Time consistent estimation of End-effectors from RGB-D data

Xiao Lin, Josep R.Casas, and Montse Pardás

Image processing Group, Technical University of Catalonia (UPC), Barcelona, Spain

Abstract. End-effectors are usually related to the location of the free end of a kinematic chain. Each of them contains rich structure information about the entity. Hence, estimating stable end-effectors of different entities enables robust tracking as well as a generic representation. In this paper, we present a system for end-effector estimation from RGB-D stream data. Instead of relying on a specific pose or configuration for initialization, we exploit time coherence without making any assumption with respect to the prior knowledge. This makes the estimation process more robust in a predict-update framework. Qualitative and quantitative experiments are performed against the reference method with promising results.

Keywords: End-effector estimation; Time coherence; Topology representation

1 Introduction

In recent years, Human Motion Analysis (HMA) has made great progress in constrained scenarios. It achieved remarkable results for pose estimation and gesture recognition with isolated human body data. But a large proportion of our visual experience involves analyzing the interactions between humans and objects. We use the term ”entities” in this paper to refer to anything that we can model with a star-graph, including humans and objects, as illustrated in Fig. 1. This requires systems to be able to represent different entities in a generic way, which makes it a difficult problem because entities could vary greatly in appearance. Those variations arise not only from changes in illumination and viewpoint, but also due to non-rigid deformations and intra-class variability in shape and other visual properties.

Previous work for generic entity representation mainly focus on two different strategies. First, global representation is usually performed by extracting global features within a bounding box of the entity. This strategy is usually employed when spatial layouts of entity appearances are roughly rigid, such as faces or pedestrians at a distance. It has limited performance in complex scenes, as it highly relies on the extracted features to not only characterize different types of entities, but also cover the intra-class variability. The other way to achieve a generic representation is to represent the global structure of local descriptors/features of the entity, such as with a Deformable Part based Model (DPM). DPM shows its potential to be generic due to the loose constraints between the part representations. Several DPM based approaches [3, 7, 1] have proved their ability to represent different kinds of entities by customizing the model with different training data. Similarly, Wang *el al.* [15] take the detection results of a selective search approach [14] as candidates, and represent each candidate with the region-let feature which stands for spatial structure and appearance of some salient small regions. However, these approaches strongly rely on training data to learn the appearance models for the local patches and the spatial models for their global structure. Employing training in the entity representation process makes the model lose its generality to represent other objects except the trained one.

With the motivation of representing generic entities, we seek for the elemental factors among them, which will be of great importance when establishing a simple and more loosely structured generic model in the future. As shown in [10], end-effector is a common factor for both deformable entities and rigid entities (See the human body and chair shown in Fig. 1) while it only changes when the geometric structure of the entity changes, which makes it possible to represent generic entities. Hence, in this paper, we propose a novel way to estimate end-effectors of generic entities based on the topological representation of its 3D point cloud from RGB-D stream data. We exploit time coherence to make the estimation process more robust in a predict-update framework, instead of relying on a specific pose or configuration for initialization. Our contributions can be summarized in the following points:

- We propose a novel way to evaluate the correctness of being an end-effector based on a new approximation of its geodesic distance to the entity centroid.
- Estimating end-effectors without training and initialization provides the proposed system the potential to estimate end-effectors of generic entities in stream data.
- We introduce temporal information to facilitate the task of point cloud end-effector estimation, which makes the proposed system more robust to topology changes caused by occlusion and body part interaction.
- We avoid strongly relying on the previous information and initialization, which provides the system the ability to recover from estimation errors.

In the rest of the paper, we first review the related work in Section 2. Then, the input data and preprocessing steps are described in Section 3. In Section 4, we present a single frame end-effector estimation approach, in which the point cloud is topologically organized into different bands. End-effectors are estimated by analyzing the point cloud topology from its band-based representation. This topology is sometimes not reliable because of the presence of occlusion and parts interaction, which will probably change the topology of the point cloud. Therefore, in Section 5, we analyze the end-effector estimation process along time and propose a time coherence guided end-effector estimation approach to address the above mentioned problems. Finally, Section 6 shows the quantitative and qualitative experiment results of the proposed method.

2 Related work

End-effector estimation from RGB-D data is currently an open task as a result of the increasing performance of consumer depth sensors. Baak *el al.* [2] construct a weighted graph for point cloud data by taking each point on the point cloud as a graph node and exploiting the neighborhood structures in the pixel domain to build edges between them. Then, edges are weighted as the Euclidean distances between connected node

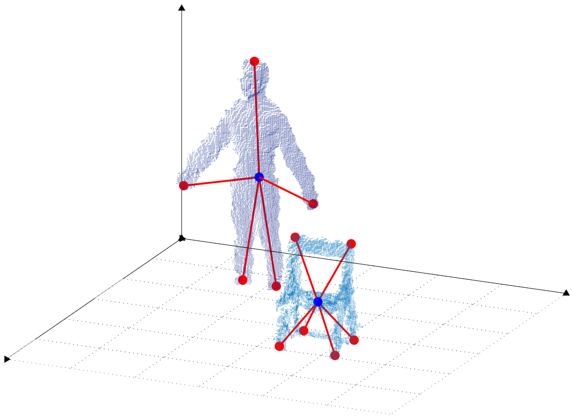


Fig. 1. Different entities represented by end-effectors

pairs. End-effectors are searched as the extrema with respect to the centroid of the point cloud using Dijkstra's algorithm. Suau *el al.* [13] employ a level set method in RGB-D data to exploit connectivities over the depth surface in order to extract a topological representation of point clouds. But these two approaches are not effective in solving the problem when deformable entities change the topology of the graph, in which case the detected extrema may not correspond to the position of the end effectors. Schwarz *el al.* [11] propose to use optical flow to handle the topology change problem. Similar to [2], a graph is constructed to represent the topology of the point cloud. Then motion information is employed to disambiguate body parts when occlusion and body part attachment (that is, when end effectors are connected to other body parts) occur. However, they introduce a human body structure in order to segment body parts and rebuild the graph, which makes the model not generic to estimate end-effectors for other entities. Besides, it requires an initialization to start the algorithm by requesting a T-pose at the beginning, which also makes it difficult to recover from the tracking errors.

3 Input data

A consumer depth sensor provides RGB-D data at video frame rates by analyzing a speckle pattern of infrared light. It captures color and relatively accurate depth data at the same time. A depth image $I_{depth} : Z^2 \rightarrow R$ contains the distance $d \in R$ for each pixel position $p \in Z^2$. Thus, given the camera parameters, we can transform the per-pixel distances into a 3D point cloud $C_I \subseteq R^3$. It contains both foreground points and background points. To extract the foreground point cloud $C_{fg} \subseteq R^3$ from the scene, a group of thresholds restricting the activity area in 3D space are involved. In the proposed system, we take the foreground point cloud as input data.

4 Single frame end-effector estimation

An end-effector is defined as the free end of a kinematic chain. Estimating end-effectors for 3D point cloud data requires representing the topology of the point cloud with a well organized structure. Thus, we exploit the approach proposed in [13] to describe the topology of point clouds. Then, a new strategy proposed in [2] for searching the end-effectors from its topology representation is integrated into the proposed system, providing a better performance than the strategy used in [13].

4.1 Point cloud topology description

Geometric Deformable Models (GDMs) [5, 8] have proved performance and flexibility on describing topology. They are based on the theory of curve evolution and level set methods [12]. Its basic idea is to evolve the initial contour on the data domain according to predefined internal and external forces. Internal forces lead the actual contour curve to expand itself while keeping its smoothness. External forces are modeled from the data and work against internal forces by countering the curve expansion, which actually makes the contour curve evolve along the topology of the data.

Following the work in [13], the external forces in the proposed system come from the foreground point cloud data $C_{fg} = \{x_i\} \subseteq R^3$ obtained in the previous step. The internal forces are defined as the expansion power. Let $\phi(x, t) : R^3 \to R$ be a level set function which provides an implicit representation of the evolving curve at time t. Let $curve(t)$ be the contour curve as the zero level set of $\phi(x, t)$, and $L_t^0 \subset C_{fg}$ is the subset enclosed by $curve(t)$. The objective is to make $curve(t)$ evolve over C_{fg} while preserving the topological properties of the point cloud data. The way to evolve it from time t to $t + 1$ is to expand and include a set of new points regarding the previous level set L_t^0, under the constraints of both proximity and density. Specifically, it is formulated as:

$$\begin{array}{l} L_{t+1} = \{x_i\} \quad if \quad \phi(x_i, t) < \delta_L \quad and \quad \rho(x_i) \\ L_{t+1}^0 = L_t^0 \cup L_{t+1} \end{array} \tag{1}$$

where,

$$\phi(x, t) = \begin{cases} 0 & \forall x \in L_t^0 \\ min\left(dist_E\left(x, curve(t)\right)\right) & \forall x \notin L_t^0 \end{cases} \tag{2}$$

$$\rho(x) = \begin{cases} True \;\; if \quad Num(x, \delta_L) >= \eta_L \\ False \quad if \quad Num(x, \delta_L) < \eta_L \end{cases} \tag{3}$$

L_{t+1} stands for the set of new points included at time $t+1$ which is called narrow band. The proximity constraint limits the Euclidean distance from points in L_{t+1} shorter than δ_L, and the density constraint is performed by requiring the number of its neighbor points within a ball area of radius δ_L larger than η_L. To complete the formulation, we should define how the contour curve is updated in the sense of point cloud data:

$$curve(t) = \{x_i\} \quad if \begin{cases} x_i \in L_{t+1} \\ \phi(x_i, t) \in \left[\frac{3}{4}\delta_L, \delta_L\right] \end{cases} \tag{4}$$

The contour curve at time t is defined as the points which are farther from the previous zero level set. Therefore, the foreground point cloud C_{fg} will be organized into bands at

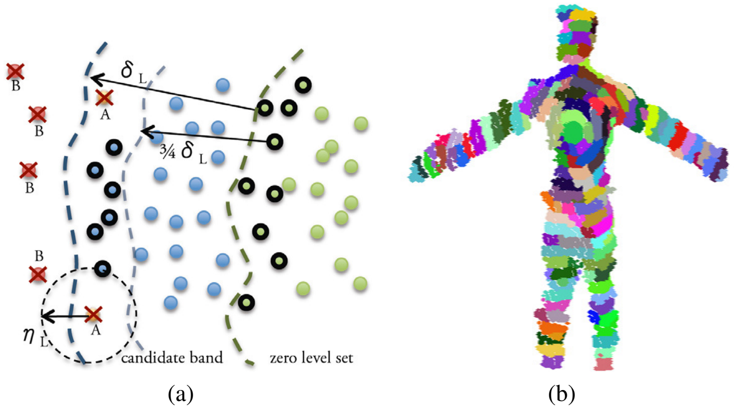


Fig. 2. Curve evolution. (a) The green points are the actual zero level set L_t^0, and those with a thick black boundary form $curve(t)$. The blue points are the organized band L_{t+1}, with the $curve(t+1)$ also marked with thick boundary. Orange points A are rejected to be included in L_{t+1}^0 because of the density constraint. Orange points B are rejected to be included in L_{t+1}^0 because of proximity constraint. (b) An example of a body topology representation, different colors stands for different sub-bands. (Fig. 2(a) is extracted from [13])

different levels by iterating through Eqs. (1)-(4) from an initial level set L_0^0. Fig. 2 shows an example of this process. Narrow bands cover the foreground point cloud, organizing its topology into different bands. However, a band may contain points belonging to different context (*e.g.* different arms of a human body). To address this, each band L_k is separated into sub-bands with respect to a constraint on the maximal number of points in a sub-band.

4.2 Estimating end-effectors in a topologically weighted graph

The sub-bands obtained in the previous step belong to visible surfaces of the analyzed entity, which also implicitly provides the approximate topology information about its point cloud (See Fig. 2(b)). To utilize these context-wise sub-bands and estimate the end-effectors of a point cloud, we construct a graph based on these sub-bands. The following three points should be considered when constructing the graph:

– Graph nodes
  In this topological graph, the centroid of each sub-band is taken as a graph node.
– Graph root
  As introduced in the second paragraph of Section 4.1, an initial zero level set L_0^0 is needed as a starting point in the topology description procedure. Such an origin set is treated as the graph root. The graph root could be a single 3D point or a set of points, depending on the application. In the case of human body, we propose to compute the centroid of the point cloud, and start with the point which has the shortest Euclidean distance to the centroid as a rough initial zero level set and obtain a rough band

structure. Then, we use a partition of the point cloud which is composed of the first N level of bands to compute the centroid again. N stands for the size of the massive part of an entity (*e.g.* N is related to the size of torso in the application of estimating human body end-effectors). In this manner, the estimated centroid is more robust to different poses.

- Edges and weights
  Graph nodes are linked in pairs with graph edges. We propose to only include those edges which link nodes attached to each other. Attachment between node n_i and n_j requires both of them to have at least P points with their closest distance to the other node lower than λ. A distance weight for an edge $w_{i,j}$ is defined as the Euclidean distance between the centroid of these two nodes (bands) in Eq. (5):

$$w_{i,j} = dist_E(n_i, n_j) \tag{5}$$

An end-effector is then searched as a node with its shortest path to the graph root longer than the other graph nodes. We exploit the strategy used in [2] to detect end-effectors on the graph. According to this strategy, we first search the node with the longest shortest path to the target node which is set to be the graph root initially. Then we add edges with zero weight between each node on this path and the target node on the graph and update the target node to the previous detected end-effector. These operations are performed iteratively till the estimated end-effector has the geodesic distance lower than a threshold g. This strategy favours possible end-effectors on the opposite side of the previous detected one while avoiding shortcuts of neighbors.

The geodesic distance of an end-effector is then calculated as the sum of all weights of the included edges on its path to the graph root:

$$dist_G(e_t) = \sum^{edge(i,j) \in path} w_{i,j} \tag{6}$$

5 Temporal coherence guided end-effector estimation

For end-effectors extracted by analyzing the topology of a point cloud, one obvious weakness is that the estimation result will be changed with the dynamic changes in the topology due to motion. The main reasons for topology changes are listed below:

- Occlusion
  In this case, part of the surface of the entity is occluded by the occluder (*e.g.* arms of human body), which results in an incomplete point cloud. This is even worse when the auto-occlusions separate naturally connected entity parts.
- Parts interaction
  Articulated entities such as human body deform by moving the rigid parts connected to the articulations. These deformations could lead to interactions between parts like merge and split. For example, the hands are not extremes of a human body when both arms attach tightly to the torso.

End-effectors can only be correctly detected for certain topological configurations of the point cloud. However, once detected, they can be tracked in order to extract the stable configurations that can be used in higher level applications. Therefore, we propose to make use of the temporal information between frames when dealing with the task of end-effector estimation from RGB-D streams data in order not to lose them when the topology changes. We favor the natural detection of end-effectors, but if one was there and has suddenly disappeared (because, for instance, a hand touched the body), we exploit temporal coherence to search back for the end effector and predict its position in the current frame. This introduces the end-effector estimation task into a predict-update framework. The purpose is to guide the single frame based end-effector estimation with temporal information, but only when needed, so that we do not rely much on temporal prediction. This strategy will avoid initialization and make the system flexible enough to escape from tracking errors.

5.1 Predict phase

The end-effectors detected at a given frame are projected into the next frame, in order to maintain the stability of the detection along time. Let $E_t = \left\{e_t^1...e_t^{M_t}\right\} \subseteq R^3$ be the end-effector set at time t, M_t be the number of end-effectors estimated. Then a dynamic model is defined as $E_{t+1}^{pr} = Dnc(E_t, \theta_t)$, in which E_{t+1}^{pr} stands for the end-effector predictions of E_t, θ_t is the parameter of the dynamic model at time t. We use a motion estimation technique to compute the velocity of each end-effector from time t to $t+1$ based on color images $\left\{I_{color}^t, I_{color}^{t+1}\right\} = \theta_t$. Specifically, forward optical flow is computed using Large Displacement Optical Flow (LDOF) [4]. The obtained optical flow is back-projected from the image plane to the 3D space using depth information and the camera parameters to obtain 3D scene flow in the real world. Consequently, the dynamic model $Dnc(E_t, \theta_t)$ propagates the previous end-effectors estimation to the current frame by using the velocity information in the scene flow.

5.2 Update phase

We take the end-effectors in E_{t+1}^{pr} obtained in the previous step as a set of end-effector candidates for the current frame. Another set of end-effector candidate E_{t+1}^{est} is estimated locally by analyzing the topology of the current point cloud as explained in Section 3.2. The final end-effectors E_{t+1} will be generated within these two sets of candidates. In order to select the best end-effectors, we propose to establish one-to-one correspondences between elements in E_{t+1}^{pr} and E_{t+1}^{est} so that we can compare each pair of them based on a selection rule and generate the final end-effectors.

End-effector correspondence assignment To establish the correspondence, we represent the end-effector candidates with features and compute the similarity in the feature space. These features may contain different information such as 3D position, color, local 3D shape etc. In our experiments, we just use the 3D position as the feature for the end-effector candidates and we define the similarity according to the geodesic distance between them with respect to the topologically weighted graph constructed in current frame, which is inversely proportional to the geodesic distance. Note that the position of predicted end-effector candidates are rectified as the closest point on the current point

Fig. 3. (a) Approximate geodesic path with mid-band. (b) Benefit of geodesic distance updating.

cloud in order to compute the geodesic distance in the current frame. As E^{pr}_{t+1} and E^{est}_{t+1} are two disjoint sets, we construct a weighted bipartite graph by adding all possible edges connecting one vertex in E^{pr}_{t+1} and one in E^{est}_{t+1} and weighting them with similarities. Then, assigning the correspondence between end-effector candidates in E^{pr}_{t+1} and E^{est}_{t+1} is converted to a maximum weighted bipartite graph matching problem. We employ the Hungarian algorithm [6] to solve it.

Evaluate the end-effector candidates Once the correspondence is established, we need to evaluate the proposed candidates in order to generate the final end-effectors. When we obtain an end-effector with a large geodesic distance to the centroid, we are in a pose which strongly hints at the actual presence of an end-effector, and that is where we can rely on, in the end-effector estimation. However, the geodesic distance calculated based on the full shortest path is sensitive to occlusions. Fig. 3(a) shows an occluded area on the torso, in which the black line stands for the full shortest path between an end-effector to the graph root and the red line is the approximated path. The occlusion leads to a longer estimated geodesic distance for the end effector candidate placed on the head. Thus, to make it more robust to occlusions, we propose to approximate the shortest path by just taking into account a mid-band in that shortest path when calculating the geodesic distance. Specifically, the approximated geodesic distance for an end-effector candidate estimated at frame t is formulated as the summation of two parts of Euclidean distance:

$$dist^{ap}_G(e^i_t) = dist_E(e^i_t, mid) + dist_E(mid, L^0_0) \tag{7}$$

where mid stands for the mid-band of the shortest path from end-effector candidate e^i_t to the centroid. For the predicted end-effector candidates, since they are not an extrema estimated in the current frame, their distance is assimilated to the approximated geodesic distance in the previous frame. Then, the approximated geodesic distance of an end-effector candidate is calculated for the comparison. The comparison is performed depending on different situations. According to the similarity between a pair of corresponding end-effector candidates, we categorize them into three cases:

- Case A: Related pair. It stands for corresponding candidates with similarity higher than a predefined threshold ε_a.

- Case B: Unrelated pair. It represents corresponding candidates with similarity larger than ε_b but lower than ε_a, which means the corresponding candidates are not the same end-effector although they were matched in the previous step. Threshold ε_b is set to filter out some extreme cases, such as when both of the two matched candidates are valid end-effectors. In this case, as the geodesic distance between two valid end-effectors $dist_G(e_1, e_2)$ is normally larger than the geodesic distance from each of them to the centroid ($dist_G(e_1)$ and $dist_G(e_2)$), we set ε_b with respect to larger one between $dist_G(e_1)$ and $dist_G(e_2)$.
- Case C: Unmatched candidate. As there may be new end-effectors detected in the current frame or tracked end-effectors that do not correspond to any extrema in the current frame, the number of candidates in E^{est}_{t+1} and E^{pr}_{t+1} may not be the same. Thus, there will be candidates which have no correspondence with any other candidates which we call unmatched candidates. Some matched pairs with similarity lower than ε_b are also included as unmatched candidates.

End-effector selection and geodesic distance update Final end-effectors are selected from the end-effector candidates depending on the above mentioned three cases. In cases A and B, we take the end-effector candidate with the longest geodesic distance as a final end-effector. In case C, the candidate will be kept as an end-effector, for a maximum of T frames. If it is not matched (thus becoming case A or B) during these frames, it will be removed. Thus, we can preserve the possibility of tracking correct estimations while avoiding permanent errors.

Finally, we update the geodesic distance of the selected end-effectors by considering both previous and current information according to Eq. (8).

$$dist^{ap}_G(e_{t+1}) = \begin{cases} \alpha \times dist^{ap}_G(e_t) + (1-\alpha) \times dist^{ap}_G(e^{pr}_{t+1}) if \quad e_t \in E^{pr}_{t+1} \\ dist^{ap}_G(e^{est}_{t+1}) \quad if \quad e_t \in E^{est}_{t+1} \end{cases} \tag{8}$$

where the updated geodesic distance of the final end-effector e_{t+1} is defined as the weighted sum of its geodesic distance in frame t and the geodesic distance of its prediction in frame $t+1$ if this end-effector is from E^{pr}_{t+1}. Otherwise, it is equal to its geodesic distance in frame $t+1$, as it is newly detected. These end-effectors are treated as the final estimation which will be predicted to the next frame when we process frame $t+2$. The weight used in Eq. (8) is defined as:

$$\alpha = \sigma^k \tag{9}$$

where k is the number of frames that this end effector has not been related with any end-effector candidate in current estimation (cases B and C). $\sigma \in [0, 1]$ is a penalty factor.

Geodesic distance updating ensures that the proposed method is capable to memorize the occurrence of tracked end-effectors and their geodesic distances at that time while not totally relying on it, which reduces the risk of getting stuck with errors in previous information. Fig. 3(b) shows an example of how the proposed system benefits from geodesic distance updating, in which the red point stands for an end-effector candidate estimated from the current data and the black point represents its corresponding candidate in E^{pr}_{t+1}. The black point indicates the correct end-effector position while not being detected from the current frame as both arms attach to the torso. The topology changes caused by body part attachment also make the approximated geodesic distance of the black candidate shorter with respect to the current data. However, as the black candidate is correctly tracked, its geodesic distance, rather than its geodesic distance calculated in the current frame, has been updated based on its information from the motion history, which will leave the black point with the correct end-effector position be favoured in the comparison.

6 Experimental results

We have evaluated our end-effector estimation approach for a human body end-effector estimation application. Berkely Multimodal Human Action Database (MHAD) [9] is employed as the benchmark data set, in which eleven different actions are performed in an indoor scenario and captured by two Kinects. These two Kinects are placed diagonally with respect to the subject. In our experiments, we use the RGB-D stream data from the front Kinect. It has been calibrated in advance and the calibration parameters are available in the database. Each frame of the stream for both depth and color image is 640×480 pixels. In the rest of this section, we present the quantitative and qualitative results of the proposed approach while comparing it with the Restricted Narrow Band Level Set (R-NBLS) approach proposed in [13].

6.1 Quantitative results

We select seven actions in the data base without human object interactions as the quantitative experiment data. Since all the actions are performed five times, we use the first two repetitions in each action sequence, which yields around 70 frames per sequence. We manually marked all the end-effectors for the selected sequences as the ground truth.

We consider that an estimated end-effector is a true positive if it is within a distance threshold with respect to the ground truth. A threshold of 15 cm has been used in the experiments presented in this paper. Three metrics including recall, average distance error and false positive rate are defined as:

$$Recall_t = M_t^{TP}/M_{human} \tag{10}$$

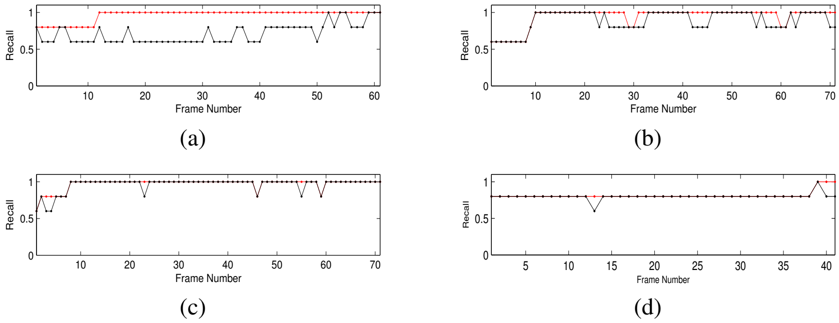
$$\bar{\epsilon}_t = \frac{1}{M_{TP}} \sum_{i=1}^{M_{TP}} dist_E(TP_t^i - GT_t^i) \tag{11}$$

$$FPR_t = M_t^{FP}/M_t \tag{12}$$

We denote the true positive end-effector estimation result at time t as set TP_t and false positive as set FP_t. In equation (10), M_t^{TP} represents the number of correctly estimated end-effectors at time t and M_{human} is a constant number which stands for the natural number of end-effectors of human body. In Eq. (11), we calculate the average Euclidean distance error between end-effector TP_t^i in TP_t and its corresponding ground truth

GT_t^i. In Eq. (12), M_t^{FP} stands for the number of end-effectors in FP_t and M_t is the number of estimated end-effectors at time t.

By considering the presence of significant occlusions, we divide the test sequences into two groups. The first group contains four actions without significant occlusions: *jumping in place*, *jumping jack*, *waving two hands*, *waving one hand*. The second group consists of three actions with significant occlusions: *bending*, *punching*, *clapping*. In Figs. 4, 5 and 6(a), we compare the proposed approach with R-NBLS by evaluating the recall of each frame in the test sequences and average recall of each sequence. The results of the proposed approach and R-NBLS are marked in red and black respectively. Fig. 4 shows the results of the first group. In these four sequences, the proposed method keeps tracking the end-effectors detected in the history while comparing them with the current estimations, which yields the continuity in end-effector estimation. This is illustrated in all the sequences as we achieve consistently better results than R-NBLS. Especially in Fig. 4(a), it is clearly showed that the proposed approach memorizes the presence of end-effectors in the history and continuously detects them in the future. Note that the memory for an occurred end-effector could be also dropped when we lose track of it, see $29th$ frame in Fig. 4(b) or $46th$ frame in Fig. 4(c). Fig. 5 shows the result of the second group. In these three sequences, there is more noise in the current estimations as significant occlusions caused by the three actions change the topology of the point cloud very often. Besides, end-effectors might not be always visible due to significant occlusions. The results show the robustness of the proposed approach with respect to noisy current estimations. Fig. 6(a) presents average recall of each sequence, which results in an overall 91.3% recall compared to 82.1% from R-NBLS. We also compute the average distance error for all the sequences as a secondary metrics in Fig. 6(b). It shows that the proposed approach has $7.5cm$ average distance error over all the test sequences compared to $7.6cm$ of R-NBLS.



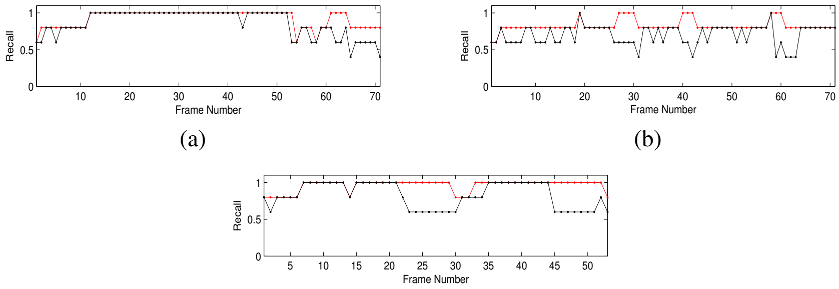
(a)

(b)

(c)

(d)

Fig. 4. (a) Jump1 (Jumping in place). (b) Jump2 (Jumping jacks). (c) Wave hand1 (Waving two hands). (d) Wave hand2 (Waving one hand). Black: R-NBLS approach [13]. Red: the proposed approach.

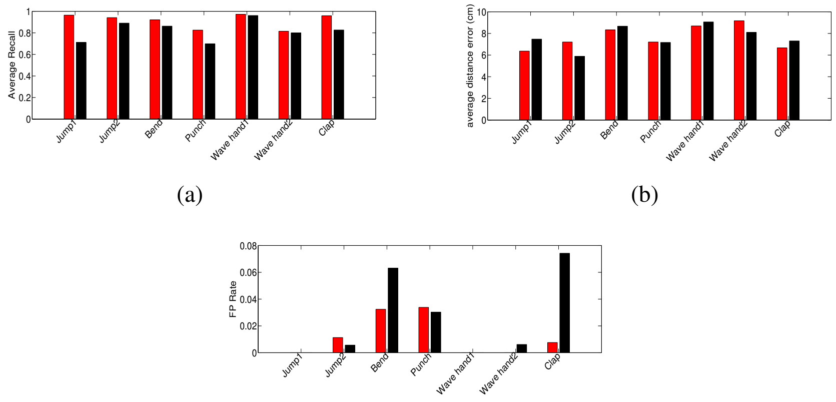


(a)

(b)

(c)

Fig. 5. (a) Bend. (b) Punch. (c) Clap.

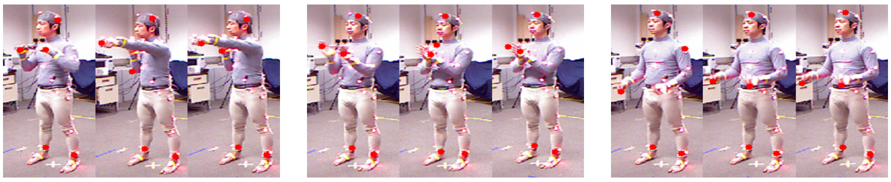


(a)

(b)

(c)

Fig. 6. (a) Average recall for all the test sequences. (b) Average distance error for all the test sequences. (c) False positive rate for all the test sequences.



(a)

(b)

(c)

Fig. 7. Three example results: (a)-(c). In each example, Left: estimation in previous frame, Center: estimation only based in current frame, Right: proposed approach.

The false positive end-effectors will affect the representation of entities in the future. Thus we also evaluate the False Positive Rate (FPR) in the experiments. As shown in Fig. 6(c), the proposed approach (bars in red) has all the FP rates lower than 4%.

6.2 Qualitative result

In Section 6.1, we compared quantitatively the performance of the proposed method with R-NBLS. To further analyze them, several example results are discussed in this section.

Fig. 7(a)-7(a) shows three examples of estimation result. In each example, the left image is the estimation result in the previous frame, the image in the middle shows end-effectors estimated just based on the current frame and the right image shows the result of the proposed approach. All the end-effectors are marked in red in this group of figures.

Fig. 7(a) presents an estimation result in punching sequence. We can see that the temporal information affects the estimation results by tracking the point in the previous frame and keeping the tracked point when we fail to detect it in the current frame. Fig. 7(b) shows the results of a clapping sequence. End-effectors are well estimated in the previous frames while collision occurs in the current frame. Once both hands attach to each other, two end-effectors on both hands are combined into one and a new end-effector on the right shoulder is detected based on the current information. This constitutes an unrelated pair between the newly born end-effector candidate on the shoulder and the end-effector candidate predicted from previous information on the right hand. To deal with collisions, the proposed approach selects the best end-effector by comparing the approximated geodesic distance between them. The right image in Fig. 7(b) shows that the correct estimation with longest approximated geodesic distance survives. Different from the first two examples, the example showed in Fig. 7(c) has an estimation error (the point on the left shoulder) in the result of the previous frame, which is then treated as the previous information of the current frame. As shown in the middle and right image in Fig. 7(c), the proposed method is capable to recover from the estimation error in the previous information when the correct information is available in the current estimation.

7 Conclusion

In this paper, we propose a time coherence guided end-effector estimation algorithm based on topological representation of RGB-D data. In order to deal with the estimation error caused by topology changes, we employed time coherence by integrating single frame estimation with previous information in a predict-update framework while avoiding strongly relying on it. Therefore, the proposed approach does not require any initialization or prior knowledge about the structure of the entity, which provides the ability to recover from estimation errors and the generality to estimate end-effectors for different entities. In our experiments, we proved that the proposed approach provides robustness for end-effector estimation, with its results outperforming the method in [13] in recall and average false positive rate while achieving similar results in terms of average distance error.

Acknowledgement: This work has been developed in the framework of the project TEC2013-43935-R, financed by the Spanish Ministerio de Economa y Competitividad and the European Regional Development Fund (ERDF).

References

1. Azizpour H., Laptev I.: Object detection using strongly-supervised deformable part models. In: Computer Vision–ECCV 2012, pages 836–849. Springer (2012)
2. Baak A., Müller M., Bharaj G., Seidel H., Theobalt C.: A data-driven approach for real-time full body pose reconstruction from a depth camera. In: Consumer Depth Cameras for Computer Vision, pages 71–98. Springer (2013)
3. Bergtholdt M., Kappes J., Schmidt S., Schnörr C.: A study of parts-based object class detection using complete graphs. International journal of computer vision, 87(1-2):93–117 (2010)
4. Brox T. and Malik J.: Large displacement optical flow: descriptor matching in variational motion estimation. Pattern Analysis and Machine Intelligence, IEEE Transactions on, 33(3):500–513 (2011)
5. Caselles V., Catté F., Coll T., Dibos F.: A geometric model for active contours in image processing. Numerische mathematik, 66(1):1–31 (1993)
6. Kuhn H.W.: The hungarian method for the assignment problem. Naval research logistics quarterly, 2(1-2):83–97 (1955)
7. Li J.L., Hao S., Lim Y.W., and Li F.F.: Object bank: An object-level image representation for high-level visual recognition. International journal of computer vision, 107(1):20–39 (2014)
8. Malladi R, Sethian J.A., Vemuri B.C.: Shape modeling with front propagation: A level set approach. Pattern Analysis and Machine Intelligence, IEEE Transactions on, 17(2):158–175 (1995)
9. Ofli F., Chaudhry R., Kurillo G., Vidal R., Bajcsy R.: Berkeley mhad: A comprehensive multimodal human action database. In Applications of Computer Vision (WACV), 2013 IEEE Workshop on, pages 53–60. IEEE (2013)
10. Plagemann C., Ganapathi V., Koller D., Thrun S.: Real-time identification and localization of body parts from depth images. In Robotics and Automation (ICRA), 2010 IEEE International Conference on, pages 3108–3113. IEEE (2010)
11. Schwarz L.A., Mkhitaryan A., Mateus D., Navab N.: Human skeleton tracking from depth data using geodesic distances and optical flow. Image and Vision Computing, 30(3):217–226 (2012)
12. Sethian J.A.: Level set methods and fast marching methods: evolving interfaces in computational geometry, fluid mechanics, computer vision, and materials science, volume 3. Cambridge university press (1999)
13. Suau X., Hidalgo J.R., Casas J.R.: Detecting end-effectors on 2.5 d data using geometric deformable models: Application to human pose estimation. Computer Vision and Image Understanding, 117(3):281–288 (2013)
14. Sande K., Uijlings J., Gevers T., Smeulders A.: Segmentation as selective search for object recognition. In: Computer Vision (ICCV), 2011 IEEE International Conference on, pages 1879–1886. IEEE (2011)
15. Wang X.Y., Yang M., Zhu S.H., Lin Y.Q.: Regionlets for generic object detection. In Computer Vision (ICCV), 2013 IEEE International Conference on, pages 17–24. IEEE (2013)