest Neighbor distances ((a) and (b)) with respect to the ground truth, for the sequences shown in Figs. 5(a) and (c) respectively.

clearly outperforms the ICP-based tracker. To create the ground-truth, we first over-segmented the video using the method proposed by [30] and then manually relabeled the segments that belong to the tracked object.

4.3. Increased Particle-Filter Effectiveness

In our approach, the translation component of the affine state matrix is refined by recomputing it after region growing (see Eq. 10). Hence, we expect a better performance from the particle-filter-based estimator when the translation component of the tracked object dominates other kinds of transformations. This can be illustrated by tracking a spherical surface, since it can undergo translation and scaling only. Figure 10(a) shows selected depth images from [2] together with the tracking results (red color). We determine the efficiency, by finding the number of effective particles, i.e., $N_{\text{eff}} = 1/\sum_i (\tilde{w}_t^i)^2$, as defined in [26, 7], where $\tilde{w}^i$ are the normalized importance weights (for details see [7]). The number of effective particles provides a measure of how well the tracker managed to predict the future state of the object. Figure 10(b) shows N_{eff} with (blue line) and without (red line) recomputing translation. We have used 15 particles in all our experiments, hence N_{eff} can vary between 1 (worst) to 15 (best). Clearly, the number of effective particles increases after applying the refinement.

4.4. Implementation Details

Currently, the algorithm is able to process $\sim$ 2 frames per second for a frame size of 200×200 pixels in Matlab on Intel Xeon 3.3 GHz processor. We have implemented the particle filter in C++ which runs at ~20 frames per second. With a complete C/C++ implementation of the method, we expect to reach real-time performance.

5. Conclusion

In this paper, we have proposed a novel approach to surface tracking by combining a state-of-the-art particle-filter-based tracker with a clustering method based on surface fitting. The combination of both methods allowed tracking of object surfaces in videos acquired with depth cameras (Kinect and PMD) despite their limited resolution and accuracy. Object surfaces could be tracked correctly even in situations where the object got in contact with other objects or got touched by the manipulator, assimilating the shape of the tracked object, which represents a highly challenging test case for tracking in depth movies. Since our method takes past measurements into account, errors arising in the method can be reduced, leading to an increased performance as compared with an ICP-based tracker, as seen in Figure 9.

The method could fail to track a complex surface that cannot be well fitted using a second order polynomial equation. Tracking performance is also dependent on the depth resolution of the range sensor. For example, the boundary of a tracked surface might become totally indistinguishable, when touching other surfaces. In such cases, the color based trackers could be used to compliment our method.

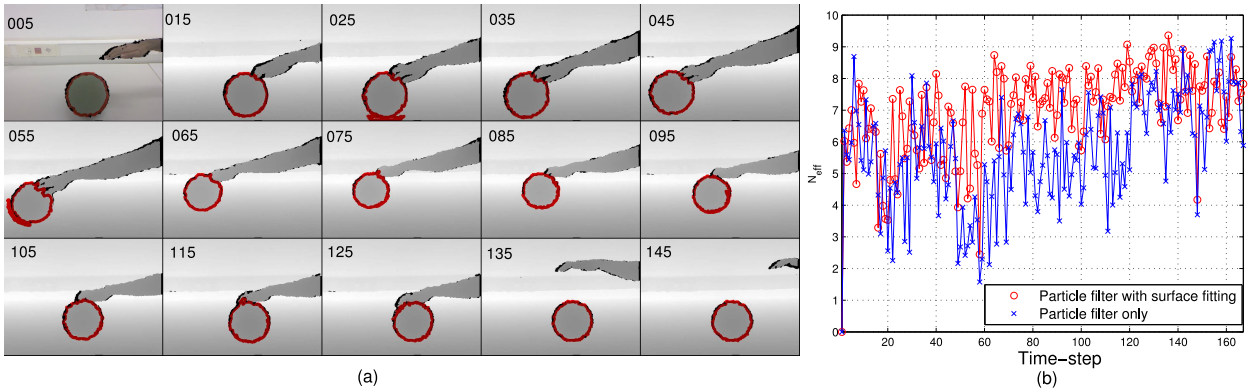

Figure 10: (a) Tracking a rolling ball. For illustration, color image has been overlaid in frame 005. (b) Plot of N_{eff} of the rolling ball.

The proposed approach could be extended to track multiple objects simultaneously while maintaining segmentation by employing multiple particle filters to model the individual motion of the different surfaces.

Acknowledgements

This work received support from the CSIC project MVOD no. 201250E028, the EU project IntellAct FP7-269959, the project PAU+ DPI2011-27510 and the project CINNOVA 201150E088. B. Dellen was supported by the Spanish Ministry for Science and Innovation via a Ramon y Cajal fellowship.

References

[1] A. Yilmaz, O. Javed, M. Shah, Object tracking: A survey, ACM Computing Surveys 38 (2006).

[2] B. Dellen, F. Husain, C. Torras, Joint segmentation and tracking of object surfaces along human/robot manipulations, Int. Conf. on Comput. Vision Theory and Applicat., 2013, pp. 244 –251.

[3] B. Sabata, J. K. Aggarwal, Surface correspondence and motion computation from a pair of range images., Comput. Vision and Image Understanding 63 (1996) 232–250.

[4] X. Jiang, S. Hofer, T. Stahs, I. Ahrns, H. Bunke, Extraction and tracking of surfaces in range image sequences, Second Int. Conf. on 3-D Digital Imaging and Modeling, 1999, pp. 252–260.

[5] D. Kim, D. Kim, A fast icp algorithm for 3-d human body motion tracking, IEEE Signal Process. Lett. 17 (2010) 402–405.

[6] R. Sandhu, S. Dambreville, A. Tannenbaum, Point set registration via particle filtering and stochastic dynamics, IEEE Trans. on Pattern Anal. and Mach. Intell. 32 (2010) 1459–1473.

[7] J. Kwon, K. M. Lee, F. Park, Visual tracking via geometric particle filtering on the affine group with optimal importance functions, IEEE Conf. on Comput. Vision and Pattern Recognition, 2009, pp. 991 –998.

[8] B. Babenko, M. Yang, S. Belongie, Robust object tracking with online multiple instance learning, IEEE Trans. on Pattern Anal. and Mach. Intell. 33 (2011) 1619–1632.

[9] J. Fan, G. Zeng, M. Body, M. S. Hacid, Seeded region growing: an extensive and comparative study, Pattern Recogn. Lett. 26 (2005) 1139–1156.

[10] E. Chen, Y. Xu, X. Yang, W. Zhang, Quaternion based optical flow estimation for robust object tracking, Digital Signal Processing 23 (2013) 118 – 125.

[11] T. Huang, A. Netravali, Motion and structure from feature correspondences: a review, Proceedings of the IEEE 82 (1994) 252–268.

[12] V. Ganapathi, C. Plagemann, S. Thrun, D. Koller, Real time motion capture using a single time-of-flight camera, IEEE Conf. on Computer Vision and Pattern Recognition, 2010.

[13] S. Knoop, S. Vacek, R. Dillmann, Sensor fusion for 3d human body tracking with an articulated 3d body model, IEEE Int. Conf. on Robotics and Automation, 2006, pp. 1686 –1691.

[14] L. V. Tsap, M. C. Shin, Dynamic disparity adjustment and histogram-based filtering of range data for fast 3-d hand tracking, Digital Signal Processing 14 (2004) 550 – 565.

[15] I. Oikonomidis, N. Kyriazis, A. Argyros, Tracking the articulated motion of two strongly interacting hands, IEEE Conf. on Comput. Vision and Pattern Recognition, 2012.

[16] M. Krainin, P. Henry, X. Ren, D. Fox, Manipulator and object tracking for in-hand 3d object modeling, Int. Journal Robotics Research 30 (2011) 1311–1327.

[17] P. Besl, H. McKay, A method for registration of 3-d shapes, IEEE Trans. on Pattern Anal. and Mach. Intell. 14 (1992) 239–256.

[18] S. Rusinkiewicz, M. Levoy, Efficient variants of the ICP algorithm, Int. Conf. on 3D Digital Imaging and Modeling, 2001.

[19] S. Foix, G. Alenyà, J. A. Cetto, C. Torras, Object modeling using a tof camera under an uncertainty reduction approach, IEEE Int. Conf. on Robotics and Automation, 2010, pp. 1306–1312.

[20] S. Granger, X. Pennec, Multi-scale EM-ICP: A fast and robust approach for surface registration, volume 2353, ECCV, 2002, pp. 418–432.

[21] S. Gold, A. Rangarajan, C.-P. Lu, S. Pappu, E. Mjolsness, New algorithms for 2d and 3d point matching: pose estimation and correspondence, Pattern Recognition 31 (1998) 1019–1031.

[22] T. Tamaki, M. Abe, B. Raytchev, K. Kaneda, Softassign and EM-ICP on gpu, International Conference on Networking and Computing, 2010, pp. 179–183.

[23] R. Lakaemper, L. Latecki, Using extended em to segment planar structures in 3d, volume 3, Int. Conf. on Pattern Recognition, 2006, pp. 1077–1082.

[24] S.-M. Rhee, Y.-B. Lee, J. D. K. Kim, T. Rhee, Split and merge

approach for detecting multiple planes in a depth image, IEEE Int. Conf. on Image Processing, 2012, pp. 1213–1216.
[25] D. A. Forsyth, J. Ponce, Computer Vision: A Modern Approach, Prentice Hall, 2003.
[26] A. Doucet, S. Godsill, C. Andrieu, On sequential monte carlo sampling methods for bayesian filtering, Statistics and Computing 10 (2000) 197–208.
[27] A. Jepson, D. Fleet, T. El-Maraghi, Robust online appearance models for visual tracking, IEEE Trans. on Pattern Anal. and Mach. Intell. 25 (2003) 1296–1311.
[28] D. Ross, J. Lim, R.-S. Lin, M.-H. Yang, Incremental learning for robust visual tracking, Int. Journal of Comput. Vision 77 (2008) 125–141.
[29] J. H. Friedman, J. L. Bentley, R. A. Finkel, An algorithm for finding best matches in logarithmic expected time, ACM Trans. Math. Softw. 3 (1977) 209–226.
[30] M. Grundmann, V. Kwatra, M. Han, I. Essa, Efficient hierarchical graph-based video segmentation, IEEE Conference on Computer Vision and Pattern Recognition, 2010, pp. 2141–2148.

Farzad Husain received a Master Degree in Electrical Engineering with emphasis on Signal Processing, in 2011, from Blekinge Institute of Technology, Sweden. He is currently pursuing his PhD in the Automatic Control, Robotics and Computer Vision programme at Universitat Politècnica de Catalunya, Spain. His research is mainly focused on the semantic analysis of range data, acquired using different range sensing devices.

Babette Dellen has studied Physics and received her PhD degree from Washington University in St. Louis (USA) in 2006. Between 2006 and 2010 she worked as a Postdoc in the Department for Computational Neuroscience at the Bernstein Center at the University of Goettingen and the Max-Planck-Institute for Dynamics and Self-Organization in the field of computer vision. From 2010 to 2014 she worked at the Institut de Robòtica i Informàtica Industrial in Barcelona, Spain. Since 2014 she is a Professor at the RheinAhrCampus of the Hochschule Koblenz in Germany. Her main research interests are computer vision methods for generating suitable visual representations for robotic applications.

Carme Torras (http://www.iri.upc.edu/people/torras) is Research Professor at the Spanish Scientific Research Council (CSIC). She received M.Sc. degrees in Mathematics and Computer Science from the Universitat de Barcelona and the University of Massachusetts, Amherst, respectively, and a Ph.D. degree in Computer Science from the Universitat Politècnica de Catalunya (UPC). Prof. Torras has published five books and about two hundred papers in the areas of computer vision, neurocomputing, geometric reasoning, and robotics. She has been local project leader of several European projects, among which the ongoing “Intelligent observation and execution of Actions and manipulations" (IntellAct). She was awarded the Narcís Monturiol Medal of the Generalitat de Catalunya in 2000, and she became ECCAI Fellow in 2007 and member of Academia Europaea in 2010.