A new 3D descriptor for human classification: application for human detection in a multi-kinect system

Kyis Essmaeel, Cyrille Migniot, Albert Dipanda, Luigi Gallo, Ernesto Damiani, Giuseppe de Pietro

Noname manuscript No.
(will be inserted by the editor)

A New 3D Descriptor For Human Classification: Application For Human Detection in a Multi-Kinect System

Kyis Essmaeel · Cyrille Migniot · Albert Dipanda · Luigi Gallo · Ernesto Damiani · Giuseppe De Pietro

Received: date / Accepted: date

Abstract In this paper we present a new 3D descriptor for human classification and a human detection method based on this descriptor. The proposed 3D descriptor allows for the classification of an object represented by a point cloud, as human or non-human. It is derived from the well-known Histogram of Oriented Gradient by employing surface normals instead of gradients. The process consists in an appropriate subdivision of the object point cloud into blocks. These blocks provide the spatial distribution modeling of the surface normal orientation into the different parts of the object. This distribution modelling is expressed in the form of a histogram. In addition we have set up a multi-kinect acquisition system that provides us with Complete Point Clouds (CPC) (i.e. 360° view). Such CPCs enable a suitable processing, particularly in case of occlusions. Moreover they allow for the determination of the human frontal orientation. Based on the proposed 3D descriptor, we have developed a human detection method that is applied on CPCs. First, we evaluated the 3D descriptor over a set of CPC candidates by using the Support Vector Machine (SVM) classifier. The learning process was conducted with the original CPC database that we have built. The results are very promising. The descriptor can discriminate human from non-human candidates and provides the frontal direction of the humans with high precision. In addition we demonstrated that using the CPCs improves significantly the classification results in comparison with Single Point Clouds (i.e. points clouds acquired with only

K. Essmaeel, C. Migniot (corresponding author) and A. Dipanda
LE2I, FRE CNRS 2005, Univ. Bourgogne Franche-Comté, Dijon, France
Tel.: +333-80-39-36-92
Fax: +333-80-39-59-69
E-mail: cyrille.migniot@u-bourgogne.fr

L. Gallo and G. De Pietro
ICAR-CNR, Naples, Italy

E. Damiani
Department of Computer Technology, University of Milan, Milan, Italy

one kinect). Second, we compared our detection method with two detection methods, namely the HOG detector on RGB images and a 3D HOG-based detection method that is applied on RGB-depth data. The obtained results on different situations show that the proposed human detection method provides excellent performances that outperform the other two detection methods.

Keywords Human classification · 3D descriptor · multi-kinect

1 Introduction

Human detection is the process of localizing the presence of one or more persons in a specific location by manipulating the information acquired by different types of sensors. This process is a pivotal element for many applications like surveillance systems [1], health monitoring [2], autonomous sport-analysis [3, 4] and driving assistance [5, 6]. Over the past decades this subject has witnessed huge advances and it continues to evolve especially with the introduction of new sensing technologies. Human detection is a challenging task with different issues to tackle. Pose, color and texture significantly vary from one person to another. Besides, the complexity of the working environment represents a sample of the challenges to overcome. Some approaches for human detection rely on special sensors attached to the subjects to determine their location that are called invasive sensors [7–9]. The use of such sensors is limited to specific applications like in animation films, but in fact most of common applications require non-invasive sensors. Color cameras have been for many years the primary non-invasive sensors employed in human detection applications [10]. However, the recent advances in depth sensing technologies added the depth sensors as another reliable source of information that are used even for high level tasks like medical applications [11]. In fact, the introduction of affordable and reliable depth sensors like the kinect from Microsoft has dramatically increased the interest in this technology and has lead to a huge number of applications employing such sensors [12–14]. Indeed, human detection was one of the first domains to leverage this new technology. However, in most of the applications depth information is only used to reduce the computation cost, while the descriptiveness of the 3D shape of the human envelop is not really exploited.

Human detection is a vast domain with different approaches and techniques. Among these approaches, the descriptor/classifier framework employs a descriptor to extract special features and characteristics from the acquired data in order to train a classifier. The classifier should be able to separate between two classes: human and non-human object. In this paper we propose a new 3D descriptor for human classification in standing/walking position. The descriptor operates on 3D point clouds and exploits exclusively the human 3D features without using color information. The proposed 3D descriptor can be considered as a generalization of the HOG descriptor [15]. The calculation of the descriptor starts by dividing the 3D cloud into 3D blocks. The 3D descriptor is then obtained by computing the histogram of orientations of the

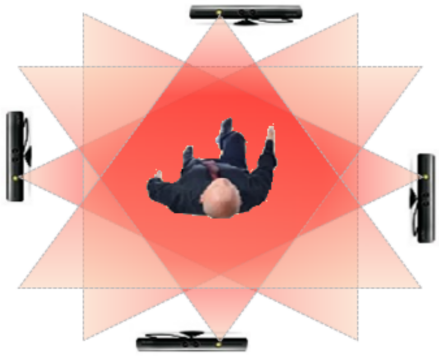

Fig. 1 An illustration of the multi-kinect platform: four kinects allowing for a 360° view.

normals on the points in each block. Moreover, the proposed descriptor provides additional information about the frontal orientation of the human. Such information is important for numerous applications namely tracking initialization, human-machine interaction and behaviour analysis.
Using this descriptor, we will present a human detection method. Our method is able to detect the locations and frontal orientations of a group of people in upright position. We use a multi-kinect system installed in an indoor location. The kinects are arranged to capture the entire scene as illustrated in Fig. 1, which in turns provides a Complete Point Cloud (CPC) of the scene (i.e 360° view).

The paper is organized as follows. Section 2 proposes a state-of-the-art of human detection methods. Two classes of methods are presented namely descriptor-based methods and body part matching methods. Section 3 presents the multi-kinect platform we used for data acquisition. It allows to obtain Complete Point Clouds (CPC) that provide a 360° view of the analyzed scenes and objetcts. Section 4 introduces the proposed human descriptor. It is calculated from 3D point clouds. Section 5 describes the classification process that allows to detect humans in a scene. The learning process required to build a new 3D CPC database of human and non-human subjects. Section 6 details the human detection procedure. Section 7 gives the experimental results to assess our descriptor and validate the effectiveness of the proposed human detection method. Finally, section 8 draws the conclusions.

2 Related works

In this section we will review the main approaches for human detection with a focus on methods that use depth data. There are two categories of methods for human detection: descriptor/classifier and matching templates.

In the first category, descriptors are computed over the acquired data to transfer it into a more descriptive space. In addition, a classifier is built from a database of positive and negative examples. Using this classifier, new candidates are classified as human or non-human objects as illustrated in Fig. 2. HOG (Histogram of Oriented Gradients) [15] is considered as one of the most successful descriptor for 2D image human detection. The descriptor is computed by first dividing the image into small spatial regions (blocks). For each block a local 1-D histogram of gradient directions or edge orientations over the pixels of the cell is computed. These histograms are combined and normalized to finally form the descriptor. In [16], HOG is applied on the 2D image space projection of clusters extracted from a point cloud. The HOD (Histogram of Oriented Depths) [17] is a well-known adaptation of the HOG which is applied on depth images, where the gradients are computed using the depth values instead of color information. HOD locally encodes the direction of depth changes and relies on a depth-informed scale-space search. The Combo-HOD is also proposed by combining depth and RGB data using a probabilistic model to detect people from RGB-D data [17]. In [13] they used the HOD with a graph-based segmentation algorithm to effectively process images captured by a moving camera. [18] use statistical pattern recognition techniques (geometric features) for the classification of candidates. [19] classifies with geometric features in the context of the pedestrian tracking from a moving vehicle. The Relational Depth Similarity Features (RDSF) [20] calculates the degrees of similarity between all of the combinations of rectangular regions inside a detection window in a single depth image only. [21] introduces Kirsch mask and Local Binary Pattern on 2D depth image to extract a local ternary direction pattern feature descriptor. The previous approaches use the depth array as a 2D image to apply image-based methods like the HOG process. However, 3D data is not exploited in their first forms, which makes them difficult to apply in scenarios where multiple sources of information are combined to produce the 3D data like in a multi-sensor system. Depth data is also exploited as a pre-processing to segment blob in 3D space and isolate candidates. For [12] the segmented 3D points are projected on 2D plane to form a color image on which HOG are processed. [22] gather 3D candidates in tracklets before person detection by HOG.

The orientation of the normals is a relevant feature to describe 3D human shape from point clouds. [23] uses also the normals to describe 3D objects from hundreds of viewpoints to obtain synthetic depth maps. [24] separates clusters of 3D points in horizontal layers and learn a classifier for each layer. [25] uses a classifier for each horizontal layer with geometrical features boosting. Each layer detects a particular body part. The final detector is composed of a probabilistic combination of the different classifiers.

Others methods [26–28] use local surface normals to describe 3D shape. Spin Image [29], FPFH [30] and SHOT [31] are well known 3D object descriptors computed from surface normals. However these methods describe static objects with standart space subdivision. The person class is so varying that they

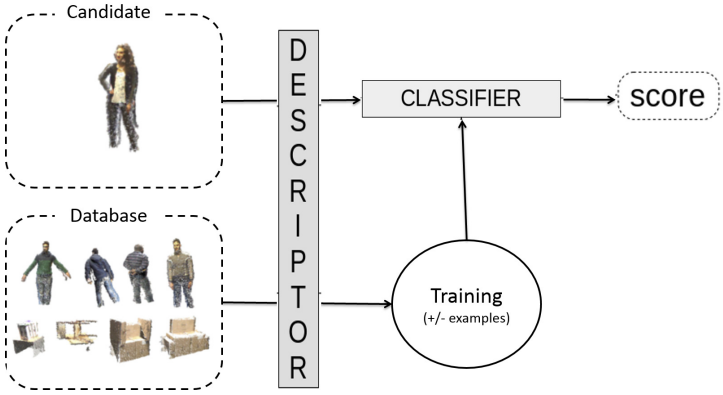

Fig. 2 Overview of the descriptor/classifier framework. The descriptor transforms the data into a more descriptive space. A classifier is built from the database of positives and negatives examples.The classifier computes for each candidate a classification score.

are not effective enough.
We finaly notice a recent trend to deep learning based approaches [32].

The second category of methods rely on matching one or many templates of certain body-parts in 2D data (images) or 3D data (point clouds). The Ω-shape of the head and shoulders of a human body are an example of descriptive templates [33,34]. To compare it to the data, [35] utilizes chamfer distance and Choi [36] exploits the Hamming distance. [37] applies chamfer distance on 2D templates as shaped-based detection on Region Of Interest (ROI). [38] minimizes the sum of squared distances between 3D points projected on 2D plane and binary templates of the head. [39] learns a 3D template from a dataset of heads. Part-based detection is sometimes used. For [40], interest points, which are based on identifying geodesic extrema on the surface mesh, coincide with salient points of the body, which can be classified as part of the body using local shape descriptors.
Numerous works exploit the temporal continuity of human displacement in video sequence: for example by defining tracklets to classify human body [22]. Scanning the ground to find candidates is a practical solution in an indoor or controlled environment. The ground can be estimated accurately from 3D using RANSAC [41], or manually by selecting three non-linear points from the ground in situations where the background is fixed.
In addition, it can be useful to obtain additional information about the detected person. The frontal orientation of the human is interesting information that can help initialize the tracking of the detected person. People orientation recognition is the topic of numerous methods. It can be inferred from the the relative poses of the human body parts [42]. Each orientation could be a class of a classification process Support Vector Machine (SVM) associated with HOG [43–45] or Aggregated Channel Features ACF [46] descriptor. Replacing the SVM by a decision tree improves the classification results [44,45].

Fig. 3 An example of a complete point cloud from two different points of view.

Lai [47] organizes 3D data into a semantically structured tree to estimate the view and pose of an object. In [48], a texture model is provided by projection onto a basis of Spherical Harmonics. The orientation is estimated by minimizing the difference between the texture model and the current texture estimate.

In this paper, we propose a new method for human detection following the descriptor/classifier approach. This approach provides better results when employing suitable data that capture the complexity of human poses and using a well constructed database to build an accurate classification model. The orientation of the normals is used to characterize the 3D human shape. In addition, we work with Complete Point Clouds (CPCs) acquired by a multi-kinect system that provides a complete view of the analysed subjects as shown in Fig. 3. Moreover, our descriptor allows for an estimation of the frontal direction without using multi-class classification.

3 Acquisition system

The goal is to reduce the percentage of occlusion by using multiple view-points which improves the process robustness. In [49], each detection window of the image of the first viewpoint is fused with the matching region on the image of the second viewpoint. The detection process on the fused region refines the detection on a single image.
In this section we will introduce our 3D acquisition system that is used to obtain a complete coverage of an indoor location. In order to achieve this complete coverage of the scene, a multi-kinect platform is constructed. The platform consists of a least three kinects arranged so that two consecutive kinects share an overlapping field of view as illustrated in Fig. 1. The multi-kinect system is then calibrated to obtain the intrinsic and extrinsic parameters for each kinect.

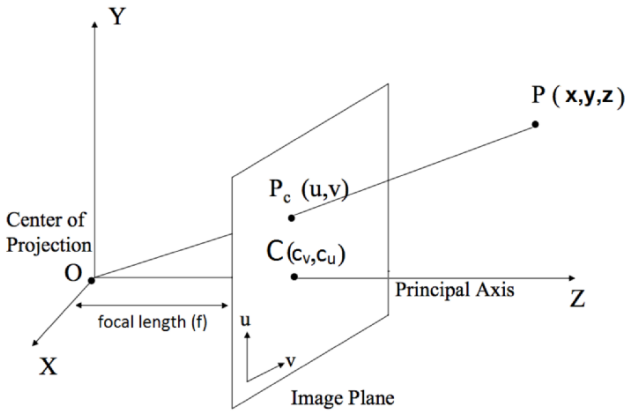

Fig. 4 Illustration of the pin-hole camera model.

The intrinsic parameters are necessary to transfer the 2D depth image into a 3D point cloud while the extrinsic parameters allows for the transformation of the point clouds from each kinect into a common coordinate system. Many efficient methods can compute these parameters [50–52]. The kinect-v1 (the version we use in the following) computes the depth image via a structure light imaging technology. For this the kinect uses an infrared light projector which projects a known pattern on the scene. The projected pattern is captured using an infrared camera. Finally, the disparity d is calculated internally by the kinect from this pattern and a pre-registered one at known distance. The depth is then calculated as the inverse of the disparity using the following equation [52]:

$$z = \frac{1}{c_v \times d + c_u} \tag{1}$$

where c_u and c_v are the image central points.
The depth camera follows a pin-hole camera model as illustrated in Fig. 4. From this, a 3D world point (x, y, z) is projected onto the 2D image point (u, v) according to the following equation:

$$\begin{pmatrix} u \\ v \\ 1 \end{pmatrix} = K \times \begin{pmatrix} \frac{x}{z} \\ \frac{y}{z} \\ 1 \end{pmatrix} \quad \text{with } K = \begin{pmatrix} f_u & 0 & c_u \\ 0 & f_v & c_v \\ 0 & 0 & 1 \end{pmatrix} \tag{2}$$

where K is the matrix of the intrinsic parameters of the camera, f_u and f_v are the focal length.
So we have [50]:

$$x = z\frac{u - c_u}{f_u} \quad \text{and} \quad y = z\frac{v - c_v}{f_v} \tag{3}$$

In a multi-kinect system the single point clouds (SPCs) produced by each kinect are combined together using the extrinsic parameters. These parameters

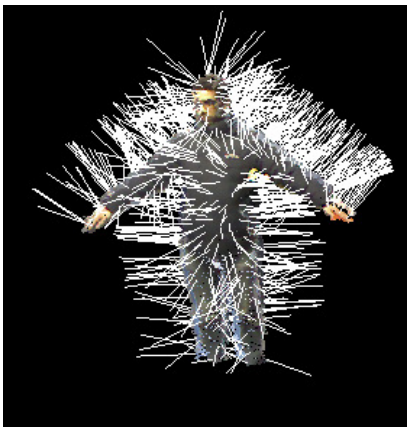

Fig. 5 Illustration of surface normals calculated at randomly chosen points from a CPC.

are the rotation $R_{i,o}$ and translation $T_{i,o}$ matrices between each kinect (i) and a reference kinect (o). Hence, the point cloud (spc_i) captured by a kinect (i) is transformed to the reference frame by means of its rotation and translation matrices. Finally the complete point cloud CP C is obtained as follows:

$$CPC = \bigcup_{i=1}^{N} (R_{i,o} \times spc_i + T_{i,o}) \tag{4}$$

where N is the number of kinects in the platform.
The complete point cloud will improve the classification accuracy of the descriptor as it provides a complete representation of the objects in the scene and reduces occlusion effects.

4 Proposed 3D descriptor

In this section we introduce our 3D descriptor and explain in details how to compute it over a 3D point cloud. The proposed 3D descriptor transposes the HOG into 3D point clouds. In HOG a window is densely subdivided into a uniform grid of blocks. In each block the gradient orientations over the pixels are computed and collected in a 1D histogram. In the 3D point cloud the gradient is meaningless. So it is replaced by the surface normal at each point (Fig. 5). The local surface normal is estimated for each point p of the point cloud using the least-mean square plane fitting [53]. The method works by fitting a plane to the set of neighbouring points of p, and the normal of the plane is assigned to the point p.
The proposed descriptor is computed in two main steps: space subdivision and 3D normal quantization.

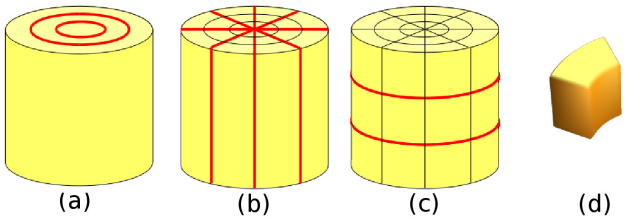

Fig. 6 Cylindric space subdivision steps: a radial cut (a), an azimuth cut (b) and an axial cut (c). The resulting block (d).

4.1 Space subdivision

The 3D space containing the point cloud is divided into sub-areas (blocks). We use a cylindrical subdivision method similar to the one proposed by Gond [54] for his work on pose recognition from voxel reconstruction. The polar subdivision (in 2D) allows rotation invariance. SRHOG [55] uses sector-ring adaptation of HOG for detection of human not only appearing in upright poses. The cylindrical subdivision allows for a division of the point cloud into basic elements that hold enough information about the local geometry of the object. The point cloud is included inside a cylinder and divided as follows:

- First, a radial cut divides the cylinder into sub cylinders(Fig. 6a).
- Second, an azimuth cut divides the cylinder into sectors (Fig. 6b).
- Third, an axial cut across the cylinder main axis subdivides the cylinder into sections (Fig. 6c).

The resulting blocks are in a form of shell sectors as represented in Fig. 6d. An illustration of this process is presented in Fig. 7. Each block contains a certain number of 3D points and then the histogram of oriented normals is computed. The three cuts will help the descriptor to characterize a human body with different sizes (radial cut), different body parts (axial cut) and the orientation of a person (azimuthal cut) whether it is front, back or side view.

4.2 3D normal quantisation

Since a normal is a 3D vector it can not be associated to a 1D histogram. To solve this problem we used the generic 3D orientation quantization proposed by Kläser [56]. In this method the 3D vector is quantized using one of the regular polyhedron shown in Fig. 8. Let $\mathcal{B}$ be the set of points in the block b, n^b is the number of points in this block. Each point p_i from this block is associated with a normal vector $\overrightarrow{n_i}$, this gives $\mathcal{N}$ the set of the corresponding normals vector in this block.

$$\mathcal{B} = \{p_1, p_2, .., p_{n^b}\}, \quad \mathcal{N} = \{\overrightarrow{n_1}, \overrightarrow{n_2}, .., \overrightarrow{n_{n^b}}\} \tag{5}$$

Given a regular n_s-side polyhedron, each face of the polyhedron corresponds to a bin of the histogram. $\overrightarrow{g_s}$ is the vector from the center of the polyhedron

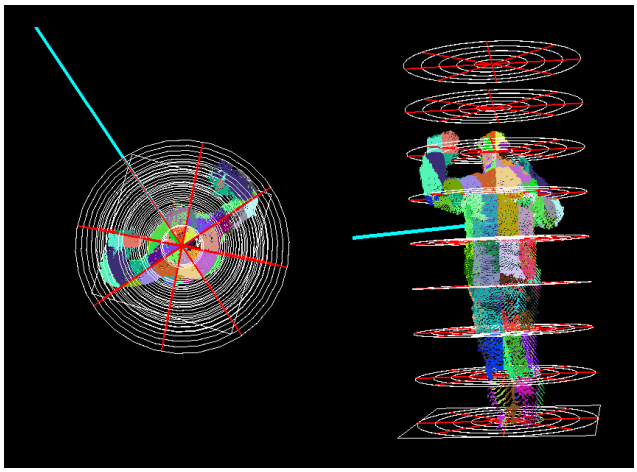

Fig. 7 Illustration of space subdivision on a CPC that represents a person. The frontal direction of the person is shown as a blue vector. a) a top view of the person shows the azimuth cut ($N_a = 8$), the division starts at the location of the frontal vector. b) a side view of the person shows the axial cut ($N_x = 8$) and the radial cut ($N_r = 5$). Points in different blocks are shown in a different color.

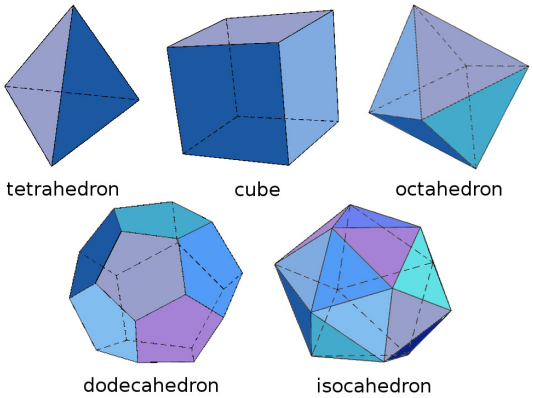

Fig. 8 The five regular polyhedrons.

to the center of its face s. To quantize a normal vector $\overrightarrow{n_i}$, it is placed in the center of the polyhedron as seen in Fig. 9a. The vector is then projected onto each of the vectors $\overrightarrow{g_s}$ as illustrated in Fig. 9b. The projection of the normal vector is computed by:

$$\widehat{q}_{i,s} = max(\overrightarrow{n_i}.\overrightarrow{g_s}, 0) \tag{6}$$

Let h_b be the histogram of the block b, the number of bins in the histogram is equal to the number of sides n_s in the polyhedron used for quantization. The bin s in the histogram h_b will hold the sum of the projection values of all the

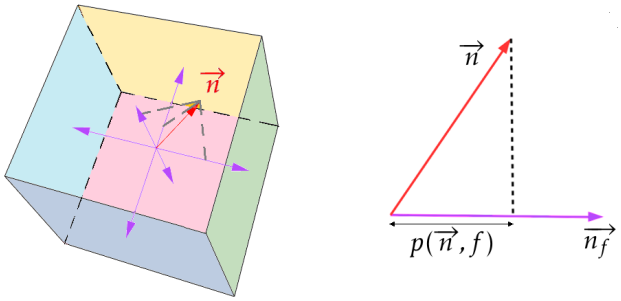

Fig. 9 Quantization of the normal vector ($\overrightarrow{n_i}$) using a cube (n_s=6): a) $\overrightarrow{n_i}$ is positioned at the center of the polyhedron to be projected on each vector $\overrightarrow{g_s}$ that connects the center of the polyhedron and the center of its face s. b) the projection of $\overrightarrow{n_i}$ on the vector $\overrightarrow{g_s}$, the result is the scalar value $\widehat{q}_{i,s}$.

normal vectors in N on the side s of the polyhedron:

$$h_b(s) = \sum_{i=1}^{n^b} \widehat{q}_{i,s} \tag{7}$$

Finally, the normalized histogram related to the block b is computed by:

$$H_b(s) = \frac{h_b(s)}{\sum_{i=1}^{n_s} h_b(s)} \tag{8}$$

The final descriptor is obtained as the concatenation of all the histograms calculated from the blocks.

$$D = \{H_1 \cdot H_2 \cdot ... \cdot H_{N^b}\} \tag{9}$$

where N^b is the number of blocks.

5 3D Database For Human Classification

In this section we introduce our database of CPCs and explain the learning process. The proposed descriptor works on 3D point cloud. Moreover, in order to improve the classification results and add the ability to detect the frontal orientation, we will use CPCs. To our knowledge, no training database has ever provided such types of point clouds. For this purpose, we have decided to build an original database for training and testing the introduced 3D descriptor.

Fig. 10 Examples of CPC of human (left) and non-human subjects (right).

5.1 CPC Database

Our database is composed of CPCs of objects acquired by our multi-kinect system. The database comprises two types of examples: positives (human) and negatives (random objects that can befound in an indoor environment) (Fig. 10). The positive part of the database dedicated to human subjects contains 1000 point clouds. This part was constructed from 34 different persons (males and females) with various poses, shapes and clothing. The negative part of the database contains the non-human examples. It is constituted of elements that could appear in an indoor scene: furniture, stacks of cartons, computer equipment, plants, lamps etc. It consists of approximatively 250 acquisitions of such objects. Actually, rotating the object provides different descriptors. Therefore, by rotating each acquisition around its main axis with a certain angle a different descriptor is produced. Hence, we can generate around 1000 negative examples by rotating each cloud by 90° four consecutive times. When building the database, we saved the frontal direction vector for each positive example in the database. These vectors will be used at a later stage when learning and testing the classifier.
Nowadays large datasets of elements from indoor scenes are available ([57]) but few of these elements have a shape close to the one of a human. We prefer to use our more chalenging database.

5.2 Learning

A Support Vector Machine (SVM) classifier [58] was chosen to train the classification model. The SVM is commonly associated with the HOG descriptor and is known to provide good results. The classification model is learned from the set of positive and negative examples in the database. The classification model will also allow the determination of the frontal orientation of the person. This is achieved with the help of the information about the frontal direction vector of each positive example in the database. As explained previously, the frontal vector will be used as a starting point to do the azimuth cut when constructing the descriptor. This will make the classification model bias to the frontal face of the 3D models that represent a human body.

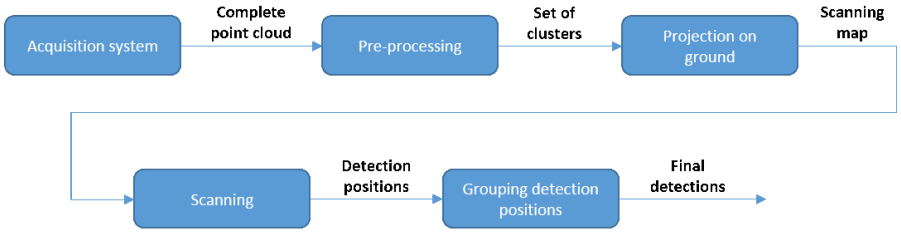

Fig. 11 Human detection procedure pipeline.

6 Proposed Human Detection

In this section we will introduce our human detection procedure that detects one or multiple persons and provides their positions and orientations in the working environment. This procedure works by scanning the ground in the CPC with a cylinder and testing the part of the CPC that falls inside the cylinder for the presence of a human. In order to speed up the detection we perform a segmentation on the CPC to remove parts with a low probability of containing a human. In addition, the scanning will be limited to certain locations on the ground that we call the Candidate Location Map (CLM). The main element in this procedure is the 3D descriptor we have introduced previously. Using this descriptor we will be able to classify parts of the CPC around each candidate location as human or non-human object. The pipeline in (Fig. 11) shows the steps of the detection procedure. In the following sections we explain each step in more details.

6.1 Complete Point Cloud Pre-processing

The multi-kinect system provides a complete point cloud composed of the different objects in the working scene. In the pre-processing step we will apply Euclidean Clustering [59] to segment the CPC into clusters and then we will use a set of heuristics to eliminate irrelevant clusters with a low probability of containing a human.
Before applying the euclidean clustering the ground is removed from the CPC in order to cut the connection between some of objects through the ground. This is done using the ground plane equation that is calculated manually for only one time after calibrating the acquisition system. The result of applying euclidean clustering to the CPC after removing the ground is a set of clusters that represents a rough segmentation of CPC.
The set of clusters is then tested to eliminate irrelevant ones. Since our goal is to detect people in standing/walking position in an indoor environment we can define simple heuristics to identify clusters with a low probability of containing people. If a cluster validates one of the following three conditions then it is removed:

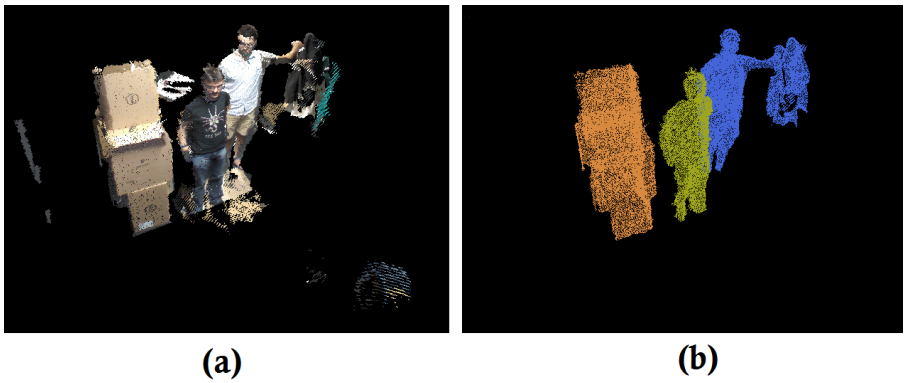

Fig. 12 CPC pre-processing: a) initial CPC, b) final clustering results.

- First, if the lowest point in a cluster relative to the ground is higher than a threshold (i.e. too high above the ground).
- Second, if the highest point relative to the ground is lower than a threshold (too short objects).
- Third, if the majority of the points belong to a planer surface.

The result from this step is a set of clusters. The processed CPC that will be tested for the presence of people is composed of these clusters. Fig. 12 shows an example of applying pre-processing on a CPC. In this example, the ground and parts of the CPC with a low probability of containing people are removed, and the result is a set of three clusters that forms the processed CPC.

6.2 Candidate Location Map (CLM) Construction

We can not apply the classification method directly on each cluster since some clusters may contain a mix of connected objects and persons. Instead, we will scan certain locations from the ground using a cylinder, and classify only the part of the processed CPC that falls inside this cylinder. The locations from the ground that correspond to the center of the cylinder define the Candidate Locations Map (CLM). The CLM is constructed in two steps: first, the CPC is projected on the ground as illustrated in Fig. 13b, the result is a dense point cloud. Second, the projected point cloud is down-sampled [59] to decrease the number of candidate locations. The result of the down-sampling is a set of relatively sparse locations on the ground which constitute the final Candidate Location Map (CLM) as shown in Fig. 13c.

6.3 Scanning of CLM

The CLM map provides a number of locations in the scene to be tested for the presence of a human. In the scanning step we will check each of these loca-

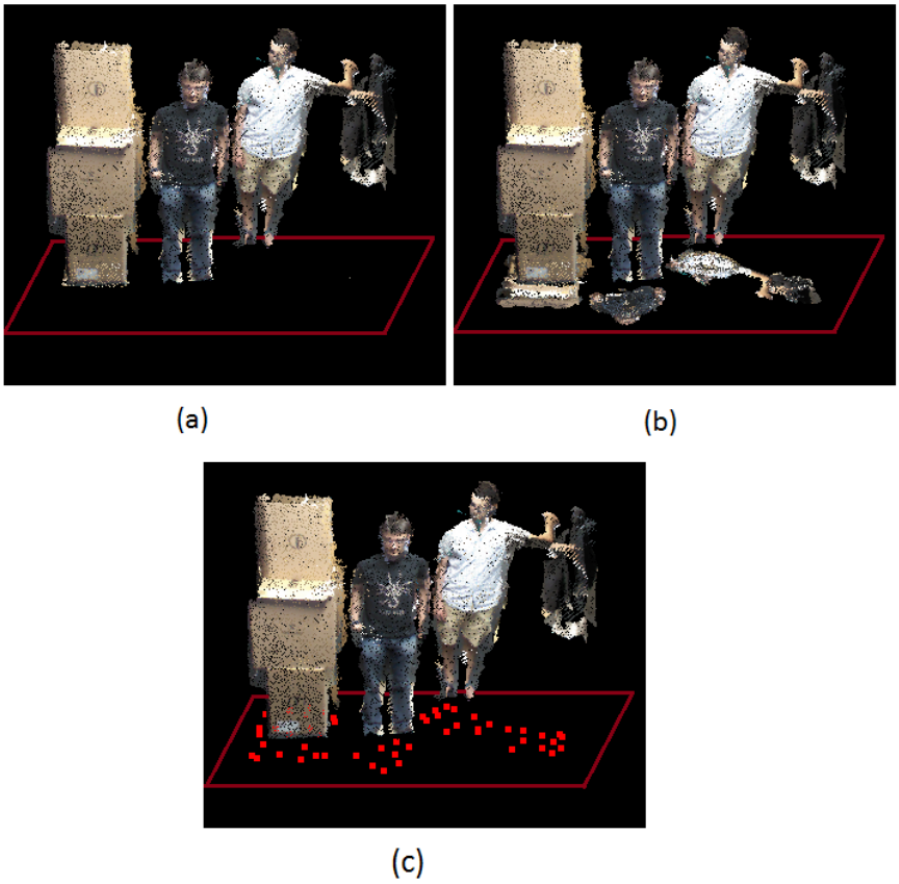

Fig. 13 CLM construction: a) CPC with the ground represented as a red rectangle, b) the projection of the CPC on the ground produces a dense point cloud, c) down-sampling of the projected cloud, the result is a set of point shown as red dots that represents the CLM.

tions. The scanning is done by using a cylinder positioned over one candidate location as illustrated in Fig. 14. The radius and height of the cylinder are determined experimentally. The parts of the point cloud inside the cylinder are extracted and an arbitrary frontal direction vector $\overrightarrow{v}$ is set for this part. Then, the vector $\overrightarrow{v}$ is rotated around the ground normal vector $\overrightarrow{g}$ by an angle θ until it makes a 360° rotation from the starting point. At each rotation we use the $\overrightarrow{v_\theta}$ as the indication of the starting point of the azimuthal cut when computing the 3D descriptor as explained in section 4. We obtain a number of descriptors that will be then classified. After classifying the descriptors, if we obtain a positive detection result we consider the vector $\overrightarrow{v_\theta}$ as frontal direction, which corresponds to the descriptor with the highest classification score. The detection position is the projection of the centroid of the detected cloud on the ground .

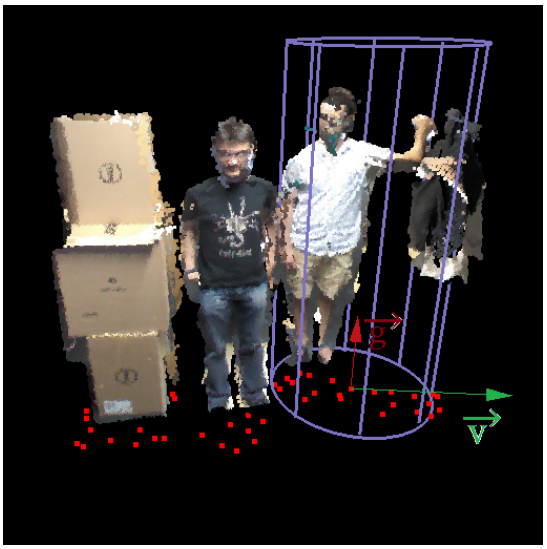

Fig. 14 A scanning cylinder is placed over one of the scanning map points perpendicular to the ground $\overrightarrow{g}$, with an arbitrary frontal direction vector $\overrightarrow{v}$.

6.4 Grouping detection positions

At the end of the scanning process, we may obtain several detection positions for the same person. Therefore we need to identify the group of detection positions that represent the same person and will produce a single detection position. We use euclidean clustering to group the detection positions of each person in a separate cluster. Each detection position is associated with a frontal direction vector, the final position result is the centroid point of a group and the direction vector associated with it is the most dominant direction vector in this group.
Fig. 15 shows an example of grouping detection results. We can see that for each person, detection positions (yellow points) are grouped inside a yellow circle. The final detection is the center of the cluster associated with a direction vector. The green cylinder is the detection cylinder positioned over the final detection point.

7 Experiments

In this part we show the results of the experiments we performed to verify the classification efficiency with the proposed descriptor and evaluate the performance of the human detection method based on this descriptor. The experimental part is organized into two sections. The first section revolves around the descriptor classification performance. In this section we first search for the best parameters that provide the best performance, second, we show the capa-

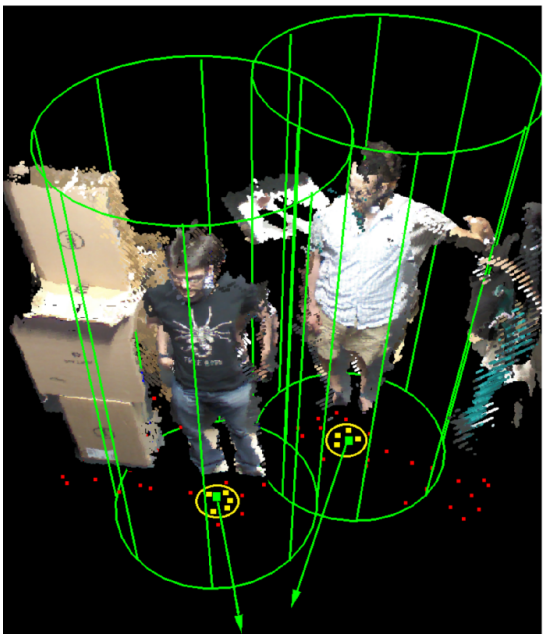

Fig. 15 Final detection results: two detected persons each marked inside a cylinder with the determined frontal vector direction (green vector).

bility of the descriptor to determine the frontal direction. Finally, we illustrate the benefits of using CPC instead of Single Point cloud (SPC) obtained only by one kinect. We do not compare our 3D descriptor with others in terms of efficiency since these descriptors are applied on images and can not be applied on CPC in our database. The second section focuses on the proposed human detection method. In this section we show the results of the comparison between our human detection method and the other two methods. The descriptor is computed in about 30ms with a non-optimized C++ implementation running on a 3GHz processor.

7.1 Classifications

Since there was no similar database in the literature, we dedicated a part of our database to evaluate the classification and to optimize the different required parameters of the method. The set that we used for testing contains 150 positive and 150 negative examples. The examples in the set were then tested by the classification model. The trained classification model returns a score that corresponds to the probability that the point cloud is a human.

Table 1 The best value for each descriptor parameter.

Parameter	Value
Cylinder Height	2 meter
Cylinder Radius	0.5 meter
Polyhedron	octahedron
Cylinder Radial Cut	5 circles
Cylinder Azimuth Cut	8 sectors
Cylinder Axial Cut	8 sections

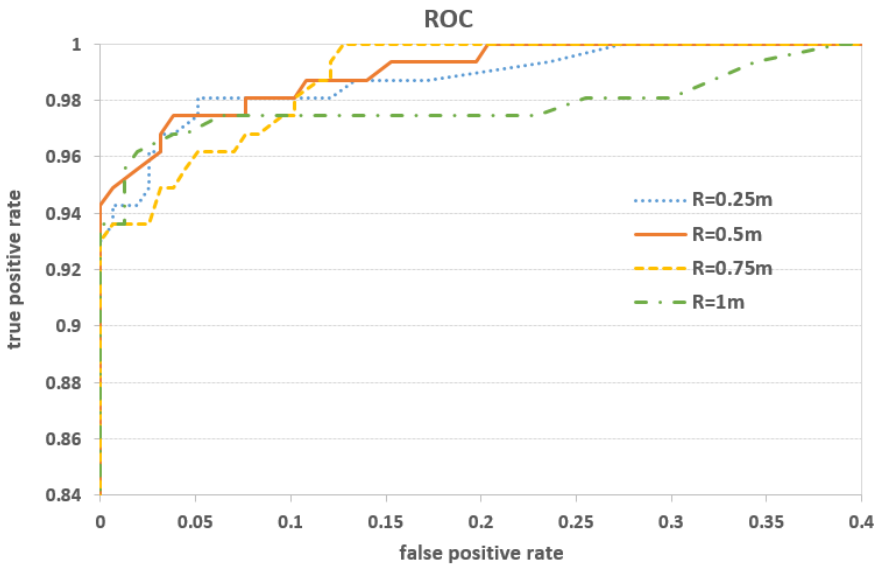

Fig. 16 ROC curves obtained with different values of cylinder radius.

7.1.1 Efficiency

Several parameters were used to compute the 3D descriptor (Table 1). We repeated the classification test several times with different combinations of descriptor parameters. Fig. 16 and Fig. 17 show the ROC curves obtained from different values for the cylinder radius and polyhedron parameters respectively. The first figure shows that a cylinder with one meter radius gives the lowest performance while the other values provides better and almost similar results with a preference for the 0.5 meter. The second figure shows that almost all types of polyhedron provide good results with a slight advantage for the octahedron polygon. Table 1 shows the chosen value for each parameter. With this configuration of parameters, we obtain a precision of 0.97 and a recall of 0.97, which gives a $F_measure$ of 0.97. These excellent results validate the efficiency of our method.

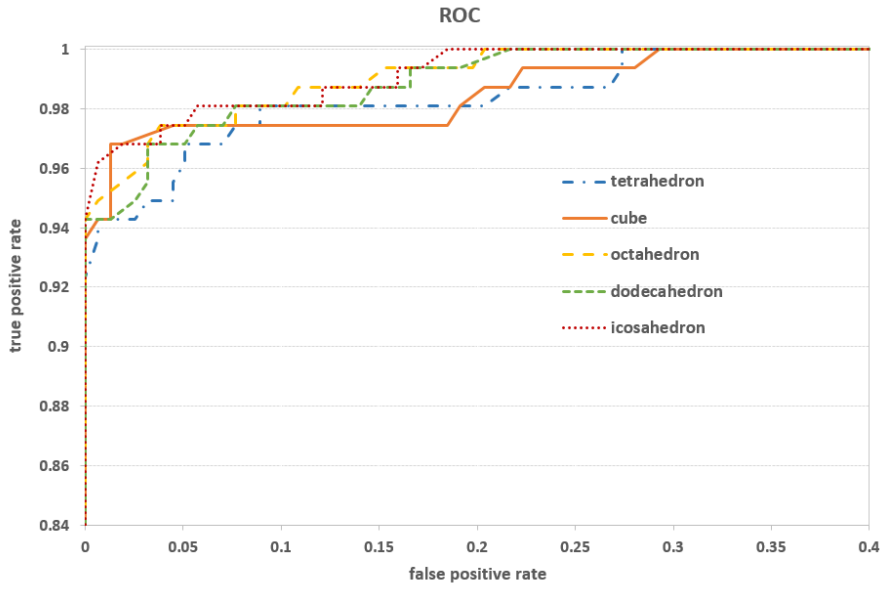

Fig. 17 ROC curves obtained with different types of polyhedrons.

$$\begin{aligned} \text{precision} &= \frac{\text{number of deteced person}}{\text{number of total persons}} \\ \text{recall} &= \frac{\text{number of deteced person}}{\text{number of detections}} \\ \text{F}_{measure} &= \frac{2\times\text{precision}\times\text{recall}}{\text{precision}+\text{recall}} \end{aligned} \tag{10}$$

Fig. 18 shows the results of the classification of a set of positive examples. In this figure we can see that the classifier is able to classify correctly the majority of the examples and determine the right frontal direction. On the other hand, the classifier fails to classify examples with complex poses and in other situations, the classifier only fails to determine the right frontal direction.

7.1.2 Orientation estimation

In order to evaluate the orientation estimation, we tested several hypothetical frontal orientations for each positive example. We chose an arbitrary direction and rotated it around the subjects vertical axis. In our case we performed the rotation 4 times (i.e we increased the rotation angle by 90°). At each rotation we computed the descriptor using the corresponding orientation vector as illustrated in Fig. 19. For each positive example from the testing dataset, we compared the orientation given by the highest score descriptor with the ground-truth orientation that was saved in the database. The orientation is

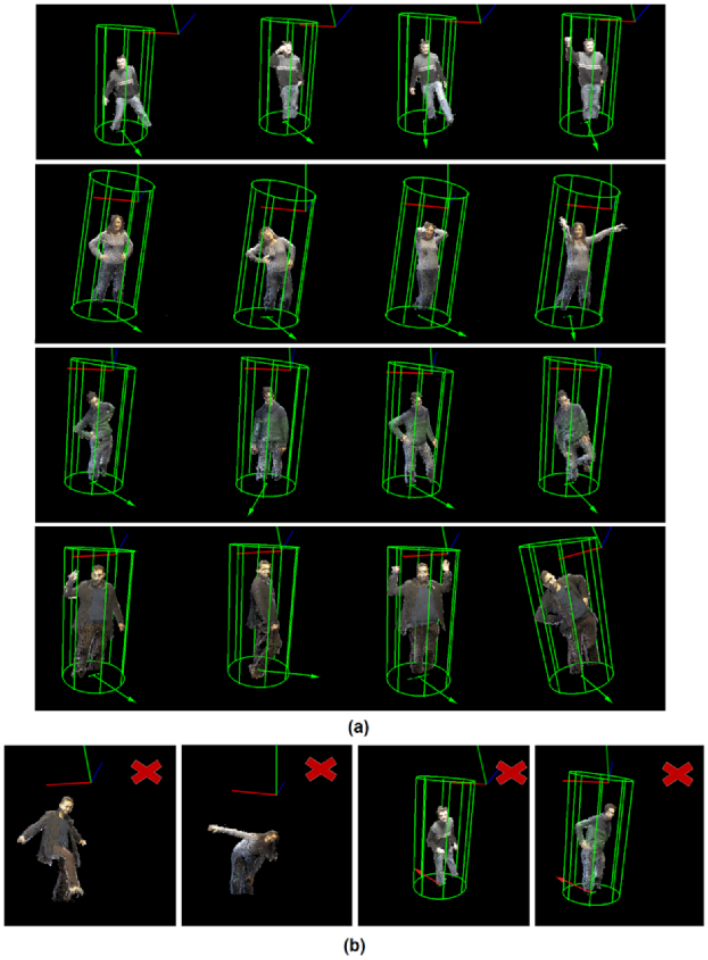

Fig. 18 A set of examples used in the classification evaluation: a) correctly classified examples (inside the green cylinder) with the correct frontal direction vector (green vector); each row contains 4 examples that belong to one person. b) examples where the classifier has fails completely (first two examples from left) or partially where it correctly classifies the examples but with wrong frontal direction (red vector) .

correctly estimated for a vast majority of examples in the database (70%). In the other situations (30%), the back is estimated as the frontal orientation resulting in a 180° error.

7.2 Comparison between CPC and SPC

To illustrate the benefits of using a CPC from a multi-Kinect system we repeated the process of classification in two different scenarios. In the first sce-

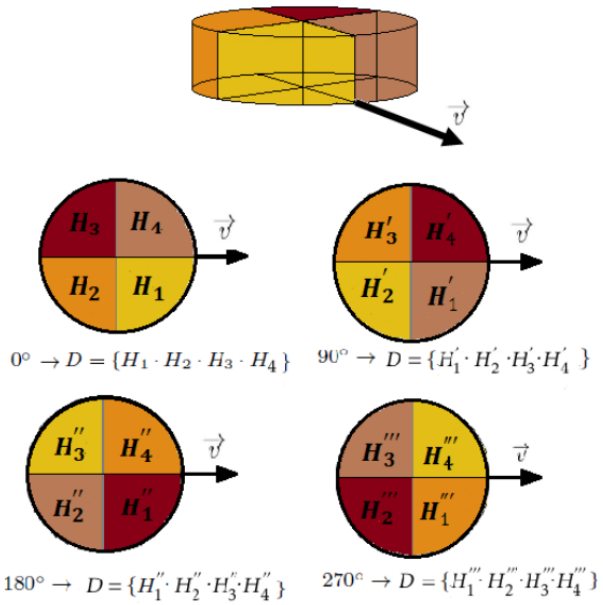

Fig. 19 Descriptor construction relative to the frontal orientation vector $\vec{v}$: (top) cylinder division into blocks; (down) each rotation of the frontal vector yields a new descriptor.

nario we assumed that the kinects of the multi-kinect system were working independently and we computed the descriptors from the single point cloud (SPC) produced by each kinect separately. In the second scenario we considered that the kinects were working together but the output of this multi-kinect system was a set of independent SPCs, and of course the number of these SPCs was equivalent to number of the kinects in the system. In this scenario, we took into account the SPC with the maximum classification result (Max-SPC). Fig. 20 shows the ROC curves for the three experiments. Concerning the first senario, we show the average curve (Average-SPC) of all the ROC curves computed form each separate kinect. It can be seen that the CPC provides the best results which confirms that working with a CPC is better than using separated point clouds independently. In the single point cloud, significant parts of the subject are missing which in turns will decrease the amount of descriptive information. As a result the descriptor ability to discriminate is reduced.

7.3 Human detection

In this section we will illustrate the efficiency of our human detection method by comparing it to two well known detection methods. The first method [15] is the traditional human detection method based on the HOG descriptor applied on RGB images. The second method [12] is also based on the HOG descriptor

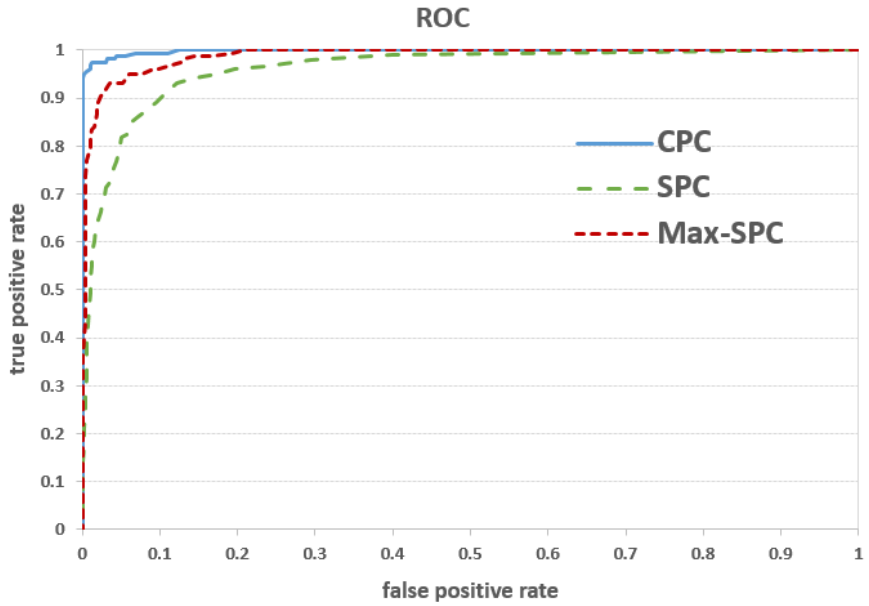

Fig. 20 ROC curves obtained from our complete point cloud (CPC) and with the point cloud of each kinect taken individually (SPC) or combined (Max-SPC). A single view decreases significantly the performances and our CPC outperforms the combination of single point cloud.

Table 2 The results of the comparison experiments.

	Our proposed Method	HOG	HOG-RGBD	
			SPC	Max-SPC
recall	**0.78**	0.4031	0.213	0.505
precision	**1**	0.4426	0.9078	0.886
$F_{measure}$	**0.88**	0.42	0.35	0.64

but it uses 3D point cloud obtained by a RGB-Depth sensors like the kinect. We will call this method HOG-RGBD (i.e HOG for RGB-Depth data). For this comparison, we constructed a dataset of people performing different positions in an indoor location. In this dataset, there are one to three persons in each example as shown in Fig. 22. In total, we have around 200 persons to be detected in all the three scenarios. All the persons used for this experiments are of course not part of the training data used to build the classification model.

7.3.1 Comparison with HOG

First we compared our method with the HOG detector. As the kinect can provide depth and RGB images, we obtained the RGB image of the scene

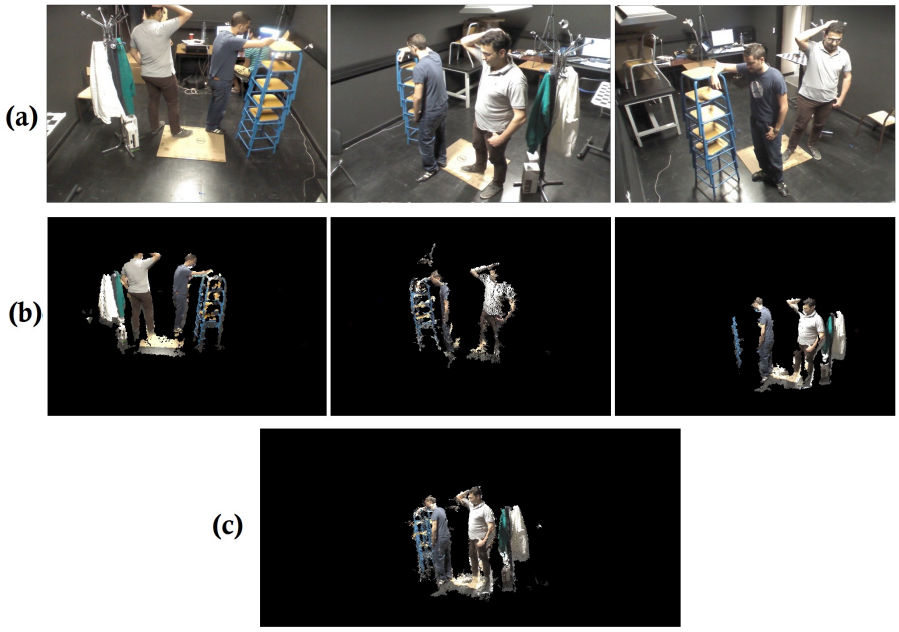

Fig. 21 Data obtained for each scene: a) RGB images b) Single Point Cloud (SPC) from each kinect, c) CPC of the scene.

from each separate kinect. Since we had multiple kinects we obtained different RGB images of the scene from different view angles (Fig. 21a). We applied, on the one hand, our method on the CPC of the scene (Fig. 21c) and on the other hand the HOG detector on the corresponding RGB images. The obtained results are shown on Table 2. The results show that our method outperforms the HOG detector, especially regarding the precision criteria.

7.3.2 Comparison with HOG-RGBD

We compared our method also with HOG descriptor for 3D camera developed in [12]. The method works by selecting a set of candidate clusters from the point clouds and then performs HOG classification method on the corresponding 2D color image of these clusters. This method can not be applied on a CPC, it can only be used with SPC especially organized coloured point clouds.
For each scene we obtained the CPC (Fig. 21c) and also the separate single point clouds (Fig. 21b) from each kinect. We applied our method on the CPC and the HOG-RGBD method was performed separately on each of the other single point clouds. The obtained results are shown on Table 2. SPC corresponds to the classification result of HOG-RGBD from a single point cloud. In Max-SPC (Combined Camera) a cluster provides a detection if it was detected from at least one kinect with HOG-RGBD method. Once again, a single point

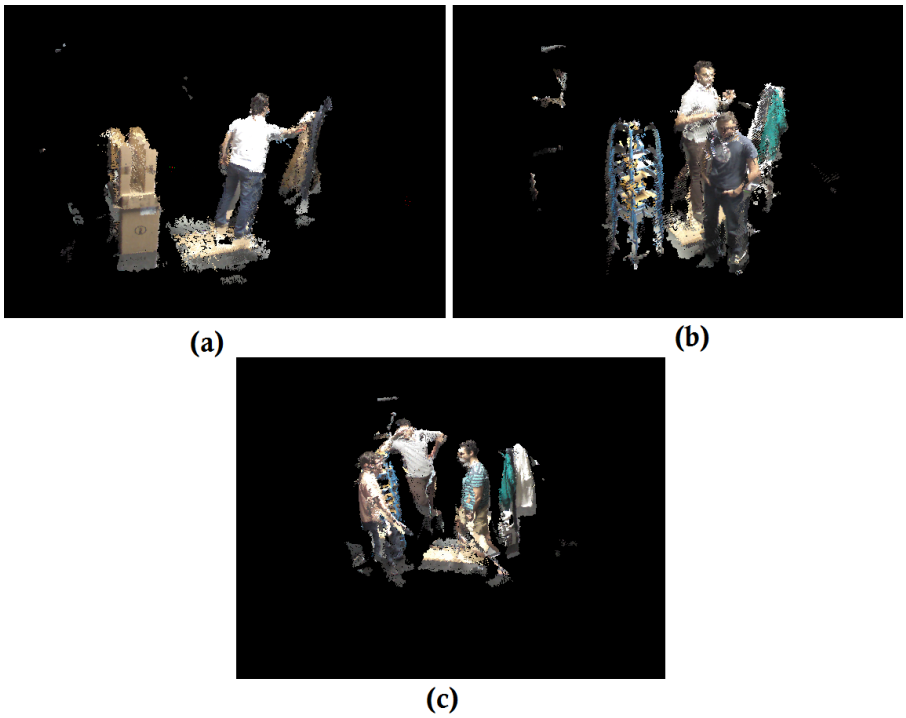

Fig. 22 Three examples from our dataset for human detection test: a)one person b) two persons c)three persons

of view provides low performances. Our method outperforms the HOG-RGBD method processed on the three kinects especially with the recall criterion which shows that our method has a low miss-rate. In Fig. 23 we show examples of the comparison experiments between our method and the other two human detection methods. The examples are shown from five different scenes; in the first scene, we have one person, in the second, third and fourth scenes we have two persons and in the last scene we have three persons. The first three rows (c_1,c_2,c_3) in the figure show the results of applying the HOG detector on the color images obtained by each kinect. The following rows (SPC_1,SPC_2,SPC_3) show the results of applying the RGBD-HOG method on the single point cloud from each kinect. The last row CP C shows the result of our method. In these results, green shapes represent correct detection, red shapes represent wrong detections, and images with no shapes indicate a failure to detect. Our method provides the best detection results as it is able to detected all the persons in the five scenes.

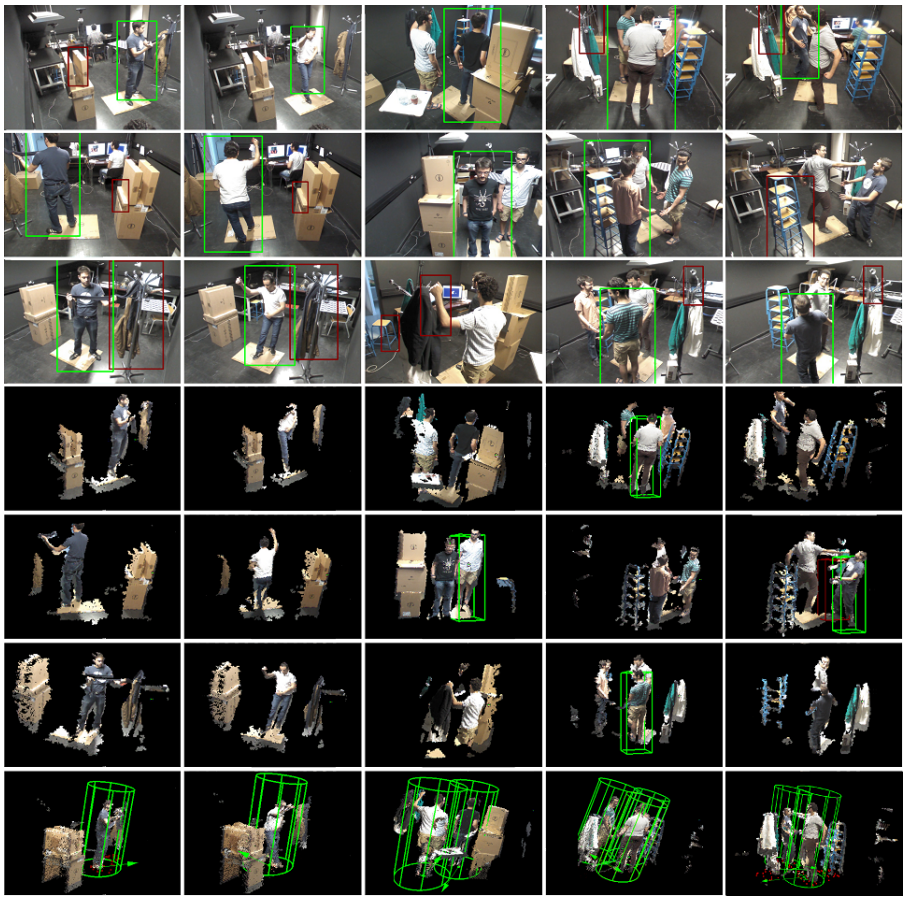

Fig. 23 Examples of the comparison experiments from five different scenes obtained by three kinects.

8 Conclusion

In this paper, we proposed a new 3D descriptor for the human detection which can also estimate the orientation of the human. The detection method is applied on complete point clouds provided by a multi-kinect system. To validate the proposed descriptor, we built an original database of CPCs that represent human and non-human objects. Our experiments show that the descriptor performs classification with an excellent precision. Starting from our 3D descriptor, we introduced a new human detection method for multi-kinect systems. The method can handle scenarios where several persons are moving and interacting in the scene. The method is based on a scanning cylinder that scans across the scene at specific positions and tests for the presence of a human. The comparison with other human detection methods show the efficiency and high performance that our method can provide. Such good results were

achieved by using the surface normal orientation to compute the proposed human descriptor and by exploiting the benefits of a multi-kinect platform. The platform provides the complete point cloud that is crucial to reach a high level performance.
The good detection results we achieved will allow us to build a tracking system which is the next step in our motion analysis system. Tracking is a challenging task in crowded environments with multiple persons and various types of obstacles. Our detection method can help initialize the tracking process for one or multiple persons. In addition, the information about the frontal orientation provided by our method will also help improve the robustness of the tracking. As the kinects asynchronously capture depth data, a temporal interpolation [60] would be required. Our future works aim at optimizing the method to reach a real-time performance in order to embedded efficiently in human tracking applications.

References

1. P. Paul, S.M.E. Haque, S. Chakraborty, EURASIP Journal on Advances in Signal Processing **1**, 1 (2013)
2. E.E. Stone, M. Skubic, IEEE Engineering in Medicine and Biology Society. pp. 5106–9 (2012)
3. A. Drory, G. Zhu, H. Li, R. Hartley, Computer Vision and Image Understanding **159**, 116 (2017)
4. P. Parisot, C.D. Vleeschouwer, Computer Vision and Image Understanding **159**(Supplement C), 74 (2017)
5. V. Campmany, S. Silva, A. Espinosa, J. Moure, D. Vazquez, A. Lopez, Procedia Computer Science **80**, 2377 (2016)
6. A. Shashua, Y. Gdalyahu, G. Hayun, IEEE Intelligent Vehicles Symposium pp. 1–6 (2004)
7. D. Roetenberg, H. Luinge, P. Slycke, (2009)
8. C. Ott, D. Lee, Y. Nakamura, IEEE-RAS International Conference on Humanoid Robots pp. 399–405 (2008)
9. K.M. Culhane, M. OConnor, D. Lyons, G.M. Lyons, Age and Ageing **6**, 556560 (2008)
10. T.B. Moeslund, A. Hilton, V. Kruger, Computer Vision and Image Understanding **23**, 90 (2008)
11. C. Zong, X. Clady, M. Chetouani, IEEE International Conference on Rehabilitation Robotics pp. 1–6 (2011)
12. M. Munaro, F. Basso, E. Menegatti, pp. 2101–2107 (2012)
13. B. Choi, . Merili, J. Biswas, M. Veloso, pp. 1108–1113 (2013)
14. A. Maimone, H. Fuchs, in *IEEE International Symposium on Mixed and Augmented Reality* (2011), pp. 137–146
15. N. Dalal, B. Triggs, in *Computer Vision and Pattern Recognition*, vol. I (IEEE, 2005), vol. I, pp. 886–893
16. B.Z. Lin, C.C. Lin, **4**, 252 (2016)
17. L. Spinello, K.O. Arras, in *International Conference on Intelligent Robots and Systems* (IEEE, 2011), pp. 3838–3843
18. L. Navarro-Serment, C. Mertz, M. Hebert, in *Tracts in Advanced Robotics*, vol. 62 (Springer, 2010), vol. 62, pp. 103–112
19. M. Bajracharya, B. Moghaddam, A. Howard, S. Brennan, L.H. Matthies, in *International Journal of Robotics Research* (2009)
20. S. Ikemura, H. Fujiyoshi, in *Asian Conference on Computer Vision* (Springer, 2011), pp. 25–38

21. Y. Shen, Z. Hao, P. Wang, S. Ma, IEEE Conference on Computer Vision and Pattern Recognition Workshops pp. 535–541 (2013)
22. J. Salas, C. Tomasi, in *Pattern Recognition*, vol. 6718 (Springer Berlin Heidelberg, 2011), vol. 6718, pp. 127–135
23. S. Song, J. Xiao, in *European Conference on Computer Vision* (2014)
24. F. Hegger, N. Hochgeschwender, G.K. Kraetzschmar, P.G. Ploeger, Lecture Notes in Computer Science **7500**, 154 (2013)
25. O.M. Mozos, R. Kurazume, T. Hasegawa, in *International Journal of Social Robotics*, vol. 2 (Springer, 2010), vol. 2, pp. 31–40
26. J. Liu, Y. Liu, G. Zhang, P. Zhu, Y.Q. Chen, in *Pattern Recognition Letters* (Elsevier, 2015), p. 1623
27. O. Oreifej, Z. Liu, IEEE Conference on Computer Vision and Pattern Recognition pp. – (2013)
28. S. Tang, X. Wang, X. Lv, T.X. Han, J. Keller, Z. He, M. Skubic, S. Lao, Asian Conference on Computer Vision **7725**, 525 (2012)
29. A. Johnson, Spin-images: a representation for 3-d surface matching. Ph.D. thesis, The Robotics Institute, Carnegie Mellon University (1997)
30. R.B. Rusu, N. Blodow, M. Beetz, in *Proceedings of the 2009 IEEE International Conference on Robotics and Automation* (2009), pp. 1848–1853
31. F. Tombari, S. Salti, L.D. Stefano, in *European Conference on Computer Vision* (2010), pp. 356–369
32. M. Engelcke, D. Rao, D.Z. Wang, C.H. Tong, I. Posner, International Conference on Robotics and Automation (2017)
33. Q. Tian, B. Zhou, W. Zhao, Y. Wei, W. Fei, in *Journal of Software*, vol. 8 (ACADEMY PUBLISHER, 2013), vol. 8, pp. 2223–2230
34. O. Hosseini Jafari, D. Mitzel, B. Leibe, IEEE International Conference on Robotics and Automation pp. – (2014)
35. L. Xia, C. Chen, J.K. Aggarwal, in *Computer Vision and Pattern Recognition Workshops* (IEEE, 2011), pp. 15–22
36. B. Choi, C. Pantofaru, S. Savarese, in *Conference on Computer Vision Workshops* (IEEE, 2011), pp. 6–13
37. D.M. Gavrila, S. Munder, in *International Journal of Computer Vision*, vol. 73 (Springer, 2007), vol. 73, pp. 41–59
38. J. Satake, J. Miura, IAPR Conference on Machine Vision Applications pp. 8–17 (2009)
39. D. Mitzel, B. Leibe, British Machine Vision Conference pp. – (2012)
40. C. Plagemann, V. Ganapathi, D. Koller, S. Thrun, IEEE International Conference on Robotics and Automation pp. 3108–3113 (2010)
41. J. Shi, J. Malik, IEEE Transaction on Pattern Analysis and Machine Intelligence **22**(8), 888 (2000)
42. M. Eichner, M. Marin-Jimenez, A. Zisserman, V. Ferrari, International Journal on Compututer Vision **99**(2), 190 (2012)
43. C. Chen, A. Heili, J. Odobez, International Conference on Advanced Video and Signal Based Surveillance pp. 5–10 (2011)
44. D. Baltieri, R. Vezzani, R. Cucchiara, European Conference on Computer Vision pp. 270–283 (2012)
45. C. Weinrich, C. Vollmer, H. Gross, International Conference on Intelligent Robots and Systems pp. 2147–2152 (2012)
46. L. Fitte-Duval, A. Mekonnen, F. Lerasle, International Conference on Computer Vision Theory and Applications pp. 439–446 (2015)
47. K. Lai, L. Bo, X. Ren, D. Fox, Conference on Artificial Intelligence pp. – (2011)
48. M.C. Liem, D.M. Gavrila, Image and Vision Computing **32**(10), 728 (2014)
49. Z. Zhang, W. Tao, K. Sun, W. Hu, L. Yao, Pattern Recognition **60**, 227 (2016)
50. J.C. Deveaux, H. Hadj-Abdelkader, E. Colle, in *International Conference on Advanced Robotics* (2013)
51. J.K.e.J.H. D. Herrera, in *Lecture Notes in Computer Science* (2011)
52. C. Raposo, J.P. Barreto, U. Nunes, in *International Conference on 3D Vision* (IEEE, 2013), pp. 342–349
53. D. Holz, S. Holzer, R.B. Rusu, S. Benke, in *Lecture Notes in Computer Science* (Springer, 2012), pp. 306–317

54. L. Gond, P. Sayd, T. Chateau, M. Dhome, in *Lecture Notes in Computer Science*, vol. 5098 (Springer, 2008), vol. 5098, pp. 370–379
55. B. Liu, H. Wu, W. Su, J. Sun, Signal Processing: Image Communication **54**, 1 (2017)
56. A. Klaser, M. Marszalek, C. Schmid, in *British Machine Vision Conference* (2008), pp. 275:1–10
57. O. Mattausch, D. Panozzo, C. Mura, O. Sorkine-Hornung, R. Pajarola, **33** (2014)
58. C. Chang, C. Lin, in *Transactions on Intelligent Systems and Technology*, vol. 27 (ACM, 2011), vol. 27, pp. 1–27
59. R. Rusu, **24** (2010)
60. M. Nakazawa, I. Mitsugami, Y. Makihara, H. Nakajima, H. Habe, H. Yamazoe, Y. Yagi, pp. 11–15 (2012)