Clustering-based refinement for 3D human body parts segmentation

Leonardo Barcellona[1], Matteo Terreran[1], Daniele Evangelista[1], and Stefano Ghidoni[1]

[1]Department of Information Engineering DEI, University of Padova, Padova, Italy
{barcellona, terreran, evangelista, ghidoni}@dei.unipd.it

Abstract. A common approach to address human body parts segmentation on 3D data involves the use of a 2D segmentation network and 3D projection. Following this approach, several errors could be introduced in the final 3D segmentation output, such as segmentation errors and reprojection errors. Such errors are even more significant when considering very small body parts such as hands. In this paper, we propose a new algorithm that aims to reduce such errors and improve 3D segmentation of human body parts. The algorithm detects noise points and wrong clusters using DBSCAN algorithm, and changes the labels of the points exploiting the shape and position of the clusters. We evaluated the proposed algorithm on the 3DPeople synthetic dataset and on a real dataset, highlighting how it can greatly improve the 3D segmentation of small body parts like hands. With our algorithm we achieved an improvement up to 4.68% of IoU on the synthetic dataset and up to 2.30% of IoU in the real scenario.

Keywords: human parsing, 3D semantic segmentation, 3D clustering

1 Introduction

Human perception is a key element in a lot of new challenges, for example, in human-robot collaboration, where any physical barrier between the human operator and the robot is removed. In this scenario, the robot needs to precisely detect the human body to avoid dangerous situations (e.g., collisions), and improve the overall task planning [1]. Another example where human perception is fundamental is motion prediction, where the objective is to estimate future movement trajectories of human body [2].

Such applications need high precision and reliability when they estimate people and their pose. Common methods to estimate people in a scene are based on people detectors and skeletal tracking algorithms [3, 4], which provide just a schematic representation of the human body made of joints and links. Considering instead a volumetric approach (e.g., segmenting human body parts in the 3D space) could provide a more informative representation of the person combining both semantics and volume information. For example, authors in [5]

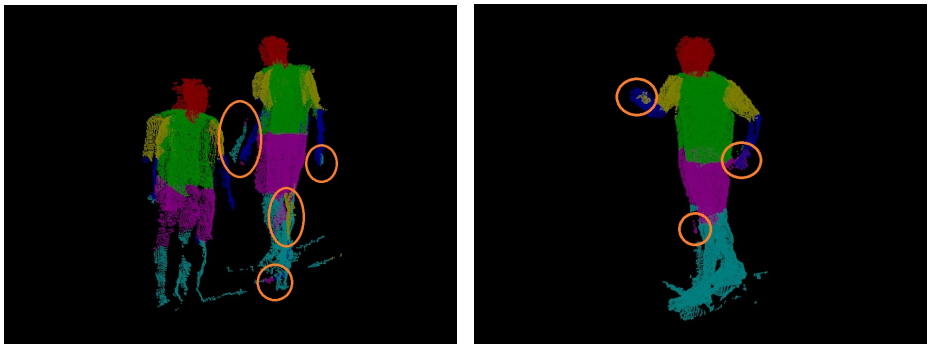
propose the use of semantic segmentation techniques to guarantee human safety during a human-robot collaborative task: by estimating which pixels in the image correspond to body parts (e.g., hands) it is possible to control the robot to take objects passed by the human while avoiding to collide with his/her hands.

Semantic segmentation of human body parts is also known as human parsing (HP) in the literature. State-of-the-art solutions for human parsing are mainly based on deep neural networks [6–8] trained on popular datasets such as Pascal-Person-Part [9] or LIP [10]. The majority of these methods work only with 2D images, due to the availability of many human parsing dataset with annotations for RGB images. Only recently, 3D data started to be directly exploited for training deep models that accurately detect human body parts [7, 11, 12]. However, the number of public 3D human parsing datasets of 3D is still limited, making it difficult to develop models that work directly on 3D data. A common strategy to tackle 3D human parsing is, hence, to reuse models trained on RGB images exploiting additional depth information to project the 2D body parts segmentation onto the 3D space. For example, we used such an approach in a previous work [13] where we developed a camera network system for 3D body parts segmentation based on SCHP network [6]. In [13] we focused on human-robot collaborative scenarios and we also proposed a manually labelled dataset acquired in a real scenario to evaluate the performance of the system.

When projecting the 2D segmentation results onto the 3D space two main types of errors should be taken into account: a 2D segmentation error of the human parsing model and a depth projection error; we named the latter Depth-Segmentation Association (DSA), which represents the wrong association between color and depth pixels. Even in calibrated cameras, the depth value of a pixel is not perfectly accurate (e.g., error in the sensor measurements), especially near the edges of small objects, so the color information may not be projected correctly. This problem is particularly significant in human body parts segmentation because, for example, the hands and fingers have a very small shape and the depth may be wrongly estimated. This may negatively affect the applicability of projecting the 2D segmentation results to 3D point clouds, especially for applications which require high accuracy and precision in segmentation.

The main effect of the DSA error is the creation of small isolated groups of 3D points, named clusters, with a different label compared to the neighboring ones. The number of isolated clusters increases in the presence of multiple people in the scene, because some wrongly projected points of a person may be attached to other people. Moreover, some clusters may also be created by segmentation errors that overestimate the dimension of the body parts. A visual examples of such errors is shown in Figure 1, where the circles highlight the small clusters created during the association between depth and segmentation mask.

In this paper, we propose a new algorithm that reduces both DSA and small segmentation errors. Such errors corresponds to clusters of 3D points in the point cloud with a wrong associated label. Our proposed algorithm aims to find such clusters by means of the DBSCAN algorithm [14], and to correct their labels exploiting the relationships between the body parts.



(a) (b)

Fig. 1: Two examples of projection errors. The orange circles highlight regions with wrong labels. The figure on the left (a) is extracted in a scene with two people and the label of the yellow points on the legs belongs to the second person but these points are wrongly associated with the depth. In the other figure (b) there is only one person, but there are still some error on the hands.

Experiments have been performed considering both synthetic data and a real scenario, highlighting a general improvement of the segmentation thanks to our algorithm. In particular, they show a marked benefit of the algorithm in improving the segmentation of smaller body parts (e.g., arms and hands), which are the parts most prone to errors described.

Another main contribution of the paper is the adaptation of the multi-view synthetic 3DPeople dataset [12] for the human parsing task. To our knowledge, there are no works that exploit such dataset to address human parsing on 3D data. Finally, it is presented a comparison of the state of the art human parsing networks for the 3D segmentation through projection using depth images.

The reminder of the paper is organized as follows. The next section is dedicated to describe the works related to human parsing, 3D projection and clustering algorithms. In the third section, the proposed algorithm is described in detail. The fourth section reports the experiments and the last section is reserved for the conclusions.

2 Related Works

2.1 Human parsing

Detecting body parts is essential. For example, it guarantees both safety measures and efficient interaction in human-robot collaboration. Some systems rely their human perception on a pose estimator [15], that only detects human keypoints around the joints. Other systems use semantic segmentation techniques to make a robot perceive the environment, labelling each pixel of an image ac-

cording to the class it represents. When the task addresses human body parts, it is called human parsing [10].

The set of body parts targeted by human parsing depends on the representation used as reference. The ATR dataset [16] considers clothes worn alongside body parts (e.g., trousers, shoes or arms) and contains mainly people from the front that are completely visible. The LIP dataset [10] also addresses the task distinguishing between body limbs and clothes, but contains images of people varying positions and actions. The extension of LIP containing instance information is called CHIP [10]. Finally, Pascal-Person-Part dataset [9] distinguishes between body limbs (e.g., head, torso, arms, legs) without considering the clothes. This last representation is strictly related to pose estimation [17] and may be useful in a lot of applications. In fact, the authors of [5] exploit the Pascal-Person-Part representation to pass objects from human hands to robot grippers.

State of the art human parsing solutions are based on deep learning. Several models achieve good performance in the Pascal-Person-Part test dataset. The standard metric to compare the networks is the mean intersection over union (IoU), a ratio that measures how the mask detected deviates from the ground truth mask. SCHP [6] is one of the first convolutional neural networks able to reach a high IoU in Pascal-Person-Part. It leverages the context information and the edges information to increase the performance of the segmentation. CDCL [7] is a network with performance similar to SCHP, but obtained in a completely different way. It merges the pose information to train the network in a synthetic dataset keeping good performance also in real datasets. Another network worth mentioning is Grapy [8], which inherited the idea of Graphonomy [18] and trains the network in the Pascal-Person-Part dataset boosting the performance using heterogeneous dataset such as the ATR dataset and CHIP dataset.

2.2 3D human parsing

Semantic segmentation is frequently addressed in 3D data processing. There are a lot of examples in literature that directly segment point clouds in 3D, but the majority do not divide people into body parts. Some of them, such as [19, 20], target object part segmentation. However, there are no point clouds of human bodies among the datasets they target.

The authors of [11] propose a multi-view dataset that also contains human point clouds. The labels of the points are divided among twenty classes focusing on the clothes worn instead of body parts.

Synthetic datasets are a possible solution to fill the scarcity of 3D human parsing datasets that target body parts. Moreover, their ground-truth does not present human errors, because are not annotated by hand. They retrieve the labels from body models rendered on a background image [21]. 3DPeople [12] is an example of a multi-view synthetic dataset, rendered from four points of view.

Directly projecting the segmentation mask extracted from the RGB image using the camera parameters is another solution to avoid the lack of 3D annotated data. This approach is exploited by [22]. In [13] we use this approach in a

human-robot collaboration scenario, exploiting the good performance of human parsing networks in 2D and re-proposing them in 3D.

2.3 Clustering algorithms and applications

When the point cloud labels are generated from the projection of the 2D data, the 3D points form groups that share the labels. These are called clusters. Clustering detection is an unsupervised learning task aiming at finding the best set of clusters given a set of data and a distance function. The distance function is used for computing how close two points are and the most common function is the euclidean distance.

In the literature, there are a lot of clustering algorithms. One of the most used is K-means [23] that tries to divide the points based on a given number of clusters and the mean position of the centers. Another important algorithm is DBSCAN [14] that divides the points into clusters based on the density. Clustering is one of the most investigated topics in computer vision and all these algorithms are currently applied [24–27]. In [24] the authors use DBSCAN as a superpixel extractor. DBSCAN is used to extract clusters with similar colors and positions. Then, the clusters are merged using a distance function to form the superpixels. In [25] the authors detect humans from lidar point clouds using an online learning technique based on clustering, while in [26] the authors divide human body parts using clustering. In [27] the human body is divided into limbs using k-means.

3 Clustering-based algorithm for 3D segmentation refinement

Human body parts segmentation is mainly addressed considering 2D images as input. A common strategy is to project such 2D segmentation masks by means of depth information. This procedure allows to reuse existing 2D human parsing models, but introduces errors (i.e., 2D segmentation and DSA errors) when the labels are wrongly associated with the 3D points. The consequence is the creation of small groups of points with incorrect labels. In this work, we aim to solve the aforementioned errors by proposing the procedure described in Algorithm 1. The algorithm takes as input a segmented point cloud, obtained by projecting in 3D the predicted segmentation masks of a 2D human parsing model, and a set of people instances in the point cloud. The final output is a point cloud with the same number of points and refined labels.

In our algorithm, people instances are described in terms of 3D bounding boxes, generated as in [13] using a people detection network and the 2D segmentation mask from each single camera. Such boxes are also used to remove outliers, namely points too far from the body person. For each person in the scene, a point cloud is obtained considering all the points inside the corresponding 3D bounding box. Then the algorithm process each person point cloud with a clustering step based on DBSCAN, which detects all the clusters of body parts

Algorithm 1: Clustering Correction

Input : start_point cloud, boxes
Output: corrected_point cloud

people_point clouds ← null

for *i=0 i < boxes.size* **do**

person ← `filter_using_box(`*start_point cloud, boxes[i]*`)`
clusters, noise ← `DBSCAN(`*person*`)`
clusters_features ← `features_extractor(`*clusters, person*`)`
graph ← `graph_extractor(`*clusters_features*`)`

cluster_features ← `merge_nodes(`*graph, cluster_features*`)`
cluster_features ← `remove_nodes(`*graph, cluster_features*`)`
corrected_person ← `correct_clusters(`*clusters, cluster_features*`)`
corrected_person ← `reassign_noise(`*corrected_person, noise*`)`

people_point clouds.`push_back(`*corrected_person*`)`

end

corrected_point cloud ← `fuse_poitclouds(`*start_point cloud, people_point clouds*`)`

and provides also groups of points marked as noise (i.e., not belonging to any of the detected clusters). For each detected cluster a set of features is extracted, considering also as a feature the temporary label assigned to the cluster itself. Moreover, a graph representing all the relations between clusters is computed. The temporary label is then updated if the cluster is contained in another one, or if it is not connected to the graph. After these, all the points previously classified as noise are updated according to the label of the nearest cluster. Finally, all the corrected person's point clouds are merged. In this last step, if the same point is wrongly associated with two or more persons and there are discordant labels, the original label is kept.

The main steps are described in detail in the following subsections. Subsection 3.1 is dedicated to DBSCAN and the procedure to detect the clusters. Subsection 3.2 explains the features and the creation of the graph to store the position of the clusters. Finally, subsection 3.3 highlights the correction performed.

3.1 Human body clusters detection

In the first step of the algorithm, we aim to group 3D points based on common features, in order to highlight all the points which do not actually belong to human limbs or have been assigned to a wrong labels. In general, the number of clusters is not known a priori and should be estimated from each person point cloud. For example, the hands are fused together when they are touching and divided when they are not. Furthermore, the DSA error increases the number of clusters. The distance metric chosen to distinguish the clusters is the Euclidean distance, which is increased when two points do not share a label so as to separate not only by point density, but also by the initially assigned label. From the

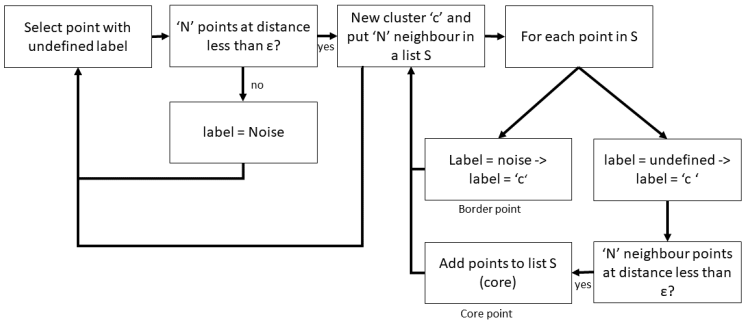


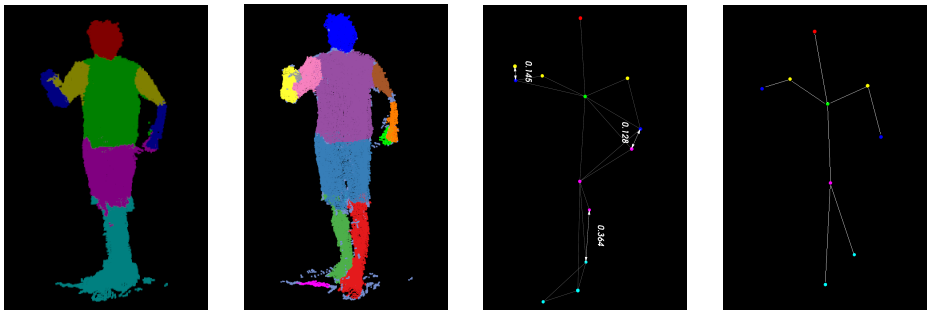
Fig. 2: A schematic overview of the DBSCAN algorithm [14]. The whole pipeline is iterated until there are unlabelled points.

description of the constraints, the chosen clustering method is DBSCAN [14], that does not need the number of clusters as input and is also capable of detecting noise points, that are points which do not belong to any clusters.

In particular, DBSCAN separates points based on density considering two main parameters: the distance ϵ used to consider two points adjacent, and the number of adjacent points N needed for a point to be considered as core points. These parameters could be considered as the scale of near points and the minimum density of a cluster. A schematic overview of the DBSCAN clustering algorithm is shown in Figure 2, highlighting the main steps performed. The algorithm starts from a random point. If such point has N or more points around with a distance $d \leq \epsilon$, it is then considered a *core point* of the cluster. Otherwise, if the random point is close to at least one core point, it is associated to the cluster and considered a border point. Finally, if a point is not a core point and is not close to any core points, it is considered as noise. The final output of the DBSCAN algorithm is a set of clusters and a set of "noise" points not associated to any of the cluster found. An example of the clusters found by DBSCAN is provided in Figure 3b, showing the result on a person point cloud. Note that in the input point cloud shown in Figure 3a, some points of the hands have been assigned to a wrong label (either due to a segmentation or a projection error). All these mislabeled points are detected by the DBSCAN algorithm and assigned to different clusters, highlighting the set of points which needs a label refinement in the next steps of the algorithm.

3.2 Feature extraction and graph construction

After the clustering step, we investigate a suitable set of features in order to understand which cluster is correctly labelled and which is not. We extract from



(a) Point Cloud (b) Clusters (c) Graph (d) Tree

Fig. 3: The cluster detected by DBSCAN and the graph constructed from them. Figure (a) is the starting point cloud. In figure (b) every color correspond to a cluster, while turquoise points the noise. Figure (c) shows the graph constructed from the clusters and figure (d) the best tree matching the person.

each cluster three main features: the actual label (shared by the whole cluster), the cluster centroid and the cluster radius. The radius of the cluster is computed as the maximum distance between the centroid and the points belonging to the cluster. The centroid and the radius do not change value during the execution of the algorithm, while the label is updated as exploited in Section 3.3.

To keep information about the relation between clusters, we use a graph. The graph is a set of nodes and edges. In our case, the nodes are the clusters, while the edges keep information about their relative position. A graph representing an human body has a particular structure: a "torso" node connected with a "head" node, two "upper arm" nodes and two "upper legs" nodes. Each upper arm is connected to a lower arm and each upper leg to a lower leg. This particular graph is a tree and we construct it starting from the biggest cluster of label torso as root and, then, connecting the arms, the hands, the legs and the head.

In our graph structure there are three types of edges distinguished by these three cases: a cluster is inside another, a cluster intersects another or a cluster belongs to the best tree that represents the human body. The last edges is computed using the torso as root as previously reported while the radius and the centroid are used to check the first two type of edges. Figure 3 shows a visualization of the constructed graph and the best tree representing the person.

3.3 Segmentation label refinement

The final step is the refinement of the segmented point cloud by correcting the labels of the noise points. The nodes of the body tree are unchangeable clusters because they represent the best body structure. The nodes of the graph that are contained in another cluster are updated, inheriting the label of the biggest cluster containing them with a different label. This is the "merge_nodes" step (M) of the pseudocode reported in Algorithm 1.

The clusters not connected to the best matching tree are labelled as background during the "remove_nodes" step (R). If a valid tree is not found, this phase, is not executed. Finally, the point cloud is refined assigning the new labels of the clusters to the points. The final step is re-labelling the points that were classified as noise. They inherit the label of the nearest point of the corrected point cloud. The search is performed using the kdtree search.

4 Experiments

The following section presents a set of experiments to evaluate the effectiveness of the proposed algorithm. In the first subsection, our segmentation refinement algorithm is tested using the synthetic multi-view dataset [12] presented in the related works, while the second subsection is focused on the evaluation of the algorithm on data acquired in a real scenario (i.e., the annotated segmentation masks from [13]).

In all the experiments, performances are evaluated in terms of intersection over union (IoU), which is computed as the ratio between the number of points with a correct label and the total number of points in the point cloud. We also compute the mean IoU of the body parts (BIoU) because the focus of the work is on correcting them. Other two reported metrics are the precision and the F1 score, which represent the number of true points of a class divided by the number of points of that class and the harmonic mean between precision and recall.

We analyze the improvement of each step of the algorithm. We indicate with "M" the merge phase, where the clusters are fused in one bigger group. We name "N" the label reassignment of the noise points. With "M - N", we refer to the algorithm with both the previous steps. Finally, "M - R - N" refers to the whole algorithm, whit an outliers-removal, as described in Algorithm 1.

4.1 Performance on synthetic data

Synthetic datasets offer a large amount of labelled data, which can be acquired and annotated in an easier and quicker way than real datasets. For example, the 3DPeople synthetic dataset [12] contains 2.5 Million frames of 80 subjects performing different actions, 40 female and 40 male models. The dataset contains high variability, with a a large range of distinct body shapes, skin tones and clothing outfits, and provides RGB-D data under different viewpoints. Regarding annotations, the dataset offers ground truth 2D segmentation masks for clothes and human body, the latter divided into fourteen classes distinguishing among the rest, left and right limbs.

For our experiments, we grouped the body parts annotation of the 3DPeople dataset in a set of 6 classes, namely *Head*, *Torso*, *Upper arms*, *Lower arms*, *Upper legs* and *Lower legs*. We then computed point clouds from the RGB-D data provided in the 3DPeople dataset for each camera; for each pair of RGB-D data the point cloud with body parts segmentation is obtained by projecting the labels predicted by the SCHP [6] network, while ground truth point cloud

(a) View 1 (b) View 2 (c) View 3 (d) Point cloud

Fig. 4: Example of the dataset created by [12]. Images (a)-(c) show the three of the four views and (d) shows the reconstructed point cloud generated by SCHP predictions [6].

are obtained by projecting the dataset annotations. Figure 4 shows an example of a model rendered on a background image from multiple points of view and the point cloud generated by using SCHP predictions. In our experiments we used the last ten male and female models to test our segmentation refinement algorithm, following the suggestion of 3DPeople dataset's authors but increasing the number of models used as test set. In particular, 1440 frames were extracted with different models, clothes, light, background and pose.

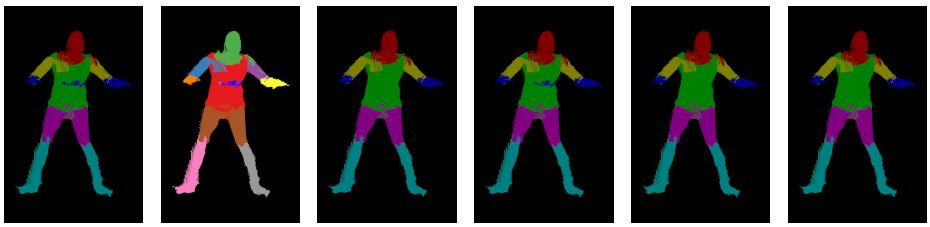
The human parsing network used for evaluating the refinement algorithm is SCHP [6]. However, one problem is the difficulty to reuse the solutions obtained in real datasets in synthetic ones or vice-versa. For example, a human parsing network trained on Pascal-Person-Part [9] may not be suitable for segmenting the 3DPeople synthetic dataset and vice-versa. For this reason, the available pretained model is not suited and a new training of the SCHP network is required before testing the proposed algorithm on the synthetic dataset. The 30 models of males and females not used for testing the algorithm were used for this purpose. 1716 frames were extracted from these to match the number of images used in Pascal-Person-Part [9] and the labels were reduced to match the set of six classes considered in the Pascal-Person-Part dataset and listed before.

The point clouds to be refined to test the proposed algorithm were computed using 1440 test frames of 3DPeople RGB-D data and the segmentation masks predicted by SCHP after the new training. Table 1 shows the results of the proposed algorithm applied to these point clouds. The parameters of DBSCAN, used in the first clustering phase of the algorithm, are 1.0 as minimum distance and 40 as the number of neighbours points. The columns show the intersection over union of the body classes, the average IoU, the average IoU of the body classes, the average precision and the F1 score.

The results reported in Table 1 are divided in rows to highlight the contribution of each step of the proposed algorithm. The letter "M" denotes the merge function that associates the labels to the cluster wrapping the initial cluster entirely. The letter "N" denotes the noise reassignment contribution and "M - N" represents the contribution of both merge and noise reassignment. The final row, denoted as "M-R-N", shows the results of the whole algorithm. For each main row, we also reported the performance increase or decrease (in green or

Table 1: The performance on the 3DPeople synthetic dataset. First columns show IoU per class. Last columns show the global performance in terms of mean IoU, mean IoU of body classes, average precision and F1 score. The second line of each row shows the change between the reference and the output of the algorithm as reported in section 3. M is the merge phase of the algorithm, N is the noise relabelling phase, M-N is the merge phase with the noise relabelling phase and M-R-N is the whole algorithm.

Type of scene	Head	Torso	Upper arms	Lower arms	Upper legs	Lower legs	Background	mIoU	BIoU	AP	F1
Reference	73.68	76.84	66.26	68.96	69.98	75.86	99.05	75.8	71.93	91.85	86.00
M	73.59	76.96	66.57	72.97	70.22	77.1	99.09	76.51	72.90	93.09	86.49
	-0.09	0.12	0.31	4.01	0.24	1.24	0.04	0.71	0.97	1.24	0.49
N	74.21	77.07	66.94	71.78	70.30	76.97	99.09	76.62	72.88	93.00	86.56
	0.53	0.23	0.68	2.82	0.32	1.11	0.04	0.82	0.95	1.15	0.56
M - N	73.66	77.06	66.62	72.27	70.27	77.11	99.09	76.58	72.83	93.03	86.54
	-0.02	0.22	0.36	3.31	0.29	1.25	0.04	0.78	0.90	1.18	0.54
M - R - N	73.75	77.13	66.76	73.64	70.41	77.83	99.12	76.95	73.25	93.6	86.79
	0.07	0.29	0.50	4.68	0.43	1.97	0.07	1.15	1.32	1.75	0.79



(a) Start (b) Clusters (c) M (d) N (e) M-N (f) M-R-N

Fig. 5: From the left to the right: the starting cloud, the clusters, the results of the merge step, the results of the noise relabeling, the results of the merge and noise relabeling, the results of the whole algorithm. The point clouds are extracted from 3DPeople [12].

red respectively) compared to the baseline, shown in the first row. Some output examples of the various steps of the algorithm are also depicted in Figure 5, together with the input point cloud and the clusters detected by DBSCAN.

From the results obtained on the synthetic dataset, it can be shown that the proposed algorithm leads to a visible improvement on some classes, especially the "Lower arms" class. Indeed, this is the class that is most frequently misclassified by the SCHP network (i.e., segmentation error). Moreover, since the lower arms class includes also the hands, for this class we have in general also a high DSA error, due to the difficulty to correctly estimate the depth of the points around the hands. The proposed algorithm is therefore able to reduce the effect of both these errors, allowing to improve the segmentation of difficult classes such as hands. The background class has a good IoU affecting also the mean, but we are more interested in the body classes because the algorithm mainly

(a) View 1 (b) View 2 (c) View 3 (d) Point cloud

Fig. 6: Example of the dataset created by [13]. Images (a)-(c) show the three view and (d) shows the ground truth point cloud.

focuses on improving them. The mean Body IoU (BIoU) is the average IoU of the body classes and may better highlight the improvement of the algorithm, with an increase of up to 1.32% of IoU. Moreover, the lower arms class reaches an improvement of 4.68%. From the table, each part of the algorithm contributes an increase in the results.

4.2 Performance on real data

The proposed algorithm proves to be effective in the experiments on synthetic data, showing an increase in performance on challenging classes. On the other hand, we also tested the proposed method on a more challenging dataset composed by real data. We used the dataset from [13] which includes manually annotated RGB-D data of a real setup. Such dataset shows up to three people moving in a scene with obstacles or a robot occluding the views. The scenes are recorded from three points of view using Microsoft Kinect One RGB-D cameras. Figure 6 shows a scene from such a dataset, where a person is standing near a robotic arm, and the corresponding ground truth point cloud.

Similarly to the previous experiments, we extracted the point cloud projecting the segmented labels from the human parsing networks. In addition to the SCHP architecture [6], we considered also two other human parsing networks, namely CDCL [7] and Grapy [8]. We did not use these networks in the previous section because there are no open source implementations suitable for training such models on custom datasets. For all the three networks, we used the pretrained weights provided by the authors, obtained after training on the Pascal-Person-Part [9] dataset. DBSCAN parameters were set to 0.05 as the maximum distance between near points and 40 as the number of neighbour points of a core point. The change is due to the different distances of the points generated by the Kinect sensors with respect to the case of synthetic depth.

Some output examples of each step in the refinement algorithm are depicted in Figure 7, showing also the input point cloud, the clusters detected by DBSCAN. Note as the noise relabeling step helps correcting some labels but the hands are not affected by it, while the merge step is able to correct it. A quantitative evaluation of the algorithm performance is reported in Table 2. For all the networks considered, our segmentation refinement approach leads to an improvement of the BIoU metric, but there are some differences in the results. In

Table 2: Performance on the real dataset. First columns show IoU per class. Last columns show the global performance in terms of mean IoU, mean IoU of body classes, average precision and F1 score. The second line of each row show the change between the reference and the output of the algorithm as reported in section 3. M is the merge phase of the algorithm, N is the noise relabelling phase, M-N is the merge phase with the noise relabelling phase and M-R-N is the whole algorithm.

SCHP [6]

Type of scene	Head	Torso	Upper arms	Lower arms	Upper legs	Lower legs	Background	mIoU	BIoU	AP	F1
Reference	80.47	76.69	57.19	50.52	71.22	59.07	99.15	70.61	65.86	81.92	82.07
M	79.77	76.56	57.48	50.75	71.91	61.32	99.09	70.98	66.30	84.74	82.35
	-0.70	-0.13	0.29	0.23	0.69	2.25	-0.06	0.37	0.44	2.82	0.28
N	80.63	77.07	57.81	51.21	72.66	60.3	99.2	71.3	66.61	82.86	82.56
	0.16	0.38	0.62	0.69	1.44	1.23	0.05	0.69	0.75	0.94	0.49
M - N	81.11	77.18	57.59	52.03	72.91	60.92	99.2	71.56	66.96	83.14	82.76
	0.64	0.49	0.40	1.51	1.69	1.85	0.05	0.95	1.10	1.22	0.69
M - R - N	81.11	77.24	57.61	51.97	72.95	61.37	99.21	71.64	67.04	83.38	82.81
	0.64	0.55	0.42	1.45	1.73	2.30	0.06	1.03	1.18	1.46	0.74

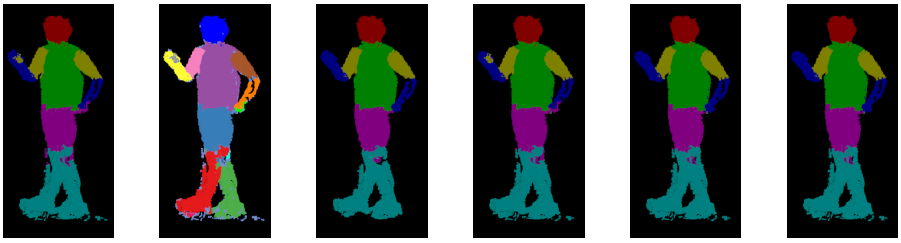
CDCL [7]

Type of scene	Head	Torso	Upper arms	Lower arms	Upper legs	Lower legs	Background	mIoU	BIoU	AP	F1
Reference	86.02	81.56	71.83	70.46	81.05	72.89	99.70	80.46	77.30	88.41	88.96
M	85.47	83.41	71.94	71.98	79.87	71.73	99.32	80.53	77.40	91.09	89.00
	-0.55	1.85	0.11	1.52	-1.18	-1.16	-0.38	0.07	0.10	2.68	0.04
N	86.36	84.43	72.40	72.08	81.58	73.54	99.47	81.41	78.40	89.61	89.53
	0.34	2.87	0.57	1.62	0.53	0.65	-0.23	0.95	1.10	1.20	0.57
M - N	86.33	84.50	72.38	72.43	81.63	73.54	99.48	81.47	78.47	89.71	89.56
	0.31	2.94	0.55	1.97	0.58	0.65	-0.22	1.01	1.17	1.30	0.60
M - R - N	86.33	84.52	72.38	72.51	81.63	73.38	99.48	81.46	78.46	89.76	89.56
	0.31	2.96	0.55	2.05	0.58	0.49	-0.22	1.00	1.16	1.35	0.60

Grapy [8]

Type of scene	Head	Torso	Upper arms	Lower arms	Upper legs	Lower legs	Background	mIoU	BIoU	AP	F1
Reference	86.13	84.84	67.82	64.79	79.13	71.05	99.48	79.04	75.63	87.46	87.87
M	84.92	84.09	67.18	64.2	79.09	70.63	99.37	78.45	75.02	89.31	87.52
	-1.21	-0.75	-0.64	-0.59	-0.04	-0.42	-0.11	-0.59	-0.61	1.85	-0.35
N	86.16	84.73	67.71	64.66	80.3	71.69	99.36	79.25	75.88	87.97	88
	0.03	-0.11	-0.11	-0.13	1.17	0.64	-0.12	0.21	0.25	0.51	0.13
M - N	86.24	84.73	67.69	64.77	80.44	71.77	99.5	79.31	75.94	88.00	88.03
	0.11	-0.11	-0.13	-0.02	1.31	0.72	0.02	0.27	0.31	0.54	0.16
M - R - N	86.24	84.81	67.69	64.82	80.41	71.22	99.5	79.25	75.87	88.04	87.99
	0.11	-0.03	-0.13	0.03	1.28	0.17	0.02	0.21	0.24	0.58	0.12

particular, the improvement is highly marked for the first two human parsing networks using all the steps of the proposed algorithm, while in case of the Grapy network the results are mostly unchanged. In particular, the merge phase for this last network does not seem effective. This means that the network is less affected by segmentation errors that go outside the edges of the objects. Secondly, the most affected body part changes based on the network used, meaning that they tend to overestimate the segmentation of some specific classes more frequently. SCHP finds more problems of over-segmentation of lower arms and legs, as found on the synthetic data, while CDCL has more segmentation problems with lower arms and torso classes. Grapy only tends to over-segment the upper legs. In this



(a) Start (b) Clusters (c) M (d) N (e) M-N (f) M-R-N

Fig. 7: From the left to the right: the input point cloud, the clusters, the results of the merge step, the results of the noise relabeling, the results of the merge and noise relabeling, the results of the whole algorithm. The input point cloud is taken from a real scenario [13].

set of experiments it is worth noticing that the real dataset is more challenging than the synthetic data, because people are recorded in occluded scenarios that may interfere with the perception, dividing people's shape and decreasing the effect of the merge procedure.

5 Conclusions

When addressing human body parts segmentation on 3D data, a common approach to deal with the lack of large 3D human parsing dataset consists in the use of 2D human parsing networks: given a input image, such networks predict a segmentation mask that is then projected in 3D by using the corresponding depth information. The resulting 3D segmentation output is however affected by several errors, namely a segmentation error and a projection error due to inaccuracies of the depth information. In this paper, we proposed a new algorithm to tackle such errors and improve the final 3D segmentation results. Our algorithm is based on a clustering technique to identify and localize 3D points which have been assigned to a wrong label or that represent noise. All the clusters are then used to define a graph structure, which allows to refine the overall segmentation based on relationships between the clusters. The proposed algorithm has been evaluated on both synthetic and real data, showing the effectiveness of our approach to refine 3D body parts segmentation outputs. In particular, the algorithm shows major improvements when considering small body parts such as lower arms and hands, which are often misclassified by 2D human parsing networks. Especially the hands are also difficult to be projected in 3D, due to the inaccuracies of the depth information around the borders of small body parts (e.g., fingers). As demonstrated experimentally, the proposed algorithm is able to reduce the effect of both segmentation and projection errors, allowing to improve the 3D segmentation of body parts, even for complex shapes like hands. This is an interesting result, especially for human-robot collaboration applications, where our segmentation refinement approach can be very useful to the robot to

get a more accurate representation of the people in the workcell, enabling a close and direct collaboration with the human. As a future research direction we will further investigate how to improve 3D body parts segmentation accuracy when several points of view are available, combining all the segmentation information by means of Bayesian fusion techniques. Moreover, we will focus on the integration of the segmentation refinement algorithm in a real robotic workcell to test our algorithm during a real human-robot collaboration task.

Acknowledgments

This research has received funding from the European Union's Horizon 2020 research and innovation program under grant agreement No. 101006732.

References

1. Merckaert, K., Convens, B., Wu, C.j., Roncone, A., Nicotra, M.M., Vanderborght, B.: Real-time motion control of robotic manipulators for safe human–robot coexistence. Robotics and Computer-Integrated Manufacturing **73** (2022) 102223
2. Zanchettin, A.M., Casalino, A., Piroddi, L., Rocco, P.: Prediction of human activity patterns for human–robot collaborative assembly tasks. IEEE Transactions on Industrial Informatics **15**(7) (2018) 3934–3942
3. Cao, Z., Hidalgo Martinez, G., Simon, T., Wei, S., Sheikh, Y.A.: Openpose: Realtime multi-person 2d pose estimation using part affinity fields. IEEE Transactions on Pattern Analysis and Machine Intelligence (2019)
4. Carraro, M., Munaro, M., Burke, J., Menegatti, E.: Real-time marker-less multiperson 3d pose estimation in rgb-depth camera networks. (10 2017)
5. Rosenberger, P., Cosgun, A., Newbury, R., Kwan, J., Ortenzi, V., Corke, P., Grafinger, M.: Object-independent human-to-robot handovers using real time robotic vision. IEEE Robotics and Automation Letters **6**(1) (2020) 17–23
6. Li, P., Xu, Y., Wei, Y., Yang, Y.: Self-correction for human parsing. IEEE Transactions on Pattern Analysis and Machine Intelligence (2020)
7. Lin, K., Wang, L., Luo, K., Chen, Y., Liu, Z., Sun, M.T.: Cross-domain complementary learning using pose for multi-person part segmentation. IEEE Transactions on Circuits and Systems for Video Technology **31**(3) (2020) 1066–1078
8. He, H., Zhang, J., Zhang, Q., Tao, D.: Grapy-ml: Graph pyramid mutual learning for cross-dataset human parsing. In: Proceedings of the AAAI Conference on Artificial Intelligence. Volume 34. (2020) 10949–10956
9. Chen, X., Mottaghi, R., Liu, X., Fidler, S., Urtasun, R., Yuille, A.: Detect what you can: Detecting and representing objects using holistic models and body parts. In: Proceedings of the IEEE conference on computer vision and pattern recognition. (2014) 1971–1978
10. Gong, K., Liang, X., Zhang, D., Shen, X., Lin, L.: Look into person: Self-supervised structure-sensitive learning and a new benchmark for human parsing. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2017)
11. Yu, Z., Yoon, J.S., Lee, I.K., Venkatesh, P., Park, J., Yu, J., Park, H.S.: Humbi: A large multiview dataset of human body expressions. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. (2020) 2990–3000

12. Pumarola, A., Sanchez, J., Choi, G., Sanfeliu, A., Moreno-Noguer, F.: 3DPeople: Modeling the Geometry of Dressed Humans. In: International Conference in Computer Vision (ICCV). (2019)
13. Terreran, M., Barcellona, L., Evangelista, D., Ghidoni, S.: Multi-view human parsing for human-robot collaboration. In: 2021 20th International Conference on Advanced Robotics (ICAR), IEEE (2021) 905–912
14. Schubert, E., Sander, J., Ester, M., Kriegel, H.P., Xu, X.: Dbscan revisited, revisited: why and how you should (still) use dbscan. ACM Transactions on Database Systems (TODS) **42**(3) (2017) 1–21
15. Wang, Y., Ye, X., Yang, Y., Zhang, W.: Collision-free trajectory planning in human-robot interaction through hand movement prediction from vision. In: 2017 IEEE-RAS 17th International Conference on Humanoid Robotics (Humanoids), IEEE (2017) 305–310
16. Liang, X., Liu, S., Shen, X., Yang, J., Liu, L., Dong, J., Lin, L., Yan, S.: Deep human parsing with active template regression. IEEE transactions on pattern analysis and machine intelligence **37**(12) (2015) 2402–2414
17. Xia, F., Wang, P., Chen, X., Yuille, A.L.: Joint multi-person pose estimation and semantic part segmentation. In: Proceedings of the IEEE conference on computer vision and pattern recognition. (2017) 6769–6778
18. Gong, K., Gao, Y., Liang, X., Shen, X., Wang, M., Lin, L.: Graphonomy: Universal human parsing via graph transfer learning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. (2019)
19. Qi, C.R., Su, H., Mo, K., Guibas, L.J.: Pointnet: Deep learning on point sets for 3d classification and segmentation. In: Proceedings of the IEEE conference on computer vision and pattern recognition. (2017) 652–660
20. Qi, C.R., Yi, L., Su, H., Guibas, L.J.: Pointnet++: Deep hierarchical feature learning on point sets in a metric space. arXiv preprint arXiv:1706.02413 (2017)
21. Varol, G., Romero, J., Martin, X., Mahmood, N., Black, M.J., Laptev, I., Schmid, C.: Learning from synthetic humans. In: Proceedings of the IEEE conference on computer vision and pattern recognition. (2017) 109–117
22. Navaneet, K., Mandikal, P., Agarwal, M., Babu, R.V.: Capnet: Continuous approximation projection for 3d point cloud reconstruction using 2d supervision. In: Proceedings of the AAAI Conference on Artificial Intelligence. Volume 33. (2019) 8819–8826
23. Hartigan, J.A., Wong, M.A.: Algorithm as 136: A k-means clustering algorithm. Journal of the royal statistical society. series c (applied statistics) **28**(1) (1979) 100–108
24. Shen, J., Hao, X., Liang, Z., Liu, Y., Wang, W., Shao, L.: Real-time superpixel segmentation by dbscan clustering algorithm. IEEE transactions on image processing **25**(12) (2016) 5933–5942
25. Yan, Z., Duckett, T., Bellotto, N.: Online learning for human classification in 3d lidar-based tracking. In: 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), IEEE (2017) 864–871
26. Buys, K., Cagniart, C., Baksheev, A., De Laet, T., De Schutter, J., Pantofaru, C.: An adaptable system for rgb-d based human body detection and pose estimation. Journal of visual communication and image representation **25**(1) (2014) 39–52
27. Haggag, H., Hossny, M., Haggag, S., Nahavandi, S., Creighton, D.: Efficacy comparison of clustering systems for limb detection. In: 2014 9th International Conference on System of Systems Engineering (SOSE), IEEE (2014) 148–153