UNIVERSITÀ
DEGLI STUDI
DI PADOVA

Sede Amministrativa: Università degli Studi di Padova

Dipartimento di Ingegneria dell'Informazione

SCUOLA DI DOTTORATO DI RICERCA IN: Ingegneria dell'Informazione

INDIRIZZO: Scienza e Tecnologia dell'Informazione

CICLO: XXVI

ROBUST PERCEPTION OF HUMANS FOR MOBILE ROBOTS RGB-DEPTH ALGORITHMS FOR PEOPLE TRACKING, RE-IDENTIFICATION AND ACTION RECOGNITION

Direttore della Scuola:	*Ch.mo Prof. Matteo Bertocco*
Coordinatore d'indirizzo:	*Ch.mo Prof. Carlo Ferrari*
Supervisore:	*Ch.mo Prof. Emanuele Menegatti*

Dottorando:
Matteo Munaro

to Marilisa

Abstract

Human perception is one of the most important skills for a mobile robot sharing its workspace with humans. This is not only true for navigation, because people have to be avoided differently than other obstacles, but also because mobile robots must be able to truly interact with humans. In a near future, we can imagine that robots will be more and more present in every house and will perform services useful to the well-being of humans. For this purpose, robust people tracking algorithms must be exploited and person re-identification techniques play an important role for allowing robots to recognize a person after a full occlusion or after long periods of time. Moreover, they must be able to recognize what humans are doing, in order to react accordingly, helping them if needed or also learning from them. This thesis tackles these problems by proposing approaches which combine algorithms based on both RGB and depth information which can be obtained with recently introduced consumer RGB-D sensors.

Our key contribution to people detection and tracking research is a depth-clustering method which allows to apply a robust image-based people detector only to a small subset of possible detection windows, thus decreasing the number of false detections while reaching high computational efficiency.

We also advance person re-identification research by proposing two techniques exploiting depth-based skeletal tracking algorithms: one is targeted to short-term re-identification and creates a compact, yet discrimative signature of people based on computing features at skeleton keypoints, which are highly repeatable and semantically meaningful; the other extract long-term features, such as 3D shape, to compare people by matching the corresponding 3D point cloud acquired with a RGB-D sensor. In order to account for the fact that people are articulated and not rigid objects, it exploits 3D skeleton information for warping people point clouds to a standard pose, thus making them directly comparable by means of least square fitting.

Finally, we describe an extension of flow-based action recognition methods to the RGB-D domain which computes motion over time of persons' 3D points by exploiting joint color and depth information and recognizes human actions by classifying gridded descriptors of 3D flow.

A further contribution of this thesis is the creation of a number of new RGB-D datasets which allow to compare different algorithms on data acquired by consumer RGB-D sensors. All these datasets have been publically released in order to foster research in these fields.

Sommario

Una delle più importanti abilità per un robot mobile che agisce in un ambiente popolato da persone è la capacità di percepire gli esseri umani. Questo non è vero soltanto per la navigazione perché le persone devono essere evitate in maniera diversa dagli altri ostacoli, ma anche perché i robot mobili devono essere in grado di interagire veramente con gli esseri umani. In un prossimo futuro, si pu immaginare che i robot saranno sempre più presenti in ogni casa e svolgeranno compiti utili al benessere delle persone. Per questo scopo, è necessario utilizzare robusti algoritmi di tracking e le tecniche di re-identificazione svolgono un ruolo importante per far sì che i robot riconoscano una persona anche dopo un'occlusione totale o dopo lunghi periodi di tempo. Inoltre, essi devono essere in grado di riconoscere le azioni delle persone per reagire in maniera adeguata, aiutandole se necessario o anche apprendendo da loro. Questa tesi affronta queste problematiche proponendo approcci che combinano algoritmi basati su informazioni RGB e di profondità che possono essere ottenute con i sensori RGB-D recentemente introdotti nel mercato.

Il nostro contributo chiave alla ricerca sulla rilevazione e il tracking di persone è un clustering basato sull'informazione di profondità che permette di applicare un rilevatore di persone robusto e basato sull'immagine solamente a un ristretto insieme delle possibili finestre di detection, quindi diminuendo il numero di falsi allarmi e raggiungendo un'elevata efficienza computazionale.

La ricerca sulla re-identificazione di persone viene avanzata proponendo due tecniche che sfruttano algoritmi di tracking dello scheletro basati sull'informazione di profondità: una è pensata per la re-identificazione a breve termine e crea una firma compatta, ma discriminativa, delle persone calcolando delle feature alle posizioni chiave dello scheletro, che sono altamente ripetibili e semanticamente significative; l'altra estrae feature a lungo termine, come la forma 3D, per confrontare le persone in base alla loro nuvola di punti 3D acquisita con un sensore RGB-D. Per tenere conto del fatto che le persone non sono oggetti rigidi, ma sono articolate, questa tecnica sfrutta l'informazione 3D dello scheletro per ricondurre le nuvole di punti delle persone ad una posa standard che le renda direttamente confrontabili mediante un fitting ai minimi quadrati.

Infine, viene descritta un'estensione al dominio RGB-D delle tecniche di riconoscimento di azioni basati sul flusso ottico. Questa estensione calcola il flusso nel tempo dei punti 3D di una persona sfruttando congiuntamente l'informazione di colore e profondità e riconosce le azioni umane classificando descrittori a griglia del flusso 3D.

Un ulteriore contributo di questa tesi è la creazione di una serie di dataset RGB-D che permettono di confrontare diversi algoritmi su dati acquisiti con sensori RGB-D di tipo consumer. Tutti questi dataset sono stati rilasciati pubblicamente per favorire la ricerca in questi settori.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Robots are about to move out of the research laboratories and enter into our everyday life, more or less like computers did in the early 1980s. We have already witnessed the mass distribution of little service robots able to navigate autonomously in indoor environments for vacuum cleaning or washing floors. A further step will see robots directly interacting with humans and dynamically reacting to their actions or requests. Some of the key abilities needed by these autonomous intelligent systems will consist in tracking people and understanding their behavior. Other than for service robotics, industrial robots are expected to benefit from similar technologies in order to become more and more intelligent and adaptive with respect to the assigned task and the working environment, which could be shared with humans. Moreover, these skills are also requested for video surveillance applications for automating the surveillance task.

In the past, these complex perception problems have been mainly tackled with approaches based on color or depth data alone. However, 2D approaches are usually slow and sensitive to clutter and occlusions, while depth-based approaches have been usually limited by the fact that 3D sensors had low resolution and high prices. With the advent of reliable and affordable RGB-D sensors, we have witnessed a rapid boosting of robots/agents capabilities. These new cameras directly provide aligned RGB and depth measurements of the scene, thus allowing for algorithms which could exploit this combined information. Microsoft Kinect sensor[1] allows to natively capture RGB and depth information at good resolution and frame rate, thus providing a very rich source of information for mobile systems. Moreover, more and more RGB-D sensors are entering the market and some of them can also work outdoors, thus paving the way for a further diffusion of RGB-D sensors in robotics and video-surveillance.

In this thesis, we will propose novel algorithms and approaches to people track-

[1] It will be described in detail in Section 2.4.1.

ing, re-identification and action recognition which exploit combined RGB and depth information for obtaining robust video analysis at high frame rate. All the techniques proposed in this thesis are targeted to real-time execution on standard machines without the need for GPU processing, in order to be applicable to mobile robots or embedded architectures. This goal is intended to allow service robots (and Intelligent Systems in general) to robustly and dynamically perceive people and their actions and react to their behavior in real time.

1.1 Thesis Outline and Contributions

In Chapter 2, we provide an overview of the sensors most used in robotics for human perception, from perspective cameras to structured-light cameras such as Microsoft Kinect, which has been extensively used in this work.

In Chapter 3, we will describe a novel approach to people detection on RGB-D data which exploits a cascade of 3D and 2D algorithms to obtain very robust results. We will also propose a tracking framework which allows to track multiple people in 3D coordinates by taking into account their 3D motion and color appearance. We will show that our method allows to reach very good performance at high framerates, unfeasible before without exploiting GPU processing.

In Chapter 4, we will propose new techniques for short-term and long-term people re-identification which are very important for extending tracking duration and allowing an agent to re-identify a person previously tracked by another agent. Short-term approaches usually assume that the person to re-identify is wearing the same clothes as at the time of the last observation, while long-term techniques should be able to recognize people after days, months or years, thus they have to rely on more permanent features of the human body and they cannot rely on clothes information. As a contribution to short-term re-identification, since robust and fast depth-based skeletal tracking algorithms have become available together with consumer RGB-D sensors, we will show how to exploit them to obtain compact, yet discrimative signatures of people based on computing features at skeleton keypoints, which are highly repeatable and semantically meaningful. Moreover, for exploiting also complementary information with respect to clothes appearance, usually encoded by short-term re-identification techniques, we will describe novel descriptors which account for long-term features of the human body, deriving from persons' 3D shape. In fact, we will propose to compare people by matching the corresponding 3D point cloud acquired with a RGB-D sensor. In order to account for the fact that people are articulated and not rigid objects, we will

exploit 3D skeleton information for warping people point clouds to a standard pose, thus making them directly comparable by means of a least square fitting. We will also compare this shape-based re-identification with a method deriving a descriptor from skeleton links lengths and we will demonstrate how a combination of skeleton and shape techniques leads to the best results.

Thanks to the proposed novel algorithms which perform people detection and tracking in real-time, mobile robots can also fulfill higher level tasks, such as recognizing actions of the tracked people. In Chapter 5, we will describe an extension of flow-based action recognition methods to the RGB-D domain which computes 3D motion flow by exploiting joint color and depth information and recognizes human actions by classifying gridded descriptors of 3D flow. We will compare this approach with an action recognition algorithm which exploits human pose estimated by consumer depth sensors and which obtains robust results while being applicable on a mobile robot.

A further contribution of this thesis is the creation of a number of new RGB-D datasets which allow to compare different algorithms on data acquired by consumer RGB-D sensors. All these datasets, described in Appendix B, have been publically released in order to foster research in these fields. In particular, we collected datasets targeted to people tracking (IAS-Lab People Tracking dataset and KTP dataset), people re-identification (BIWI RGBD-ID dataset and IAS-Lab RGBD-ID dataset) and action recognition (IAS-Lab Action dataset).

Moreover, implementations of our people detection and tracking algorithms described in Section 3 have been made publically available as part of the Point Cloud Library [102] and of the ROS-Industrial Human Tracker project [2] and as the basis for the OpenPTrack library[3].

1.2 Publications

The work described in this thesis has also been presented in the publications listed here below, divided by topic.

People detection and tracking:

- [81] M. Munaro and E. Menegatti. *Fast RGB-D people tracking for service robots*. To appear in Autonomous Robots Journal, Springer, 2014.

[2] `https://github.com/ros-industrial/human_tracker/tree/develop`.
[3] `http://openptrack.org`.

- [40] S. Ghidoni, S. M. Anzalone, M. Munaro, S. Michieletto and E. Menegatti. *A distributed perception infrastructure for robot assisted living*. To appear in Robotics and Autonomous Systems (RAS) Journal, Elsevier, 2014.

- [78] M. Munaro, F. Basso, S. Michieletto, E. Pagello and E. Menegatti. *A software architecture for RGB-D people tracking based on ROS framework for a mobile robot*. Frontiers of Intelligent Autonomous Systems, Volume 466, pp 53-68, Springer 2013.

- [77] M. Munaro, F. Basso and E. Menegatti. *Tracking people within groups with RGB-D data*. In Proceedings of the International Conference on Intelligent Robots and Systems (IROS) 2012, Vilamoura (Portugal), 2012.

- [7] M. Munaro*, F. Basso*, S. Michieletto and E. Menegatti. *Fast and robust multi-people tracking from RGB-D data for a mobile robot*. In Proceedings of the 12th Intelligent Autonomous Systems (IAS) Conference, Jeju Island (Korea), 2012 (* means equal contribution.)

Person re-identification:

- [76] M. Munaro, A. Basso, A. Fossati, L. Van Gool and E. Menegatti. *3D Reconstruction of freely moving persons for re-identification with a depth sensor*. To be presented at IEEE International Conference on Robotics and Automation (ICRA), Hong Kong (China), 2014.

- [80] M. Munaro, S. Ghidoni, D. Tartaro Dizmen and E. Menegatti. *A feature-based approach to people re-identification using skeleton keypoints*. To be presented at IEEE International Conference on Robotics and Automation (ICRA), Hong Kong (China), 2014.

- [79] M. Munaro, A. Fossati, A. Basso, E. Menegatti and L. Van Gool. *One-shot person re-identification with a consumer depth camera*. Person Re-Identification, pp 161-181, Springer 2014.

Action recognition:

- [75] M. Munaro, G. Ballin, S. Michieletto and E. Menegatti. *3D flow estimation for human action recognition from colored point clouds*. Journal on Biologically Inspired Cognitive Architectures, vol. 5, pp 42-51, 2013.

- [82] M. Munaro, S. Michieletto and E. Menegatti. *An evaluation of 3D motion flow and 3D pose estimation for human action recognition*. In Proceedings of Robotics Science and Systems 2013: Workshop on RGB-D - Advanced Reasoning with Depth Cameras, Berlin (Germany), 2013.

- [4] G. Ballin, M. Munaro and E. Menegatti. *Human action recognition from RGB-D frames based on real-time 3D optical flow estimation*. In Proceedings of Biologically Inspired Cognitive Architectures (BICA) 2012, Advances in Intelligent Systems and Computing, Volume 196, Springer, 2012, pp 65-74.

Chapter 2

RGB and Depth Sensors

In this chapter, we provide an overview of the sensors most used in robotics for human perception. In Section 2.1, perspective monocular and stereo cameras are described, while Section 2.2 introduces laser range finders. In Section 2.3, an overview of matricial time-of-flight technology is given, while Section 2.4 details structured-light sensors and Microsoft Kinect in particular.

2.1 Perspective Cameras

A perspective camera is a device in which the 3D scene is projected down onto a 2D image following a 3D perspective projection. In Figure 2.1, perspective projection based on the pinhole camera model is illustrated. The 3D point $\mathbf{X}$ is projected to the image point $\mathbf{x_C}$, which is the intersection of the image plane with the camera ray passing though $\mathbf{X}$ and the camera center $\mathbf{C}$. The map from 3D world points into pixel coordinates can be thought as a cascade of three successive stages:

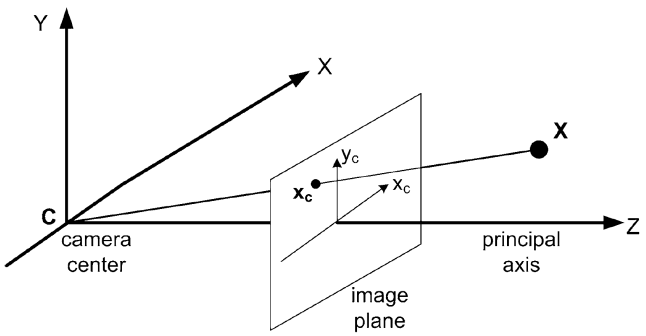


Figure 2.1: Perspective projection based on the pinhole camera model. Figure courtesy of [92].

1. a rigid transformation from points in world coordinates to the same points expressed in camera coordinates

2. a perspective projection from the 3D world to the 2D image plane

3. a mapping from metric image coordinates to pixel coordinates.

These three transformations are synthesized in the following equation:

$$\lambda \overbrace{\begin{bmatrix} x \\ y \\ 1 \end{bmatrix}}^{\substack{\textit{homogeneous} \\ \textit{image} \\ \textit{coordinates}}} = \mathsf{K}\,[\mathsf{R}\ \mathbf{t}]\,\mathbf{X} = \overbrace{\begin{bmatrix} \alpha_x & s & x_0 \\ 0 & \alpha_y & y_0 \\ 0 & 0 & 1 \end{bmatrix}}^{\substack{\textit{intrinsic} \\ \textit{camera} \\ \textit{parameters}}} \overbrace{\begin{bmatrix} r_{11} & r_{12} & r_{13} & t_x \\ r_{21} & r_{22} & r_{23} & t_y \\ r_{31} & r_{32} & r_{33} & t_z \end{bmatrix}}^{\substack{\textit{extrinsic} \\ \textit{camera} \\ \textit{parameters}}} \overbrace{\begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}}^{\substack{\textit{homogeneous} \\ \textit{world} \\ \textit{coordinates}}} \tag{2.1}$$

where K is the matrix containing the intrinsic parameters of the camera (focal lengths, optical center and skew), while R and $\mathbf{t}$ contain the rotation and translation parameters from world to camera reference frame [92].

Standard monocular or color cameras are widely used in robotics, because they can provide data with high resolution and low signal-to-noise ratio. However, understanding a 3D scene from a single image it is a very hard challenge and usually requires highly computationally expensive algorithms. For this reason, 3D information is usually extracted if the camera is moving or if more cameras observe the same scene. For the case where image sequences are captured by the same camera over a period of time, *structure from motion* (SfM) techniques are applied. If images are captured at a high frame rate, optical flow can be computed, which estimates the motion field from the image sequences, based on the spatial and temporal variations of the image brightness. These techniques present three subproblems:

- correspondences between consecutive image frames have to be estimated;

- camera motion (ego-motion) has to be estimated from these correspondences, together with the 3D structure of the observed scene;

- moving objects have to be segmented out from the scene before computing camera motion.

If using consecutive video frames, poor 3D accuracy can be obtained because of the short baseline between cameras, but if the time increment between the images is too

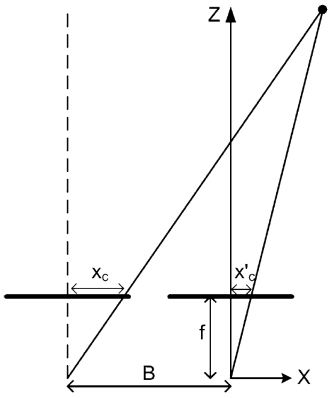


Figure 2.2: Stereo geometry. Figure courtesy of [92].

large, the camera could have moved significantly and image correspondences estimation could fail. One possible solution is then to track features over consecutive frames using a local area-based search and compute 3D from a pair of frames tracked over a significantly longer baseline.

The problems of segmenting out dynamic objects and of jointly estimating camera motion and 3D structure can be avoided if two or more cameras are used. Two perspective cameras mounted on a rigid platform separated by a fixed baseline are called a stereo pair or stereo cameras. In Figure 2.2, the geometry of a stereo system is reported. 3D reconstruction from a stereo system must solve two problems:

- given that a 3D point is projected to a point $\mathbf{x}$ in the left image, determine the corresponding point $\mathbf{x}'$ in the right image;

- given two corresponding points in the left and right image, find the 3D position of the corresponding scene point $\mathbf{X}$.

The former problem is simpler with respect to the structure from motion method, because stereo images can be rectified, that is calibration parameters can be exploited to warp the images so that corresponding points in the left and right images lie on the same image row. At this point, correspondences can be estimated by means of correlation-based or feature-based approaches [92]. The difference in horizontal image coordinates between the corresponding left and right image points is called *disparity* and it is inversely proportional to the distance of the 3D point from the camera. In particular, it can be expressed as

$$d = x_c - x_c^{'} = f\frac{B}{Z}, \tag{2.2}$$

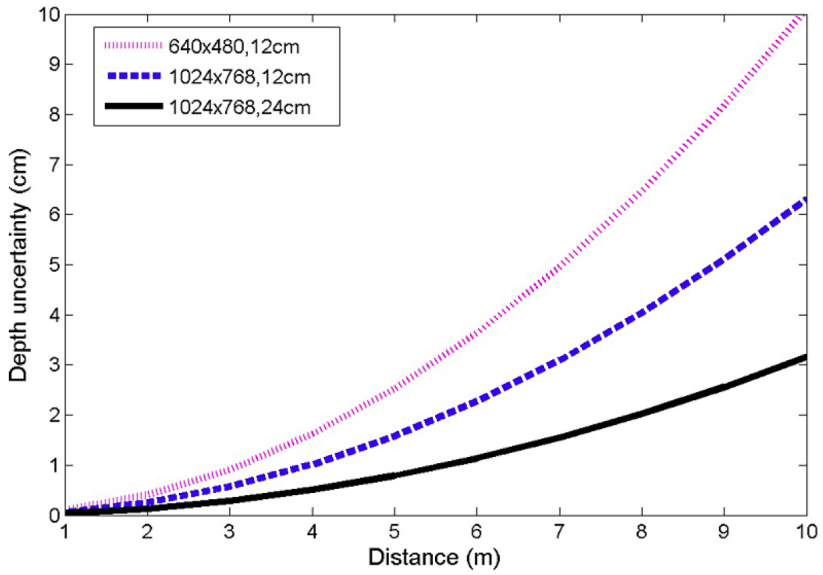


Figure 2.3: Plot illustrating depth estimation uncertainty with respect to distance from the sensor for three different stereo configurations. Figure courtesy of [92].

where f is the camera focal length, B is the baseline and Z is the distance of the 3D point from the image plane. From that formula, that of the standard deviation of the depth estimates with respect to the standard deviation of the disparity can be derived to be

$$\Delta Z = \frac{Z^2}{Bf}\Delta d. \tag{2.3}$$

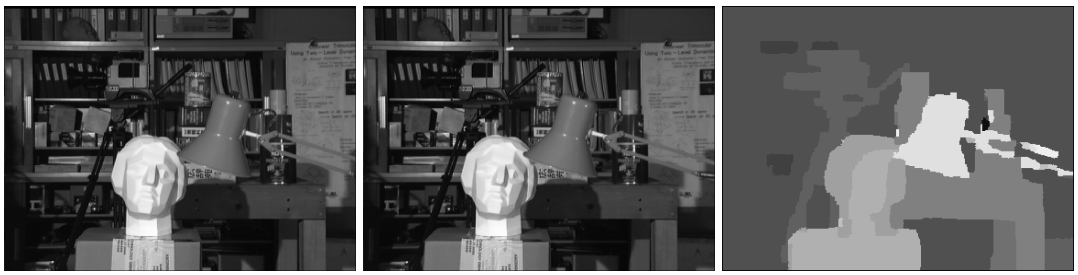
In Figure 2.3, we report the uncertainty in depth estimation with respect to the distance from the sensor for three different stereo configurations which vary in image resolution and baseline. It can be noticed that uncertainty increases with the distance squared and that larger baseline and higher resolution provide better accuracy.

In Figure 2.4, examples of monocular and stereo perspective cameras are reported, while, in Figure 2.5, a pair of stereo images is reported, together with the disparity map which can be computed from them.

The main drawbacks of stereo cameras raise from the need for finding correspondences between left and right image points, which is a computationally expensive operation and it can fail for scenes where texture is poor. For such situations, an active sensor is preferable, such as those which will be presented in the next sections.

(a) PointGrey Grasshopper 3 (b) PointGrey Bumblebee 2

Figure 2.4: Example of (a) monocular and (b) stereo perspective cameras.



(a) Left image (b) Right image (c) Disparity image

Figure 2.5: Example of data which can be obtained with a stereo camera: (a) left image, (b) right image, (c) disparity image.

2.2 Laser Range Finders

Since passive sensors usually fail to obtain 3D information for textureless objects, active sensors are also used in mobile robotics. These sensors exploit artificial illumination to acquire accurate range data even when passive methods do not work. They are usually based on two main measurement principles: time-of-flight and triangulation.

Laser range finders as those shown in Figure 2.6 operate by measuring the time of flight of laser light pulses: as illustrated in Figure 2.7, a pulsed laser beam is emitted and reflected if it meets an object. The reflection is registered by a receiver and the time between transmission and reception of the impulse is directly proportional to the distance between the sensor and the object. In particular, the distance of an object from the sensor is

$$d = c\,t/2 \tag{2.4}$$

where c is the speed of the wave emitted by the sensor and t is the time passed from the emission to the reception of that wave. The confidence in distance estimation is inversely proportional to the square of the received signal amplitude. It is important to point out that an electromagnetic wave travels 3 meters in 10 nanoseconds, thus

(a) Sick LMS100 (b) Sick LMS500 (c) Hokuyo UTM 30LX

Figure 2.6: Example of commercial laser range finders.

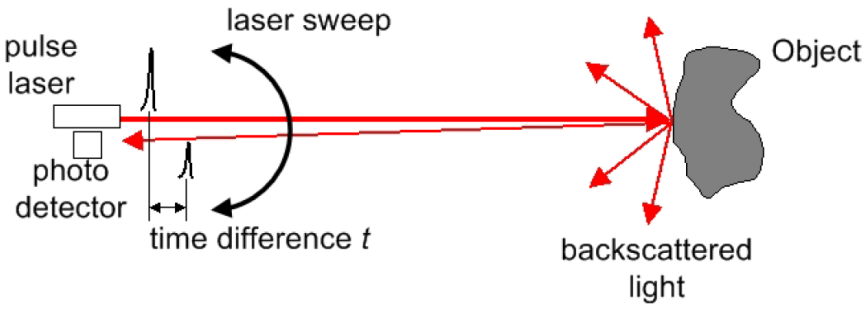


Figure 2.7: Illustration of a 2D laser range finder with rotating mirror based on the laser pulse time-of-flight principle. Figure courtesy of [50].

time of flight estimation is not an easy task and for this reason laser range sensors are usually expensive and delicate. The laser range finders of Figure 2.6 do not estimate just a single distance, but a set of points along one direction. In fact, the pulsed laser beam is deflected by an internal rotating mirror so that a fan-shaped scan is made of the surrounding area. Sometimes, these laser range finders are also mounted on a pan-tilt unit which actuates the laser by making it rotate and tilt. With such a setup, a matrix of measurements can be obtained instead of a line. However, this configuration has some drawbacks:

- the refresh rate is slow because the pan-tilt unit has to complete its movement before having a whole set of measurements. A trade-off between range data resolution and acquisition frequency is usually needed;
- the whole set of laser and pan-tilt unit is quite a heavy payload for a mobile robot;
- there are a high number of moving parts (rotating mirror, pan-tilt unit) which

(a) Mesa SR4000 (b) PMD CamCube 2 (c) Creative Senz3D

(d) Dinast Cyclopes II (e) Microsoft Kinect 2

Figure 2.8: Example of matricial Time-Of-Flight cameras.

could be damaged during robot movements.

2.3 Matricial Time-of-Flight Cameras

Matricial time-of-flight (ToF) cameras are active sensors capable of performing a matrix of depth measurements at framerates up to 50 Hz. As actuated laser range finders, ToF cameras can measure the 3D structure of a whole scene, but they don't have moving parts. In Figure 2.8, some commercial ToF cameras are shown.

These ToF cameras measure depth with a different technique with respect to that applied by the laser range finders presented in Section 2.2. In fact, they measure the phase shift between an emitted infrared light and the reflected signal. If the emitted infra-red (IR) signal is modulated by a sinusoid of frequency f_{mod}, namely

$$s_E(t) = A_E\left[1+\sin(2\pi f_{mod}t)\right], \tag{2.5}$$

the reflected signal which is received by the sensor will be both attenuated and shifted in phase:

$$s_R(t) = A_R\left[1+\sin(2\pi f_{mod}t+\Delta\phi)\right]+B_R. \tag{2.6}$$

An illustration of the emitted and received signals is reported in Figure 2.9. The phase

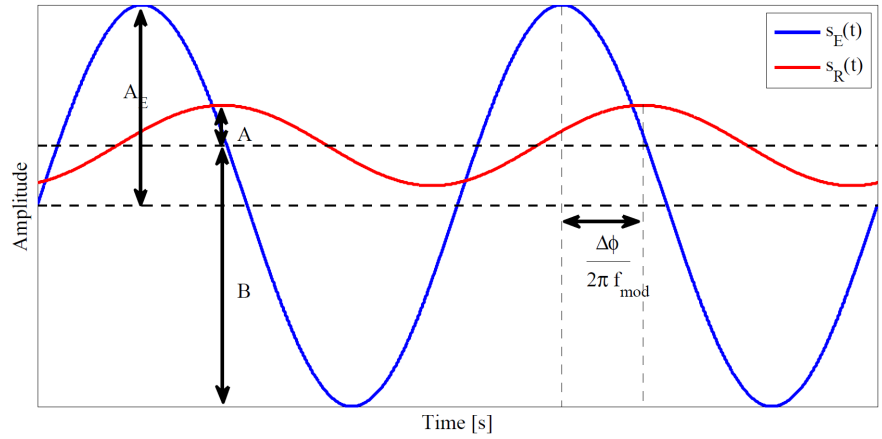


Figure 2.9: Example of emitted signal $s_E(t)$ (in blue) and received signal $s_R(t)$ (in red). Figure courtesy of [29].

shift $\Delta\phi$ used for inferring the distance d can thus be expressed as

$$\Delta\phi = 2\pi f_{mod}\tau = 2\pi f_{mod}\frac{2d}{c}, \tag{2.7}$$

thus, expliciting the distance, we obtain

$$d = \frac{c}{4\pi f_{mod}}\Delta\phi \tag{2.8}$$

where c is the signal propagation speed.

A ToF camera can be thought as a matricial organization of a multitude of single devices, each one made by an emitter and a co-positioned receiver. In practice, this configuration is not possible, thus some emitters are positioned on the sensor in order to mimick the presence of a single emitter co-positioned with the center of a matrix of receivers, which are implemented as CCD/CMOS lock-in pixels [29].

In Figure 2.10, we report an example of data which are provided by a ToF camera. They are images representing *amplitude*, *intensity* and *depth* estimated by all the ToF sensor pixels and scaled to interval $[0,1]$. Since the receivers matricial organization is far more complicated than for time of flight sensors with a single receiver, ToF cameras usually provide low resolution images, e.g. 176 x 144 pixels for the MESA SR4000 of Figure 2.8 (a). However, recently Microsoft released a new generation of Microsoft Kinect (Figure 2.8 (e)) which exploits the ToF technology and estimate depth with a resolution of 512 x 424 pixels.

(a) Amplitude (b) Intensity (c) Depth

Figure 2.10: Example of data obtainable with a matricial Time-Of-Flight camera. Figure courtesy of [29].

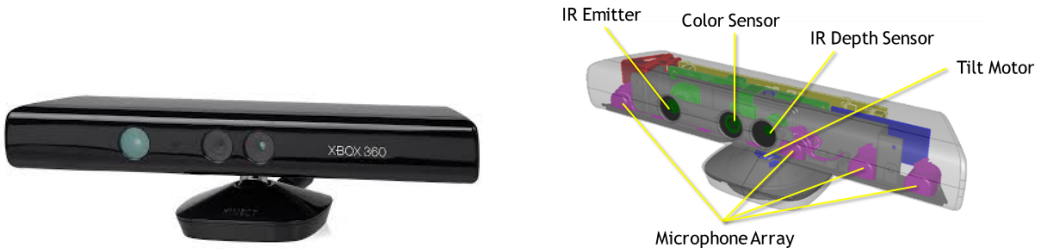
2.4 Structured-Light Cameras

Structured-light sensors are active sensors which use artificial illumination to measure the scene 3D structure by means of triangulation. Active cameras are similar to the stereo system presented in Section 2.1, but replace one camera by a projection device. This projection device can be a digital video projector, an analogue slide projector or a laser and the projected pattern can be a spot, a stripe or a structured light pattern. Spot and stripe scanners usually need to be moved over one or two directions in order to scan the entire scene, while sensors based on projecting a structured light pattern provide matricial depth measurements without the need of moving them, thus minimizing distortion due to motion in dynamic scenes. The correspondence problem, however, is more challenging, thus requiring a coding strategy to recover which camera pixel views a given pattern area. Coding can be either spatial or temporal, but spatial has the advantage to require shorter capture times.

2.4.1 Microsoft Kinect

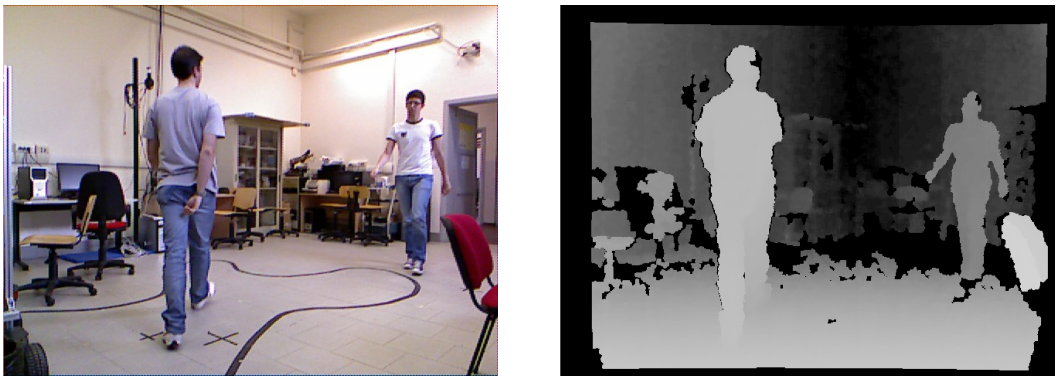
The range camera of the first-generation Microsoft Kinect[1] is a structured-light camera composed of an infrared projector and an infrared camera. The projector emits a light-coded pattern of 640 x 480 pixels at 30 fps. The infrared camera observes the scene and allows to obtain 3D position of the pattern points by means of matricial active triangulation. The minimum measurable depth is 0.8 m, while the maximum is 15 m. In Figure 2.11, an external view of the Kinect is shown, together with a description of its internal components. Kinect is also composed of a RGB camera of VGA resolution which can be calibrated with the range camera in order to have a direct correspondence between points belonging to the RGB and depth images. An example of RGB image

[1]The same hardware is shared with Asus Xtion Pro Live and PrimeSense Carmine 1.08.



(a) (b)

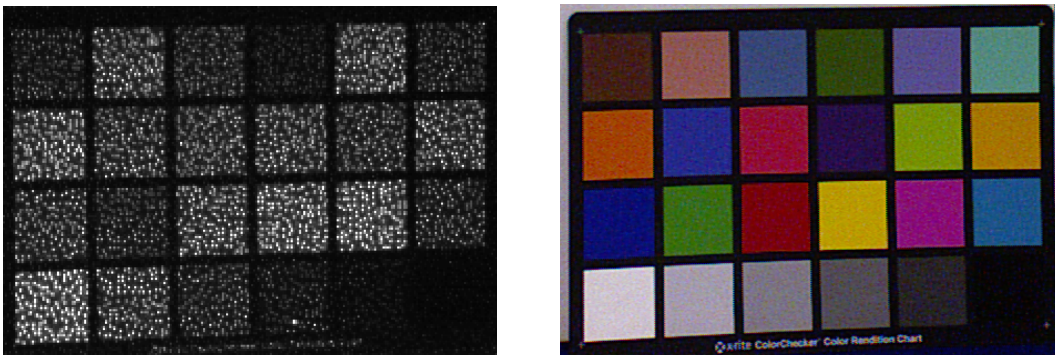
Figure 2.11: (a) External view of the Microsoft Kinect sensor and (b) description of its internal components.



(a) RGB image (b) Depth image

Figure 2.12: Example of RGB and depth images obtained with a Microsoft Kinect. The depth image has been registered to the RGB frame so that there is a direct correspondence between RGB and depth pixels.

and depth image registered to the RGB frame is reported in Figure 2.12, while, in Figure 2.13, it can be seen how the infrared pattern is reflected by surfaces of different colors. Black is the color which absorbs most the infrared light, thus Kinect can hardly estimated depth of objects of this color. Depth is also impossible to estimate if another source of infrared light is projected to the scene. In fact, this would cause the pattern to disappear, thus making the Kinect unable to perform triangulation. Since sunlight also contains infrared light, this kind of cameras cannot be used outdoors. Other than for these reasons, also perspective distortion, occlusions and camera noise can prevent Kinect from recognizing part of the projected pattern in the infrared image. In order to be robust to these non-idealities, a particular code is used in the pattern, which helps in correspondence estimation. The more the code-words are different, the more the code is robust against disturbances. Moreover, the differences between the various codewords become greater as the number of code-words decreases. Thus, the smaller is the number of used code-words, the more robust is the code. For allowing this,



(a) (b)

Figure 2.13: (a) Reflection of Kinect infrared pattern when projected on the color palette in (b). Figure courtesy of [29].

pattern windows at different rows are uncorrelated. Moreover, also different pattern windows on the same rows are uncorrelated, in order to simplify the solution of the correspondence problem.

Thanks to this technology, Kinect-like sensors allow to obtain a big amount of RGB and depth data at high framerate. Moreover, the price of these sensors is very low with respect to laser range finders and ToF cameras, thus they experienced a wide diffusion from the very first moment they appeared in the market, becoming the most used vision sensor for mobile robotics applications.

Chapter 3

People Detection and Tracking

People detection and tracking are among the most important perception tasks for an autonomous mobile robot acting in populated environments. Such a robot must be able to dynamically perceive the world, distinguish people from other objects in the environment, predict their future positions and plan its motion in a human-aware fashion, according to its tasks. In this chapter, we propose a fast and robust approach to people detection and tracking for wheeled robots equipped with RGB-D sensors. Section 3.1 reviews existing work on people detection and tracking, while Section 3.2 gives and overview of our approach. Section 3.3 details depth-based clustering, Section 3.4 explains RGB-based classification, Section 3.5 describes the online classifier we use for describing a person's appearance and Section 3.6 explains the data association process with a three-term joint likelihood. Section 3.7 proposes a tracking policy based on the detector confidence and details about our implementation are given in Section 3.8. Finally, Section 3.9 reports the experiments we performed and conclusions are written in Section 3.10.

3.1 Related Work

Many works exist about people detection and tracking by using monocular images only ([30], [18]) or range data only ([74], [110], [111], [22], [84]). However, when dealing with mobile robots, the need for robustness and real time capabilities usually led researchers to tackle these problems by combining appearance and depth information. In [10], both a PTZ camera and a laser range finder are used in order to combine the observations coming from a face detector and a leg detector, while in [71] the authors propose a probabilistic aggregation scheme for fusing data coming from an omnidirectional camera, a laser range finder and a sonar system. These works,

(a) (b)

Figure 3.1: Example of our system output: (a) a 3D bounding box is drawn for every tracked person on the RGB image, (b) the corresponding 3D point cloud is reported, together with the estimated people trajectories.

however, do not exploit sensors which can precisely estimate the whole 3D structure of a scene. Ess *et al.* [35], [36] describe a tracking-by-detection approach based on a multi-hypothesis framework for tracking multiple people in busy environments from data coming from a synchronized camera pair. The depth estimation provided by the stereo pair allowed them to reach good results in challenging scenarios, but their approach is limited by the time needed by their people detection algorithm which needs 30 seconds to process each image. Stereo cameras continue to be widely used in the robotics community ([3], [103]), but the computations needed for creating the disparity map always impose limitations to the maximum frame rate achievable, especially when further algorithms have to run in the same CPU. Moreover, they do not usually provide a dense representation and fail to estimate depth in low-textured scenes.

With the advent of reliable and affordable RGB-D sensors as those described in Section 2.4, we have witnessed a rapid boosting of robots capabilities. Even though the depth estimation becomes very poor over eight meters and this technology cannot be used outdoors, they constitute a very rich source of information for a mobile platform.

Kinect SDK[1] performs people detection based on the distance of the subject from the background, while NiTE middleware[2] relies on motion detection. Both these approaches work in real time with CPU computation, but they are thought to be used with a static camera and they work for people up to 4 meters of distance from the sensor, thus they are not suitable for mobile robotics applications.

In [109], a people detection algorithm for RGB-D data is proposed, which exploits

[1] `http://www.microsoft.com/en-us/kinectforwindows/develop`.
[2] `http://www.primesense.com/solutions/nite-middleware`.

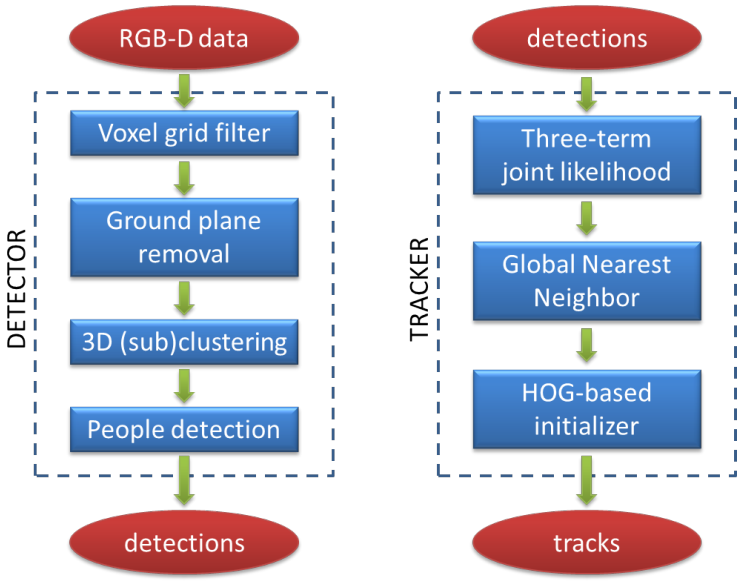


Figure 3.2: Block diagram describing input/output data and the main operations performed by our detection and tracking modules.

a combination of *Histogram of Oriented Gradients* (HOG) and *Histogram of Oriented Depth* (HOD) descriptors and is not limited to static sensors or to a restricted distance range. However, each RGB-D frame is densely scanned to search for people, thus requiring a GPU implementation for being executed in real time. Also [26] and [27] rely on a dense GPU-based object detection, while [73] investigates how the usage of the people detector can be reduced using a depth-based tracking of some *Regions Of Interest* (ROIs). However, the obtained ROIs are again densely scanned by a GPU-based people detector.

In [69], a tracking algorithm on RGB-D data is proposed, which exploits the multi-cue people detection approach described in [109]. It adopts an on-line detector that learns individual target models and a multi-hypothesis decisional framework. No information is given about the computational time needed by the algorithm and results are reported for some sequences acquired from a static platform equipped with three RGB-D sensors.

3.2 System Overview

This work is targeted to develop a people detection and tracking technique for mobile robots working in real time with standard CPU computation. Works reported in Section 3.1 mainly rely on GPU-based dense scanning of the RGB and depth images [109] or on assuming hypothesis not met when dealing with mobile robots (i.e. static

background, as for NiTE). Our approach only assumes that a single ground plane is present in the scene and people are on top of it. This hypothesis is often met in indoor environments, except for people walking on stairs. Thanks to this assumption, we could implement the detection pipeline reported in Figure 3.2. The RGB-D data are processed by a detection module that filters the point cloud data, removes the ground and performs a 3D clustering of the remaining points. Then, we apply a HOG-based people detection algorithm to the projection onto the RGB image of the 3D clusters extended till the ground, in order to keep only those that are more likely to belong to the class of people. The resulting output is a set of detections that are then passed to the tracking module.

In [69], a multi-hypothesis framework is implemented for recovering from tracking errors. However, this choice usually leads to high computational complexity and to a variable tracking framerate. In order to perform fast detection-track association, we used a Global Nearest Neighbor approach for data association. This technique is very fast but does not allow for recovery, thus we prevent errors by computing a robust joint likelihood composed by three terms: motion, color appearance and people detection confidence. For evaluating color appearance, a person classifier for every target is learned online by using features extracted from the color histogram of the target and choosing as negative examples also the other detections inside the image.

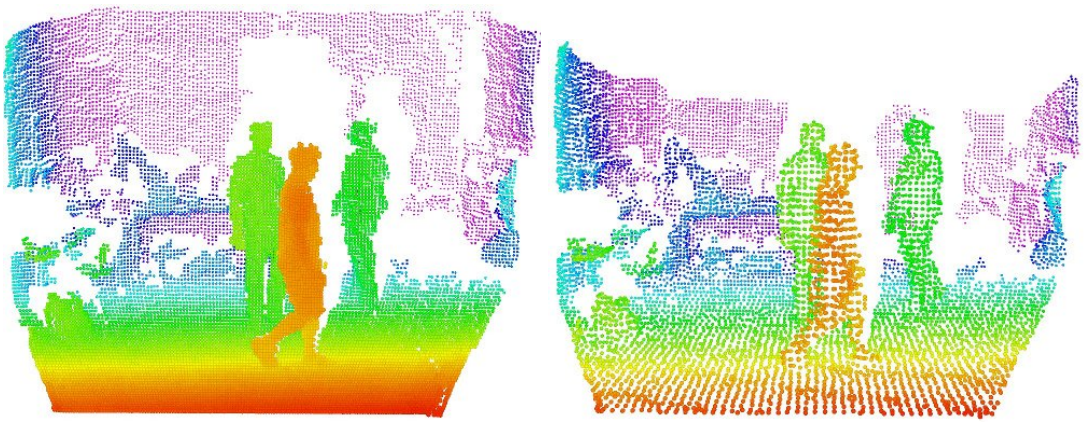
The policies of creation/update/deletion of the tracks are also very important to get good results from the whole tracking process. As we will see in Section 3.7, we exploit the confidence obtained by the people detector for robustly initializing new tracks when no association with existing tracks is found or for detecting that a track has drifted to part of the background.

3.3 Depth-based Clustering

In this section, we explain how we process the Kinect depth point cloud in order to find some 3D clusters to further analyze with a people detection algorithm based on the RGB image. This process allows to obtain a number of clusters two or three orders of magnitude lower than the number of windows analyzed by a dense scanning algorithm as that in [109].

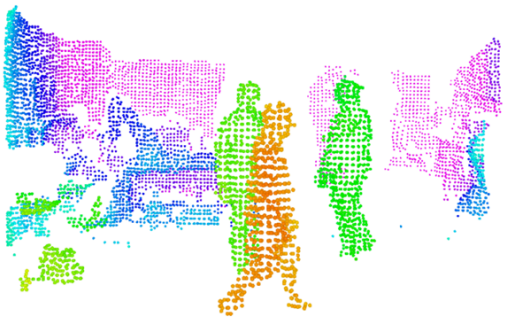
3.3.1 Basic Clustering

As a pre-processing step of our people detection algorithm, we downsize the input point cloud by applying a voxel grid filter, that is a filter which subdivides the space



(a) Raw Kinect point cloud.

(b) After voxel grid filtering.



(c) After ground plane removal.

Figure 3.3: The effect of voxel grid filtering and ground removal on Kinect 3D data.

into a set of voxels (volumetric pixels) and approximates all points inside each voxel with the coordinates of their centroid. This filter is also useful for obtaining point clouds with approximately constant density, where points density no longer depends on their distance from the sensor. In that condition, the number of points of a cluster is directly related to its real size. As an example, in Figure 3.3 (a-b), we compare the raw depth point cloud of the Kinect with the result of the voxel grid filtering when choosing the voxel size to be of 0.06 m.

Since we make the assumption that people walk on a ground plane, our algorithm estimates and removes this plane from the point cloud provided by the voxel grid filter. The plane coefficients are computed with a RANSAC-based least square method [100] and they are updated at every frame by considering as initial condition the estimation at the previous frame, thus allowing real time adaptation to small changes in the floor slope or camera oscillations typically caused by robot movements. An example of point cloud after the ground plane removal is reported in Figure 3.3 (c).

Once this operation has been performed, the different clusters are no longer connected through the floor, so they could be calculated by labeling neighboring 3D points on the basis of their Euclidean distances, namely performing an Euclidean clustering [100]. However, this procedure can lead to two typical problems illustrated in Figure 3.4: (i) the points of a person could be subdivided into more clusters because of occlusions or some missing depth data (Figure 3.4 (a-b)); (ii) more persons could be merged into the same cluster because they are too close or they touch themselves (Figure 3.4 (c-e)) or, for the same reason, a person could be clustered together with the background, such as a wall or a table.

3.3.2 Sub-Clustering Groups of People

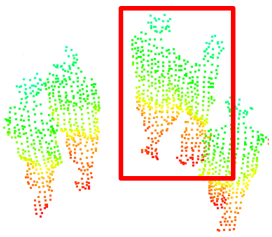
For solving problem (i), after performing the Euclidean clustering, we remove clusters too high with respect to the ground plane and merge clusters that are very near in ground plane coordinates[3], so that every person is likely to belong to only one cluster. For what concerns problem (ii), when more people are merged into one cluster, the more reliable way to detect individuals is to detect the heads, because there is a one to one person-head correspondence and the head is the body part least likely to be occluded. Moreover, the head is usually the highest part of the human body. From these considerations we implemented the following algorithm, that detects the heads from a cluster of 3D points and segment it into sub-clusters according to heads position:

[3]For ground plane coordinates, we mean x and y coordinates in the ground plane reference frame.

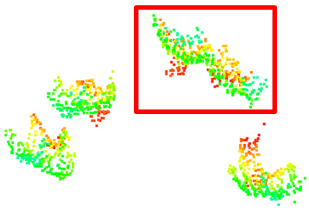
(a) Person point cloud.

(b) Euclidean clustering result.

(c) RGB image.



(d) Point cloud.



(e) Point cloud (top view).

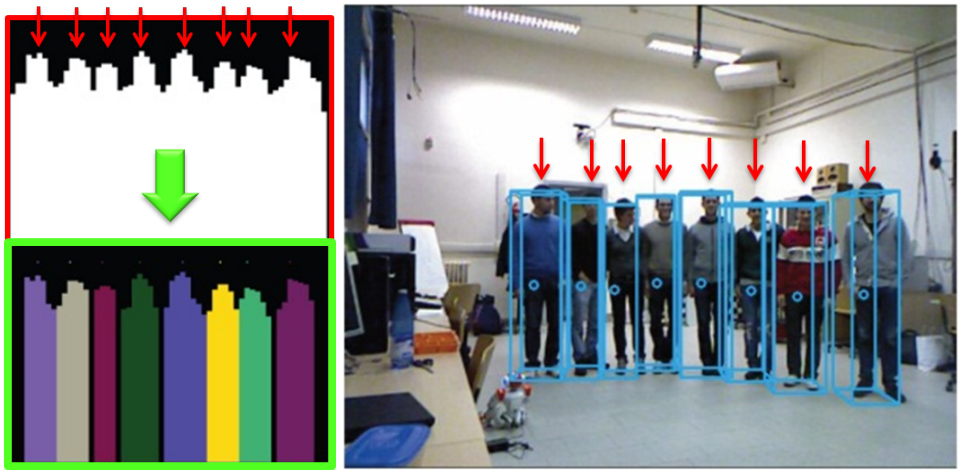
Figure 3.4: Problems of basic clustering techniques: if some depth data are missing (a), a person's point cloud could result split into more clusters (b). If two people touch each other (figures (c-e), red rectangle), they could be merged into a single cluster.

1. for every cluster a height map[4] is created along the direction corresponding to the image x axis[5];
2. local maxima are searched for within the height map;
3. only maxima farther than a threshold distance in ground plane coordinates are kept because people heads are not often nearer than the intimate distance [44], usually equal to 0.3 m;
4. a sub-cluster is created for every remaining maximum (head) by adding cluster points distant to the maximum less than the intimate distance when considering ground plane coordinates;
5. sub-clusters with too few points or not enough high are discarded.

In Figure 3.5, we report an example of sub-clustering of a cluster that was composed by eight people very close to each other. In particular, we show: (a) the black and white height map which contains in every bin the maximum heights from the ground

[4]For every bin, it contains the maximum height from the ground plane.

[5]For speed considerations, we chose to limit the height map along one axis, but it can easily extended to take into account both x and y axis.



(a) Height map and segmentation.

(b) People detection output.

Figure 3.5: Sub-clustering of a cluster containing eight people standing very close to each other.

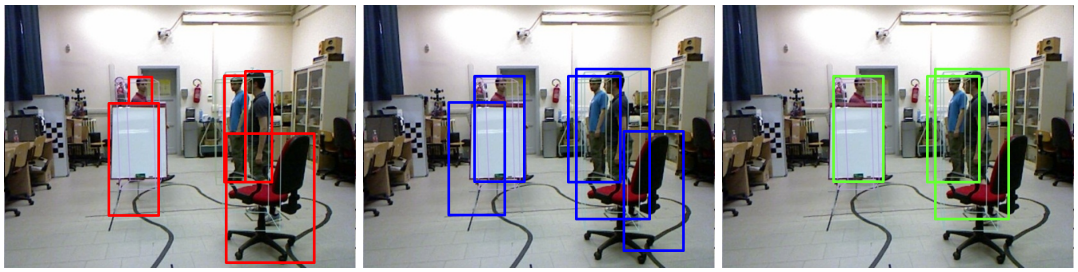
plane of the points of the original cluster, the estimated head positions pointed by red arrows above the height map, the cluster segmentation into sub-clusters explained with colors and (b) the final output of the people detector on the whole image.

3.4 RGB-based Classification

The sub-clustering procedure of Section 3.3.2 allows to obtain a small number of clusters candidate to belong to people. To these clusters we can then apply a more sophisticated people detection technique without considerably decreasing the detection framerate.

HOG detection on clusters extended to the ground

In this work, we exploit a people detector based on the Histogram of Oriented Gradients (HOG) descriptor and Support Vector Machine (SVM) classifier [30]. Given a 3D cluster of points, we project to the RGB image the theoretical bounding box of the cluster, namely the bounding box with fixed aspect ratio that should contain the whole person, from the head to the ground. In Figure 3.6 (a), we report a RGB image where the projection of the real bounding box of the clusters is drawn with red rectangles, while, in Figure 3.6 (b), the projection of their theoretical bounding box is drawn with blue rectangles. These image patches are those sent to the HOG+SVM classification, whose people detection result is shown in Figure 3.6 (c) with green rectangles. It is



(a) Clusters projection to image. (b) Classification input. (c) Classification output.

Figure 3.6: Example of (a) projection to the RGB image of sub-clusters bounding boxes found with the sub-clustering method of Section 3.3.2 (red rectangles), (b) detection windows provided as input to the HOG+SVM classification (blue rectangles) and (c) detection windows which are classified as containing people (green rectangles).

worth noting that using the theoretical bounding boxes allows to obtain a more reliable HOG confidence when a person is occluded, with respect to applying the HOG detector directly to the cluster real bounding box.

For the people detector, we used Dollár's implementation of HOG[6] and the same procedure and parameters described by Dalal and Triggs [30] for training the detector with the *INRIA Person Dataset*[7]. In Figure 3.7, we report the performance of our people detection module evaluated on the KTP dataset (Section B.1.2). The DET curve [33] relates the number of False Positives Per Frame (FPPF) with the False Rejection Rate (FRR) and compares the detection performance with that obtained by applying also the tracking algorithm on top of it. The ideal working point would be in the bottom-left corner (FPPF = 0, FRR = 0%). From the figure, it can be noticed that the tracker performs better than the detector for FPPF > 0.001. Although tested on a different dataset, our people detection algorithm for RGB-D data seems to achieve from one to two order of magnitudes less False Positives Per Frame than state-of-the-art sliding window approaches for RGB images [33].

Figure 3.8 shows an example of how our sub-clustering method allows to correctly detect a person otherwise merged with the background. It can be noticed how the sub-clustering technique splits the cluster into three sub-clusters and only the sub-cluster containing a person is validated by the HOG+SVM classification procedure.

In Figure 3.9, the effect on detection results of merging close clusters and imposing constraints on clusters height, dimension and HOG confidence is shown for a frame of the KTP dataset.

[6]Contained in his Matlab toolbox `http://vision.ucsd.edu/~pdollar/toolbox`.
[7]`http://pascal.inrialpes.fr/data/human`.

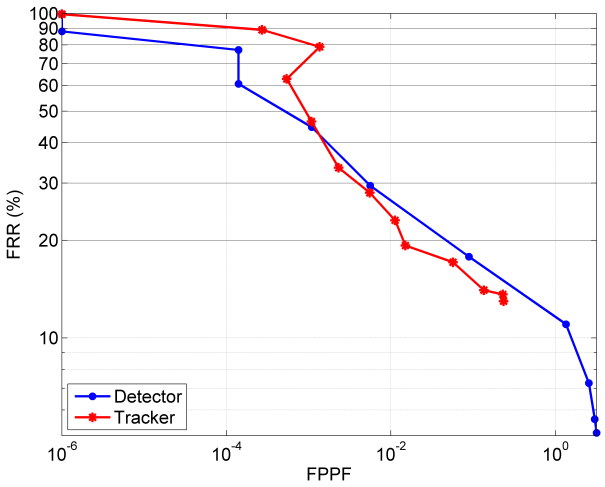
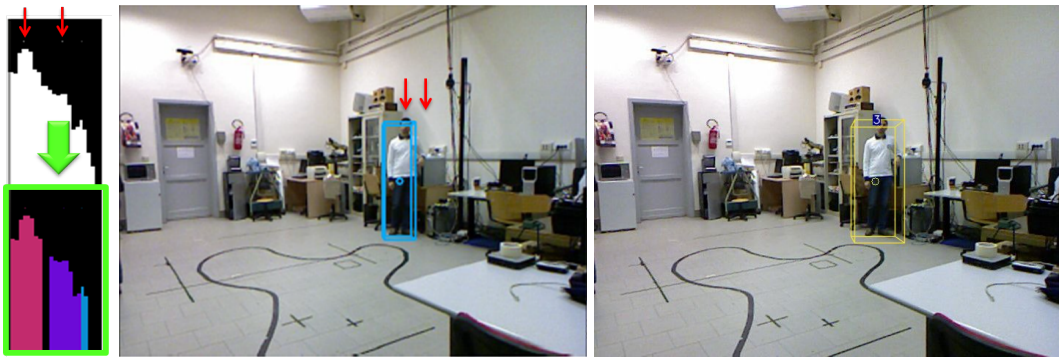
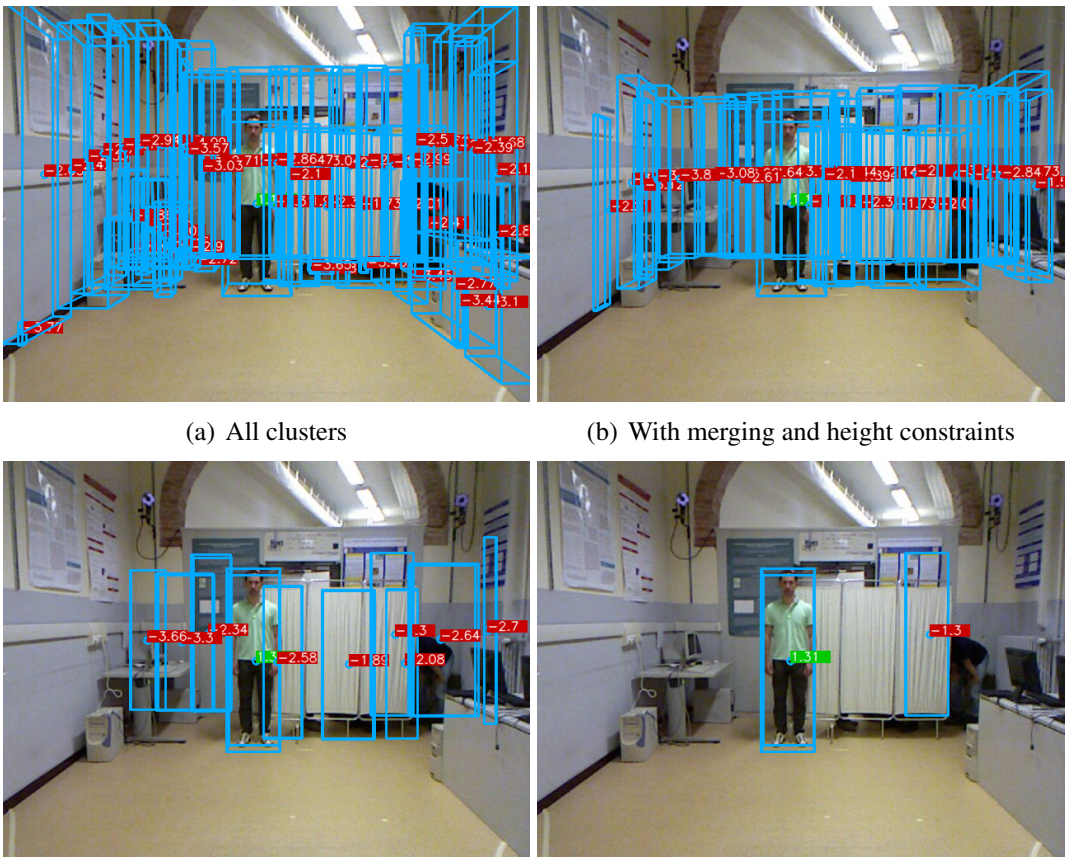


Figure 3.7: DET curve comparing the detector and tracker performance on the KTP dataset in terms of False Positives Per Frame (FPPF) and False Rejection Rate (FRR).



(a) (b) (c)

Figure 3.8: Example of how the sub-clustering method allows to correctly detect a person otherwise merged with the background. a) Height maps, b) detector output, c) tracker output.



(a) All clusters

(b) With merging and height constraints

(c) With dimension constraints

(d) With enough HOG confidence

Figure 3.9: People detection results when (a) no merging is performed and no constraints are imposed and when (b) merging close clusters and adding constraints on height, (c) number of points and (d) HOG confidence. For every cluster, its HOG confidence is reported.

We released an implementation[8] of our people detection algorithm as part of the open source *Point Cloud Library* ([102] in order to allow comparisons with future works and its use by the robotics and vision communities. It can be noticed that, if people are too close to the camera and the heads are not visible, the subclustering algorithm can fail to separate individuals, but also sliding window approaches can fail in these conditions. The height map we used for detecting heads can be easily extended to the *x*-*z* plane, thus allowing to separate also a short person in front of a tall person or two heads at the same height close together in *x* but not *z*.

As person position, we compute the centroid of the cluster points belonging to the head of the person and we add 0.1 m in the viewpoint direction, in order to take into account that the cluster only contains points of the person's surface.

3.5 Online Classifier for Learning Color Appearance

For every initialized track, we maintain an online classifier based on Adaboost, like the one used in [41] or [69]. But, unlike these two approaches, that make use of features directly computed on the RGB (or depth) image, we calculate our features in the color histogram of the target, as following:

1. we compute the color histogram ($\mathscr{H}$) of the points corresponding to the current detection associated to the track. This histogram can be computed in RGB, HSV or other color space; here, we assume to work on the RGB space. If B is the number of bins chosen, 16 by default, then

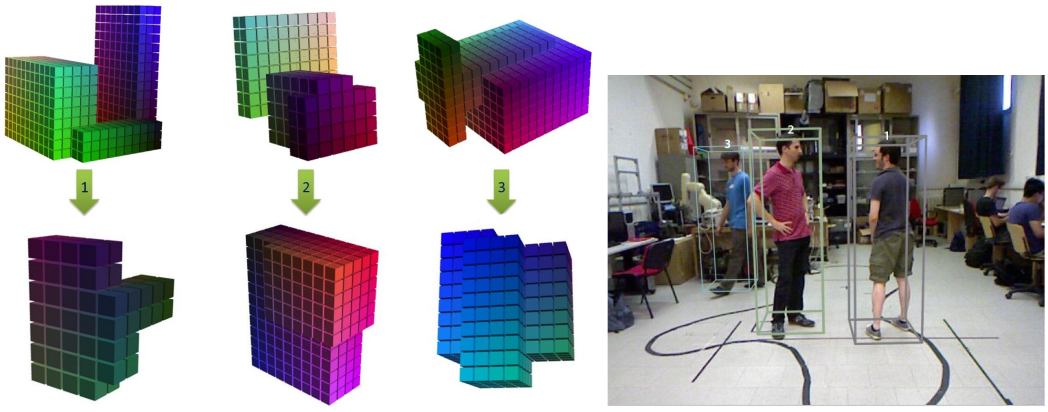
$$\mathscr{H} : [1...B]\text{x}[1...B]\text{x}[1...B] \rightarrow \mathbb{N} \tag{3.1}$$

2. we select a set of randomized axis-aligned parallelepipeds (one for each weak classifier) inside the histogram. The feature value is given by the sum of histogram elements that fall inside a given parallelepiped. If B_R, B_G and B_B are the bins ranges corresponding to the R, G and B channels, the feature is computed as

$$f(\mathscr{H}, B_R, B_G, B_B) = \sum_{i \in B_R} \sum_{j \in B_G} \sum_{k \in B_B} \mathscr{H}(i, j, k). \tag{3.2}$$

With this approach, the color histogram is computed only once per frame for all feature computations. In Figure 3.10 (a) we report the three most weighted features (par-

[8] `http://pointclouds.org/documentation/tutorials/ground_based_rgbd_people_detection.php`.



(a) (b)

Figure 3.10: (a) From left to right: visualization of the features selected by Adaboost at the first frame (first row) and after 150 frames (second row) for the three people shown in (b).

allelepipeds in the RGB color space) for each one of the three people of Figure 3.10 (b) at the initialization (first row) and after 150 frames (second row). It can be easily noticed how the most weighted features after 150 frames highly reflect the real targets colors.

For the training phase, we use as positive sample the color histogram of the target, but, instead of selecting negative examples only from randomly selected windows of the image as in [41], we consider also as negative examples the histograms calculated on the detections not associated to the current track. This approach has the advantage of selecting only the colors that really characterize the target and distinguish it from all the others. Figure 3.11 clearly shows how this method increases the distance between the confidences of the correct track and the other tracks.

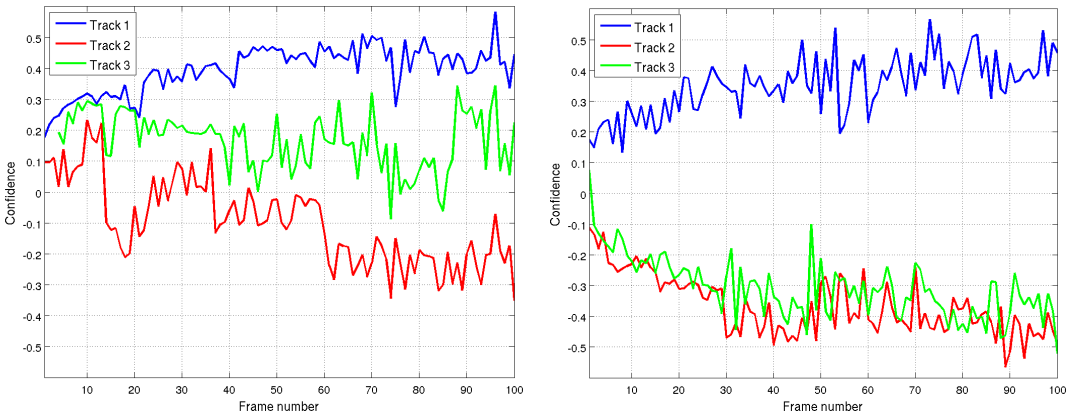
3.6 Three-Term Joint Likelihood for Data Association

For performing data association, we use the Global Nearest Neighbor approach (solved with the Munkres algorithm), described in [56] and [10]. Our cost matrix derives from the evaluation of a three-term joint likelihood for every target-detection couple.

As motion term, we compute the Mahalanobis distance between track i and detection j as

$$D_M^{i,j} = \tilde{\mathbf{z}}_k^{\mathrm{T}}(i,j) \cdot \mathbf{S}_k^{-1}(i) \cdot \tilde{\mathbf{z}}_k(i,j) \tag{3.3}$$

where $\mathbf{S}_k(i)$ is the covariance matrix of track i provided by a filter and $\tilde{\mathbf{z}}_k(i,j)$ is the



(a) Random windows.

(b) Other detections and random windows.

Figure 3.11: Confidence obtained by applying to the three people of Figure 3.10 (b) the color classifier trained on one of them (Track 1) for two different methods of choosing the negative examples.

residual vector between measurement vector based on detection j and output prediction vector for track i:

$$\tilde{\mathbf{z}}_k(i,j) = \mathbf{z}_k(i,j) - \hat{\mathbf{z}}_{k|k-1}(i). \tag{3.4}$$

The values we compare with the Mahalanobis distance represent people positions and velocities in ground plane coordinates. Given a track i and a detection j, the measurement vector $\mathbf{z}_k(i,j)$ is composed by the position of detection j and the velocity that track i would have if detection j were associated to it.

An Unscented Kalman Filter is exploited to predict people positions and velocities along the two ground plane axes (x,y). For human motion estimation, this filter turns out to have estimation capabilities near those of a particle filter with much smaller computational burden, comparable to that of an Extended Kalman Filter, as reported by [10]. As people motion model we chose a constant velocity model because it is good at managing full occlusions, as described in [10].

Given that the Mahalanobis distance for multinormal distributions is distributed as a chi-square [83], we use this distribution for defining a gating function for the possible associations.

For modeling people appearance we add two more terms:

1. the color likelihood, that helps to distinguish between people when they are close to each other or when a person is near the background. It is provided by the online color classifier learned for every track;

2. the detector likelihood, that helps keeping the tracks on people, without drifting

to walls or background objects when their colors look similar to those of the target. For this likelihood, we use the confidence obtained with the HOG detector of Section 3.4.

The joint likelihood to be maximized for every track i and detection j is then

$$L_{TOT}^{i,j} = L_{motion}^{i,j} \cdot L_{color}^{i,j} \cdot L_{detector}^{j}. \tag{3.5}$$

For simpler algebra we actually minimize the log-likelihood

$$l_{TOT}^{i,j} = -log\left(L_{TOT}^{i,j}\right) = \gamma \cdot D_M^{i,j} + \alpha \cdot c_{online}^{i,j} + \beta \cdot c_{HOG}^{j}, \tag{3.6}$$

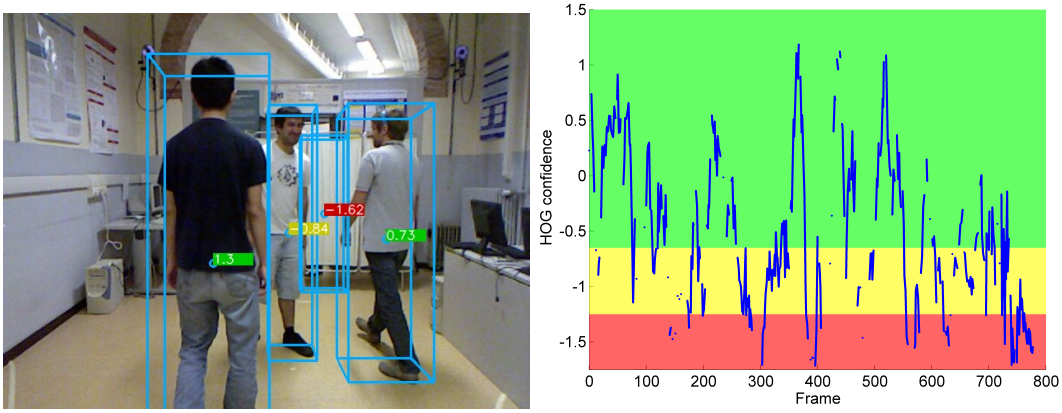
where $D_M^{i,j}$ is the Mahalanobis distance between track i and detection j, $c_{online}^{i,j}$ is the confidence of the online classifier of track i evaluated with the histogram of detection j, c_{HOG}^{j} is the HOG confidence of detection j and γ, α and β are weighting parameters empirically chosen.

3.7 HOG-based Tracking Policy

As stated in Section 3.2, people detection confidence is used by our algorithm for robust track initialization and for detecting track drifts. We divide HOG confidence values into three groups:

- *red*: if they are near the minimum confidence used for people detection;
- *green*: if they are above the value required for initializing a track;
- *yellow*: if they are between *red* and *green* values.

After data association, if there are unassociated detections with *green* HOG confidence, new tracks are created. Then, detections with *yellow* or *red* confidence are also used for data association. Nevertheless, if too many detections with *red* confidence are consecutively associated to a track, that track is likely to have drifted to a background object, thus our algorithm starts to require at least a detection with *yellow* confidence for updating that track. In Figure 3.12 (a), we show the people detection output for a frame of the KTP dataset and the HOG detection values colored as explained before. In Figure 3.12 (b), we report the HOG confidence values for all the detections associated to the occluded person with a white t-shirt of Figure 3.12 (a), from its track creation to its deletion. This person undergoes various occlusions and it often turns on itself in



(a) (b)

Figure 3.12: (a) People detection output and HOG confidences colored according to their values. (b) HOG confidence trend for the track relative to the central person in (a).

order to change movement direction. We noticed that the HOG value enters the *yellow* zone when the person is partially occluded or seen from a lateral point of view, while enters the *red* zone just for few frames in presence of very strong occlusion or at the end, when the person exits the room and its track drifts to the room divider, which has a similar color. However, given that the room divider has *red* HOG confidence, after a few frames the track is not associated to it any more.

3.8 Software Architecture

All our people detection and tracking algorithms are implemented in `C++` and exploit the Robot Operating System (ROS) [96]. This section focuses on how ROS tools and functionalities have been used for the architectural design of our system, its implementation, debug and testing.

3.8.1 Modularity with Nodes, Topics and Message-Passing

ROS nodes are processes that perform computation and communicate with each other by asynchronous message passing. Messages are exchanged within a network by means of TCP or UDP and written/read to/from topics, that are named buses over which nodes exchange well defined types of messages.

ROS nodes structure highly incentives software modularity, so that most of the code can be reused also for other applications. In particular, sensors drivers are usually confined into single nodes, while data processing is implemented within other nodes.

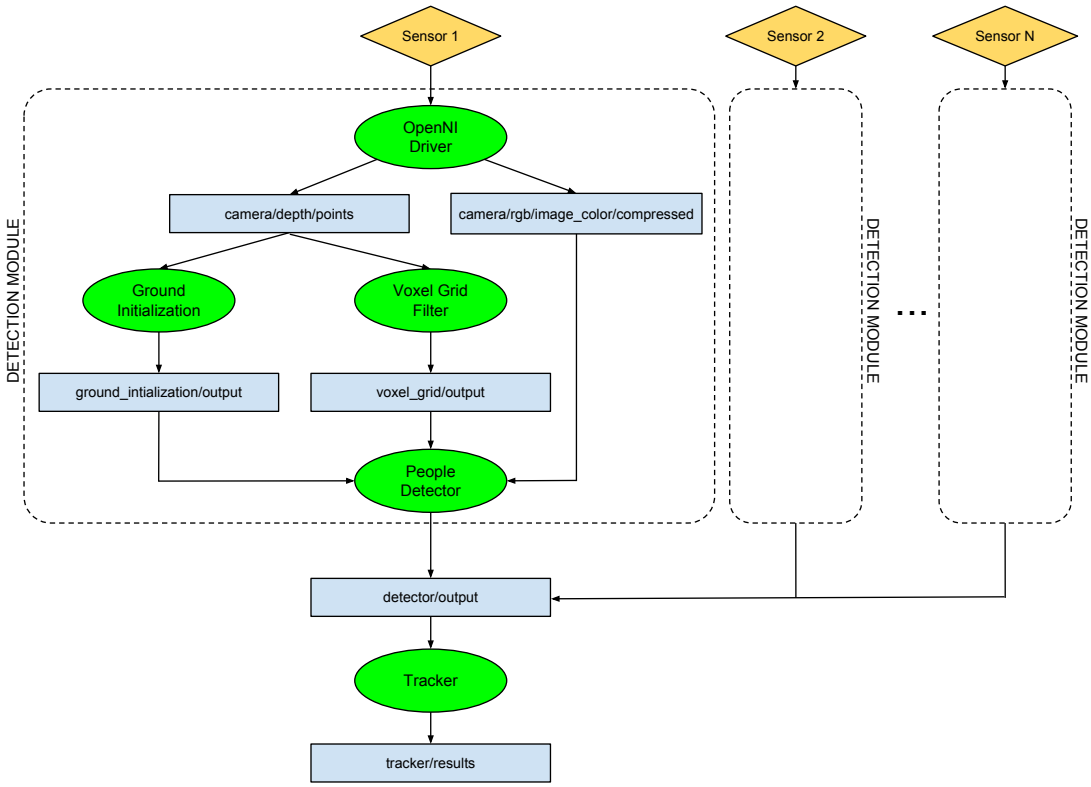


Figure 3.13: Block scheme of nodes (green) and topics (cyan) interconnections.

Following this policy, we structured our code as composed by five different nodes interconnected as shown in Figure 3.13. This figure is very similar to what can be obtained at runtime with the ROS tool `rxgraph`, but it has been re-drawn for better visualization with colors. Nodes are reported in green, topics in cyan and sensors in orange. `OpenNI Driver` and `Voxel Grid Filter` nodes come as part of ROS libraries and they are respectively used for acquiring data from a RGB-D sensor and for providing a voxelization of the point cloud as explained in Section 3.3.1. `Ground Initialization`, `People Detector` and `Tracker` nodes, instead, have been implemented for initializing the ground plane equation, perfoming people detection and tracking. It is worth noting that the ground initialization node is executed only once, thus it is usefully structured as a separated node that terminates after the ground plane has been initialized.

People detections are published to a particular topic (in this case named `/detector/output`) which is then read by the `Tracker` node, which has the task to perform prediction and data association. In a multi-sensor scenario, as in Figure 3.13, a detection module for each sensor is run and all people detections are published to the same output topic. Thus, detections can come from different sensors connected to different computers and they are all used by the `Tracker` node for performing people tracking.

Table 3.1: Comparison between nodes and nodelets versions of our people detection software in terms of framerate (fps) and at different depth resolutions on a CPU Xeon E3-1220 Quad Core 3.1 GHz.

	VGA	**QVGA**	**QQVGA**
nodes	14.5	21.7	28.1
nodelets	18.3	24.8	29.9

The main part of the algorithms performed by the described nodes is contained in callback functions, executed every time a new message is published to their corresponding input topic.

3.8.2 Nodelets for Avoiding Data Copying

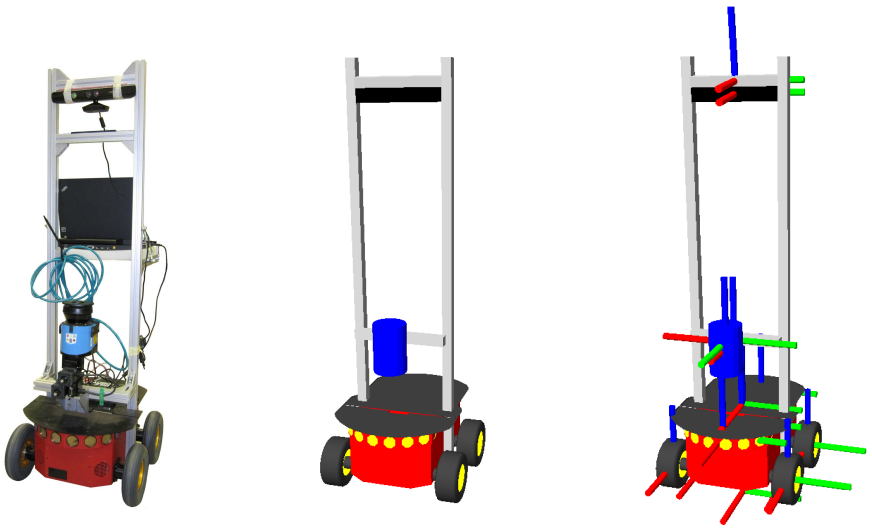
Every node in ROS runs in a different process, thus interprocess communication is needed in order to exchange data between nodes. That means that the computer has to spend more time and memory for passing data between them. For such a reason, as an alternative to nodes, nodelets have been designed to provide a way to run multiple algorithms in the same process with zero copy transport between algorithms.

We implemented a nodelet version of our detection module where we substituted the nodes with nodelets managed by the same nodelet manager. In Table 3.1, we report the framerate of the people detector for our two different implementations and for three different resolutions of the sensor depth point cloud. As it can be noticed, the nodelets implementation led to a higher framerate and the higher is the resolution, the higher is the framerate gain because we avoid to copy large amounts of data between different processes[9].

3.8.3 Easy Configuration with `tf`

Some of the most frequent errors when dealing with robots with multiple joints or with multiple sensors is related to wrong reference frames transformations. ROS `tf` package has been designed to easily refer data to multiple reference frames, that can vary over time, and maintains the relationship between coordinate frames in a tree structure buffered in time. This package is distributed, in the sense that there is no central source of information and all the coordinate frames are known to all ROS components on any computer in the system. Moreover, there is no loss in accuracy when transforming data multiple times. These transforms can be published or listened to with

[9] All the other tests reported in this work refer to the implementation based on nodes.



(a) Test platform. (b) URDF model in `rviz`. (c) URDF model and `tf` in `rviz`.

Figure 3.14: The robotic platform (a) used in the tests, its model displayed with the ROS visualizer (b) and the reference frames of every model joint (c).

a fixed rate over time and they can be visualized with both ROS visualizer (`rviz`), that shows their position in real time, and with the command `rosrun tf view_frames`, that saves to a file the full tree of coordinate transforms. In Figure 3.14, we report a picture of the robotic platform (a) we use for people tracking and following applications, together with its *Unified Robot Description Format* (URDF) model as it is visualized in `rviz` (b). In addition, also `tf` reference axes are drawn for every robot joint and every sensor (c). This kind of visualization allows to easily check the correctness of the reference frames. Moreover, `rviz` makes possible visualization with hardware in the loop, so that a direct comparison between the simulated and the real behavior of the robot can be performed.

3.9 Experiments

In this section, we present a number of tests we perfomed for validating our people detection and tracking technique and compare it to the best works in literature. Some of the datasets we used were publically available (RGB-D People dataset, ETH dataset), others have been collected as part of this work and made public (KTP dataset, IAS-Lab People Tracking dataset). The people tracking datasets we collected are described in Appendix B.1.

3.9.1 Evaluation Procedure

For the purpose of evaluating tracking performance, we adopted the CLEAR MOT metrics [12], that consists of two indices: MOTP and MOTA. The MOTA index gives an idea of the number of errors that are made by the tracking algorithm in terms of false negatives, false positives and mismatches, while the MOTP indicator measures how well exact positions of people are estimated. The higher these indices are, the better is the tracking performance. We computed the MOTA index with the following formula

$$MOTA = 1 - \frac{\sum_t (fn_t + fp_t + ID_t^{sw})}{\sum_t g_t} \tag{3.7}$$

where fn_t is the number of ground truth people instances (for every frame) not found by the tracker, fp_t is the number of output tracks instances that do not have correspondences with the ground truth, ID_t^{sw} represents the number of times a track corresponding to the same person changes ID over time and g_t is the total number of ground truth instances present in all frames. When evaluating tracking results with respect to the 2D ground truth referred to the image, we computed the MOTP index as the average PASCAL index [37] (intersection over union of bounding boxes) of the associations between ground truth and tracker results by setting the validation threshold to 0.5.

With the KTP dataset, we also compared people 3D position estimates with the 3D ground truth obtained with the motion capture system. For doing this, we computed the MOTP index as the mean 3D distance obtained considering only those results that are correctly associated to the ground truth. Since we were interested in evaluating estimates of people position within the ground plane indipendently from people's height estimates, we computed distances of only the x and y coordinates, disregarding the z. We considered an estimate to match with the ground truth if their distance was below 0.3 meters, which seemed to be a fair 3D extension of the area ratio threshold of 0.5 used for the PASCAL test.

3.9.2 Tests on the IAS-Lab People Tracking dataset

We present here some results obtained with our tracking system on the IAS-Lab People Tracking dataset (Appendix B.1.1), that has been collected in an indoor environment with the mobile robot shown in Figure 3.14 (a). In this work, odometry readings are used to refer people detections to a common (global) reference frame used by the tracking algorithm.

In Table 3.2, we report the MOTP and MOTA indices, the percentage of false posi-

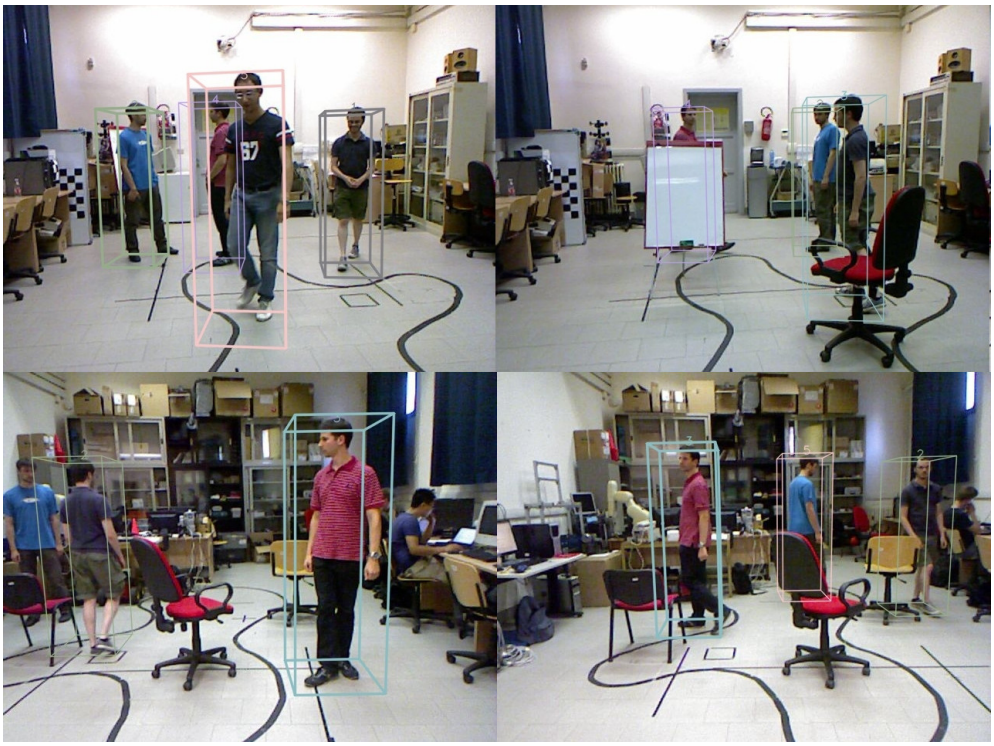


Figure 3.15: Tracking output on some frames extracted from the IAS-Lab People Tracking dataset.

Table 3.2: Tracking results for tests with the IAS-Lab People Tracking dataset.

	MOTP	**MOTA**	**FP**	**FN**	**ID Sw.**
Simple traj.	82.2%	95.8%	2.5%	1.6%	3
Complex traj.	83.5%	90.9%	4.7%	4.4%	1
With obstacles	83.3%	94.3%	4.7%	0.9%	3

tives and false negatives and the number of ID switches for the test sequences acquired with a moving robot. Even if depth data were at a reduced resolution (160 x 120 pixels) with respect to Kinect maximum resolution (640 x 480 pixels), our system obtained high tracking accuracy: the only ID switches are due to people who change motion direction when occluded by other people or outside the camera field of view. Since we use a constant velocity model to model human motion, these situations are hard to handle by our short-term tracking. In Figure 3.15, we report some examples of correctly tracked frames from our test set. Different IDs are represented by different colors and the bounding box is drawn with a thick line if the algorithm estimates a person to be completely visible, while a thin line is used if a person is considered occluded.

Table 3.3: Tracking results for the KTP dataset with different algorithms.

	MOTP	**MOTA**	**FP**	**FN**	**ID Sw.**
Full	84.24%	86.1%	0.8%	12.7%	60
No sub.	84.2%	83.02%	0.6%	15.9%	56
[7]	86%	82.41%	0.9%	16.1%	82

3.9.3 Tests on the Kinect Tracking Precision Dataset

In the following paragraphs, we report a detailed study we perfomed on the RGB-D videos of the KTP dataset (Appendix B.1.2), which allows to measure also 3D accuracy of people tracking algorithms thanks to a comparison with measurements obtained with a motion capture system. We computed results for every dataset sequence and we studied how different parameters and implementation choices can influence the tracking results. We will use the term *2D* tracking results when referring to tracking results computed on the image, while we will write *3D* tracking results when referring to the evalutation performed in 3D coordinates and described in Section 3.9.1.

On the first row of Table 3.3 we report the 2D tracking results we obtained on the KTP dataset with our full algorithm, a variant of our algorithm which does not use the sub-clustering method and the algorithm we presented in [7], which does not perform sub-clustering and uses a different data association procedure. Other than the MOTP and MOTA indices, we report also the false positive and false negatives percentages and the total number of identity (ID) switches, which represents the number of times tracks change ID over time. It can be noticed how the sub-clustering algorithm allowed to separate people very close (mostly present in the *Side-by-side* and *Group* situations), thus reducing the percentage of false negatives. It should be noted that some false positives for the *Group* video are generated by tracks positions not perfectly centered on the person. When not using sub-clustering, fewer detections were produced because people close to each other or touching were merged into a single detection, thus also less false positives were counted. For the same reasons, also slightly less ID switches were produced. The same difference in false negative rate holds when comparing the algorithm described in this work with the one in [7]. Moreover, with this work we obtain 14% less ID switches, because we introduced in the log-likelihood computation the confidence given by the appearance classifier that in [7] was used, instead, for a re-identification module decoupled from the motion estimate.

In Table 3.4, we report the 2D tracking results divided by video. It can be noticed that our algorithm reaches very similar tracking performances for static and moving videos, thus proving to be very good at dealing with robot planar movements.

Table 3.4: Tracking results for the KTP dataset divided by video.

	MOTP	**MOTA**	**FP**	**FN**	**ID Sw.**
Still	83.76%	88.86%	0.9%	9.8%	15
Translation	84.39%	88.03%	0.8%	10.7%	15
Rotation	84.19%	83.16%	0.9%	15.4%	19
Arc	84.89%	83.24%	0.8%	15.5%	13

Table 3.5: Tracking results for the KTP dataset divided by situation.

	MOTP	**MOTA**	**FP**	**FN**	**ID Sw.**
Back and forth	84.23%	89.17%	0.9%	9.9%	1(0)
Random walk	84.4%	86.42%	1%	12.3%	22(0)
Side by side	84.6%	79.12%	0.6%	20.2%	5(5)
Running	80.79%	90.44%	0.9%	8.7%	4(4)
Group	83.59%	63%	0.8%	35.8%	32(16)

In Table 3.5, we present results divided by type of sequence performed. As expected, we can notice that the worst result is obtained for the *Group* situation, where people are often visible only for a small portion. It is worth distinguishing between ID switches caused by an ID previously associated to a track and then to another and those generated when a person exits the scene and re-appears after several seconds. In this table, we reported the total number of ID switches and then we wrote inside the parenthesis the number of ID switches due to the latter reason. We can notice that the random walk sequences produce the highest number of ID switches of the first type. This fact can be explained because we use a constant velocity model to predict people position. This model results to work well for all the other situations, but performs worse in the *Random walk* situation because people abruptly change their direction. The constant velocity model is also not suited for avoiding ID switches of the second type, because it leads to a very low likelihood for people who exit the room going in one direction and re-enter the room walking towards the opposite direction. In Table 3.6, we report the same table, but using the 3D evaluation method. We can notice that MOTA changes considerably, increasing for the *Syde-by-side* and *Running* situation, but decreasing for the *Random walk* and *Group* ones. The *Group* situation, other than for very strong occlusions, is also challenging because three people enter the field of view of the camera from close to the robot and are not fully visible in the image for some seconds.

In Figure 3.16, we report also some examples of tracked frames from the KTP dataset. Different IDs are represented by different colors and the bounding box is drawn with a thick line if the algorithm estimates a person to be completely visible,

Table 3.6: 3D tracking results for the KTP dataset divided by situation.

	MOTP	**MOTA**	**FP**	**FN**	**ID Sw.**
Back and forth	0.196m	88.97%	2.4%	8.5%	1(0)
Random walk	0.171m	70.93%	9.8%	18.9%	20(0)
Side by side	0.146m	87.22%	1.2%	11.6%	5(5)
Running	0.143m	94.57%	1.1%	4.4%	4(4)
Group	0.181m	47.91%	9.1%	42.53%	26(16)

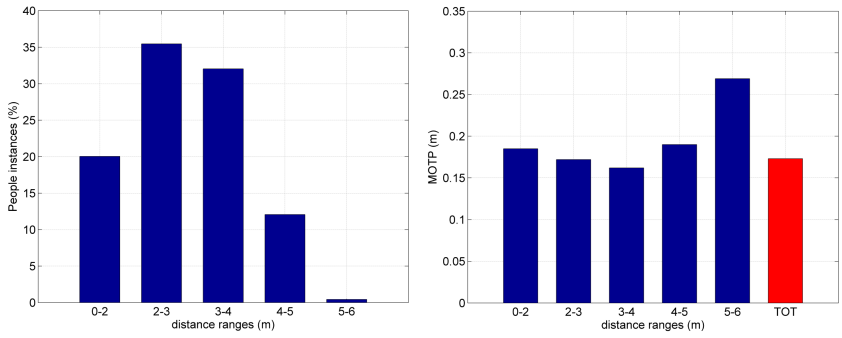
(a) Back and forth (b) Random walk (c) Side-by-side

(d) Running (e) Group

Figure 3.16: Example of tracking output for every situation represented in the KTP dataset. Different colors are used for different IDs. On top of every bounding box the estimated ID and distance from the camera are reported.

Table 3.7: Tracking results for the KTP dataset for different colorspaces.

	MOTP	**MOTA**	**FP**	**FN**	**ID Sw.**
RGB	84.24%	86.1%	0.8%	12.7%	60
HSV	84.22%	85.8%	0.7%	12.5%	53
CIELab	84.22%	86.48%	0.9%	12.2%	56
CIELuv	84.25%	86.72%	0.9%	12.9%	65



(a) (b)

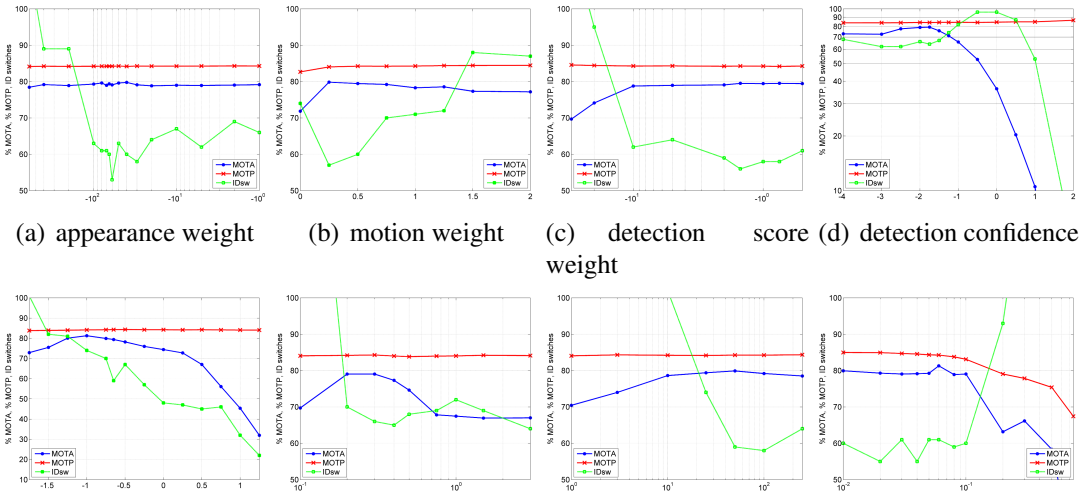
Figure 3.17: (a) Percentage of people instances for variuos distance ranges from the sensor and (b) MOTP value for every distance range.

while a thin line is used if a person is considered partially occluded.

We evaluated different colorspaces to be used for computing the color histogram of the clusters. As it can be seen in Table 3.7, the HSV space turned out to work better than RGB, CIELab and CIELuv, especially for reducing the number of ID switches.

The Microsoft Kinect estimates distances by means of a triangulation method. That means that its precision decreases with the squared distance from the camera. In order to analyze accuracy and precision of our tracking algorithm at various distances, we report in Figure 3.17 (a) the percentage of people instances and in Figure 3.17 (b) the 3D tracking results obtained on the KTP dataset for different distance ranges from the camera. If a person is too close to the camera, the head could be not visible, thus the tracking algorithm could produce a worse position estimate. For this reason, the optimal range of distances for people tracking with Kinect turned out to be of 3-4 meters. Under five meters, the mean tracking precision is below 20 centimeters, which is a fair localization error for robotics applications, while, over five meters, the MOTP rapidly increases, because Kinect depth estimates lose in accuracy.

In Figure 3.18, we report graphs of MOTA, MOTP and ID switches deriving from the 2D evaluation of our algorithm while varying its main parameters. The optimum is reached when MOTA and MOTP are maximum and ID switches are minimum. The (a-c) graphs show the effects of varying the weights given respectively to the color,



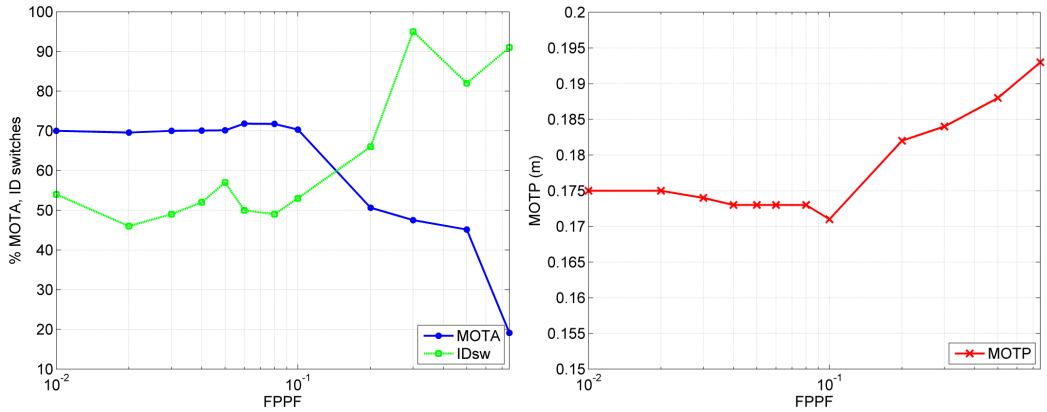
(a) appearance weight (b) motion weight (c) detection score weight (d) detection confidence

(e) min confidence for initialization (f) min distance for sub-clustering (g) selectors (h) voxel size

Figure 3.18: Performance of our algorithm on the KTP dataset when varying its main parameters. MOTA (blue) and MOTP (red) curves represent percentages, while ID switches (green) is the number obtained by summing up the ID switches for every video of the dataset. The optimum is reached when MOTA and MOTP are maximum and ID switches are minimum. In some of them, the logarithmic scale is used for the x or y axis for better visualization. Please, see text for details.

motion and detector likelihoods. It can be noticed how MOTP is almost invariant to these parameters, thus the main criteria for choosing the weights is to look at MOTA and ID switches. In (d), we report results for various values of the minimum confidence for people detection. Minimum confidences under -1.25 work well, while above this value the performance rapidly worsen because many people are missed. For what concerns the track initialization, a value between -1 and -0.5 seems to be the best choice. Below this value some false tracks are generated, above this value many people instances are missed. The results reported in (f) confirm that the intimate distance (0.3 meters) is the best choice for the minimum distance between people to be used in the sub-clustering method. If this value is too low, false positives are generated because a person can result splitted in many clusters. Instead, if using a too high threshold, people close to each other would remain merged in a single cluster.

In order to test our appearance classifier, we varied the number of features (selectors) that are chosen at every iteration from a pool of 250 weak classifiers for composing Adaboost strong classifier. It can be seen in (g) that selecting from 50 to 100 features from a pool of 250 gives the best tracking results. Given that the higher this number, the higher the computational cost, 50 has been chosen as the default parameter. As a further test for evaluating the effectiveness of using an online learning scheme



(a) (b)

Figure 3.19: 3D evalutation on the KTP dataset when varying the voxel size.

for computing the appearance likelihood needed for data association, we compared our results with a method which does not exploit learning and keeps the person classifier fixed after its initialization. With this approach, we measured a MOTA decrease of 7%, in particular due to an increased number of ID switches and missed people. Moreover, if we compute the appearance likelihood of a track given a detection as the Bhattacharyya distance between their global histograms, we obtain a 13% drop in MOTA, mainly due to an increase of ID switches with respect to our approach which exploits an online classifier.

Finally, an important trade-off between precision and computational speed must be taken into account for the choice of the voxel size. The higher this value is, the faster is the algorithm because less points have to be processed, but precision is lost in estimating people position. From the graph in (h) it can be noticed that the algorithm still performs well with a voxel size of 0.1 meters, then the performance degrades very quickly, this time involving also the MOTP index. Moreover, we show in Figure 3.19 how the voxel size impacts on the 3D tracking results. Even when evaluating the results in 3D coordinates, we could see that until a voxel size of 0.1 meters the obtained results are good, while, for bigger values, all the three indices rapidly get worse.

In Table 3.8, we summarize the parameters values of our detection and tracking algorithm which gave the best tracking results on the KTP dataset.

3.9.4 Tests on the RGB-D People Dataset

For the purpose of comparing with other state-of-the-art algorithms, we tested our tracking system on the RGB-D People dataset[10] ([109], [69]), that contains about

[10] `http://www.informatik.uni-freiburg.de/~spinello/RGBD-dataset.html`.

Table 3.8: Best parameters values resulting from our study on the KTP dataset.

Parameter	**Value**
Appearance weight	[30; 100]
Motion weight	0.5
Detection weight	1
Detection confidence	$<$ -1.25
Min confidence for initialization	[-1; -0.5]
Distance for sub-clustering	0.3
Number of selectors	50
Voxel size	$<$ 0.1
Colorspace	HSV

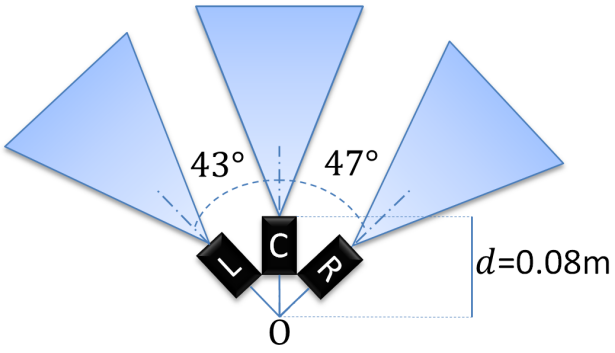


Figure 3.20: Sensory setup configuration for the RGB-D People dataset.

4500 RGB-D frames acquired from three vertically mounted Kinect sensors pointing towards adjacent, but not overlapping regions. Even if the exact position between the sensors is not provided in the dataset webpage, we deduced it to be as shown in Figure 3.20. For this test, we used three independent people detection modules (one for each Kinect), then detections have been fused at the tracking stage.

In Table 3.9, we report the results obtained with our default system against those obtained in [69], which uses GPU computation. A video with our tracking results can be found at this link: `http://youtu.be/b70vLKFsriM`. and in Figure 3.21 all the estimated trajectories are shown from a top view. Our MOTA and ID switches indices are 71.8% and 19, while for [69] they are 78% and 32, respectively.

For a correct interpretation of these results, the following considerations must be

Table 3.9: Tracking evaluation with RGB-D People Dataset.

	MOTP	**MOTA**	**FP**	**FN**	**ID Sw.**
Ours	73.7%	71.8%	7.7%	20.0%	19
[69]	N/A	78%	4.5%	16.8%	32

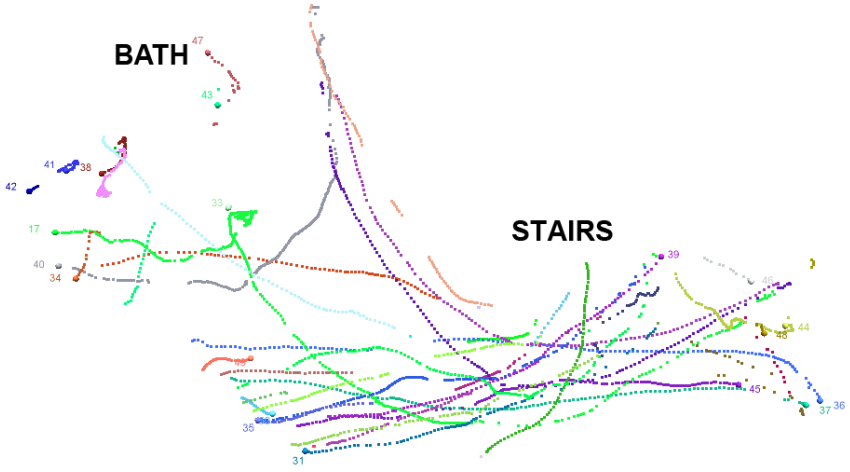


Figure 3.21: Top view of the resulting estimated trajectories for the RGB-D People dataset.

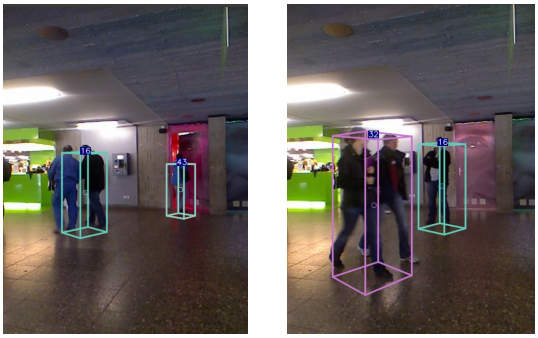
taken into account:

- half of our ID switches are due to tracks re-initialization just after they are created because of a poor initial estimation for track velocity. If we do not use the velocity in the Mahalanobis distance for motion likelihood computation the ID switches decrease to 9, while obtaining a MOTA of 70.5%;
- 10% of people instances of this dataset appear on the stairs, but tracking people who do not walk on a ground plane was out of our scope. It is then worth noting that half of our false negatives refer to those people, thus reducing to 10% the percentage of missed detections on the ground plane;
- in the annotation provided with the dataset some people are missing even if they are visible and, when people are visible in two images they are annotated only in one of these. Our algorithm, however, correctly detects people in every image they are visible. Examples of these kinds of annotation errors are reported in Figure 3.22. Actually, 90% of our false positives are due to these annotation errors, rather than to false tracks. Without these errors, the FP and MOTA values would be 0.7% and 78.9%. If we do not consider people on the stairs, the MOTA value raises to 88.9%.

Part of the success of our tracking is due to sub-clustering. In fact, if we do not use the sub-clustering method described in Section 3.3.2, the MOTA index decreases by 10%, while the ID switches increase by 17. In Figure 3.23, we report two examples of people merged together when not using the sub-clustering technique.

(a) Track 1 (b) Track 16

Figure 3.22: Examples of people missing in the ground truth of the RGB-D People dataset (first row) while detected by our algorithm (second row).



(a) (b)

Figure 3.23: Examples of people merged together when not using the sub-clustering method for the RGB-D People dataset.

Table 3.10: Computers of the wired network used for the distributed tests.

	PC1	PC2	PC3	PC4
CPU	Xeon Quad 3.10GHz	i5-520M 2.40 GHz	i7-620M 2.67Ghz	Core 2 Quad 2.66Ghz
RAM	4GB DDR3	4GB DDR3	4GB DDR3	4GB DDR3
Type	Desktop	Laptop	Laptop	Desktop

For this particular dataset the online classifier has not been very useful because most of the people are dressed with the same colors and Kinect auto-exposure function makes the brightness to considerably and suddenly change among frames.

Tests with a distributed tracking system

When doing people tracking from multiple cameras, it is useful to distribute the computational cost to multiple agents/computers. The advantage is two-fold: there is no need for a powerful computer as central unit and only selected data have to be sent over the network. If a software has been developed using ROS, distributing the computation among different machines becomes an easy task because a node makes no assumption about where in the network it runs, thus computation can be relocated at run-time to match the available resources.

In order to test our people tracking algorithm in a distributed configuration, we used the RGB-D People dataset. As sketched in Figure 3.13, we used three independent people detection modules (one for each sensor) that publish detection messages to the same topic and a tracking node that fuses them in tracks as they arrive. Detections coming from different sensors are referred to the same reference system by means of static transforms (`tf`). Data were pre-recorded in ROS `bag` files containing Kinect depth at QQVGA resolution and RGB at VGA resolution. We evaluated the tracking framerate of the whole process both when executed on a single computer and also when distributed on a heterogeneous computer network composed by PCs whose main specifications are reported in Table 3.10. PC1 and PC4 are desktop computers, while PC2 and PC3 are laptops suitable to be used on a mobile robot. As it can be seen in Table 3.11, the stream of a single Kinect of the dataset can be processed on PC1 at 28 fps by our detection module, while the whole detection and tracking algorithm runs at 26 fps. When processed on PC2, these framerates decrease at 23 and 19 fps respectively. When processing the whole RGB-D People dataset (three detection modules and one tracking node) only on PC2, the framerate decreased at 15 fps only. Therefore, we tested a distributed configuration where the three detection modules have been run on PC2, PC3 and PC4 and the tracking node have been executed on PC1. All the computers were connected within a gigabit ethernet network. In this

Table 3.11: Framerate of the detection and tracking modules with different test configurations.

	Detector (fps)	**Tracker (fps)**
Single stream on PC1	28	26
Single stream on PC2	23	19
Three streams centralized on PC2	20	15
Three streams distributed	58	31

configuration, we measured a framerate of 58 fps as the sum of the frames processed by the three detection modules in the three computers and 31 fps for the tracker node in the fourth computer. As a result, we can notice the doubling of the tracking framerate. Moreover, it is worth noting that the centralized processing of three Kinect streams has been possible because data were pre-recorded, while dealing with live acquisition in a centralized configuration could further decrease the tracking framerate or could not be possible due to bandwidth limitations of the USB bus on standard computers.

3.9.5 Tests on the ETH Dataset

In addition to the experiments we reported with Kinect-style RGB-D sensors, we also tested our algorithm on a part of the challenging ETH dataset [35], where data were acquired from a stereo pair mounted on a mobile platform moving at about 1.4 m/s in a densely populated outdoor environment.

In Figure 3.24, we show our tracking results on the *Bahnhof* sequence, which is composed of 1000 frames acquired at 14Hz at a resolution of 640 x 480 pixels. We compare our FPPF vs miss-rate DET curve with those of the state-of-the-art approaches already reported in [27]. We remind that the ideal working point would be in the bottom-left corner (FPPF = 0, FRR = 0%). In Figure 3.25 (a-b), we also show a qualitative result for two frames of this dataset. It is worth noting that depth data obtained from stereo are more noisy and present more artifacts with respect to those obtainable with Kinect-style sensors. Standard state-of-the-art algorithms highly rely on a dense scanning of the RGB image, thus being less sensible to bad depth data, while our approach processes only patches of the RGB image corresponding to valid depth clusters, thus being more dependent on the quality of depth estimates. Nevertheless, we obtain performance near that of the best state-of-the-art approach, doing better than [35] for FPPF less than 0.1. The algorithm which performs best is a variant of the method in [27] which uses the *Deformable Parts Model* [39] detector. Similar results are also obtained by the standard approaches in [27], [132] and [126]. However, the

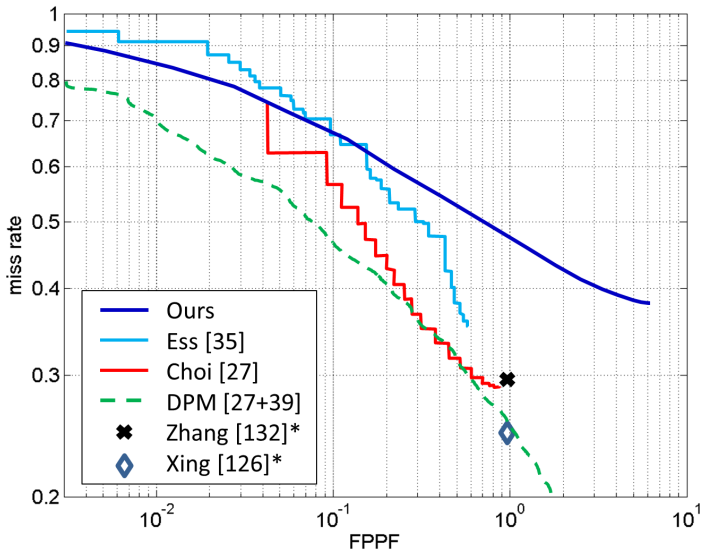


Figure 3.24: FPPF vs miss-rate curve showing tracking results on the *Bahnhof* sequence of the ETH dataset. The papers with * require all the images in a batch as an input.

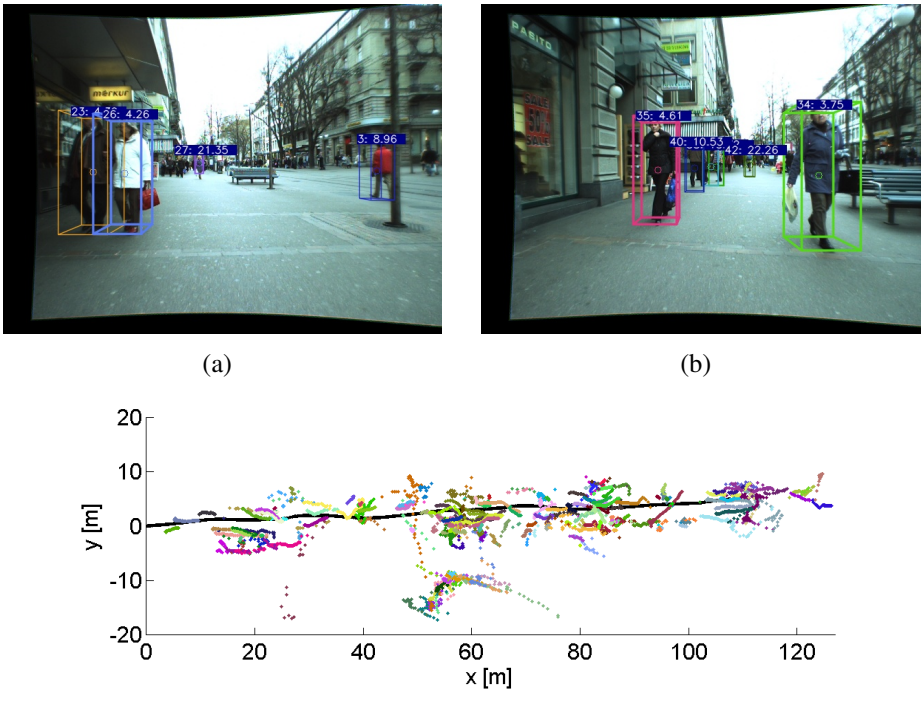


Figure 3.25: (a-b) Example of our people tracking results on the stereo data of the ETH dataset; (c) all estimated people trajectories (various colors) and the robot trajectory (in black).

latter two algorithms performs a global optimization of all the tracked frames, thus requiring all images in a batch as an input. Among all these methods, our approach is the only one which works in real time on a standard laptop CPU. In Figure 3.25 (c), we also reported the robot path given by the odometry (in black) and all people trajectories estimated by our algorithm.

3.9.6 People Following Tests

For proving the robustness and the real time capabilities of our tracking method, we tested the system on board of the service robot of Figure 3.14 (a) moving in crowded indoor environments. The robot's task was to follow a specific tracked person and its speed was only limited by the manufacturer to 0.7 m/s. We tested our whole system both in a laboratory setting and in a crowded environment at the Italian robotics fair *Robotica 2011*, held in Milan on the 16-19th November 2011. Our robot successfully managed to detect and track people within groups and to follow a particular person within a crowded environment by means of only the data provided by the tracking algorithm. In Figure 3.26, we report some tracking results while our robot was following a person along a narrow corridor with many light changes (first row) or when other people were walking near the person to follow (second row). Finally, also some tracking results collected at *Robotica 2011* fair are shown (third row), together with an external view of our robot following the person with the blue sweater (Figure 3.27).

3.9.7 Time Performance Analysis

Our system has been developed making use of highly optimized libraries for 2D computer vision [17], 3D point cloud processing [102] and bayesian estimation[11]. In Table 3.11, we reported the frame rates we measured for the detection algorithm and for our complete system (detection and tracking). The most demanding operations are Euclidean clustering (Section 3.3.1) and HOG descriptors computation (Section 3.4), which require 46% and 23% of time with QQVGA resolution, respectively. The tracking algorithm is less onerous since it occupies 8-17% of the CPU.

Our implementation does not rely on GPU processing, nevertheless, our overall algorithm is faster than other state-of-the-art algorithms such as [36], [73] and [27], respectively running at 3, 10 and 10 fps with GPU processing. This suggests that even a robot with limited computational resources could use the same computer for people tracking and other tasks like navigation and self localization. It is worth noting that also

[11] `http://bayesclasses.sourceforge.net/Bayes++.html`.

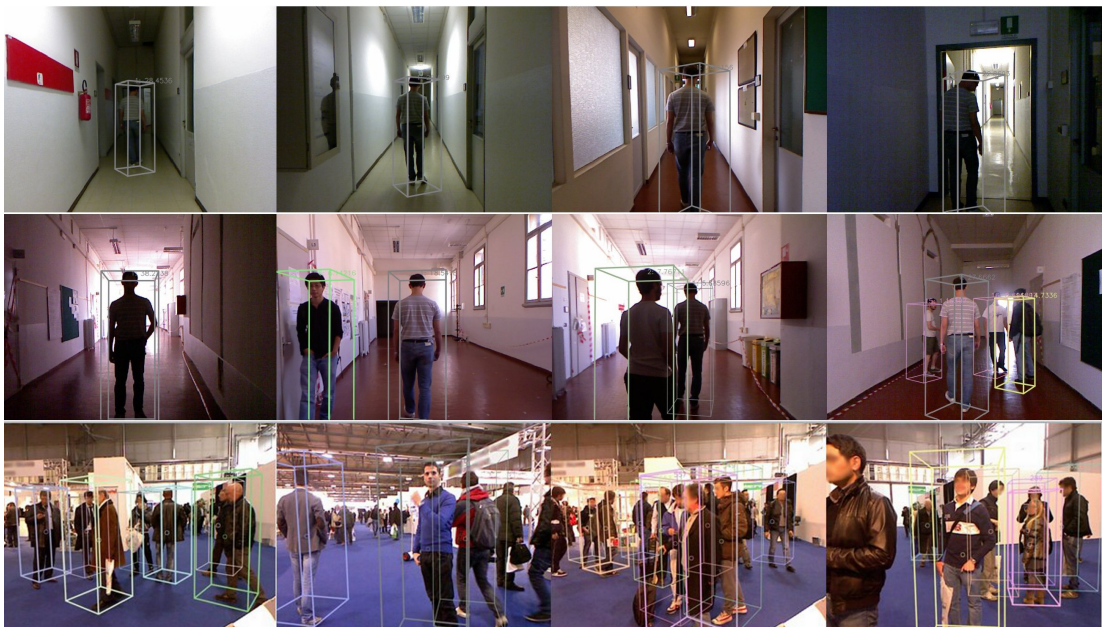


Figure 3.26: People following tests. First row: examples of tracked frames while a person is robustly followed along a narrow corridor with many light changes. Second row: other examples of correctly tracked frames when other people are present in the scene. Third row: tracking results from our mobile robot moving in a crowded environment. In these tests, the top speed of the robot was 0.7 m/s.

Figure 3.27: Sample images of our mobile robot following the person with the blue sweater within a crowded environment. It is worth noting that the sweater has the same color of the carpet on the floor, thus algorithms based only on color information would have easily failed.

our algorithm would benefit from GPU computing. For example, voxel grid filtering, Euclidean clustering and HOG descriptors computation, which take about 80% of the computation, are all highly parallelizable. These data refer to Kinect QQVGA depth resolution, while the frame rate is half that given in Table 3.11 when using Kinect VGA resolution (640 x 480 pixels).

We exploit ROS multi-threading capabilities and we explicitly designed our system for real time operation and for correctly handling data coming from multiple sensors, delays and lost frames. For testing these properties, we forced our system to process in real time the 3x30 Hz stream of the RGB-D People dataset on a single computer. Even if only 32% of the images could be processed (68% of frames was lost) it produced good results, in fact the MOTA index computed for those images was 69.3% with 24 ID switches.

3.10 Conclusions

In this chapter, we presented a fast and robust algorithm for multi-people tracking with RGB-D data designed to be used on mobile service robots. It can track multiple people with state-of-the-art accuracy and beyond state-of-the-art speed without relying on GPU computation. Morever, we perfomed tests on the newly introduced KTP dataset, the first RGB-D dataset with 2D and 3D ground truth for evalutating accuracy and precision of people tracking algorithms also in terms of 3D coordinates and we found that the average 3D error of our people tracking system is lower enough for robotics applications. From the extensive evaluation of our method on this dataset and on another public dataset acquired from three static Kinects, we demonstrated that Kinect-style sensors can replace sensors previously used for indoor people tracking from service robotics platforms, such as stereo cameras and laser range finders, while reducing the required computational burden.

Our people detection software has been publically released as part of the *people* module of the Point Cloud Library [102] and as part of ROS-Industrial *Human Tracker*[12], whose aim is to provide robust and fast people detection and tracking algorithms for industrial environments. Moreover, it served as a basis for OpenPTrack[13], an open source library for scalable and low-latency people tracking.

[12] `https://github.com/ros-industrial/human_tracker/tree/develop`.
[13] `http://openptrack.org`.

Chapter 4

Person Re-Identification

Person re-identification is the ability to recognize a person observed in the past or from a different point of view in a multi-sensor scenario. This is a high-level capability that is crucial in several fields including service robotics, intelligent video surveillance systems and smart environments. Unlike tracking, the person could have disappeared for a short or long timespan and motion information could be not reliable any more. Strong efforts have been spent by the research community to improve performance and reliability of re-identification algorithms, and several different techniques have been developed. These techniques are mostly grouped into short-term and long-term re-identification, based on the features which are used to recognize people. Short-term approaches usually assume that the person to re-identify is wearing the same clothes as at the time of the last observation, while long-term techniques should be able to recognize people after days, months or years, thus they have to rely on more permanent features of the human body and they cannot rely on clothes information.

In this chapter, we will propose two novel approaches to person re-identification which exploit data obtainable from consumer RGB-D sensors and the depth-based skeletal tracking algorithms described in Appendix A.

4.1 Short-Term Re-Identification

In this section, we will describe a novel methodology for short-term people re-identification based on skeletal information. Features are evaluated on the skeleton joints and a highly distinctive and compact feature-based signature is generated for each user by concatenating descriptors of all visible joints. We will compare a number of state of the art 2D and 3D feature descriptors to be used with our signature on two newly acquired public datasets for people re-identification with RGB-D sensors

described in Appendix B.2. Moreover, we will test our approach against the best re-identification methods in the literature and on a widely used public video surveillance dataset. As we will see, our approach is robust to strong illumination changes and occlusions. It achieves very high performance also on low resolution images, overcoming state of the art methods in terms of recognition accuracy and efficiency. These features make our approach particularly suited for mobile robotics.

Section 4.1.1 reviews existing work, Section 4.1.2 gives an overview of our method, and Section 4.1.3 introduces the skeletal tracking algorithms we exploit. In Section 4.1.4, feature descriptors are presented and matching methods are outlined in Section 4.1.5. In Section 4.1.6, the new signature we propose is described and similarity metrics are introduced in Section 4.1.7. The extension from single-frame to multi-frame evaluation is defined in Section 4.1.8, while experiments are reported in Section 4.1.9 and Section 4.1.10 contains conclusions.

4.1.1 Related Work

People re-identification in images is addressed observing three main characteristics: color, texture and shape, either considered singularly or mixed together. Color undergoes clear changes from person to person, and is usually measured by means of global or partial histograms. A color-based state-of-the-art approach divides the body of each target into smaller parts and evaluates multiple histograms, one for each part [25, 38]. This method is simple and effective, but suffers from two main flaws: it fails upon strong illumination changes, and it is a global (or semi-global) method that is not able to describe the target in detail.

Texture-based and shape-based approaches usually make use of local features: they exploit descriptors evaluated on a set of keypoints to generate the signature of a target. Performance are therefore strongly related to the characteristics of the set of descriptors selected, including the capability of the keypoint detector to select stable features. This approach is widely used in the literature [8, 53, 130] thanks to its superior capability of providing a detailed description of each target; moreover, it overcomes the two main drawbacks of the color-based approach previously discussed. Such approach has also been used together with histograms [47].

Very recently, computer vision for robotics was revolutionized by the introduction of affordable high-resolution three-dimensional sensors, that generate color point clouds instead of images. This had a strong impact on a number of applications, including people re-identification: for example, approaches based on three-dimensional features were developed. This new type of features can include information about both

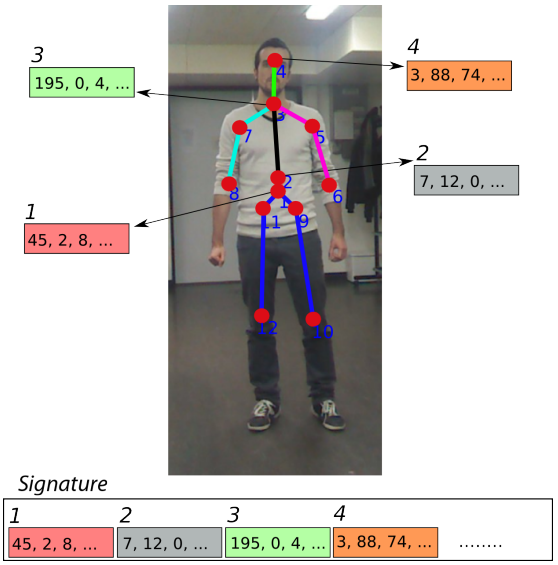


Figure 4.1: An overview of our approach to re-identification. Feature descriptors are computed around human skeleton joints, which serve as keypoints. These descriptors are then concatenated to obtain the whole person signature.

color and shape, which can provide superior performance over standard 2D features; however, noise affecting the location of the point cloud elements is usually not negligible for low-cost sensors like the Microsoft Kinect: in this case, shape descriptors can provide performance worse than expected, thus global approaches are usually preferred [5, 6, 90].

4.1.2 System Overview

In this section, we present a novel three-dimensional feature-based re-identification system capable of providing superior performance while preserving computational efficiency, that is crucial for real-time applications. Our approach is particularly suited for robotics applications, that have strong requirements on computational load and processing time. Moreover, it makes use of a sensor that is often available on robotic platforms, and is already used by a large number of people in the research community. We demonstrate that our system is able to exploit the high amount of information provided by RGB-D sensors to achieve better performance with respect to other state of the art approaches.

As discussed above, several feature-based approaches presented in the literature compute the signature of a target by finding keypoints and evaluating the features, either on 2D or 3D data, which is a general-purpose technique that works for any type of target.

Differently, we base our approach on the assumption that the target is a human, whose body has a well-known structure. Building on such hypothesis, we exploit a skeletal tracker to evaluate the body pose, and use this information to guide the keypoint selection, as shown in Figure 4.1, and compute the proposed *Skeleton-based Person Signature* (SPS). A small number of points are found at specific body positions: head, neck, shoulders, elbows, hands, ankles, knees, feet. Only 15 or 20[1] keypoints need to be evaluated and compared for every target: this dramatically reduces the computational load and the amount of memory for storing target models. The small number of keypoints is compensated by their high stability: since their positions come from a high-level knowledge of the target shape, their variance is extremely low compared to the typical values achievable with low-level keypoint detectors.

We thoroughly studied and tested our novel skeleton-based approach. Several state of the art 2D and 3D features were considered, leading to a mixed 3D-2D approach, which provides the best performance: the skeleton is evaluated from 3D data, while the signature is obtained from 2D descriptors. The presented system is meant to provide the best performance on high-resolution RGB-D data; however, benchmark datasets available for testing re-identification algorithms are often made of low-resolution 2D images. We collected the IAS-Lab RGBD-ID dataset (AppendixB.2.2), a novel dataset for testing our approach, acquired using a Microsoft Kinect installed on a robot, so that people are seen from the typical perspective used in robotics applications. This dataset highlights the system performance in the targeted working conditions. We also performed further experiments on a part of the BIWI RGBD-ID dataset (AppendixB.2.1), which contains a high number of people, and we validated our algorithms with respect to previous approaches in the literature: to do so, we chose the CAVIAR4REID video-surveillance dataset [25], that is widely used in the literature. Since our approach is based on a skeletal tracker, we added the joints to the dataset, applied our method, and compared our approach to others in the literature, showing it advances the state of the art also in this case. As a further contribution to the research community, we released both the new dataset and the skeletal joints of the CAVIAR4REID dataset.

Our re-identification approach was designed to be accurate and fast at the same time: for this reason, it is suitable for being included in real-time people tracking systems used in robotics, offering a superior performance with respect to appearance classifiers commonly exploited in tracking systems.

The proposed re-identification technique characterizes each target based on several feature vectors, evaluated on a set of keypoints. We call *keypoints* the salient points

[1] Depending on how many joints are estimated by the skeletal tracking algorithm.

taken as reference to characterize the target, i.e. the locations where the target is observed. A *descriptor* is a vector that describes the characteristics (like color, shapes, etc.) of a keypoint neighborhood, and is the output of a feature extraction algorithm. The number and type of characteristics taken into account, as well as the size of the neighborhood, depend on the specific feature. Finally, the *signature* is the data that describes the whole target, normally obtained by concatenating all descriptors in an organized way.

The two main components of our method are the skeletal tracker, that determines the keypoint location, and the type of feature chosen for characterizing them. Both elements have a strong impact on the re-identification performance, and have been deeply studied in this work.

4.1.3 Skeletal Tracker

The stability of the keypoints is extremely important to obtain similar descriptors in different observations of the same target, thus ensuring a reliable re-identification. Dealing with human people is very different than working with objects, since the human body is very flexible and deformable, and usually presents a number of different texture and shape patterns on the different body parts.

To properly describe the body shape and pose we adopted a skeletal tracker, that is an algorithm capable of understanding how the human body is placed; its output is a set of points, called *joints*, that represent a small subset of the real joints of a body.

In our work, we exploited two skeletal trackers, that are the most commonly used in computer vision and robotics and that are described in Appendix A: one is provided by Microsoft as a complement to its Kinect sensor and will be called Kinect skeletal tracker [107]; the other is included in the OpenNI SDK, is implemented inside the NiTE Middleware and will be called NiTE skeletal tracker. Both trackers work on 3D data, which ensures an accurate description of the body shape and superior performance over the 2D-based approaches.

4.1.4 Descriptor Evaluation

Once keypoints have been located by the skeletal tracker, features are exploited to characterize their appearance in terms of texture, color and three-dimensional shape. Features considered in this work were taken from the most recent literature in the fields of robotics and computer vision; both 2D and 3D variants were considered, because they offer advantages and disadvantages.

Generally speaking, 3D features are often more computationally expensive to evaluate and noisy, because the additional noise source affecting the point position in a point cloud is not negligible when low-cost sensors like the Microsoft Kinect are exploited for acquisition. On the other hand, 3D features are obviously highly descriptive and provide richer information. For a more detailed discussion we refer to the literature presenting keypoint and feature detectors that is reported later in this section.

Dealing with features, the radius on which they are evaluated is an important parameter that has a strong impact on the performance. For 3D features, such radius R is a metric value which can be chosen once and kept constant, while in the image domain the feature radius r is measured in pixels and cannot be fixed because the size of a person in the image varies depending on its 3D position. A great advantage offered by our 3D-based approach is the capability of relating r expressed in pixels to R expressed in world coordinates and fixed for all targets in all views: this sensibly enhances the feature stability, because the same volume is always considered for evaluating the features. This is possible because the relationship between the two radii r and R can be expressed by:

$$r = f \times \frac{R}{z}, \tag{4.1}$$

where z is the distance of the target to the sensor plane and f is the sensor focal length.

The implementations chosen for evaluating the features discussed in the following are the standard ones provided by the widely diffused OpenCV[2] [17] and PCL[3] [102] libraries, using default parameters when available.

2D Descriptors

The 2D features considered can be split into two categories based on the data type composing the descriptor vector: either real or binary. The first category includes SIFT and SURF while BRIEF, ORB and FREAK are examples of binary descriptors.

SIFT was presented by Lowe [68] in 1999. It is a keypoint detector and a descriptor invariant to image scale and rotation and also robust to changes in illumination, noise, and minor affine transformations. It is computed as a 8-binned histogram of gradient distribution within the region around each keypoint. The descriptor is normalized to unit length to reach illumination invariance.

[2] `www.opencv.org`
[3] `www.pointclouds.org`.

SURF was developed by Bay et al [9] in 2006, is invariant to image scale, illumination changes and orientation, and is partially inspired by SIFT descriptor. To compute the descriptor vector, the horizontal and vertical Haar-wavelet responses are computed and summed within a region around the keypoint. The absolute values of the responses are also summed and concatenated. The illumination invariance is obtained normalizing the descriptor to unit length.

BRIEF was presented in 2010 by Calonder et al [21], is composed by a binary string. Each bit of the vector is the result of an intensity test on a pair of a particular patch centered on the keypoint. The final descriptor is the concatenation of the results of N intensity tests.

ORB is the evolution of Calonder's BRIEF algorithm. Since BRIEF is not invariant to image rotation, Rublee et al [97] in 2011 suggested a method to overcome this drawback. To add the rotation invariance, the keypoint is described according to its orientation. Then, the intensity tests are computed similarly to the BRIEF algorithm on an oriented patch.

FREAK was introduced by Alahi et al [1] in 2012. Intensity tests are computed over a retinal sampling pattern, that is circular and has a higher density of points and a minor gaussian smoothing near the center of the pattern. Similarly to BRIEF and ORB, the final descriptor is obtained by concatenating the results of the intensity tests.

3D Descriptors

The 3D features considered are computed on a neighborhood of the keypoints that belong to a point cloud. Some features only consider the shape of such neighborhood, while others also exploit color information.

PFH was introduced by Rusu et al [99]. It is invariant to the 6D pose of the surface. It aims at characterizing a keypoint based on its k-nearest neighbors by considering the relationships between all point pairs and their estimated surface normals, that generate a quadruplet with angular features. The PFH descriptor is then obtained as a histogram of such quadruplets.

PFHRGB adds color information to the standard PFH descriptor. Including chromatic information increases the robustness of the descriptor, but has stronger compu-

tational requirements.

FPFH is another variant of PFH, developed by Rusu et al [98] in 2009. The goal of this descriptor is to reduce the computational complexity of PFH. A quadruplet of angular values is computed between the keypoint and its k-nearest neighbors (not between all pairs of neighbors). The final descriptor is obtained similarly to PFH.

SHOT was proposed in 2010 by Tombari et al [116]. It defines an isotropic spherical grid centered on the keypoint and computes a histogram of normals for each sector of the grid. The final descriptor is obtained by grouping all the histograms into a vector and normalizing it to unit norm.

SHOTRGB enriches SHOT descriptor by adding chromatic information. It was developed by Tombari et al [117] in 2011. The descriptor is a concatenation of two histograms, obtained from the points shape and the color channels within a spherical grid around the keypoint. This descriptor is also normalized to unit norm.

4.1.5 Matching Methods

The re-identification task is achieved by comparing the signatures of each target found in the test frame with those found in the training frames; the best-matching one is finally selected. We propose two ways of considering the skeleton joints in the matching phase: one considers tracked joints only, the other all of them.

Tracked Joints Matching Method

The first proposed method works by matching the reliable joints only, i.e. those that are in the tracked state in the test frame, and is therefore named TJ (Tracked Joints). This method achieves high performance, as it will be discussed in Sec. 4.1.9 because it relies on visible and stable joints only. Additionally, this method deals with almost all frames in each dataset, because it does not require the complete skeleton to process a frame, but rather, is able to select the part that has been reliably detected. The only exception is when the target is partially outside the field of view, because the whole skeleton is poorly detected.

All Joints Matching Method

The second approach is called AJ (All Joints) because it considers also the unstable keypoints – recall that all of them are always located by both skeletal trackers considered in this work. The performance of this approach is lower with respect to TJ because unreliable keypoints are considered in the matching. This approach is nevertheless proposed to test our system on the CAVIAR4REID dataset, in which training and testing frames are often acquired from different viewpoints. This represents a special case, because some joints that are detected in the test frames are never seen in the training set, and would therefore never be matched, thus reducing the number of joints actually usable: the AJ matching method is therefore used for recovering this situation.

When the AJ matching is used, an additional operation is performed. If a joint is not tracked, its descriptor is replaced with the descriptor computed at its symmetric joint, if this one is tracked. This change relies on the assumption that descriptors at symmetric joints are similar, thus this replacement can increase recognition performance. For instance, if the right hand is not tracked while the left hand is tracked, the descriptor around the left hand is used in place of that of the right hand. This obviously applies to all symmetric joint couples.

4.1.6 Skeleton-based Person Signature

The signature we propose for describing a target is called Skeleton-based Person Signature (SPS). It is built according to the matching method adopted, therefore two versions of the signature were developed: one evaluated only on the tracked keypoints, the other on all of them. In the first case we first define the feature tracking indicator for the i-th joint J_i on the k-th target T_k as:

$$I(J_i, T_k) = \begin{cases} 1 & \text{if } i\text{-th joint of frame } k \text{ is tracked} \\ 0 & \text{otherwise} \end{cases}, \tag{4.2}$$

where $i \in [0, \ldots, N-1]$ and N is the number of joints considered; note that Kinect skeletal tracker and NiTE skeletal tracker automatically provide $I(J_i, T_k)$. The signature on tracked joints $\mathbf{SPS}^{\mathrm{TJ}}$ of target k is obtained as:

$$\mathbf{SPS}_k^{\mathrm{TJ}} = \bigcup_{i=0}^{N-1} \{D(J_i, T_k) : I(J_i, T_k) = 1\}, \tag{4.3}$$

where $D(J_i, T_k)$ is the descriptor obtained evaluating the selected feature on the i-th joint for target k. Following (4.3), the signature is therefore obtained concatenating the descriptors of all tracked joints. The second version of the signature, built considering all joints, is analogously defined by simply omitting the feature tracking indicator:

$$\mathbf{SPS}_k^{\mathrm{AJ}} = \bigcup_{i=0}^{N-1} \{D(J_i, T_k)\} . \tag{4.4}$$

4.1.7 Similarity Metrics

The algorithms used at low level for comparing features are the standard methods commonly used, and already available in the libraries OpenCV and PCL we exploited for computing the features. In particular, two similarity metrics are exploited at this stage, depending on the data type of the feature descriptor, that can be composed either by real values, or binary ones. In the first case, the Euclidean distance is used, defined as:

$$d_{\mathrm{E}}(T_i, T_j) = \sqrt{\sum_{m=0}^{L-1} \left[\mathbf{SPS}_i(m) - \mathbf{SPS}_j(m)\right]^2}, \tag{4.5}$$

where $\mathbf{SPS}_i(m)$ is the m-th element of the signature, and $\mathbf{SPS}_i$ stands either for $\mathbf{SPS}_i^{\mathrm{TJ}}$ or $\mathbf{SPS}_i^{\mathrm{AJ}}$, depending on the type of signature to be considered; finally, L is the number of descriptors available in the signature, that can be smaller than the number of joints if $\mathbf{SPS}_i^{\mathrm{TJ}}$ is employed.

In the second case, binary features provide descriptors that are bitstrings, and we compare them using the Hamming distance:

$$d_{\mathrm{H}}(T_i, T_j) = \sum_{m=0}^{L-1} \left[\mathbf{SPS}_i(m) - \mathbf{SPS}_j(m)\right]^2 . \tag{4.6}$$

4.1.8 Single-Frame vs Multi-Frame Re-Identification

As discussed in the previous section, re-identification means associating a target found in the current frame (test set) with others observed in the past, or a set of pre-recorded examples (training set). We already discussed how to classify a new test frame by matching it with the training frames.

When multiple frames of a person are available in the testing set, they can be jointly exploited to provide a more robust classification. In [25], the authors propose a multi-shot modality which, for comparing M probe signatures of a given subject against N gallery signatures of another one, simply calculates all the possible $M \times N$ single-

shot distances, and keeps the smallest one. This approach does not offer an efficient scalability over the number of frames. The computational time grows over time since an increasing number of frames are considered for finding the best match.

Since our purpose is to use our re-identification approach in a real time scenario, we perform single-frame classification and adopt a voting scheme that associates each test sequence to the subject voted by the highest number of frames. This approach merges single-frame results into a sequence-wise result without adding further computational costs and leading to considerable improvements (10-30%) in the recognition rate, as we will see in Sec. 4.1.9.

4.1.9 Experiments

For validating the approach we described in Sec. 4.1.2, we performed a number of experiments on publically available datasets presenting different challenges related to the re-identification task. In particular, the BIWI RGBD-ID dataset (Appendix B.2.1) is targeted to people re-identification from a robot point of view when the training set is composed by many people.

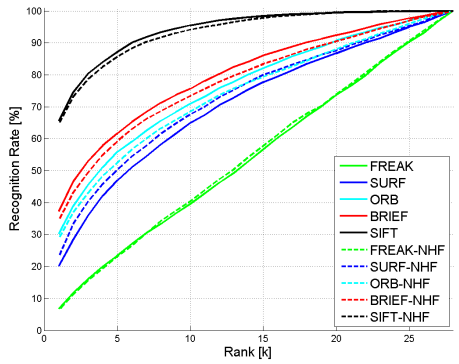
The novel IAS-Lab RGBD-ID dataset (Appendix B.2.2) presents strong illumination changes because training and testing sets were acquired in different rooms; finally, the CAVIAR4REID dataset[4] is made of very low resolution images and contains occlusions, considerable pose changes between training and testing set and a high number of people since it is targeted to people re-identification in a video surveillance scenario.

For evaluation purposes, we compute *Cumulative Matching Characteristic (CMC) Curves* [42], which are commonly used for evaluating re-identification algorithms. For every k from 1 to the number of training subjects, these curves express the mean person recognition rate computed when considering a classification to be correct if the ground truth person appears among the subjects who obtained the k best classification scores. The typical evaluation parameters for these curves are the *rank-1* recognition rate and the *normalized Area Under Curve* (nAUC), which is the integral of the CMC. In this work, the recognition rates are separately computed for every subject and then averaged to obtain the final recognition rate.

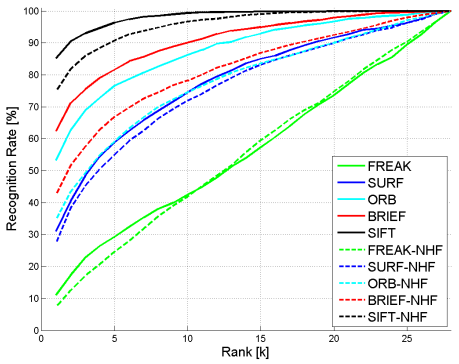
Performance Analysis on the BIWI RGBD-ID Dataset

The BIWI RGBD-ID dataset (Appendix B.2.1) was originally targeted to long-term re-identification, thus people wear different clothes in the training video with respect

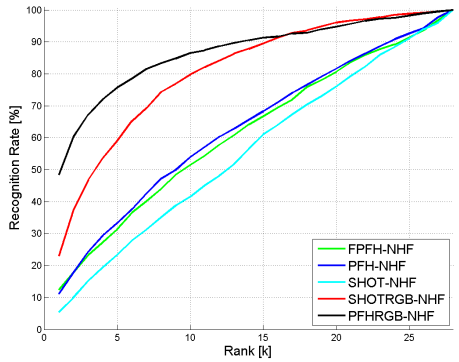
[4] `http://www.lorisbazzani.info/code-datasets/caviar4reid`.



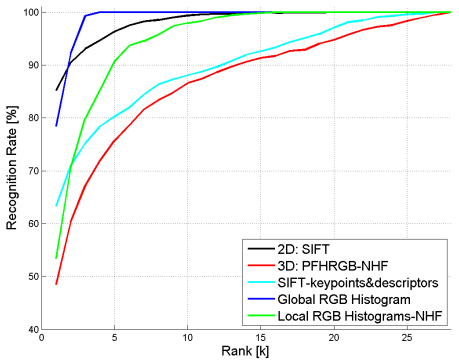
(a) 2D features - all frames



(b) 2D features - fully tracked skeletons



(c) 3D features - fully tracked skeletons



(d) Main tested approaches - fully tracked skeletons

Figure 4.2: Re-identification results on the BIWI RGBD-ID dataset.

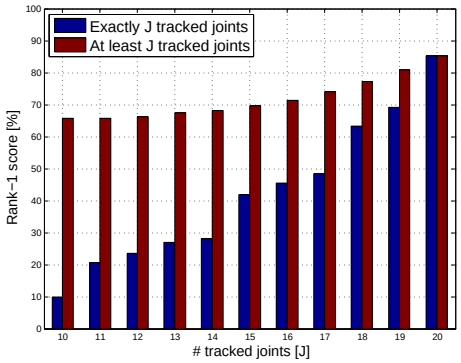
to their two testing sequences. For our purposes, we used only the sequences in which people are wearing the same clothes. In particular, for every person, we exploited the testing video where the person is still as our training set and the testing video where the person is walking as our testing set. Thus, our training set was composed of 28 people and all the people in the testing set were also present in the training set.

In Figure 4.2 (a), we report the CMCs we obtained on this dataset with the $\mathbf{SPS}_k^{\mathrm{TJ}}$ signature computed exploiting the five 2D descriptors we tested and exploiting the TJ matching approach. The solid curves refer to a SPS which concatenates descriptors at all the 20 skeleton keypoints, while the dashed curves are obtained by removing from the signature the keypoints at wrists, hands, ankles and feet, that are the joints which are more often misplaced by the skeletal tracker. We labeled this approach *No Hands and Feet* (NHF). As it can be noticed, these two configurations lead to similar results, with SIFT obtaining the best rank-1 (65.67%) and nAUC (94.26%), followed in order by BRIEF, ORB and SURF. Unlike the others, the FREAK descriptor led to random results, with a nAUC around 50%, thus showing not to be discriminative for the re-

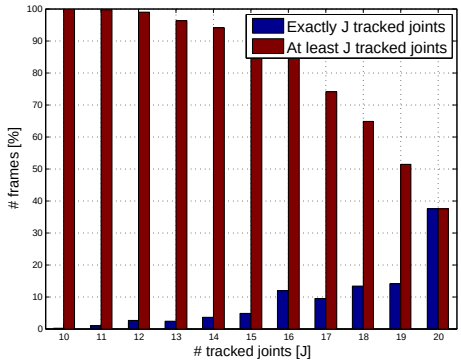
identification task. These results are obtained performing re-identification on nearly all the frames where a skeleton is provided[5], with a number of tracked joints ranging from 10 to 20 for the solid curves. A complete analysis of the re-identification performance when varying the number of tracked joints (and thus of skeleton keypoints) is reported in Figure 4.3. In particular, we report the rank-1 score when performing re-identification on only those frames for which *at least* J joints are tracked and when *exactly* J joints are tracked. As expected, this score increases when exploiting more skeleton keypoints in our SPS signature. However, the more joints are requested for computing the signature, the fewer frames are considered valid for performing re-identification. For understanding the trade-off between rank-1 and number of analyzed frames, in Figure 4.3 (b), we report the percentage of *valid* frames vs the number of tracked joints for our testing set from the BIWI RGBD-ID dataset. Moreover, Figure 4.3 (c) shows the number of frames in which each skeleton joint is tracked. It can be noticed that, if re-identification is only performed when all the 20 joints are tracked, only 37% of the frames with a skeleton are analyzed. That is the case reported in Figure 4.2 (b), where we can notice that rank-1 scores are considerably higher than when all frames are considered, and of Figure 4.2 (c), where 3D descriptors are computed at skeleton keypoints given by the NHF approach, because bad keypoint localization at hands and feet could easily lead to singularities in 3D descriptors. As expected, descriptors only encoding 3D local shape (PFH, FPFH, SHOT) achieve random performance, because of the strong noise affecting point clouds when consumer depth sensors are employed. Instead, descriptors encoding both shape and color achieve good performance, even if lower than those obtained with 2D descriptors. In particular, PFHRGB obtains the best rank-1 (48.49%) and nAUC (87.01%), while SHOTRGB stops at a rank-1 of 23.01% and a nAUC of 81.14%.

In Figure 4.2 (d), we compare the best 2D (SIFT) and 3D (PFHRGB) approaches of Figure 4.2 (b) and (c) with other methods widely used in literature. In order to validate our choice of skeleton joints as keypoints, we also reported the CMC obtained when selecting keypoints with the standard SIFT keypoint detector and then matching the SIFT descriptors for re-identification, as often done for object recognition. It can be noticed how our approach to keypoint selection allows to obtain a rank-1 20% higher while also avoiding the feature matching step, which is the process of finding corresponding features among training and testing descriptors, thus saving a considerable amount of time, as we will see in Sec. 4.1.9. We also compare these approaches to the use of color histograms, which are highly used in people re-identification litera-

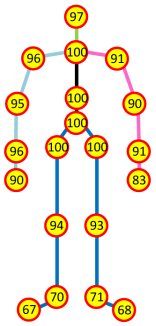
[5]We only discard those frames where the person is partially out of the image to the right or left or where the person is farther than 3.5 m because the skeleton is poorly estimated in these conditions.



(a) Rank-1 score VS number of tracked joints



(b) Number of frames VS number of tracked joints



(c) Tracking percentage for every joint

Figure 4.3: (a) Re-identification results and (b) number of frames of the BIWI RGBD-ID dataset when varying the number of tracked joints. No frames contain a skeleton with less than 10 tracked joints. In (c), we report the percentage of frames in which each joint is tracked.

ture [25,38], matched with the Bhattacharyya distance [14]. The best result is obtained by computing a global RGB histogram on all points belonging to a target, while a lower result is achieved when concatenating local RGB histograms extracted from each body part. However, such performance is still lower in terms of rank-1 score to our best SPS signature exploiting SIFT descriptors.

Robustness to Illumination Changes on the IAS-Lab RGBD-ID Dataset

The IAS-Lab RGBD-ID dataset (Appendix B.2.2) consists of 33 sequences of 11 people acquired using the OpenNI SDK and the NiTE skeletal tracker. For every subject, the *Training* and *TestingB* sequences were collected in different rooms, with strong illumination changes caused by the different auto-exposure level of the Kinect in the two rooms. Given that the NiTE skeletal trackerdoes not provide reasonable skeletons when some joints are not tracked, we only performed re-identification when all the 15 joints are considered to be tracked. Since for this dataset we obtained very similar results when considering all the 15 joints and when using the NHF approach, we report here the results for the NHF approach, because it is more computationally efficient since it computes a lower number of descriptors[6]. In Figure 4.4, we show the CMCs obtained with the main 2D and 3D descriptors we tested and with the approach we explained in Sec. 4.1.9 which computes a global RGB histogram for every person. As it can be noticed, SIFT is again the best descriptor, with a rank-1 very close to 100%. BRIEF and ORB also obtain very good results, directly followed by the PFHRGB descriptor. Unlike in the BIWI RGBD-ID dataset, the color histogram approach obtains poor results on this dataset, probably because of the strong illumination changes, while our texture-based approach maintains high performance.

Multi-Frame Results

In Table 4.1, all re-identification results are reported for all the approaches and datasets we tested in this work. The multi-frame score presented in Sec. 4.1.8 is also reported for the BIWI RGBD-ID dataset and the IAS-Lab RGBD-ID dataset. It can be noticed how, if we consider this sequence-wise result, our SPS signature coupled with the SIFT descriptor allows to re-identify all the people of the two tested datasets (R1-Multi $= 100\%$) and in general the re-identification percentage is 10-30% higher than the rank-1 computed for the single-frame re-identification.

[6]We recall that the NHF approach removes hands and feet, thus considering 12 joints for the Kinect skeletal tracker and 11 joints for the NiTE skeletal tracker.

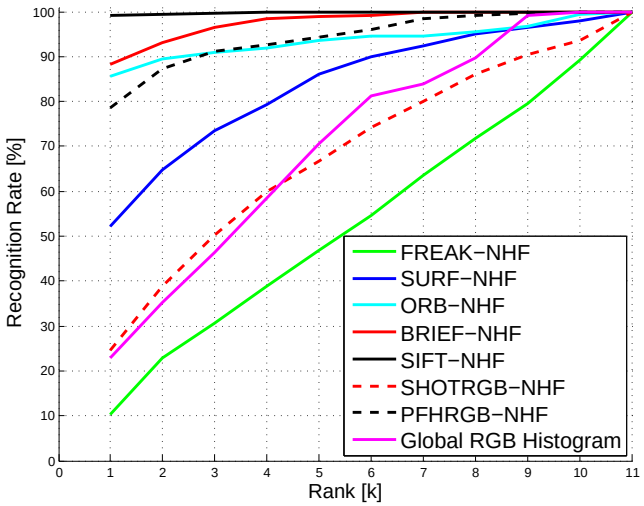


Figure 4.4: Re-identification results on the IAS-Lab RGBD-ID dataset.

Further Comparisons on the CAVIAR4REID Dataset

In order to validate our re-identification approach with respect to the best re-identification approaches in literature, we performed some tests on one of the most challenging datasets which is used for evaluating re-identification in video surveillance scenarios, the CAVIAR4REID dataset. This dataset contains RGB information only, thus we manually annotated the skeleton joints on all the images in order to compute our SPS signature at skeleton keypoints. We distinguished between visible and non-visible joints and we also released our annotations[7] in order to allow further comparisons with our method. Some examples of training and testing sequences and of annotated skeletons are shown in Figure 4.5.

Since the training and testing frames are collected from different cameras in this dataset, many joints which are visible in the testing images are not visible in the training images and vice-versa. Therefore our TJ matching approach is not applicable to this dataset and we selected the AJ matching instead, which computes the $\mathbf{SPS}^{\mathrm{AJ}}$ signature by concatenating descriptors at all keypoints, regardless if they are tracked or not and substitutes the descriptors for the occluded joints with those computed at their symmetric joints. We performed the same single-frame and multi-frame tests (with M=3 and M=5) described in [25] and we reported the results in Figure 4.6. Once again, our SPS signature which exploits the SIFT descriptor obtained the best rank-1 scores for all the tests, thus outperforming both the CPS approach [25] and the SDALF descriptor [38], even though this dataset was targeted to video surveillance applications

[7] `http://robotics.dei.unipd.it/reid`.

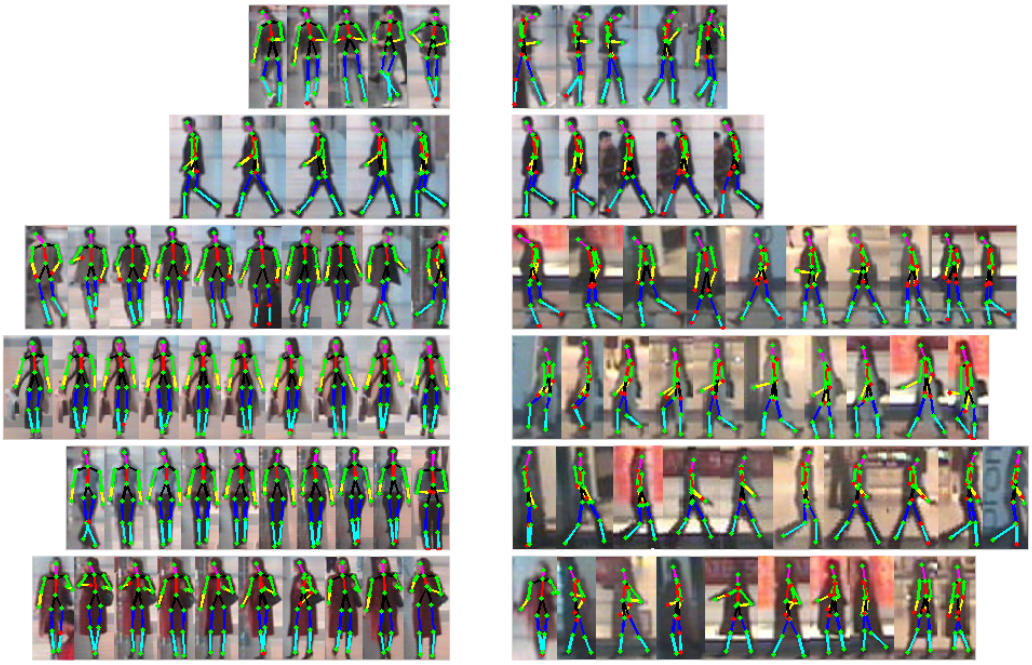


Figure 4.5: Examples of training (left) and testing (right) frames of the CAVIAR4REID dataset with the skeleton annotations we provided.

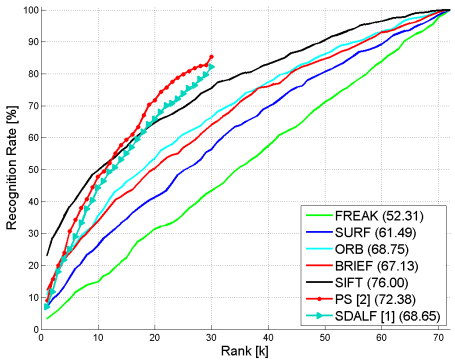
and the image resolution was very low.

Discussion on Computational Complexity

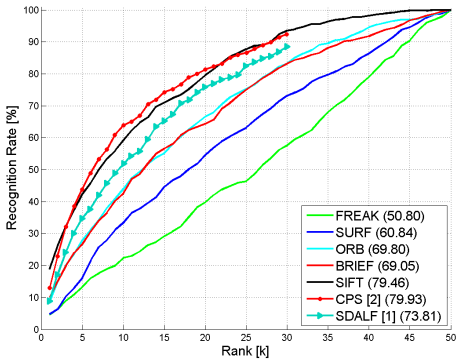
Table 4.1 also reports the time needed for classifying one frame for our SPS signature and the other approaches we tested. For matching with Nearest Neighbor, we exploited KD-Trees and FLANN[8] based matcher, which improve time performance of about one order of magnitude with respect to the brute force algorithm. Our tests were performed with a C++ implementation running on an Intel®Core™i3 CPU M330 @ 2.13 GHz with 4 GB DDR3 RAM.

The SPS signature with 2D descriptors results to be the fastest approach among all the techniques evaluated in this work. In particular, BRIEF is the fastest algorithm and obtains very good re-identification results, even though inferior to SIFT. 3D features are almost one order of magnitude slower than 2D features, thus preventing their use in real time applications. Finally, we show that the algorithm which uses SIFT keypoint detector to select keypoints results to be 2 times slower in the extraction phase and 10 times slower in the matching phase with respect to our skeleton-based approach.

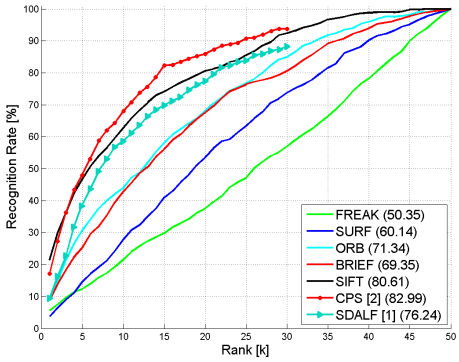
[8] `http://www.cs.ubc.ca/research/flann`.



(a) Single-frame



(b) Multi-frame (M=3)



(c) Multi-frame (M=5)

Figure 4.6: Re-identification results on the CAVIAR4REID dataset. For every approach, nAUC is reported within brackets.

Table 4.1: Summary of re-identification accuracy and computational times for the main approaches proposed and compared in this work. For the BIWI RGBD-ID dataset and the IAS-Lab RGBD-ID dataset, the results refer to tests performed on frames with all joints tracked. 20 joints are used for the BIWI RGBD-ID dataset, while the NHF approach is reported for the IAS-Lab RGBD-ID dataset. For the CAVIAR4REID dataset, tests have been perfomed on all frames and with all skeleton joints (AJ signature).

	Timings (ms)		**BIWI RGBD-ID**			**IAS-Lab RGBD-ID**			**CAVIAR4REID (M=5)**	
Approach	**Extraction**	**Matching**	**Rank-1**	**nAUC**	**R1-Multi**	**Rank-1**	**nAUC**	**R1-Multi**	**Rank-1**	**nAUC**
SIFT	185.45	0.00045	85.2	98.23	100.0	99.2	99.86	100.0	21.4	80.61
SURF	12.45	0.00072	31.0	78.52	75.0	52.3	84.35	72.7	3.8	60.14
BRIEF	9.58	0.00024	62.4	91.16	92.9	88.3	97.68	90.9	8.8	69.35
ORB	12.84	0.00024	53.3	88.12	89.3	85.6	93.83	90.9	9.2	71.34
FREAK	15.64	0.00048	11.1	56.90	28.6	10.2	55.33	9.1	5.6	50.35
PFHRGB-NHF	894.74	0.00137	48.5	87.01	67.9	78.7	94.31	100	-	-
SHOTRGB-NHF	622.17	0.01482	23.0	81.14	39.3	24.6	69.56	45.46	-	-
FPFH-NHF	1317.27	0.00021	12.4	62.18	32.1	-	-	-	-	-
PFH-NHF	765.37	0.00095	11.1	63.54	21.4	-	-	-	-	-
SHOT-NHF	616.09	0.00347	5.4	56.38	10.7	-	-	-	-	-
SIFT-keypoints	413.28	0.0035	63.4	89.99	71.43	-	-	-	-	-
Global RGB Histogram	352.16	0.0075	78.4	98.93	85.7	23.0	71.60	27.3	-	-
Local RGB Histogram	309.15	0.0085	53.4	94.84	75.0	-	-	-	-	-
SDALF [38]	-	-	-	-	-	-	-	-	9.4	76.24
CPS [25]	-	-	-	-	-	-	-	-	17.2	82.99

4.1.10 Conclusions

In this section, a novel approach to short-term people re-identification in RGB-D data has been presented. This approach builds on the assumption that very stable keypoints can be detected on human targets by means of a skeletal tracker, and exploited to evaluate signatures by means of 2D and 3D feature extractors. This idea was developed considering several features and matching methods and overcoming the instabilities that still affect skeletal trackers.

The novel re-identification system presented has been extensively tested using both video-surveillance datasets, for comparing this novel approach to the state of the art, and newly created datasets that are capable of highlighting the great advantages offered by our approach. This re-identification method is particularly suited for robotic applications dealing with humans, since it offers superior performance, it exploits sensors commonly available on most autonomous robots and runs in real-time.

4.2 Long-Term Re-Identification

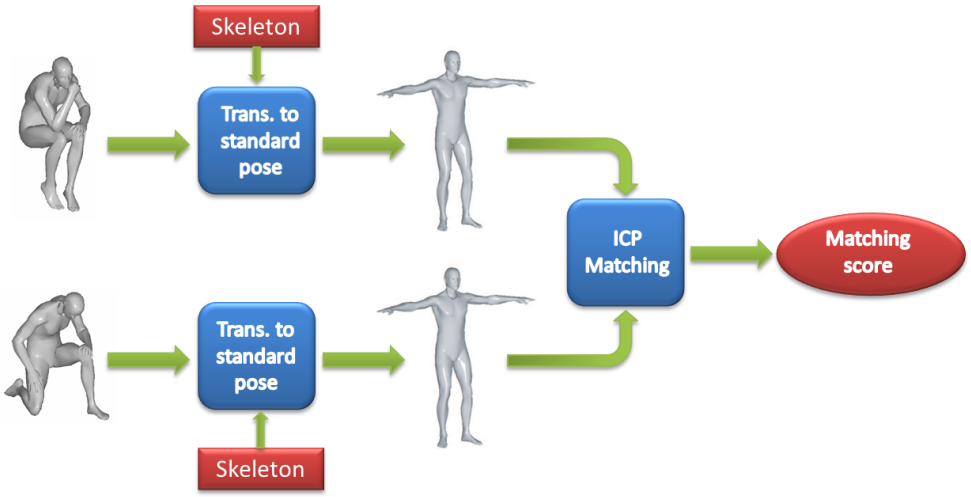


Figure 4.7: Illustration of the pipeline we developed for comparing body shapes exploiting a skeletal tracker.

Unlike short-term techniques, long-term person re-identification is targeted to recognize people after days, months or years, thus it has to rely on permanent features of the human body and not on clothes information. When long-term re-identification is performed without having access to reliable and discriminative data such as the DNA sequence and fingerprints, and without the collaboration of the person to be analyzed, we refer to the branch of non-invasive and non-collaborative biometrics. In particular, in this work, we simply rely on the input provided by a cheap consumer depth camera.

The set of features that we adopt to identify a specific person are commonly known as soft biometrics. This means that each feature alone is not a univocal identifier for a subject. Still, the combination of several soft biometrics features can show a very good discriminative performance even with large sets of persons.

Global shape is a *soft biometric* feature [49] that can allow to discriminate between people in a long-term time span when other cues are not available or are not enough for obtaining accurate results. In recent years, it became easily available with consumer RGB-D sensors, such as Microsoft Kinect, which equips the majority of modern mobile robots. However, little effort has been spent so far to develop methods for people re-identification based on this particular cue. In order to identify a person within a training set of known people given a partial point cloud obtained by a depth sensor, complete 3D models of the people in the training set are necessary. Moreover, the matching between test clouds and training models can fail because a person is not rigid

and can assume a great variety of poses. The main sources of shape variability among clouds belonging to the same person are differences in pose and clothing between training and testing point clouds. If we assume that the differences in clothing shape are negligible, we could compare people as we do for rigid objects if they all had the same pose. The main idea investigated with this work is then to exploit the information provided by state of the art skeletal tracking algorithms to transform persons point clouds to a neutral pose, as illustrated in Figure 4.7. The proposed method is useful both for creating full body models which can be used as training set and for matching new point clouds with the training models and is efficient so that both training and testing can be done online onboard a mobile robot. Moreover, our approach does not require the cooperation of the scanned person, that can freely move.

The contribution of this section is three-fold: On one hand we propose a novel technique for exploiting skeleton information to efficiently transform persons' point clouds to a standard pose. On the other hand, we describe how to use these transformed point clouds for composing 3D models of freely moving people which can be used for re-identification by means of an ICP matching with new test clouds. Moreover, we propose a method to combine the ICP matching scores with a descriptor of body skeleton lengths which improves the recognition rates obtainable with the single approaches. We compare these techniques with skeleton-based and face-based baselines on the newly acquired *BIWI RGBD-ID* and *IAS-Lab RGBD-ID* re-identification datasets, which exploit two different state of the art skeletal tracking algorithms.

Section 4.2.1 describes related work, while Section 4.2.2 introduces the new datasets we collected. The Point Cloud Matching approach is detailed in Section 4.2.3, while the skeleton-based approach is outlined in Section 4.2.4 and a combination of these two approaches is proposed in Section 4.2.5. The classification process is presented in Section 4.2.6, Section 4.2.7 contains several experiments and conclusions are drawn in Section 4.2.8.

4.2.1 Related Work

Depth-based and Multi-Modal Re-Identification

Due to the very recent availability of cheap depth sensing devices, only a few works exist that focused on identification using such multi-modal input. Most of these works rely on combinations of soft biometric features, that are in general not discriminative enough to identify a subject, but can be very powerful if combined with other traits. In [87], it is shown that anthropometric measures are discriminative enough to obtain

a 97% accuracy on a population of 2000 subjects. The authors apply Linear Discriminant Analysis to very accurate laser scans to obtain such performance. Also the authors of [118] studied a similar problem. They in fact used a few anthropometric features, manually measured on the subjects, as a pre-processing pruning step to make face-based identification more efficient and reliable. In [104], the authors have recently proposed an approach which uses the input provided by a network of Kinect cameras: The depth data in their case is only used for segmentation, while their re-identification techniques purely relies on appearance-based features. The authors of [6] propose a method that relies only on depth data, by extracting a signature for each subject. Such signature includes features extracted from the skeleton as the lengths of a few limbs and the ratios of some of these lengths. In addition, geodesic distances on the body shape between some pairs of body joints are considered. The choice of the most discriminative features is based upon a few extensive experiments carried on a validation dataset. The signatures are extracted from a single training frame for each subject, which renders the framework quite prone to noise. The dataset used in the paper has also been made publicly available, but this does not contain facial information of the subjects and skeleton links orientation, in contrast with the datasets proposed in our work. Also Kinect Identity [64], the software running on the Kinect for XBox360, uses multi-modal data, namely the subject's height, a face descriptor and a color model of the user's clothing to re-identify a player during a gaming session. In this case, though, the problem is simplified as such re-identification only covers a very short time-span and the number of different identities is usually very small. A recent work also applied gait recognition [95] techniques to classify sequences of descriptors of joints information obtained with Kinect skeletal tracker.

Bronstein *et al.* ([19], [20]) exploit global body shape for re-identification and tackle the person matching problem by applying an isometric embedding which allows to get rid of pose variability (extrinsic geometry) by warping shapes to a canonical form where geodesic distances are replaced by Euclidean ones. In this space, an ICP matching is applied to estimate similarity between shapes. However, a geodesic masking which retains the same portion of every shape is needed for this method to work well. In particular, for matching people's shape, a complete and accurate 3D scan has to be used, thus partial views cannot be matched with a full model because they could lead to very different embeddings. Moreover, this approach needs to solve a complicated optimization problem, thus requiring several seconds to complete. Our method, instead, exploits the information provided by a skeletal tracking algorithm for rapidly transforming persons point clouds to a standard pose, so that techniques studied for

reconstructing objects can then be applied.

3D Reconstruction of People

For recognizing people based on their global shape, full 3D models have to be composed from sequences of depth frames, so that they can be used as training set. Even though human body reconstruction from multiple sensors (motion capture) has been thoroughly studied, little literature exists on person reconstruction from a single moving sensor. As discussed in Sec. 4.2, people are articulated and locally deformable, thus recent techniques for real-time RGB-D reconstruction [24, 48], which assumes to reconstruct rigid scenes, are doomed to fail if the person is moving. In [123], the authors propose a method which exploits the SCAPE body model for obtaining 3D models of people from range scans acquired by a Microsoft Kinect. However, their approach is computationally expensive and still requires the person to be collaborative during the scanning process since it is targeted to fitness and apparel applications.

4.2.2 RGB-D Re-Identification Datasets

The vast majority of publicly available RGB-D datasets are targeted to human activity analysis and action recognition, and for this reason they are generally composed by many gestures performed by few subjects [89, 114, 120, 121, 124, 131]. The only dataset explicitly thought for the RGB-D re-identification task has been proposed in [6]. It consists of 79 different subjects collected in 4 different scenarios. However, this dataset contains very few frames for each subject, faces are blurred for privacy reasons and skeleton links orientation are not available. For overcoming these limitations, we collected our own RGB-D re-identification datasets: the BIWI RGBD-ID dataset (Appendix B.2.1) contains 50 people and data from Kinect skeletal tracker, while the IAS-Lab RGBD-ID dataset features 11 people and NiTE skeletal tracker (Appendix B.2.2). Both of them provide also RGB and depth information collected with a Microsoft Kinect.

4.2.3 Point Cloud Matching

In this section, we propose a method which takes into account the whole human body point cloud for the re-identification task. In particular, given two point clouds, we try to align them and then compute a similarity score between the two, as illustrated in Figure 4.7. As a fitness score, we compute the average distance of the points of a

cloud to the nearest points of the other cloud. If $\mathscr{P}_1$ and $\mathscr{P}_2$ are two point clouds, the fitness score of $\mathscr{P}_2$ with respect to $\mathscr{P}_1$ is then

$$f_{2\to1} = \sum_{p_i \in \mathscr{P}_2} \|p_i - q_i^*\|, \tag{4.7}$$

where q_i^* is defined as

$$q_i^* = \arg\min_{q_j \in \mathscr{P}_1} \left\|p_i - q_j\right\|. \tag{4.8}$$

It is worth to notice that this fitness score is not symmetric, that is $f_{2\to1} \neq f_{1\to2}$.

For what concerns the alignment, the position and orientation of a reference skeleton joint, e.g. the hip center, is used to perform a rough alignment between the clouds to compare. Then, that alignment is refined by means of an ICP-based registration [13], which should converge in few iterations if the initial alignment is good enough. When the input point clouds have been aligned with this process, the fitness score between them should be minimum, ideally zero if they coincide or if $\mathscr{P}_2$ is contained in $\mathscr{P}_1$.

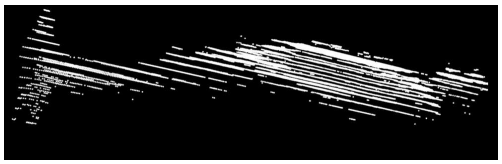
For the purpose of re-identification, this procedure is used to compare a testing point cloud with the point cloud models obtained from the training set as we will describe in Sec. 4.2.3 and to select the training subject whose point cloud model has the minimum fitness score when matched with the testing cloud.

Point Cloud Smoothing

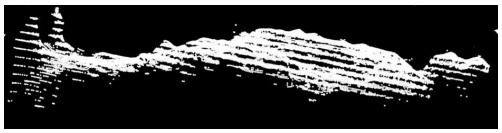
3D point clouds acquired with consumer depth sensors have good resolution but the depth quantization step increasing quadratically with the distance does not allow to obtain smooth people point clouds beyond two meters from the sensor. In Figure 4.8 (a), the point cloud of a person three meters from the sensor is reported. It can be noticed that the point cloud results divided into slices produced by the quantization steps. As a pre-processing step, we improve the person point cloud by applying a voxel grid filter and a Moving Least Squares surface reconstruction method to obtain a smoothing, as reported in Figure 4.8 (b).

Transformation to Standard Pose

Composing 3D models of persons usually requires multiple cameras or at least that the person remains still during the scanning process. It is because typical reconstruction techniques for rigid objects are usually exploited [43]. However, when dealing with moving people, the rigidity assumption does not hold any more, because people are articulated and they can appear in a very large number of different poses, thus these



(a) Raw



(b) Smoothed

Figure 4.8: (a) Raw person point cloud at 3 meters of distance from the Kinect and (b) point cloud after the pre-processing step.

approaches would be doomed to fail.

To overcome this problem, our method proposes to exploit the information provided by a skeletal tracking algorithm for rapidly transforming persons' point clouds to a standard pose, so that techniques studied for reconstructing objects can then be applied.

This result is obtained by rototranslating each body part according to the positions and orientations of the skeleton joints and links given by the skeletal tracking algorithm. A preliminary operation consists in segmenting the person's point cloud into body parts. Even if Kinect skeletal tracker estimates this segmentation as a first step and then derives the joints position, it does not provide to the user the result of the depth map labeling into body parts. For this reason, we implemented the reverse procedure for obtaining the segmentation of a person point cloud into parts by starting from the 3D positions of the body joints. In particular, we assign every point cloud point to the nearest body link. For a better segmentation of the torso and the arms, we added two further fictitious links between the hips and the shoulders. The body links considered and an example of body segmentation are reported in Figure 4.9 (a).

Once we performed the body segmentation, we warp the pose assumed by the person to a new pose, which is called *standard pose*. The standard pose makes the point clouds of all the subjects directly comparable, by imposing the same orientation between the links. On the other hand, each link length is person-dependent and is estimated from a valid frame of the person and then kept fixed. The transformation consists in rototranslating the points belonging to each body part according to the corresponding skeleton link position and orientation[9]. In particular, every body part is

[9]It is worth noting that all the links belonging to the torso have the same orientation, as the hip center.

rotated according to the corresponding link orientation and translated according to its joints coordinates. If Q_c is the quaternion containing the orientation of a link in the current frame given by the skeletal tracker and Q_s is the one expressing its orientation in standard pose, the whole rotation to apply can be computed as

$$R = Q_s \left(Q_c\right)^{-1}, \tag{4.9}$$

while the full transformation applied to a point p can be synthesized as

$$p' = T_{V_s}\left(R\left(\left(T_{V_c}\right)^{-1}(p)\right)\right), \tag{4.10}$$

where T_{V_c} and T_{V_s} are the translation vectors of the corresponding skeleton joint at the current frame and in the standard pose, respectively.

As the standard pose, we chose a typical frontal pose of a person at rest. In Figure 4.9 (b-c), we report two examples of point clouds before and after the transformation to standard pose.

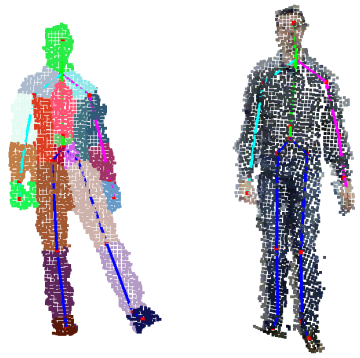
It is worth noting that the point cloud transformation to standard pose, that is the process of rototranslating each body part according to the skeleton estimation, can have two negative effects on the point cloud: some body parts can intersect each other and some gaps can appear around the joint centers. However, the parts intersection is tackled by voxel grid filtering the transformed point cloud, while the missing points do not represent a problem for the matching phase, since a test point cloud is considered to perfectly match a training point cloud if it is fully contained in it, as explained in Sec. 4.2.3.

Creation of Human Body Models

The transformation to standard pose is not only useful because it allows to compare people clouds disregarding their initial pose, but also because more point clouds belonging to the same moving person can be easily merged to compose a more complete body model. In Figure 4.11 (a), a single person point cloud is compared with the model we obtained by merging together several point clouds acquired while the person was moving and transformed to the standard pose. Moreover, we show the union cloud we obtained without any smoothing and after a voxel grid filter and a Moving Least Squares surface reconstruction method are applied. It can be noticed how the union cloud is denser and more complete with respect to the single cloud.

For the re-identification task, we create a point cloud model for every person from a sequence of training frames where the person is moving freely, as illustrated in Fig-

(a) (b)



(c)

Figure 4.9: (a) Body links considered and body segmentation obtained. (b-c) Two examples of standard pose transformation. On the left, the body segmentation is shown with colors, on the right, the RGB texture is applied to the point cloud obtained after the transformation.

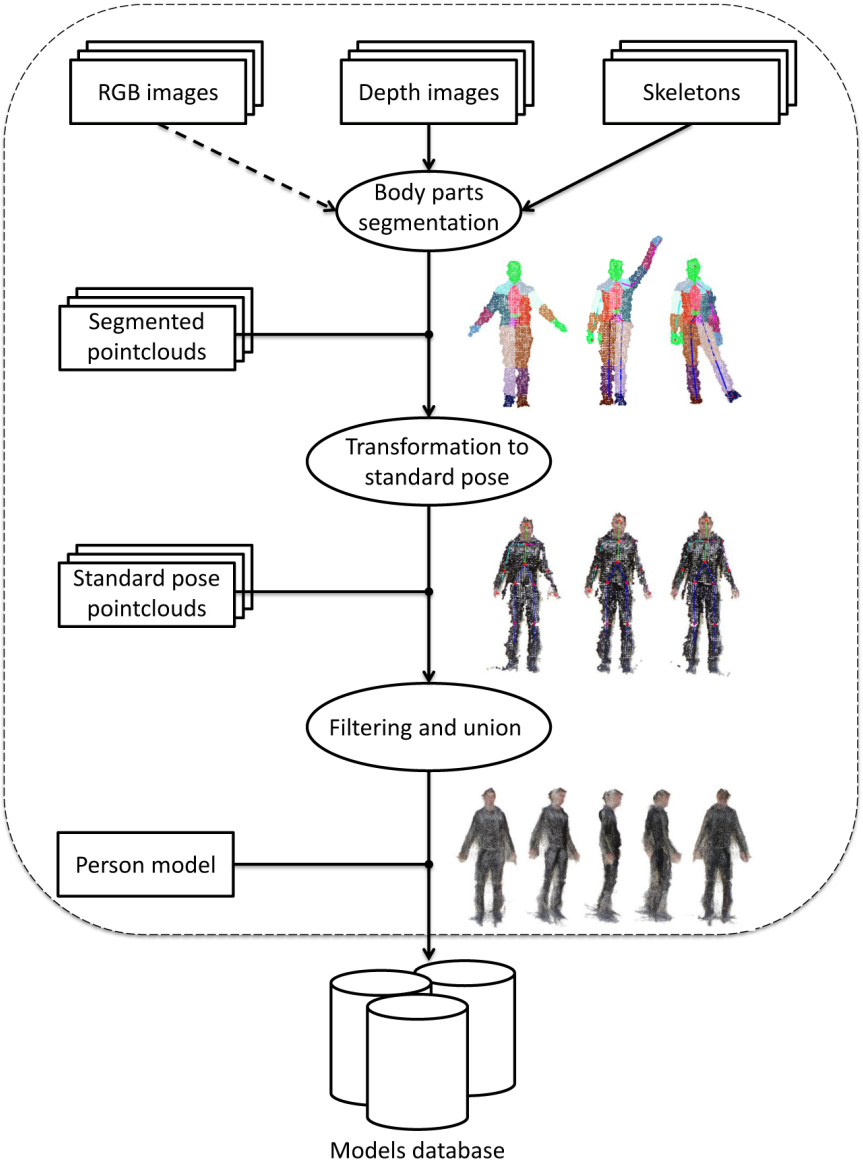


Figure 4.10: Illustration of the pipeline we developed for creating full 3D models of freely moving persons. RGB information is added to the point cloud models only for a better visualization, but in this work it is not used for matching.

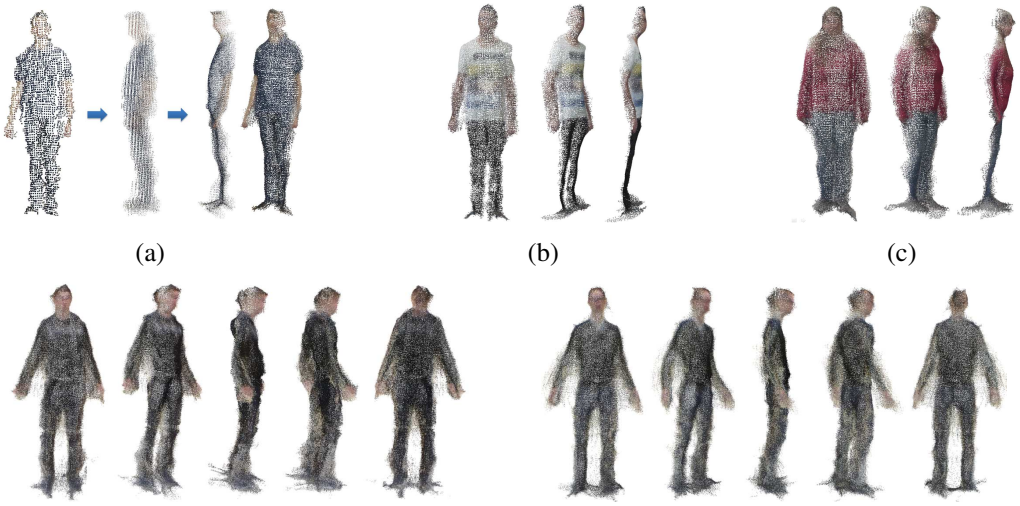


Figure 4.11: (a) Steps of model creation (from left to right: a single person's point cloud, the union of point clouds before and after smoothing); (b-c) examples of 180° person models obtained with Kinect skeletal tracker; (d-e) examples of 360° person models obtained with NiTE skeletal tracker.

ure 4.10. For tackling the problem of noisy skeleton estimates, every body part is further registered to the model already composed by means of a local ICP algorithm. Given that, with Kinect skeletal tracker, we do not obtain valid frames if the person is seen from the back, we can only obtain 180° people models (Figure 4.11 (b) and (c)), while with NiTE skeletal tracker we can reconstruct a 360° people model (Figure 4.11 (d) and (e)). It can be noticed how the front and back of the models obtained with NiTE skeletal tracker are too close to each other. This is due to a limitation of the skeletal tracking algorithm, which does not estimate correctly the joints position inside the bones. However, this offset could be measured and compensated as a future work.

4.2.4 Skeleton Descriptor

In this work, we compare the proposed shape-based technique with a feature-based method, similar to [6], which computes a descriptor composed of skeleton links lengths and joints distances to the ground and classifies it by means of a Nearest Neighbor classifier based on the Euclidean distance. In particular, our skeleton descriptor is composed of the following 13 distances (also illustrated in Figure 4.12):

a) head height,

b) neck height,

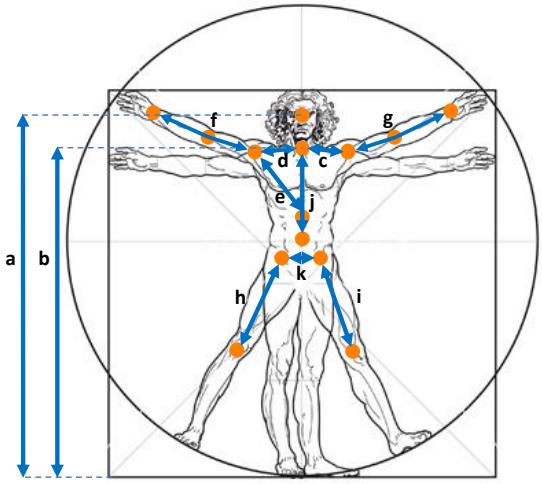


Figure 4.12: Illustration of the links lengths and joints distances to the ground which constitute the skeleton descriptor.

c) neck to left shoulder distance,

d) neck to right shoulder distance,

e) torso to right shoulder distance,

f) right arm length,

g) left arm length,

h) right upper leg length,

i) left upper leg length,

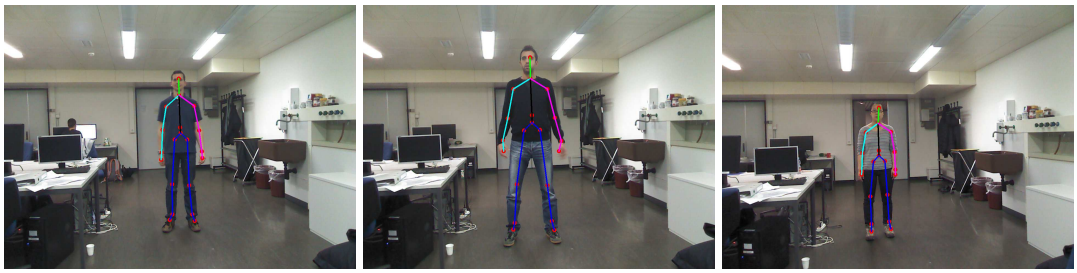
j) torso length,

k) right hip to left hip distance,

l) ratio between torso length and right upper leg length (j/h),

m) ratio between torso length and left upper leg length (j/i).

In Figure 4.13, the skeleton computed with Kinect skeletal tracker is reported for three very different people of our dataset, while in Figure 4.14 and 4.15, we show how the value of some skeleton features varies along time when these people are still and walking, respectively. We also report the average standard deviation of these features for the people of the two testing sets. As expected, the heights of the head and the neck



(a) (b) (c)

Figure 4.13: Examples of estimated skeletons for three people of the testing videos of the *BIWI RGBD-ID* dataset.

from the ground are the most discriminative features. What is more interesting is that the standard deviation of these features doubles for the walking test set with respect to the test set where people are still, thus suggesting that the skeleton joint positions are better estimated when people are static and frontal.

When a person is seen from the side or from the back, Kinect skeletal tracker [107] does not provide correct estimates because it is based on a random forest classifier which has been trained with examples of frontal people only. For this reason, in this work, we discard frames with at least one not tracked joint[10]. Then, we keep only those where a face is detected [119] in the proximity of the head joint position. This kind of selection is needed for discarding also those frames where the person is seen from the back, which come with a wrong skeleton estimation.

4.2.5 Combined Approach

Since the skeleton lengths and the body shape are complementary features, a combination of these approaches could lead to results superior to those obtained with the single techniques. For this reason, we also propose a mixed approach which uses the fitness scores obtained by matching the testing cloud with the training models as weights for the distances between the testing and training skeleton descriptors. Given f^i_{ICP} the fitness score obtained by comparing a test point cloud warped to standard pose with the i^{th} model of the training dataset and be d^i_{skel} the minimum distance between the test skeleton descriptor and the descriptors of the i^{th} person of the training set. Then, according to our combined approach the distance of the test frame to the i^{th} person of the training set is computed as

$$d^i_{PCM+skel} = f^i_{ICP} \cdot d^i_{skel}, \quad i = 1...N_{train}. \tag{4.11}$$

[10]Kinect skeletal tracker provides a flag for every joint stating if it is tracked, inferred or not tracked.

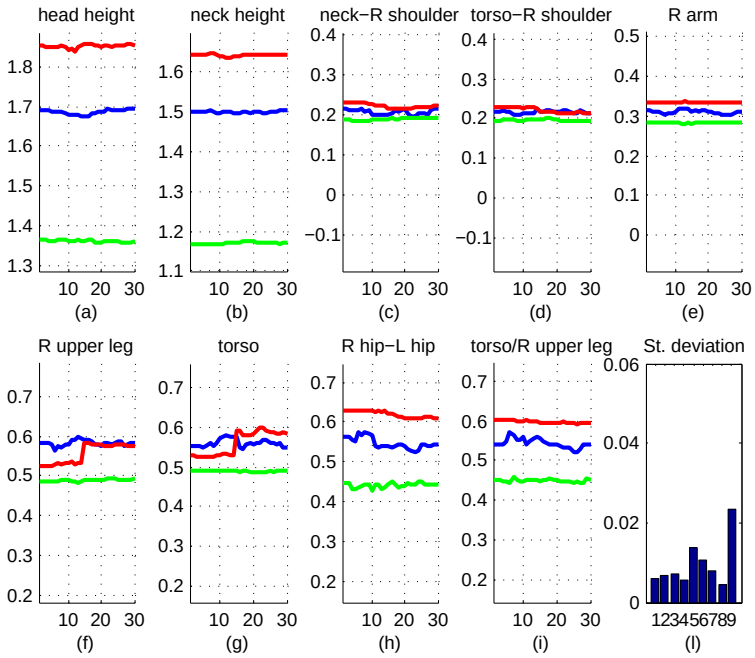


Figure 4.14: (a-i) Estimated skeleton features for some frames of the *Still* test sequence for the three subjects of Figure 4.13. Those subjects are represented by blue, red and green curves, respectively. In (l), the standard deviation of these features is reported.

where N_{train} is the number of training people. This procedure weights the skeleton classification differently for every person of the training set according to how well the current testing point cloud matches the training models obtained as described in Sec. 4.2.3. The current test frame is then associated to the training person obtaining the minimum $d^i_{PCM+skel}$.

4.2.6 Classification

For classifying descriptors presented in the previous sections, we tested three different classification approaches. The first method compares descriptors extracted from the testing dataset with those of the training dataset by means of a Nearest Neighbor classifier based on the Euclidean distance. The second one consists in learning the parameters of a Support Vector Machine (SVM) [28] for every subject of the training dataset. As SVMs are originarily designed for binary classification, these classifiers are trained in a *One-vs-All* fashion: For a certain subject i, the descriptors computed on that subject are considered as positive samples while the descriptors computed on all the subjects except i are considered as negative samples.

The One-vs-All approach requires all the training procedure to be performed again

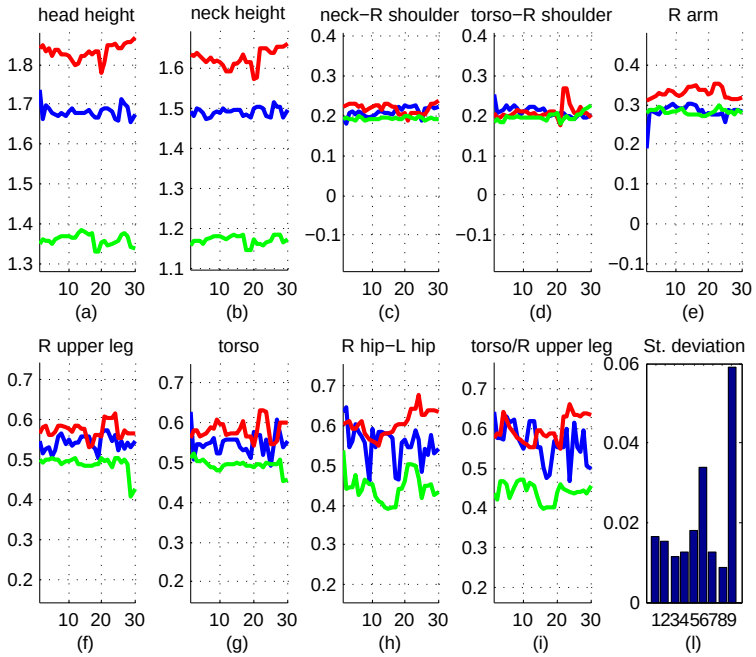


Figure 4.15: (a-i) Estimated skeleton features for some frames of the *Walking* test sequence for the three subjects of Figure 4.13. Those subjects are represented by blue, red and green curves, respectively. In (l), the standard deviation of these features is reported.

if a new person is inserted in the database. This need makes the approach not suitable for a scenario where new people are inserted online for a subsequent re-identification. For this purpose, we also trained a *Generic SVM* which does not learn how to distinguish a specific person from all the others, but it learns how to understand if two descriptors have been extracted from the same person or not. The positive training examples which are fed to this SVM are of the form

$$pos = \left| d_1^i - d_2^i \right|, \tag{4.12}$$

where d_1^i and d_2^i are descriptors extracted from two frames containing the same subject i, while the negative examples are of the form

$$neg = \left| d_1^i - d_2^j \right|, \tag{4.13}$$

where d_1^i and d_2^j are descriptors extracted from frames containing different subjects. At testing time, the current descriptor d_{test} is compared to the training descriptors d_k^i of every subject i by using this Generic SVM for classifying the vector $\left| d_{test} - d_k^i \right|$ and the test descriptor is associated to the class for which the maximum SVM confidence is obtained.

4.2.7 Experiments

We present here some person re-identification tests we performed with the datasets described in Sec. B.2.1 and B.2.2. For evaluation purposes, we compute *Cumulative Matching Characteristic (CMC) Curves* [42], which are commonly used for evaluating re-identification algorithms. For every k from 1 to the number of training subjects, these curves express the mean recognition rate, computed when considering a classification to be correct if the ground truth person appears among the subjects who obtained the k best classification scores. The typical evaluation parameters for these curves are the *rank-1* recognition rate and the *normalized Area Under Curve* (nAUC), which is the integral of the CMC. In this work, the recognition rates are separately computed for every subject and then averaged to obtain the final recognition rate.

Tests on the BIWI RGBD-ID dataset

Kinect skeletal tracker is based on a random forest classifier which has been trained with examples of frontal people only, thus it does not provide correct estimates when the person is seen from the side or from the back. For this reason, in this work, we

discarded the frames with at least one not tracked joint[11]. Then, we kept only those where a face was detected in the proximity of the head joint position. This kind of selection is needed for discarding also those frames where the person is seen from the back, which come with a wrong skeleton estimation.

For testing the point cloud matching approach of Sec. 4.2.3, we built one point cloud model for every person of the training set by merging together point clouds extracted from their training sequences and transformed to standard pose. At every frame, a new cloud was added and a voxel grid filter was applied to the union result for resampling the cloud and limiting the number of points. At the end, we exploited a moving least squares surface reconstruction method for smoothing. At testing time, every person's cloud was transformed to standard pose, aligned and compared to the 50 persons training models and classified according to the minimum fitness score $f_{test \rightarrow model}$ obtained. It is worth noting that the fitness score reported in Eq. 4.7 correctly returns the minimum score (zero) if the test point cloud is contained in the model point cloud, while it would return a different score if the test cloud would only partially overlaps the model. Also for this reason, we chose to build the persons models described in Sec. 4.2.3. In Figure 4.17, we compare the described method with a similar matching method which does not exploit the point cloud transformation to standard pose. For the testing set with still people, the differences are minor because people are often in the same pose, while, for the walking test set, the transformation to standard pose considerably outperforms the method which does not exploit it, reaching a rank-1 performance of 22.4% against 7.4% and a nAUC of 81.6% against 64.3%.

In the same figure, we report also the results obtained by classifying the skeleton descriptor and with the combined *PCM+Skeleton* approach of Sec. 4.2.4. It can be noticed how the point cloud matching technique obtained a slightly better rank-1 performance with respect to the skeleton classification, while the combined approach outperformed both methods, thus proving to exploit the complementarity of joint lengths and body shape. As a further reference, we report also the results obtained with a face recognition technique. This technique extracts the subject's face from the RGB input using a standard face detection algorithm [119]. To increase the computational speed and decrease the number of false positives, the search region is limited to a small neighborhood of the 2D location of the head, as provided by the skeletal tracker. Examples of skeleton-aided face detection are reported in Figure A.3. Once the face has been detected, a real-time method to extract the 2D location of the 10 fiducials points shown in Figure 4.16 is applied [31]. Finally, SURF descriptors [9] are computed at the

[11] Kinect skeletal tracker provides a flag for every joint stating if it is tracked, inferred or not tracked.

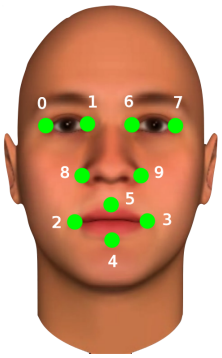
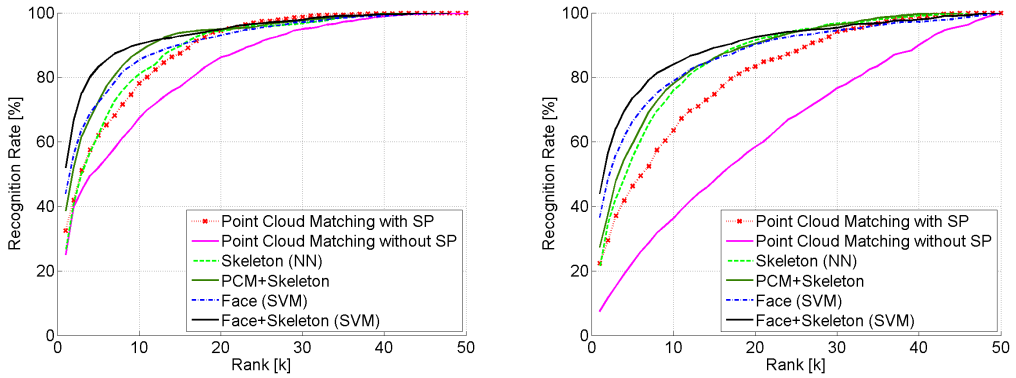


Figure 4.16: Fiducial points detected in a face with the algorithm in [31] and used for extracting a face descriptor.



(a) Still (b) Walking

Figure 4.17: Cumulative Matching Characteristic Curves obtained with the main approaches described in this section for the BIWI RGBD-ID dataset.

location of the fiducials and concatenated forming a single vector. Unlike the skeleton descriptor, the face descriptor has been classified with a One-VS-All SVM classifier, reaching 44% of rank-1 for the *Still* testing set and 36.7% for the *Walking* set. Finally, by concatenating the skeleton and face descriptors and classifying them with a One-VS-All SVM, a further 8% gain of rank-1 for the *Still* test set and 7.2% for the *Walking* test set can be obtained. In Table 4.2, all the numerical results are reported with also the cross validation outputs.

The re-identification methods we described are all based on a one-shot re-identification from a single test frame. However, when more frames of the same person are available, the results obtained for each frame can be merged to obtain a sequence-wise result. In Table 4.2, we also report the rank-1 performances which can be obtained with a simple multi-frame reasoning, that is by associating each test sequence to the subject voted by the highest number of frames. On average, this voting scheme allows to obtain a per-

Table 4.2: Evaluation results obtained in cross validation and with the testing sets of the BIWI RGBD-ID dataset.

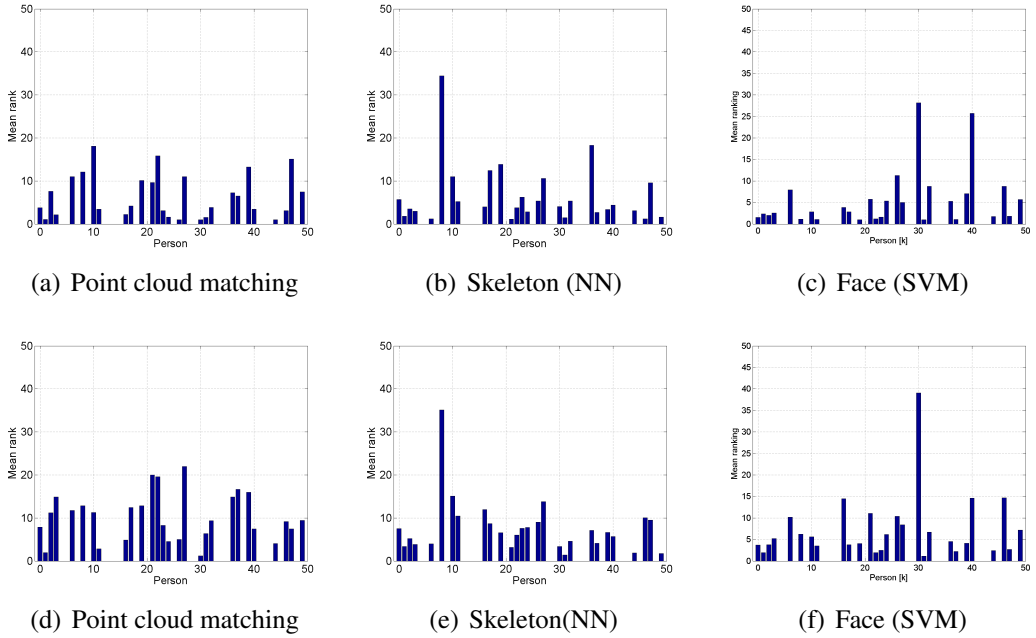
	Cross validation			Test - Still			Test - Walking		
	Rank-1	**nAUC**	**R1 Multi**	**Rank-1**	**nAUC**	**R1 Multi**	**Rank-1**	**nAUC**	**R1 Multi**
Point Cloud Matching	93.7%	99.6%	100%	32.5%	89.0%	42.9%	22.4%	81.6%	39.3%
Skeleton (NN)	80.5%	98.2%	100%	26.6%	89.7%	32.1%	21.1%	86.6%	39.3%
PCM+Skeleton	-	-	-	38.6%	91.8%	46.4%	27.4%	87.4%	42.9%
Face (SVM)	97.8%	99.4%	100%	44.0%	91.0%	57.1%	36.7%	87.6%	57.1%
Face+Skeleton (SVM)	98.4%	99.5%	100%	52.0%	93.7%	67.9%	43.9%	90.2%	67.9%

formance improvement of about 10-20%. The best performance is again obtained with the SVM classification of the combined face and skeleton descriptors, which reaches 67.9% of rank-1 for both the testing sets, while the combined PCM+skel approach obtained a rank-1 of 46.4%, thus proving to be the best option when face is not available.

To analyze how the re-identification performance differs for the different subjects of our dataset, we report in Figure 4.18 the histograms of the mean ranking for every person of the testing dataset, which is the average ranking at which the correct person is classified. The missing values in the x axis are due to the fact that not all the training subjects are present in the testing set. It can be noticed that there is a correspondence between the mean ranking obtained in the *Still* testing set and that obtained in the *Walking* test set. It is also clear that different approaches lead to mistakes on different people, thus showing to be partially complementary.

Tests on the RGB-D Person Re-Identification dataset

As explained in Section 4.2.2, the RGB-D Person Re-Identification dataset is the only other public dataset for person re-identification using RGB-D data. Unfortunately, there are only few examples available for each of the subjects, which make the use of many machine learning techniques, including SVMs trained with a One-VS-All approach, quite complicated. However, given that the Generic SVM described in Sec. 4.2.6 is one for all the subjects, we had enough examples to train it correctly. In Table 4.3, we compare the results reported in [6] with our results obtained when classifying the skeleton descriptor with the Nearest Neighbor and the Generic SVM. Unfortunately, the authors of [6] report performances only in terms of normalized Area Under Curve (nAUC) of the Cumulative Matching Curve (CMC), thus their rank-1 scores are not available except for one result that can be inferred from a figure. The classification of our skeleton descriptor with the Generic SVM performed better than [6] and of our Nearest Neighbor classifier for the tests which do not involve the *Collaborative* set, where people walk with open arms. We also tested the geodesic features the au-



(a) Point cloud matching (b) Skeleton (NN) (c) Face (SVM)

(d) Point cloud matching (e) Skeleton(NN) (f) Face (SVM)

Figure 4.18: Mean ranking histograms obtained with different techniques for every person of the *Still* (top row) and *Walking* (bottom row) test sets of the BIWI RGBD-ID dataset.

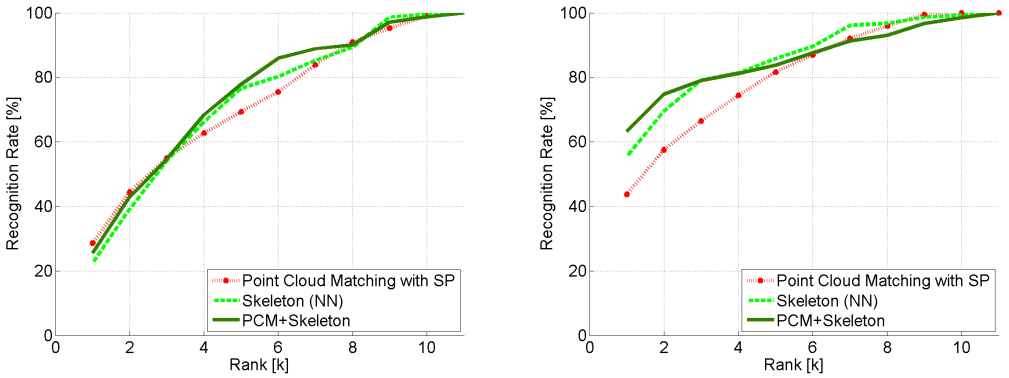
Table 4.3: Evaluation results on the RGB-D Person Re-Identification dataset.

Training	**Testing**	**[6]**		**Ours - NN**		**Ours - Generic SVM**	
		Rank-1	**nAUC**	**Rank-1**	**nAUC**	**Rank-1**	**nAUC**
Collaborative	**Walking1**	N/A	90.1%	7.8%	81.1%	5.3%	79.0%
Collaborative	**Walking2**	13%	88.9%	4.8%	81.3%	4.1%	78.6%
Collaborative	**Backwards**	N/A	85.6%	4.6%	78.8%	3.6%	76.0%
Walking1	**Walking2**	N/A	91.8%	28.6%	89.9%	35.7%	92.8%
Walking1	**Backwards**	N/A	88.7%	17.8%	82.7%	18.5%	90.6%
Walking2	**Backwards**	N/A	87.7%	13.2%	84.1%	22.3%	91.6%

thors propose, but they did not provide substantial improvement to the skeleton alone. We did not test the point cloud matching and the face recognition techniques on this dataset because the links orientation information was not provided and the face in the RGB image was blurred.

Tests on the IAS-Lab RGBD-ID dataset

For the IAS-Lab RGBD-ID dataset, we performed the same evaluation described for the BIWI RGBD-ID dataset. However, we used also the frames where persons were seen from the back. NiTE skeletal tracker provides much more noisy estimates than Kinect skeletal tracker and sometimes estimates impossible poses, e.g. with legs turned backwards when the torso is turned forward or with legs crossing and compenetrating.



(a) TestingA (b) TestingB

Figure 4.19: Cumulative Matching Characteristic Curves obtained with the main approaches described in this section for the IAS-Lab RGBD-ID dataset.

Table 4.4: Evaluation results obtained in cross validation and with the testing sets of the IAS-Lab RGBD-ID dataset.

	Cross validation			**TestingA**			**TestingB**		
	Rank-1	**nAUC**	**R1 Multi**	**Rank-1**	**nAUC**	**R1 Multi**	**Rank-1**	**nAUC**	**R1 Multi**
Point Cloud Matching	92.0%	98.7%	100%	28.6%	73.2%	63.6%	43.7%	81.7%	72.7%
Skeleton (NN)	86.2%	96.4%	100%	22.5%	73.8%	27.3%	55.5%	86.3%	81.8%
PCM+Skeleton	-	-	-	25.6%	75.5%	27.3%	63.3%	86.3%	81.8%

These erroneous frames are automatically detected by our algorithm and discarded. In Figure 4.19, we report the CMC curves obtained with the point cloud matching, the skeleton descriptor and the combined approach with the IAS-Lab RGBD-ID dataset, while all the numerical results are listed in Table 4.4. It can be noticed that the absolute performance of the three methods considerably decreases for the testing set with people wearing different clothes (*TestingA*), but the point cloud matching technique provides a very good multi-frame rank-1 score (63.6%) which doubles that obtained with the skeleton descriptor (27.3%). For the *TestingB* set, the best results are again obtained by the combined approach, which obtained a rank-1 score of 63.3%, against 55.5% of the skeleton descriptor classification and 43.7% of the point cloud matching.

Runtime Performance

In Table 4.5, the runtime of the single algorithms needed for the point cloud matching method of Sec. 4.2.3 are reported. They refer to a C++ implementation running on a standard workstation with an Intel Core i5-3570k@3.40GHz processor. The most demanding operation is the matching between the test point cloud transformed to standard pose and the models of every subject in the training set, which takes 250 ms for performing 50 comparisons. The overall frame rate is then of about 2.8 fps, which sug-

Table 4.5: Runtime performance of the algorithms used for the point cloud matching method.

	time (ms)
Face detection	42.19
Body segmentation	3.03
Transformation to standard pose	0.41
Filtering and smoothing	56.35
ICP and fitness scores computation	254.34

gests that this approach could be used in a real time scenario with further optimization and with a limited number of people in the database.

4.2.8 Conclusions

In this section, we proposed an efficient method for composing 3D models of persons while moving freely. For overcoming the problem of the different poses a person can assume, we exploited the skeletal information provided by a skeletal tracking algorithm for warping persons point clouds to a standard pose, such that point clouds coming from different frames can be merged to compose a model. We showed how these models can be effectively used for the long-term re-identification task by means of a rigid comparison based on a ICP-like fitness score. We also compared the proposed technique with other state of the art approaches in terms of re-identification results on two newly created datasets targeted to RGB-D re-identification. Moreover, we proposed a method for combining skeleton lengths and body shape information to further improve re-identification results. Experimental results show that shape information can be used for effectively re-identifying subjects in a non-collaborative scenario, reaching performances near those of face recognition if PCM is combined with a classification of skeleton lengths. More accurate depth sensors and skeletal tracking algorithms would be helpful for obtaining more correct and realistic 3D models, so that a mobile robot could compose 3D models which could be used for re-identification by both robots and humans.

Chapter 5

Action Recognition

In recent years, robotics perception has grown very fast and has made possible applications unfeasible before. This success has been fostered by the introduction of RGB-D sensors with good resolution and framerate like those introduced in Section 2.4 and open source software for robotics development [96]. Thanks to these progresses, we can now think about robots capable of smart interaction with humans. One of the most important skills for a robot interacting with a human is the ability to recognize what the human is doing. For instance, a robot with this skill could assist elderly people by monitoring them and understanding if they need help or if their actions can lead to a dangerous situation.

As a basis for this high level skill, robust and fast algorithms for people detection, tracking and re-identification are needed, such as those presented in Section 3 and 4, which allow to collect RGB-D information of a person (or of more persons) over time.

In this chapter, we propose a method to compute 3D motion flow of points belonging to person's point clouds and to encode it in a 3D grid-based descriptor which could be classified to recognize actions. We also compare this technique with an approach which classifies information collected by a skeletal tracker as those presented in Appendix A on the newly created IAS-Lab Action dataset.

In Section 5.1, we review existing work, while the 3D motion flow technique is presented in Section 5.2 and the skeleton-based approach is outlined in Section 5.3. Section 5.4 reports how to compute descriptors of frame sequences and Section 5.5 contains experiments on two datasets. Conclusions are drawn in Section 5.6.

5.1 Related Work

Human action recognition is an active research area in computer vision. First investigations about this topic began in the seventies with pioneering studies accomplished by Johansson [52]. From then on, the interest in the field rapidly increased, motivated by a number of potential real-world applications such as video surveillance, human-computer interaction, content-based video analysis and retrieval. Moreover, in recent years, the task of recognizing human actions has gained increasingly popularity thanks to the emergence of modern applications such as motion capture and animation, video editing and service robotics. Most of the works on human action recognition relies on information extracted from 2D images and videos [94]. These approaches mostly differ in the features representation. We can distinguish between methods which exploit global traits of the human body and those which extract local information from the data. Popular global representations are edges [23], silhouettes of the human body [15, 101, 129], 2D optical flow [2, 34, 127] and 3D spatio-temporal volumes [15, 54, 66, 101, 106, 129]. Conversely, effective local representations mainly refer to [32, 55, 61, 62, 86, 105, 122].

The recent spread of inexpensive RGB-D sensors has paved the way to new studies in this direction. These 3D recognition systems are inherently related to the acquisition of a more informative input content with respect to their 2D counterparts and thus it is expected they outperform the traditional approaches. The first RGB-D related work is signed by Microsoft Research [65]. In [65], a sequence of depth maps is given as input to the system. Next, the relevant postures for each action are extracted and represented as a bag of 3D points. The motion dynamics are modeled by means of an action graph and a Gaussian Mixture Model is used to robustly capture the statistical distribution of the points. Subsequent studies mainly refer to the use of three different technologies: Time of Flight cameras [45, 46], motion capture systems [58, 88, 115] and active matricial triangulation systems (i.e.: Kinect-style cameras) [16, 63, 72, 85, 93, 113, 114, 125, 128, 131, 133]. We can further distinguish these works by means of the features used during the recognition process. The most used features are related to the extraction of the skeleton body joints [16, 88, 113–115, 125, 128]. Usually, these approaches first collect raw information about the body joints (e.g.: spatial coordinates, angle measurements). Next, they summarize the raw data into features through specific computations, in order to characterize the posture of the observed human body. Differently from the other joints-related publications, [88] distinguishes itself by computing features which carry a physical meaning. Indeed, in [88], a *Sequence of Most Informative Joints* (SMIJ) is computed based on measures like the

mean of joint angles, the variance of joint angles and the maximum angular velocity of body joints.

Other popular features are the result of the extension to the third dimension of typical 2D representations. Within this feature category, we shall perform a further distinction between local representations and global representations. Features in [72, 85, 131, 133] are actually local representations since they aim to exploit *Space-Time Interest Points* (STIPs) [60–62] by extending it with the depth information. Examples of global representations in the 3D field can be found in [45, 46, 93]. In [93], Popa *et al.* propose a Kinect-based system able to continuously analyze customers' shopping behaviours in malls. Silhouettes for each person in the scene are extracted and then summarized by computing moment invariants. In [45, 46], a 3D extension of 2D optical flow is exploited for the gesture recognition task. Holte *et al.* compute optical flow in the image using the traditional Lukas-Kanade [70] method and then extend the 2D velocity vectors to incorporate also the depth dimension. At the end of this process, the 3D velocity vectors are used to create an annotated velocity cloud. 3D Motion Context and Harmonic Motion Context serve the task of representing the extracted motion vector field in a view-invariant way. With regard to the classification task, [45] and [46] do not follow a learning-based approach, instead a probabilistic distance classifier is proposed in order to identify which gesture best describes a string of primitives. Note that [46] differs from [45] because the optical flow is estimated from each view of a multi-camera system and is then combined into a unique 3D motion vector field.

Finally, works in which trajectory features are exploited [58, 63] recently emerged. While in [63] trajectory gradients are computed and summarized, in [58] an action is represented as a set of subspaces and a mean shape.

From the application point of view [113, 114], and [85] are targeted to applications in the personal robotics field, while [65] and [128] are addressed to human-computer interaction and gaming applications. Finally, [131] and [93] are primarily addressed to applications in the field of video surveillance.

Unlike [45] and [46], which compute 2D optical flow and then extend it to 3D, we propose a method to compute the motion flow directly on 3D points with color. From the estimated 3D velocity vectors, a motion descriptor is derived and a sequence of descriptors is concatenated and classified by means of Nearest Neighbor.

5.2 3D Motion Flow

Optical flow is a powerful cue to be used for a variety of applications, from motion segmentation to structure-from-motion passing by video stabilization. As reported in Section 5.1, some researchers proved its usefulness also for the task of action recognition [2, 34, 127]. The most famous algorithm for optical flow estimation was proposed by Lukas and Kanade [70]. The main drawbacks of this approach are that it only works for highly textured image patches and, if repeated for every pixel of an image, it results to be highly computational expensive. Moreover, 2D motion estimation in general has the limitation to be dependent on the viewpoint and closer objects appear to move faster because they appear bigger in the image.

When depth data are available and aligned with a RGB/intensity image, the optical flow computed in the image can be extended to 3D by looking at the corresponding points in the depth image or point cloud [45, 46]. This procedure allows to compute 3D velocity vectors, thus overcoming some of the limitations of 2D-only approaches, such as viewpoint and scale dependence. However, the motion estimation process is still performed on the RGB image, and it does not exploit the available 3D information for obtaining a better estimate. Moreover, the computational burden remains high.

In this work, we introduce a novel technique for computing 3D motion of points in the 3D-color space directly. This method consists in estimating correspondences between points of clouds belonging to consecutive frames. Our approach is fast and able to overcome some singularities of optical flow estimation in images by relying also on 3D points coordinates. Moreover, it is applicable to any point cloud containing XYZ and RGB information, and not only to those derived from a 2D matrix of depth data (projectable point clouds).

5.2.1 3D Flow Estimation Pipeline

Given two point clouds (called *source* and *target*) containing 3D coordinates and RGB/HSV color values of an object of interest (in this work, a person), the following pipeline is applied:

1. correspondence finding: for every point of the target point cloud, we select K nearest neighbors in the source point cloud in terms of Euclidean distance in the XYZ space; among the resulting points, we select the nearest neighbor in terms of HSV coordinates. We preferred HSV to RGB because it is more perceptually uniform. If $\mathcal{N}_{\mathbf{p}_i^{target}}$ is the set of K nearest neighbors in XYZ space in the source point cloud to the point $\mathbf{p}_i$ in the target point cloud, then $\mathbf{p}_i^{target}$ is said to match

with

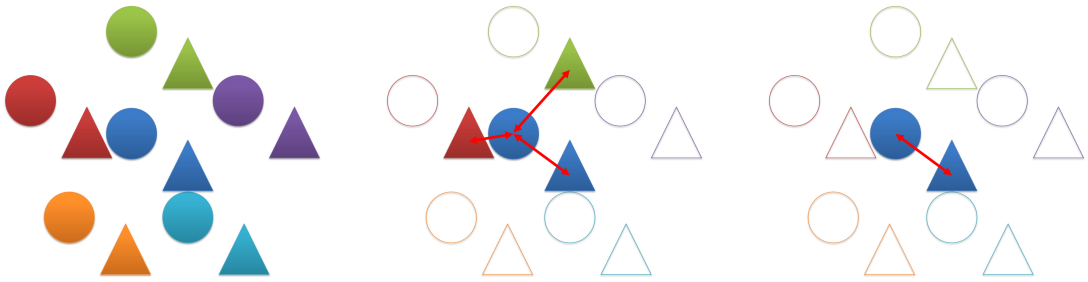
$$\mathbf{p}_*^{source} = \underset{\mathbf{p}_i^{source} \in \mathcal{N}_{\mathbf{p}_i^{target}}}{\operatorname{argmin}} \; d_{HSV}\left(\mathbf{p}_i^{target}, \mathbf{p}_i^{source}\right), \tag{5.1}$$

where d_{HSV} is the distance operator in the HSV space. The number of neighbors K is a function of the point cloud density. In this work, we filter the point clouds to have a voxel size of 0.02 m and we set K to 50. An illustration of this method with $K = 3$ is reported in Figure 5.1.

2. outlier rejection by means of reciprocal correspondences: this method consists in estimating correspondences from target to source and from source to target. Then, points which match in both directions are kept.

3. computation of 3D velocity vectors $\mathbf{v}_i$ for every match i as spatial displacement over temporal displacement of corresponding 3D points $\mathbf{p}_i$ from target and source:

$$\mathbf{v}_i = \left(\mathbf{p}_i^{target} - \mathbf{p}_i^{source}\right) / \left(t_i^{target} - t_i^{source}\right) \tag{5.2}$$



(a) Target (circles) and source (triangles) point clouds

(b) 3-NN in XYZ space

(c) NN in HSV space

Figure 5.1: Illustration of the matching process between points of two point clouds represented by circles and triangles, respectively. In particular, a point of the target point cloud (blue circle) is matched with the corresponding point of the source point cloud (blue triangle) after (b) K-Nearest Neighbor (K-NN) in XYZ space with K=3 and (c) Nearest Neighbor (NN) in HSV space among the points obtained at the K-NN stage.

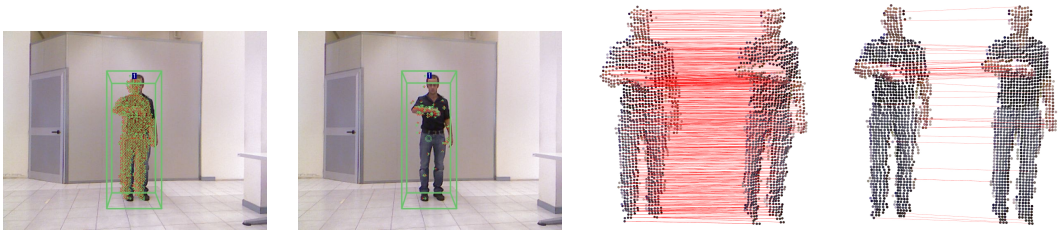
4. further outlier rejection: points with 3D velocity magnitude $\|\mathbf{v}_i\|$ below a threshold are discarded. Isolated moving points (not near to other moving points) are also deleted. In particular, points moving faster than 0.3 m/s are retained and a moving point is considered to be isolated if none of its neighbors moves faster than 0.75 m/s.

The reciprocal correspondence technique for outlier rejection can be considered as a 3D extension of the *Template Inverse Matching* method [67], which has been widely

used to estimate the goodness of 2D optical flow estimation. The constraints we apply on the flow magnitude and on the proximity to other moving points are thought to remove spurious estimates which can be generated from the noise inherent in the depth values.

In this work, we segment persons' point clouds from the rest of the scene by means of the people detection and tracking method for RGB-D data described in Section 3, then we apply the flow estimation algorithm to the detected persons' clusters.

In Figure 5.2, we report two consecutive RGB frames of a person performing the *Check Watch* action from the IAS-Lab Action dataset. Green arrows show magnitude and direction of the estimated flow when reprojected to the image. It can be noticed how outlier rejection manages to remove most of the noisy measurements, while preserving the real motion at the right arm position.



(a) Before OR (b) After OR (c) Before OR (d) After OR

Figure 5.2: Example of 3D flow estimation results reprojected to the image (a-b) for action *Check watch*. Flow is visualized as green arrows in the image, (a) before and (b) after outlier removal (OR). Also correspondences between point clouds are visualized (c) without and (d) with outlier removal.

5.2.2 Motion Flow Feature Descriptors

In this section, we describe the frame-wise descriptors we extract for describing actions with the 3D motion flow estimated as reported in Section 5.2.1.

Gridded Flow Descriptor

In order to compute a descriptor accounting for direction and magnitude of motion of every body part, we center a 3D grid of suitable dimensions around a person point cloud. This grid divides the space around the person into a number of cubes. In Figure 5.3, a person point cloud is reported, together with the 3D grid which divides its points into different clusters represented with different colors. The size of the grid is proportional to the person height in order to contain the whole limbs motion and to

make the flow descriptor person-independent. For a person 1.75 m tall, the grid results to be 2 m width, 2.3 m tall and 1.8 m deep.

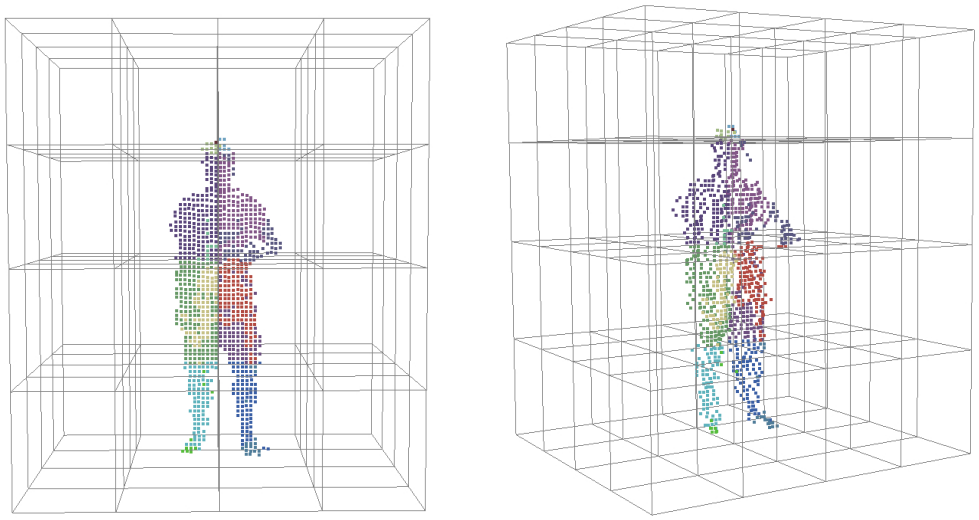


Figure 5.3: Two different views of the computed 3D grid: 4 partions along the x, y and z axis are used.

MEANFLOW and SUMFLOW

For every cube of the grid, we extract flow information from all the points inside the cube. In this work, we propose two descriptors: the former computes the mean 3D motion vector from every cube, while the latter computes the sum of the motion vectors of the current cube. We will refer to these desciptors with the name *MEANFLOW* and *SUMFLOW*, respectively. For both of them, the resulting vectors for all the cubes are concatenated into a single descriptor which is then L2-normalized for making the descriptor invariant to the speed at which an action is performed. In this work, the grid is divided into four parts in every dimension, thus the total number of cubes is $C = 64$. If $x_i^{sF}, y_i^{sF}, z_i^{sF}$ are the coordinates of the flow sum vector for the $i-th$ cube, the SUMFLOW descriptor can be written as

$$\mathbf{d}_{SUMFLOW} = \left[x_1^{sF}\ y_1^{sF}\ z_1^{sF}\ \ldots\ x_C^{sF}\ y_C^{sF}\ z_C^{sF}\right]. \tag{5.3}$$

The MEANFLOW can be expressed in a similar form, but with the mean flow vectors instead of the sum vectors **sF**. As we will see in Section 5.5, the MEANFLOW descriptor is used in combination with a pipeline for computing motion flow which does not perform the last step of outlier rejection described in Section 5.2.1. If also the last step would be performed, noise velocity vectors passing through the outlier

rejection would have too much weight in the mean flow computation.

5.3 3D Pose

As we reported in Section 5.1, most of the works on action recognition exploiting consumer RGB-D sensors are based on classifying how person's 3D pose varies over time. For doing this, they use the depth-based skeletal trackers described in Appendix A. Thus, in our experiments, we also tested descriptors based on skeleton information in order to compare them with our technique based on 3D motion flow.

5.3.1 Skeleton Descriptors

NiTE skeletal tracker (Appendix A.2) provides $N = 15$ joints from *head* to *foot*. Each joint is described by position (a point in 3D space) and orientation (a quaternion). On these data, we perform two kinds of normalization: the former scales the joints positions in order to report the skeleton to a standard height, thus achieving invariance to people height, the latter makes every feature to have zero mean and unit variance. Starting from the normalized data, we extracted three kinds of descriptors: a first skeleton descriptor ($\mathbf{d}_P$) is made of the set of joints positions concatenated one to each other; for the second one ($\mathbf{d}_O$), normalized joints orientations are gathered. Finally, we tested also a descriptor ($\mathbf{d}_{TOT}$) concatenating both position and orientation of each normalized joint:

$$\mathbf{d}_P = [x_1\ y_1\ z_1\ \ldots\ x_N\ y_N\ z_N]\,, \tag{5.4}$$

$$\mathbf{d}_O = \left[q_1^1\ q_1^2\ q_1^3\ q_1^4\ \ldots\ q_N^1\ q_N^2\ q_N^3\ q_N^4\right], \tag{5.5}$$

$$\mathbf{d}_{TOT} = \left[\mathbf{d}_P^1\ \mathbf{d}_O^1\ \ldots\ \mathbf{d}_P^N\ \mathbf{d}_O^N\right]. \tag{5.6}$$

5.4 Sequence Descriptor

Since an action actually represents a sequence of movements over time, the use of multiple frames can provide more discriminant information to the recognition task with respect to approaches in which only a single-frame classification is performed. For this reason, we compose a single descriptor from every sequence of frames to be classified. In particular, we select a fixed number of frames evenly spaced in time from every pre-segmented sequence and we concatenate the single-frame descriptors

to form a single sequence descriptor. Thanks to this approach, we take into account the temporal order in which the single frame descriptors occur. As it will be highlighted in the experiments section, we obtained the best results by choosing 30 frames for composing sequence descriptors. That is, the total dimension of these descriptors is $F * C * 3 = 30 * 64 * 3 = 5760$, where F is the number of frames extracted from a sequence, C is the number of cubes of the descriptor grid and 3 is the dimension of every cube descriptor.

5.5 Experiments

5.5.1 Nearest Neighbor Classification

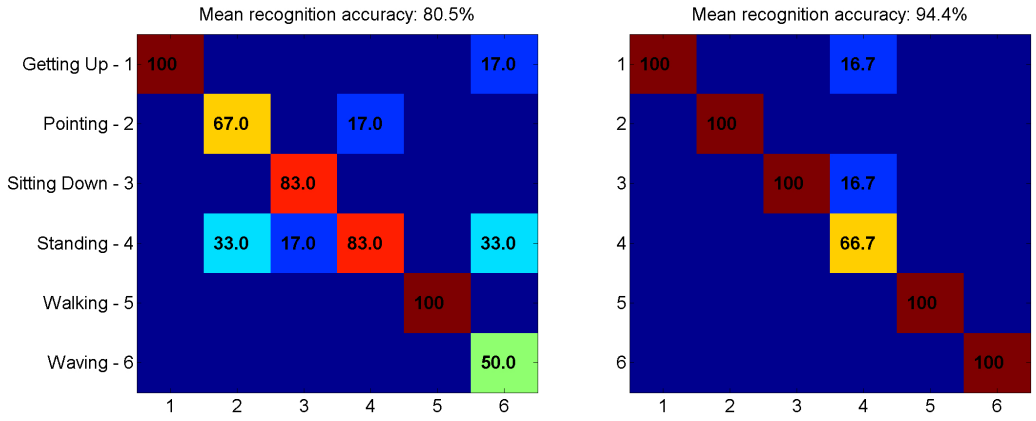
For validating the descriptors we presented in this work, we adopted a Nearest Neighbor classification between the sequences in a testing set and those of a training set. In particular, at the training stage, we learn how to recognize actions by storing descriptors computed from labeled frame sequences containing one action each. This implies to use a dataset where videos have been pre-segmented by a human, which is the case of most action recognition datasets. For classifying a test sequence, we compute its sequence descriptor and compare it with those of the training set by means of the Euclidean distance and we assign to it the action label of the nearest training descriptor. It is worth noting that the whole recognition procedure remains the same if we substitute Nearest Neighbor with a Support Vector Machine classifier, which scales better when increasing the number of training examples.

5.5.2 Results

We first evaluated our 3D motion flow descriptor on an action dataset with six types of human actions: *getting up, pointing, sitting down, standing, walking, waving*. Each action is performed once by six different actors and recorded from the same point of view. Every action is already segmented out into a video containing only one action. Each of the segmented video samples spans from about 1 to 7 seconds. This dataset was acquired with a Microsoft Kinect in order to provide RGB and depth data. No skeleton information was collected, thus the skeleton-based descriptors could not be tested.

For assigning an action label to every test sequence, we performed the Nearest Neighbor classification described in Section 5.5.1 with a *leave-one-person-out* approach, that is we trained the actions classifiers on the videos of all the persons except

one and we tested on the video containing the remaining person. Then, we repeated this procedure for all the people and we computed the mean of all the rounds for obtaining the mean recognition accuracy. In Figure 5.4 (b), we report the confusion matrix obtained on this dataset with our approach based on the *SUMFLOW* descriptor when using 10 frames evenly spaced in time for composing the sequence descriptor. The mean accuracy is 94.4% and the only errors occur for recognizing the *standing* action, which is sometimes confused with the *getting up* and *sitting down* actions. The accuracy we obtained with this approach is considerably higher (14% more) than that obtained by using another approach (Figure 5.4 (a)) which computes the *MEANFLOW* descriptor without performing the last step of outlier rejection described in Section 5.2.1 and which selects the central frames of every sequence for composing the sequence descriptor, thus encoding only the central part of an action.

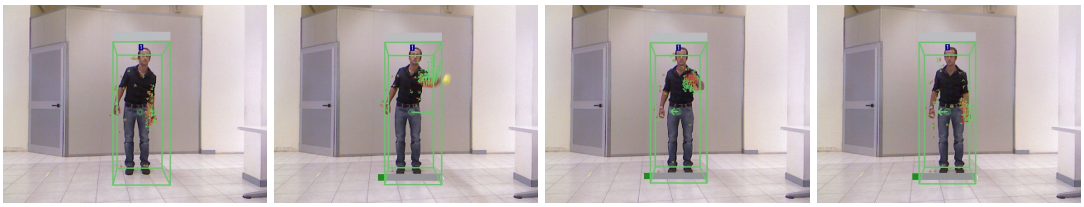


(a) (b)

Figure 5.4: Confusion matrix obtained on a dataset of six actions with the two approaches described in the text.

In order to better evaluate the single contributions of this work and compare motion flow descriptors with skeleton-based descriptors, we performed tests on the newly acquired IAS-Lab Action dataset (Appendix B.3.1), which contains 15 actions performed by 12 people. In Figure 5.5, we show an example of 3D flow estimation for some key frames of the *Throw from bottom up* action. We adopted the same leave-one-person-out approach described above for computing the recognition accuracy.

For this dataset, we tested both the *SUMFLOW* and the *MEANFLOW* descriptors with the same approach to compose the sequence descriptors. In particular, 30 frames evenly spaced in time have been selected from every action sequence to compute the sequence descriptor. It can be noticed how the *MEANFLOW* descriptor (Figure 5.6 (a)) reaches considerably lower recognition accuracy with respect to the *SUMFLOW*



(a) (b) (c) (d)

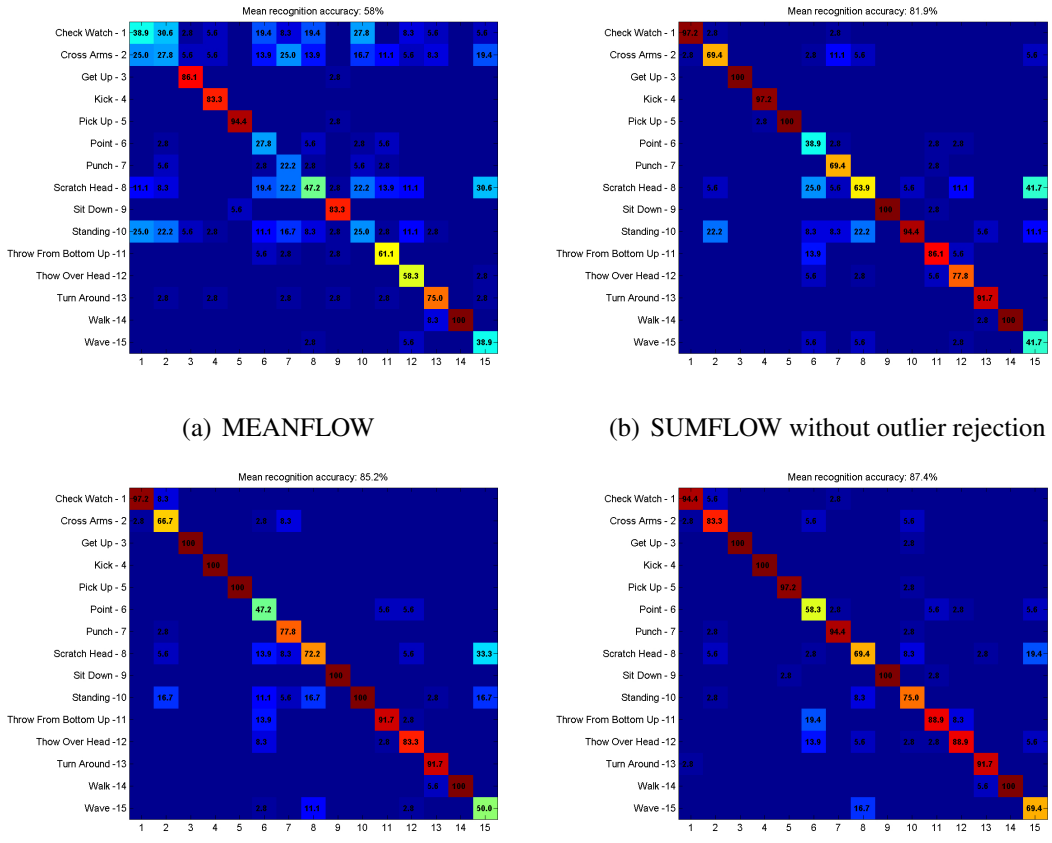
Figure 5.5: Example of 3D flow estimation for some key frames of the *Throw from bottom up* action of the IAS-Lab Action dataset.

(Figure 5.6 (c)), 58% against 85.2%. We also report the results obtained when the outlier rejection described in the pipeline section is not performed (Figure 5.6 (b)), thus leading to a drop of 3.3% in perfomance.

A further improvement can be obtained by performing a Principal Component Analysis (PCA) projection of the frame descriptors. The descriptor length reduced from 192 to 61 elements when retaining the 99% of information. This reduction allowed a faster comparison between descriptors and a reduction of the noise influence. We show in Figure 5.6 (d) the confusion matrix obtained when performing this PCA projection. The overall accuracy increased by 2.2%, reaching 87.4%, which can be considered as a very good score given that the people used as test set were not present in the training set. These results prove that 3D local motion is highly discriminative for the action recognition task. We can notice that most of errors occurred for the action *Point* and *Wave*, and in particular that actions with little motion can be confused with the *Standing* action. It is worth noting that, even if the *Standing* action is not included in all the datasets we reported in the datasets section, it is very important for the task of action detection: an algorithm able to reliably distinguish this action from the rest could be easily extended to detect actions from an online stream, rather than needing pre-segmented sequences.

For what concerns the skeleton-based descriptors, we report the confusion matrices on the IAS-Lab Action dataset relative to the three descriptors introduced in Section 5.3.1. The classification of links orientation ($\mathbf{d}_O$) and joints position ($\mathbf{d}_P$) descriptors reaches, respectively, 55.9% and 76.7% accuracy, while the combined use of links angles and joints position ($\mathbf{d}_{TOT}$) leads to a result which is in the middle of the two, namely 66.9%. For $\mathbf{d}_P$, the skeleton-based descriptor obtaining the best results, the most of recognition errors are due to the fact that the *Standing*, *Turn Around* and *Walk* actions are featured by very similar skeleton poses.

These results prove that 3D local motion is highly discriminative for the action recognition task and can also lead to better results than those which can be obtained by



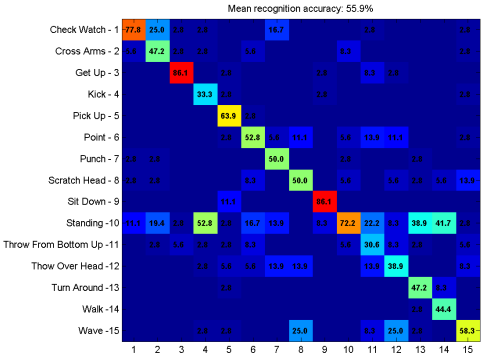
(a) MEANFLOW

(b) SUMFLOW without outlier rejection

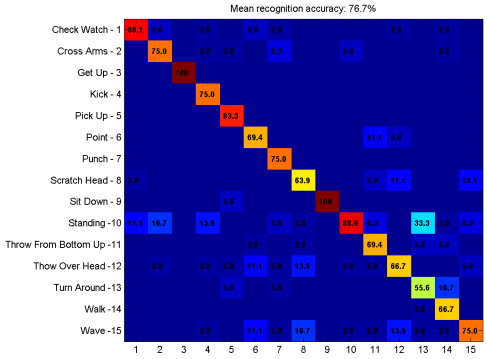
(c) SUMFLOW with outlier rejection

(d) SUMFLOW after PCA projection

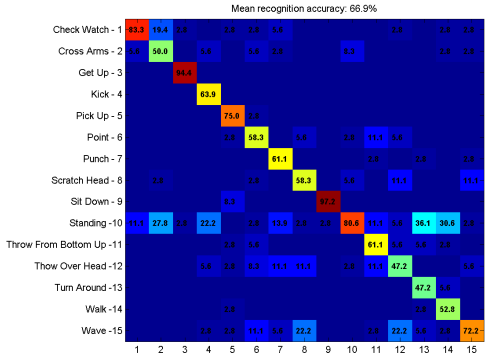
Figure 5.6: Confusion matrix obtained with the MEANFLOW and some variants of the SUMFLOW descriptor: a) MEANFLOW, b) SUMFLOW without outlier rejection, c) SUMFLOW with outlier rejection, d) SUMFLOW after projection on a PCA subspace.



(a) Skeleton links orientation descriptor $\mathbf{d}_O$



(b) Skeleton joints position descriptor $\mathbf{d}_P$



(c) Skeleton joints position and links orientation descriptor $\mathbf{d}_{TOT}$

Figure 5.7: Confusion matrix obtained on the IAS-Lab Action dataset with skeleton-based descriptors.

exploiting skeleton information. This is also due to the fact that the noise intrinsic in a consumer depth camera and some challenging human poses can make the skeleton to be sometimes unreliably estimated.

In Figure 5.8, we report the mean recognition accuracy obtainable on the IAS-Lab Action dataset when using the *SUMFLOW* frame-wise descriptor and varying the number of frames used for composing the sequence descriptor. It can be noticed how the accuracy rapidly increases until 5 frames per sequence and continues to considerably improve until 30 frames are used. By comparing the curves with and without PCA projection, we can also observe that the accuracy obtainable without PCA and 30 frames per sequence can be obtained with PCA and half (15) of the frames, thus allowing faster comparison between sequence descriptors.

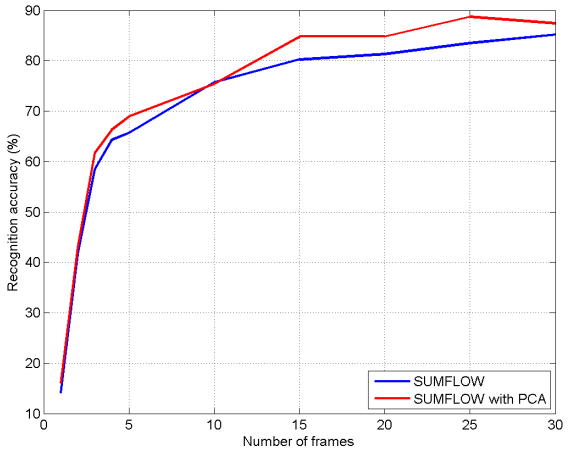


Figure 5.8: Mean recognition accuracy when varying the number of frames used for composing the sequence descriptor.

5.5.3 Runtime Performance

In terms of runtime performance, the skeleton description is very fast to compute, because all the information is already provided by the skeletal tracker algorithm.

In Table 5.1, the computation times needed by every step of our motion flow algorithm for processing one frame are shown. These timings are measured on a notebook with an Intel i7-620M 2.67Ghz processor and 4GB of RAM. The most demanding operation is the matching between two point clouds, that is the search for correspondences, which takes 0.015 s for initializing the octree used for the search and 0.23 s for the actual matching. On the contrary, the outlier removal and the frame descriptor computation are very fast operations. The overall runtime is then of about 0.25

Table 5.1: Runtime for every step of our action recognition algorithm (seconds).

Octree initialization	0.0150
Point clouds matching	0.2300
Flow vectors computation	0.0001
Outlier removal	0.0004
Descriptor computation	0.0003
Total	**0.2458**

s, meaning a framerate of 4 frames per second. The nearest neighbor classification is also rapidly performed in 0.0015 s if we consider the case where the descriptors are reduced in length by means of PCA projection.

The runtime of our algorithm is highly dependent on the number of points belonging to the persons cluster cloud. In this work, we filtered the Kinect point cloud with a voxel grid filter with voxel size of 0.02 m, thus obtaining person's clusters of about 1000 points. If we use the same voxel size used for people detection and tracking in Section 3 (0.06 m), the medium person cluster size is of 400 points and our whole algorithm runs at 20 fps. However, for achieving a good accuracy in finding correspondences between point clouds, the voxel size had to be reduced with respect to that chosen for people tracking purposes.

5.6 Conclusions

In this chapter, we presented a novel method for real-time estimation of 3D motion flow from colored point clouds and a complete system for human action recognition which exploits this motion information. In particular, we also proposed two 3D grid-based descriptors as frame-wise motion descriptors and a sequence descriptor which is then classified with a Nearest Neighbor classifier for performing action recognition. Moreover, we compared this method with an action recognition technique which classifies skeleton information obtained with a skeletal tracker. We also presented results on a new and publically released dataset with high variability in the number of people performing the actions and providing both RGB-D data and skeleton pose for every frame. The tested 3D flow technique reported very good results in classifying all the actions of the dataset, reaching 85.2% of accuracy and outperforming of 7.5% the skeleton-based method. A further improvement of 2.2% in accuracy is obtained if performing a PCA projection of the frame descriptors.

As future work, we schedule to implement an automatic method for online segmentation of action sequences, so that our algorithm could run in real time on a mobile

robot without no need for human intervention.

Chapter 6

Conclusions

In a near future, we can imagine that robots will be more and more present in every house and will perform services useful to the well-being of humans. They will need to recognize people and their actions and interact with them in order to help them in the daily tasks and to learn new tasks from them in a natural way. This thesis wanted to tackle these problems and to propose solutions which could be both robust and efficient in order to be directly applied to mobile robotics. For this reason, we developed approaches to people tracking, re-identification and action recognition which exploit both depth and color data, which can be jointly acquired for example with recently introduced consumer RGB-D sensors.

In Chapter 3, we presented a fast and robust algorithm for multi-people tracking with RGB-D data designed to be used on mobile service robots. It can track multiple people with state-of-the-art accuracy and beyond state-of-the-art speed without relying on GPU computation, which has high power requirements and is not always available in embedded systems. Morever, we perfomed tests on the newly introduced KTP dataset, the first RGB-D dataset with 2D and 3D ground truth for evalutating accuracy and precision of people tracking algorithms also in terms of 3D coordinates and we found that the average 3D error of our people tracking system is lower enough for robotics applications. From the extensive evaluation of our method on this dataset and on another public dataset acquired from three static Kinects, we demonstrated that Kinect-style sensors can replace sensors previously used for indoor people tracking from service robotics platforms, such as stereo cameras and laser range finders, while reducing the required computational burden. Our people detection software has been publically released as part of the *people* module of the Point Cloud Library [102] and as part of ROS-Industrial Human Tracker, whose aim is to provide robust and fast people detection and tracking algorithms for industrial environments. Moreover, it served

as a basis for OpenPTrack[1], an open source library for scalable and low-latency people tracking. Future works will go in the direction of extending tracking to networks with a high number of agents/robots cooperating with each other.

In Section 4.1, a novel approach to short-term people re-identification in RGB-D data has been presented. This approach builds on the assumption that very stable keypoints can be detected on human targets by means of a skeletal tracker, and exploited to evaluate signatures by means of 2D and 3D feature extractors. This idea was developed considering several features and matching methods and overcoming the instabilities that still affect skeletal trackers. The novel re-identification system presented has been extensively tested using both video-surveillance datasets, for comparing this novel approach to the state of the art, and newly created datasets that are capable of highlighting the great advantages offered by our approach. This re-identification method is particularly suited for robotic applications dealing with humans, since it offers superior performance, it exploits sensors commonly available on most autonomous robots and runs in real-time. As a future work, we envision to integrate this short-term re-identification technique within the data association scheme used for the people tracking approach described in Chapter 3. This combination should lead to lowering identity switches when people undergo full occlusions or they are seen from a different camera.

In Section 4.2, we proposed an efficient method for composing 3D models of persons while moving freely. For overcoming the problem of the different poses a person can assume, we exploited the skeletal information provided by a skeletal tracking algorithm for warping persons' point clouds to a standard pose, such that point clouds coming from different frames can be merged to compose a model. We showed how these models can be effectively used for the long-term re-identification task by means of a rigid comparison based on a ICP-like fitness score. We also compared the proposed technique with other state-of-the-art approaches in terms of re-identification results on two newly created datasets targeted to RGB-D re-identification. Moreover, we proposed a method for combining skeleton lengths and body shape information to further improve re-identification results. Experimental results show that shape information can be used for effectively re-identifying subjects in a non-collaborative scenario, reaching performances near those of face recognition if Point Cloud Matching is combined with a classification of skeleton lengths. More accurate depth sensors and skeletal tracking algorithms would be helpful for obtaining more correct and realistic 3D models, so that a mobile robot could compose 3D models which could be used for re-identification by both robots and humans.

[1] `http://openptrack.org`.

In Chapter 5, we presented a novel method for real-time estimation of 3D motion flow from colored point clouds and a complete system for human action recognition which exploits this motion information. In particular, we also proposed two 3D grid-based descriptors as frame-wise motion descriptors and a sequence descriptor which is then classified with a Nearest Neighbor classifier for performing action recognition. Moreover, we compared this method with an action recognition technique which classifies skeleton information obtained with a skeletal tracker. We also presented results on a new and publically released dataset with high variability in the number of people performing the actions and providing both RGB-D data and skeleton pose for every frame. The tested 3D flow technique reported very good results in classifying all the actions of the dataset, reaching about 85% of accuracy and outperforming the skeleton-based method. A further improvement in accuracy has been obtained by performing a PCA projection of the frame descriptors. As future work, we schedule to implement an automatic method for online segmentation of action sequences, so that our algorithm could run in real time on a mobile robot without the need for human intervention.

The results we obtained in this thesis are encouraging and prove the effectiveness of combining RGB-depth approaches for reaching robust performance while preserving efficiency. Since consumer RGB-D sensors are going to be more and more precise and resoluted, we expect these techniques to obtain even better accuracy in the future, so that they could be effectively applied to mobile robotics. For what concerns the proposed person re-identification methods, their future accuracy and applicability will depend also on improvements of skeletal tracking algorithms we exploited as a basis for them.

Appendices

Appendix A

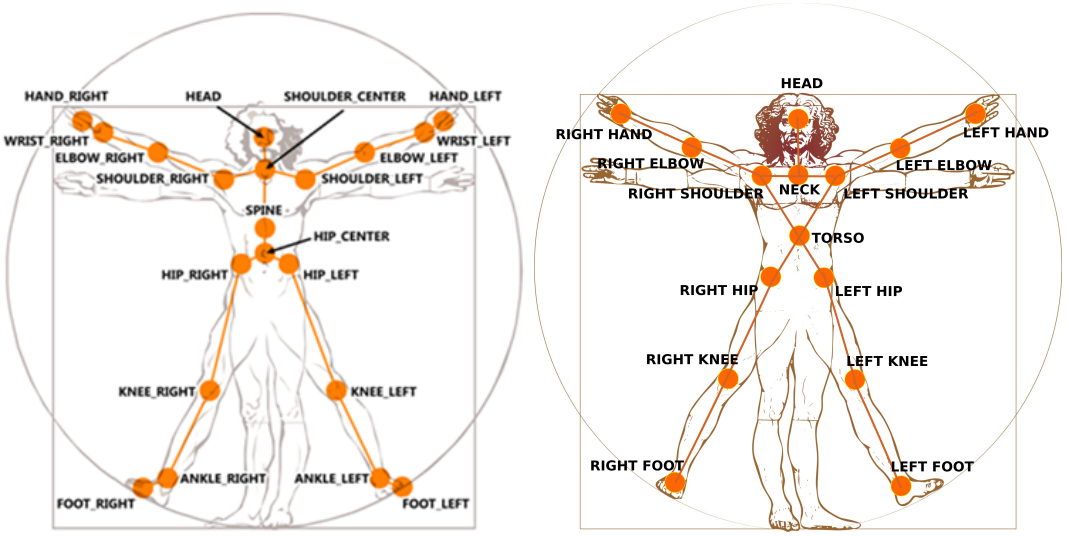
Depth-based Skeletal Trackers

Together with consumer RGB-D sensors, two software development kits have been released by Microsoft (*Kinect SDK*) and OpenNI (*OpenNI SDK*) for developing middlewares and applications. Kinect[1] and NiTE[2] middlewares, built on top of Kinect and OpenNI SDK respectively, provide user detection and tracking functionalities working with static sensors. In particular, both of them implement skeletal tracking algorithms which are able to estimate the position of several joints of the human body. Since these techniques are based on depth data, they can reach very good performances in terms of precision and framerate with respect to their image-based counterparts and they are invariant to illumination changes. In our work, we tested both these skeletal tracking algorithms and exploited them for people re-identification and action recognition. The next paragraphs will give an overview of these methods and will highlight the main differences between them.

A.1 Kinect Skeletal Tracker

The Kinect skeletal tracker[3] [107] tracks at 30 fps the 20 joints of the human body illustrated in Figure A.1 (a). It provides both joints position and links orientation. Orientation can be represented in Kinect camera coordinates (*absolute orientation*) or based on a bone relationship defined on the skeleton joint structure (*hierarchical orientation*). It allows to obtain a precise estimation from a single depth image, but it assumes that people are facing the camera, thus the skeleton is flipped left-right if the person is seen from the back side. The maximum working range of the skeletal tracking

[1] `http://www.microsoft.com/en-us/kinectforwindows/develop`.
[2] `http://www.primesense.com/solutions/nite-middleware`.
[3] In this work, we used version 1.6 of Kinect SDK.



(a) Kinect skeletal tracker (b) NiTE skeletal tracker

Figure A.1: Position and names of the human skeleton joints estimated with (a) Kinect skeletal tracker and (b) NiTE skeletal tracker.

is between 0.8 and 4 meters, but the best results can be obtained in the range 1.2-3.5 meters. Bad skeleton estimations are provided mainly when the person is partially occluded or out of the image to the right or left side, or it is farther than 3.5 m from the sensor. In Figure A.2 we report two examples of these situations. Depending on the

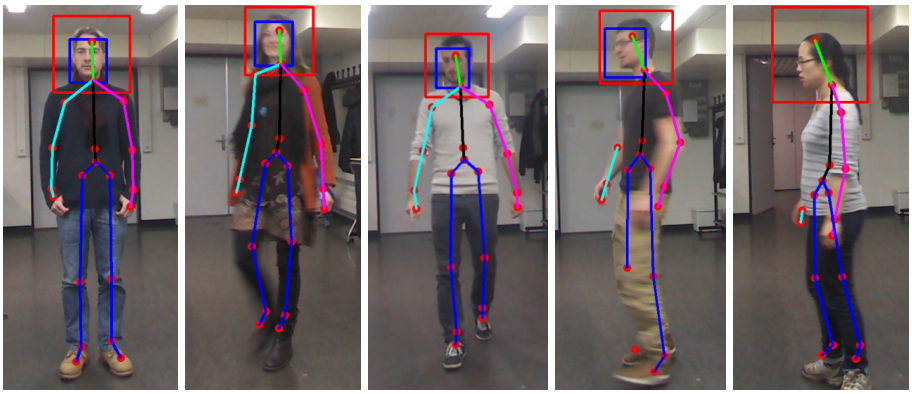
(a) Out of image (b) Too far

Figure A.2: Example of two situations in which Kinect skeletal tracker estimates wrong skeletons, caused by the target being only partially visible (a) or too far from the sensor (b).

quality of the segmented target and the level of occlusion, the skeletal tracker might not detect all the joints: in this case, some of them are marked as non tracked, meaning their location is not reliable, but they are anyway part of the whole skeleton.

In this work, skeleton estimation is also used for helping in reducing the time needed by face detection. In fact, faces are searched only in an image patch around

the head joint estimated by a skeletal tracker. With this method, face detection can be performed at about 20-30 Hz depending on the distance of the person from the camera. In Figure A.3, some examples of skeleton estimation obtained with Kinect skeletal tracker are reported, together with the region where a face is searched for (red rectangle) and the detected face (blue rectangle). Only the joints marked as tracked by the



(a) (b) (c) (d) (e)

Figure A.3: Example of skeleton estimation obtained with Kinect skeletal tracker and face detection guided by the skeleton tracking. No face could be detected in (e).

algorithm are plotted. It can be seen that the skeletal tracker behaves reasonably well even when part of the human body is not visible ((d) and (e)).

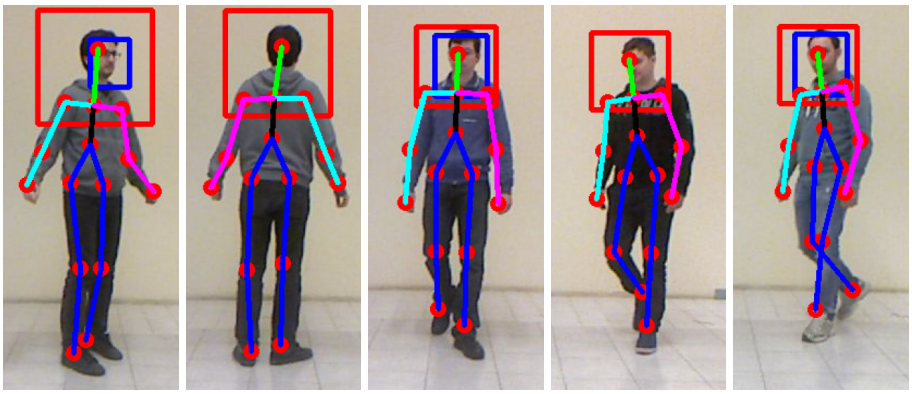
A.2 NiTE Skeletal Tracker

NiTE skeletal tracker[4] tracks the 15 joints of Figure A.1 (b) (no hands and feet) and exploits the information from multiple depth frames to improve the tracking performance.

As Kinect skeletal tracker, it works on CPU at 30 Hz, tracks people in the range 1.2-3.5 m and provides a label for every joint stating if it is tracked or inferred (not visible). Unlike Kinect skeletal tracker, people detection is based on motion detection, thus a person has to move at startup for being detected. However, it allows to track skeletons also for people seen from the back side and it can be used with the Robot Operating System (ROS) [96]. In Figure A.3, some examples of skeleton estimation obtained with NiTE skeletal tracker are reported.

From our experience, while Kinect skeletal tracker provides reasonable skeletons also when some joints are not tracked, NiTE skeletal tracker often poorly estimates the

[4]In this work, we used version 1.5.2 of NiTE.



(a) (b) (c) (d) (e)

Figure A.4: Example of skeleton estimation obtained with NiTE skeletal tracker and face detection guided by the skeleton tracking.

whole skeleton when some joints are not visible. For this reason, when using NiTE skeletal tracker in this work, we used only frames with all joints marked as tracked.

Appendix B

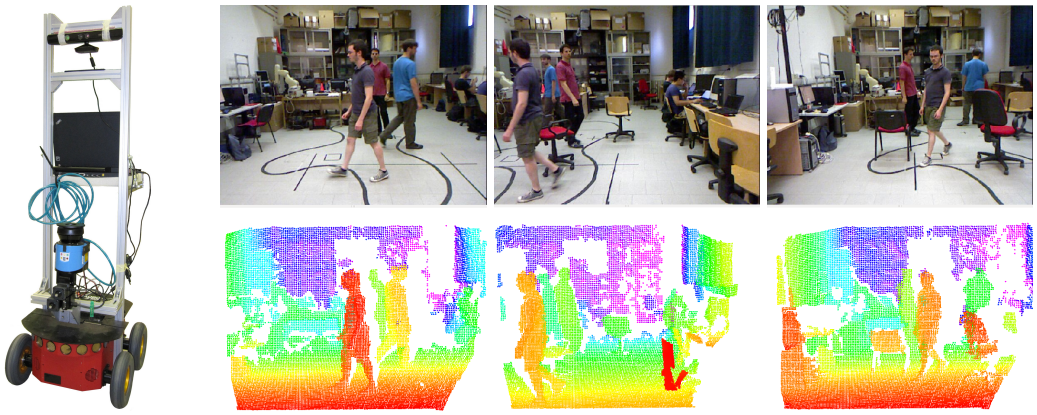
RGB-D Datasets

Since Microsoft Kinect and other consumer RGB-D sensors have been introduced, new datasets have been created in order to provide their aligned RGB and depth streams for a variety of indoor applications. The need for new datasets was driven by the fact that these sensors provide depth data with resolution, framerate, precision and artifacts which are considerably different from those of existing datasets, such as those collected with stereo cameras. [59], [51] and the organizers of the 2011 *Solutions in Perception Challenge*[1] proposed datasets acquired with Kinect suitable for object recognition and pose estimation, while [114] and [131] describe datasets expressly thought for RGB-D action recognition. In [112], the authors propose a new dataset for creating a benchmark of RGB-D SLAM algorithms and, in [57] and [108], Kinect data have been released for evaluating scene labeling algorithms. For people tracking evaluation, the only dataset acquired with native RGB-D sensors is proposed in [109] and [69], while [6] proposed the first dataset for RGB-D people re-identification. We refer to [11] for a wider survey on datasets collected with consumer RGB-D sensors, while we report here below the new datasets we collected for validating the algorithms developed in this work.

B.1 RGB-D Datasets for People Tracking

The authors of [109] recorded data in a university hall from three static Kinects with adjacent, but non overlapping field of view and tracking performance can be evaluated in terms of accuracy (false positives, false negatives, ID switches) and precision in localizing a person inside the image. Before Kinect release, the most popular datasets for evaluating people detection and tracking algorithms which exploited aligned RGB and

[1] `http://solutionsinperception.org`.



(a) (b) (c) (d)

Figure B.1: (a) The platform we used for collecting the IAS-Lab People Tracking dataset and (b-d) some sample RGB images and corresponding 3D point clouds from the dataset.

depth data were acquired from stereo cameras or reciprocally calibrated laser range finders and colour cameras. That is the case of [35], which proposed videos acquired from a stereo pair mounted on a mobile platform in order to evaluate people detection in an outdoor scenario, or [91], where a dataset collected with Willow Garage PR2 robot is presented, with the purpose of training and testing multi-modal person detection and tracking in indoor office environments by means of stereo cameras and laser range finders.

B.1.1 IAS-Lab People Tracking dataset

The IAS-Lab People Tracking dataset[2] is a dataset targeted to people detection and tracking applications and composed of RGB-D video sequences collected in an indoor environment with the mobile robot shown in Figure B.1 (a). It consists of a Pioneer P3-AT platform equipped with a Microsoft Kinect sensor. RGB images are recorded at 640 x 480 pixel resolution, while depth data are saved at 160 x 120 pixel resolution. Both streams have been saved at 30 Hz and the robot odometry is also available.

Sequences have been collected both while keeping our platform as static and while moving it on a predefined path. For both cases, we acquired videos in three different scenarios of increasing difficulty:

1. no obstacle is present, people move with simple (linear) trajectories;

2. no obstacle is present, people move with complex trajectories and interact with

[2] `http://www.dei.unipd.it/~munaro/iaslab-people-tracking-dataset.html`.

each other;

3. obstacles are present, people move with complex trajectories and interact with each other.

Every video sequence extends over about 750 frames, thus the total dataset includes 4671 frames, 12272 instances of people and 26 tracks that have been manually annotated on the RGB image and that constitute the ground truth. The minimum distance between people is 0.2 m while the minimum people-object distance is 0.05 m. In Figure B.1 (b-d), some sample RGB images and corresponding 3D point clouds from the dataset are shown.

B.1.2 Kinect Tracking Precision dataset

As reported earlier in this section, some datasets have been recently released for testing computer vision algorithms on data from consumer RGB-D sensors. Among these datasets, the one proposed in [111] and [69] has been created for evaluating people tracking algorithms in terms of accuracy (false positives, false negatives, ID switches) and precision in localizing a person inside the image. However, this dataset is not exhaustive for mobile robotics applications. Firstly, RGB-D data are recorded from a static platform, thus robustness to camera vibrations, motion blur and odometry errors cannot be evaluated, secondly, a 3D ground-truth is not reported, i.e. the actual position of people neither in the robot frame of reference nor in the world frame of reference is known. For many applications, and in particular when dealing with a multi-camera scenario, it becomes also important to evaluate how accurate and precise a tracking algorithm is in 3D coordinates. In [27], a sort of 3D ground truth is inferred from the image bounding boxes and the depth images computed from stereo data, but these measures are correlated to the sensor depth estimate. For these reasons, we collected a new RGB-D dataset called *Kinect Tracking Precision (KTP) dataset*[3] acquired from a mobile robot moving in a motion capture room. This dataset has been realized to measure 2D/3D accuracy and precision of people tracking algorithms based on data coming from consumer RGB-D sensors and it has been publically released.

Data Collection and Ground Truthing

We collected 8475 frames of a Microsoft Kinect at 640 x 480 pixel resolution and at 30 Hz, for a total of 14766 instances of people. The Kinect was mounted on a mobile

[3] `http://www.dei.unipd.it/~munaro/KTP-dataset.html`.

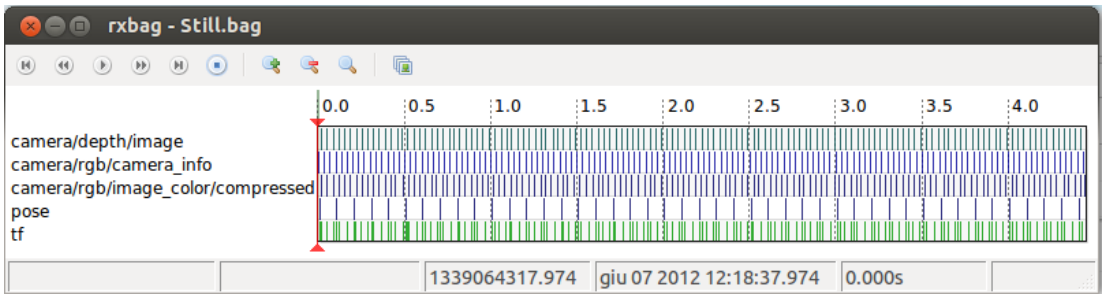


Figure B.2: A bag from the KTP dataset as it is visualized by ROS rxbag tool. The messages published to every topic can be inspected.

robot moving inside a 3 x 5 meters room equipped with a BTS[4] marker-based motion capture system composed of six infrared cameras. The spatial extent of the dataset was limited by the dimensions of the motion capture room. The dataset provides RGB and depth images, with the depth images already registered to the RGB ones, and robot odometry. They are made available both as single files with timestamp and as ROS bag files, that are recordings containing RGB, depth, odometry and transforms among reference frames as a synchronized stream. In Figure B.2, a screenshot of the ROS tool rxbag allows to see the publishing rate of every data source. While Kinect data have been published at 30 frames per second, the robot pose was limited to 10 frames per second. As ground truth, image and 3D people positions are given, together with a further ground truth for robot odometry obtained by placing some markers also on the robot. Image ground truth is in the form of bounding boxes and has been created with the annotating tool and the procedure described in [33]. We annotated only people who are at least half visible in the RGB image. Except when people are partially out of the image, we made the bounding boxes width to be half of the height and centered on the person's head.

3D ground truth consists of people 3D position obtained by placing one infrared marker on every person's head as depicted in Figure B.3 and tracking them with the motion capture system. Then, we referred people 3D position to the robot odometry reference frame and we synchronized the time with that of the images. At this point, we attempted to assign a 3D ground truth to every acquired Kinect frame. When some people were missing (out of the field of view or fully occluded) in the image ground truth, they have been deleted also in the 3D ground truth. When some people were present in the image ground truth, but missing in the 3D ground truth because of occlusions, no 3D ground truth have been associated to those frames. With this process, we assigned 3D ground truth to about 70% of the acquired Kinect frames. In Table B.1,

[4] http://www.btsbioengineering.com.

Table B.1: Statistics of the ground truth provided with the *KTP Dataset*.

	Image	**3D**
Annotated frames	8475	6287
Frames with people	7058	4870
People instances	14766	10410
Number of tracks	20	20

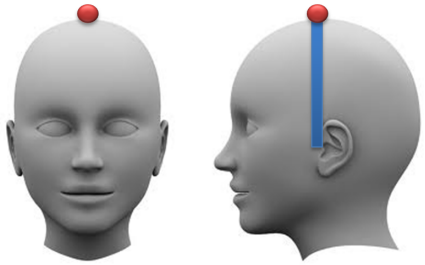


Figure B.3: Marker position (in red) on a person's head.

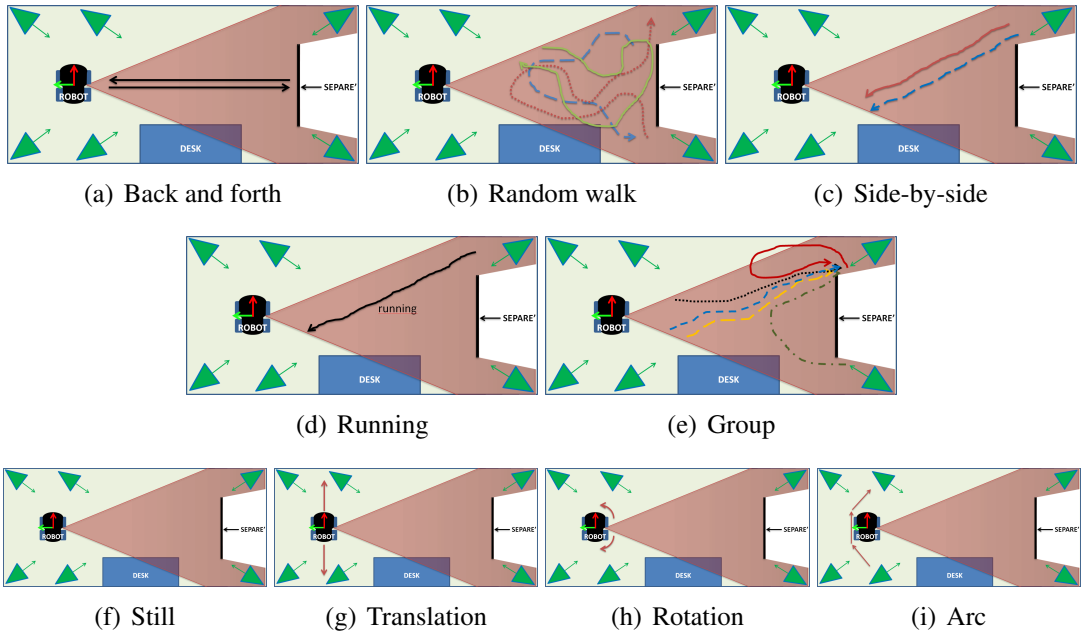
some statistics are reported about the image and 3D ground truth.

Content Description

The dataset consists of four videos of about one minute each. In each video, the same five situations are performed:

- *back and forth*: a person walks back and forth once;
- *random*: three persons walk with random trajectories for about 20 seconds;
- *side-by-side*: two persons walk side-by-side with a linear trajectory;
- *running*: one person runs across the room;
- *group*: five persons gather in a group and then leave the room.

The robot moves differently in every video, in order to test tracking performance for different robot motions. The four videos are named according to the movement the robot performs: *Still*, *Translation*, *Rotation*, *Arc*. The robot maximum translation and rotation speeds have been respectively set to 0.15 m/s and 0.11 rad/s for avoiding stability issues due to the high friction produced by the plastic floor of the motion capture room on the robot wheels. In Figure B.4, a pictorial representation of (a-e) the five situations contained in the dataset and (f-i) the movements the robot performed inside the motion capture room is reported. In Figure B.5 some annotated RGB and depth images are reported as representative of the five situations characterizing the dataset.



(a) Back and forth (b) Random walk (c) Side-by-side

(d) Running (e) Group

(f) Still (g) Translation (h) Rotation (i) Arc

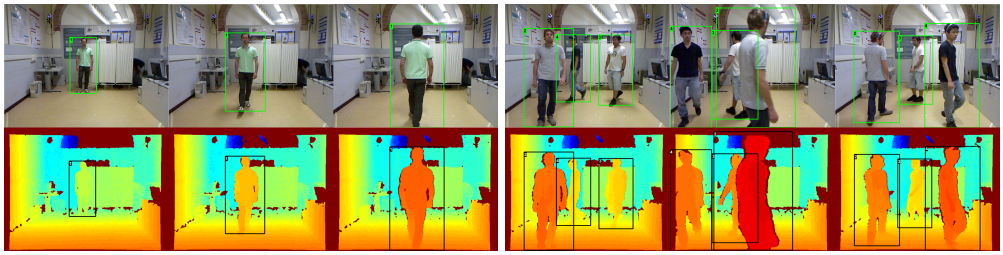
Figure B.4: Illustration of (a-e) the five situations featured in the KTP dataset and (f-i) the four movements the robot performed inside the motion capture room. Motion capture cameras are drawn as green triangles, while Kinect field of view is represented as a red cone.

Robotic Platform

The dataset has been collected with the mobile robot represented in Figure B.6 (a). It consists of a Pioneer P3-AT platform equipped with a side mounted Kinect sensor. In Figure B.6 (b) a model of the robot in the *Unified Robot Description Format*[5] and a representation of the main reference frames associated to it are depicted.

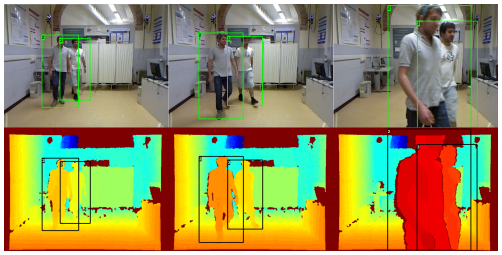
With the KTP dataset, we provide both the odometry of this robot and its real position in 3D measured with the motion capture system. This double source of information allowed us to estimate the errors in robot odometry by taking the motion capture measurements as ground truth. In Figure B.7, we report the mean odometry errors measured in estimating *x*-*y* position and *yaw* orientation. As we expected, the error in *x*-*y* is maximum (22 mm) when the robot both translates and rotates for performing an arc movement, while the maximum *yaw* error (1°) is reached when the robot performs more rotations (namely in *Rotation*).

[5]URDF: `http://www.ros.org/wiki/urdf`.

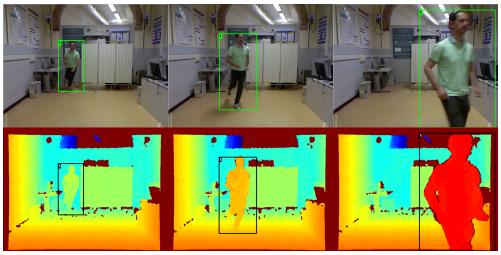


(a) Back and forth

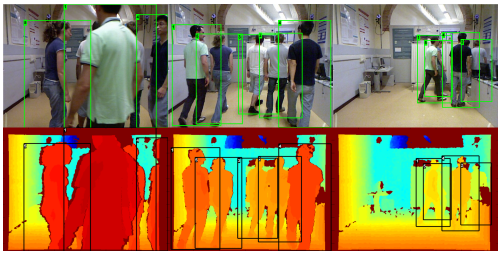
(b) Random walk



(c) Side-by-side

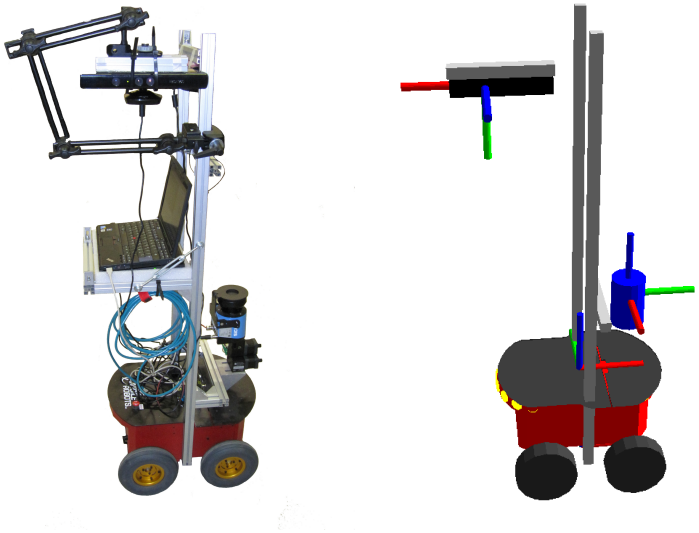


(d) Running



(e) Group

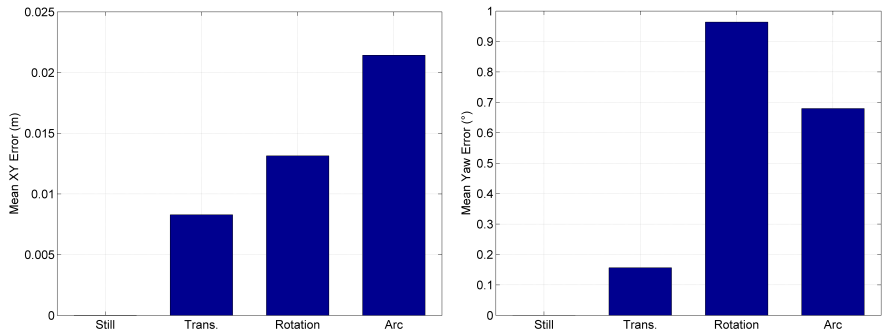
Figure B.5: RGB and Depth sample images showing the five situations of the KTP dataset, together with the corresponding image annotations.



(a) Test platform.

(b) URDF model and main reference fames.

Figure B.6: (a) The robotic platform used in the KTP dataset and (b) its URDF model together with the main reference frames. Note that for this dataset we only acquired RGB-D data from a Microsoft Kinect sensor and did not make use of other sensors such as Laser Range Finder or sonars.



(a) (b)

Figure B.7: Histograms of the odometry error ((a) in *x*-*y* and (b) in *yaw*) with respect to the ground truth obtained with the marker-based motion capture system.

B.2 RGB-D Datasets for People Re-Identification

The challenges of people re-identification are usually different from those of short-term tracking, thus different datasets are also needed for testing re-identification techniques. In particular, a good re-identification dataset should contain a high number of people and feature different conditions between training and testing set.

As stated in Section B, a dataset explicitly thought for the RGB-D re-identification task has been proposed in [6]. It consists of 79 different subjects collected in 4 different scenarios. For allowing to use shape and skeletal tracking information for re-identification, the dataset contains a 3D mesh and skeleton joints position for every subject, in addition to the RGB image and a foreground segmentation mask. However, this dataset contains very few frames for each subject, faces are blurred for privacy reasons and skeleton links orientation is not available. With [90], another dataset for RGB-D re-identification has been released, which contains RGB, depth and mask images at 320 x 240 pixel resolution at 6-15 Hz. However, no skeleton information is provided with this dataset.

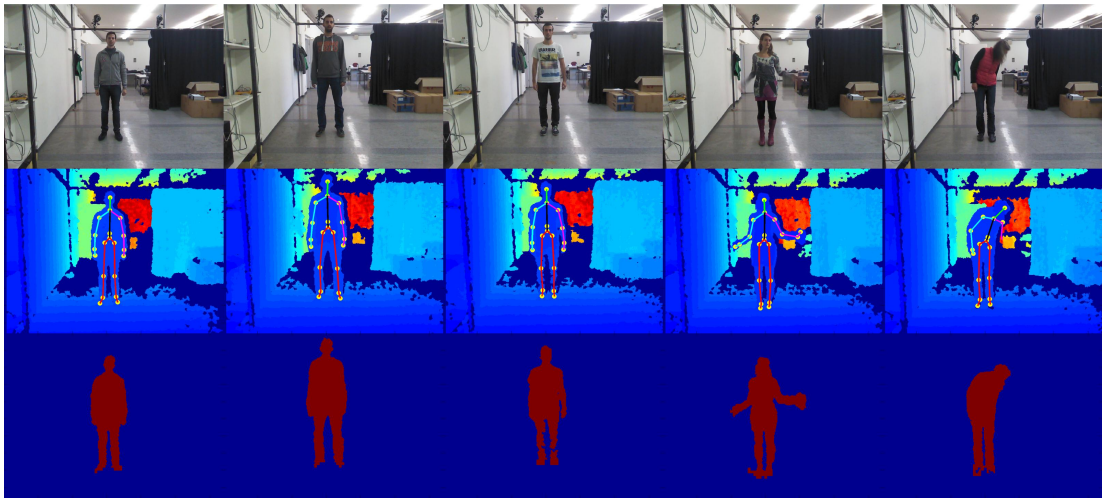
For overcoming the limitations of existing datasets, we collected two new publically released re-identification datasets which provide all the information available with a consumer RGB-D sensor and that exploit two different skeletal tracking algorithms.

B.2.1 BIWI RGBD-ID dataset

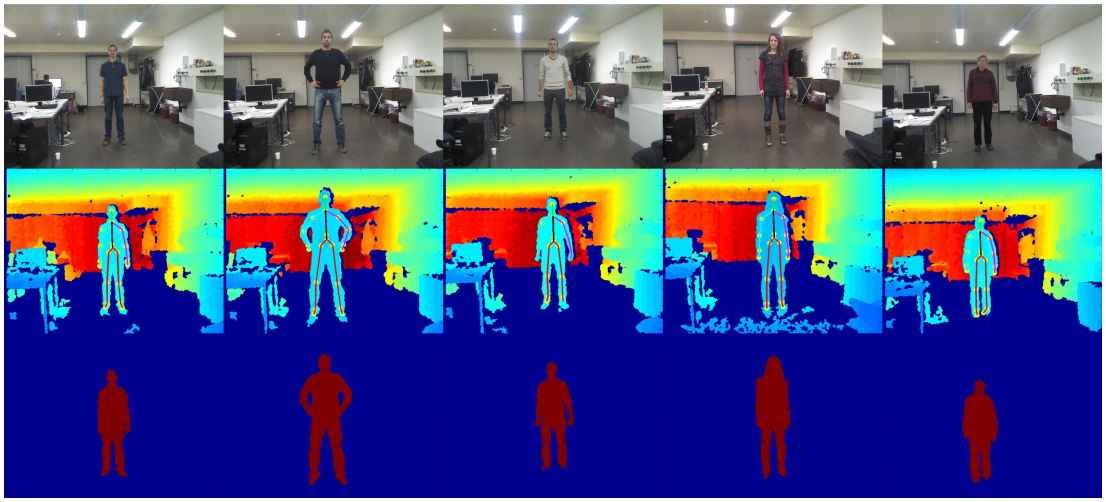
The BIWI RGBD-ID dataset[6] consists of video sequences of 50 different subjects, performing a certain routine of motions and walks in front of a Kinect. The dataset includes synchronized RGB images (at 1280 $\times$ 960 pixels), depth images (at 640 $\times$ 480 pixels), persons segmentation masks and skeletal joints position and links orientation (as provided by the Microsoft Kinect SDK), in addition to the ground plane coordinates. These videos have been acquired at about 8-10 fps and last about one minute for every subject in order to provide enough information for training most of re-identification algorithms.

Moreover, we have collected a *Still* and a *Walking* test sequence for 28 subjects already present in the dataset. In the *Walking* video, every person performs two walks frontally and other two walks diagonally with respect to the Kinect. These have been collected on a different day and therefore most subjects are dressed differently. These sequences are also shot in a different location than the studio room where the training

[6] `http://robotics.dei.unipd.it/reid`.



(a) Training set.



(b) Testing set.

Figure B.8: Samples of RGB, depth, skeleton and user mask for five people from the (a) training and the (b) testing set of the BIWI RGBD-ID dataset.

dataset had been collected.

In Figure B.8, we report some sample images for five people from the training and testing set. For each person, RGB, depth, skeleton and segmentation mask are shown.

B.2.2 IAS-Lab RGBD-ID dataset

The skeletal tracking algorithm provided by Microsoft SDK gives the best accuracy, but does not allow to estimate the skeleton of non-frontal people. For this reason, we also collected the IAS-Lab RGBD-ID dataset[7], which contains 33 RGB-D sequences of 11 people and skeleton tracking information obtained with the OpenNI SDK[8] and the NiTE middleware, which does not have this limitation. For every subject, we recorded three sequences, where the person rotates on himself and performs some walks. The first (*Training*) and the second (*TestingA*) sequences were acquired with people wearing different clothes, while the third one (*TestingB*) was collected in a different room, but with the same clothes as in the first sequence. These two different testing sets allow to validate both short-term and long-term re-identification techniques on this dataset. Since NiTE skeletal tracking often poorly estimates the whole skeleton when some joints are not visible, we kept only those frames where all the joints are marked as tracked by the algorithm.

In Figure B.9, a sample frame from every sequence and for every person is reported. It can be noticed that strong illumination changes are present between training and testing videos of some people because of Kinect auto-exposure function. It can also be noticed that both the BIWI RGBD-ID dataset and the IAS-Lab RGBD-ID dataset have been acquired from a point of view typical of robotics applications.

B.3 RGB-D Datasets for Action Recognition

Currently, the following datasets for RGB-D action recognition have been released: RGBD-HuDaAct Database [131], Indoor Activity Database [114], MSR-Action3D Dataset [121], MSR-DailyActivity3D Dataset [120], LIRIS Human Activities Dataset [124] and Berkeley MHAD [89]. All these datasets are targeted to recognition tasks in indoor environments. The first two are thought for personal or service robotics applications, while the two from MSR are also targeted to gaming and human-computer interaction. The LIRIS dataset concerns actions performed from both single persons and groups, acquired in different scenarios and changing the point of view. The last one

[7] `http://robotics.dei.unipd.it/reid`.

[8] `http://www.openni.org/openni-sdk`.

(a) Training set. (b) Testing set A.

(c) Testing set B.

Figure B.9: Samples RGB frames and skeleton from the (a) training and the (b-c) two testing sets of the IAS-Lab RGBD-ID dataset.

Table B.2: Datasets for 3D Human Action Recognition.

	#actions	**#people**	**#samples**	**RGB**	**skel**
[131]	6	1	198	yes	no
[114]	12	4	48	yes	yes
[121]	20	10	567	no[10]	yes
[120]	16	10	320	no	yes
[124][11]	10	21	461	yes[12]	no
[89]	11	12	660	yes	yes[13]
Ours	15	12	540	yes	yes

was acquired using a multimodal system (mocap, video, depth, acceleration, audio) to provide a very controlled set of actions to test algorithms across multiple modalities.

B.3.1 IAS-Lab Action dataset

Two key features of a good dataset are size and variability. Moreover, it should allow to compare as many different algorithms as possible. For the RGB-D action recognition task, that means that there should be enough different actions, many different people performing them and RGB and depth synchronization and registration. Moreover, the 3D skeleton of the actors should be saved, given that it is easily available and many recent techniques rely on it. Hovewer, we noticed the lack of a dataset having all these features, thus we acquired the IAS-Lab Action dataset[9], which contains 15 different actions performed by 12 different people. Each person repeats each action three times, thus leading to 540 video samples. All these samples are provided as ROS `bags` containing synchronized and registered RGB images, depth images and point clouds and ROS `tf` for every skeleton joint as they are estimated by the NiTE middleware. Unlike [89], we preferred NiTE's skeletal tracker to a motion capture technology in order to test our algorithms on data that could be easily available on a mobile robot and, unlike [124], we asked the subjects to perform well defined actions, because, beyond a certain level, variability could bias the evaluation of an algorithm performance.

In Table B.2, the IAS-Lab Action dataset is compared to the datasets mentioned in B.3, while in Figure B.10 a sample image for every action is reported.

[9] `http://robotics.dei.unipd.it/actions`.

[10] The RGB images are provided, but they are not synchronized with the depth images.

[11] Only the set provided with depth information was considered.

[12] The RGB information has been converted to grayscale.

[13] Obtained from motion capture data.

(a) Check watch (b) Cross arms (c) Get up

(d) Kick (e) Pick up (f) Point

(g) Punch (h) Scratch head (i) Sit down

(j) Standing (k) Throw from bottom up (l) Throw over head

(m) Turn around (n) Walk (o) Wave

Figure B.10: Sample images for the 15 actions present in the IAS-Lab Action dataset.

Bibliography

[1] A. Alahi, R. Ortiz, and P. Vandergheynst. Freak: Fast retina keypoint. In *Proceedings of the 2012 IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2012)*, pages 510–517, 2012.

[2] S. Ali and M. Shah. Human action recognition in videos using kinematic features and multiple instance learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 32(2):288–303, 2010.

[3] M. Bajracharya, B. Moghaddam, A. Howard, S. Brennan, and L. H. Matthies. A fast stereo-based system for detecting and tracking pedestrians from a moving vehicle. In *International Journal of Robotics Research*, volume 28, pages 1466–1485, 2009.

[4] G. Ballin, M. Munaro, and E. Menegatti. Human action recognition from rgb-d frames based on real-time 3d optical flow estimation. In A. Chella, R. Pirrone, R. Sorbello, and K. R. Jóhannsdóttir, editors, *Biologically Inspired Cognitive Architectures 2012*, volume 196 of *Advances in Intelligent Systems and Computing*, pages 65–74. Springer Berlin Heidelberg, 2012.

[5] D. Baltieri, R. Vezzani, R. Cucchiara, A. Utasi, C. Benedek, and T. Sziranyi. Multi-view people surveillance using 3d information. In *Proceedings of the 2011 IEEE International Conference on Computer Vision Workshops (ICCV Workshops 2011)*, pages 1817–1824, 2011.

[6] I. B. Barbosa, M. Cristani, A. Del Bue, L. Bazzani, and V. Murino. Re-identification with rgb-d sensors. In *European Conference on Computer Vision (ECCV) Workshops 2012*, pages 433–442. Springer, 2012.

[7] F. Basso, M. Munaro, S. Michieletto, E. Pagello, and E. Menegatti. Fast and robust multi-people tracking from rgb-d data for a mobile robot. In *12th Intelligent Autonomous Systems Conference (IAS-12)*, pages 265–276, Jeju Island, Korea, June 2012.

[8] M. Bauml and R. Stiefelhagen. Evaluation of local features for person re-identification in image sequences. In *8th IEEE International Conference on Advanced Video and Signal-Based Surveillance (AVSS 2011)*, pages 291–296. IEEE, 2011.

[9] H. Bay, A. Ess, T. Tuytelaars, and L. Van Gool. Speeded-up robust features (surf). *Computer Vision and Image Understanding*, 110(3):346–359, June 2008.

[10] N. Bellotto and H. Hu. Computationally efficient solutions for tracking people with a mobile robot: an experimental evaluation of bayesian filters. *Autonomous Robots*, 28:425–438, May 2010.

[11] K. Berger. The role of rgb-d benchmark datasets: an overview. *arXiv preprint arXiv:1310.2053*, 2013.

[12] K. Bernardin and R. Stiefelhagen. Evaluating multiple object tracking performance: the clear mot metrics. *Journal of Image Video Processing*, 2008:1:1–1:10, January 2008.

[13] P. J. Besl and N. McKay. A method for registration of 3-d shapes. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 14:239–256, 1992.

[14] A. Bhattacharyya. On a measure of divergence between two statistical populations defined by their probability distributions. *Bulletin of Calcutta Mathematical Society*, 35:99–109, 1943.

[15] M. Blank, L. Gorelick, E. Shechtman, M. Irani, and R. Basri. Actions as space-time shapes. In *Proc. Tenth IEEE Int. Conf. Computer Vision ICCV 2005*, volume 2, pages 1395–1402, 2005.

[16] V. Bloom, D. Makris, and V. Argyriou. G3d: A gaming action dataset and real time action recognition evaluation framework. In *2012 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 7 –12, june 2012.

[17] G. Bradski. The OpenCV Library. *Dr. Dobb's Journal of Software Tools*, 2000.

[18] M. D. Breitenstein, F. Reichlin, B. Leibe, E. Koller-Meier, and L. V. Gool. Robust tracking-by-detection using a detector confidence particle filter. In *International Conference on Computer Vision (ICCV) 2009*, volume 1, pages 1515–1522, October 2009.

[19] A. M. Bronstein, M. M. Bronstein, , and R. Kimmel. Three-dimensional face recognition. *International Journal of Computer Vision*, 64:5–30, 2005.

[20] A. M. Bronstein, M. M. Bronstein, and R. Kimmel. Topology-invariant similarity of nonrigid shapes. *International Journal of Computer Vision*, 81:281–301, March 2009.

[21] M. Calonder, V. Lepetit, C. Strecha, and P. Fua. Brief: binary robust independent elementary features. In *Proc. of the 2010 European Conference on Computer Vision (ECCV 2010)*, pages 778–792. Springer, 2010.

[22] A. Carballo, A. Ohya, and S. Yuta. Reliable people detection using range and intensity data from multiple layers of laser range finders on a mobile robot. *International Journal of Social Robotics*, 3(2):167–186, 2011.

[23] S. Carlsson and J. Sullivan. Action recognition by shape matching to key frames. In *IEEE Computer Society Workshop on Models versus Exemplars in Computer Vision*, 2001.

[24] J. Chen, D. Bautembach, and S. Izadi. Scalable real-time volumetric surface reconstruction. *ACM Transactions on Graphics (TOG)*, 32(4):113, 2013.

[25] D. S. Cheng, M. Cristani, M. Stoppa, L. Bazzani, and V. Murino. Custom pictorial structures for re-identification. In *British Machine Vision Conference*, volume 2, page 6, 2011.

[26] W. Choi, C. Pantofaru, and S. Savarese. Detecting and tracking people using an rgb-d camera via multiple detector fusion. In *International Conference on Computer Vision (ICCV) Workshops 2011*, pages 1076–1083, 2011.

[27] W. Choi, C. Pantofaru, and S. Savarese. A general framework for tracking multiple people from a moving camera. *Pattern Analysis and Machine Intelligence (PAMI)*, 35(7):1577–1591, 2012.

[28] C. Cortes and V. N. Vapnik. Support-vector networks. *Machine Learning*, 20, 1995.

[29] C. Dal Mutto, P. Zanuttigh, and G. M. Cortelazzo. *Time-of-Flight Cameras and Microsoft Kinect*. Springer, 2012.

[30] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *Computer Vision and Pattern Recognition (CVPR) 2005*, volume 1, pages 886–893, June 2005.

[31] M. Dantone, J. Gall, G. Fanelli, and L. V. Gool. Real-time facial feature detection using conditional regression forests. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2578–2585, 2012.

[32] P. Dollar, V. Rabaud, G. Cottrell, and S. Belongie. Behavior recognition via sparse spatio-temporal features. In *Proc. 2nd Joint IEEE Int Visual Surveillance and Performance Evaluation of Tracking and Surveillance Workshop*, pages 65–72, 2005.

[33] P. Dollár, C. Wojek, B. Schiele, and P. Perona. Pedestrian detection: A benchmark. In *Computer Vision and Pattern Recognition (CVPR) 2009*, pages 304–311, 2009.

[34] A. Efros, A. Berg, G. Mori, and J. Malik. Recognizing action at a distance. In *9th IEEE International Conference on Computer Vision (ICCV) 2003*, pages 726 –733 vol.2, oct. 2003.

[35] A. Ess, B. Leibe, K. Schindler, and L. Van Gool. A mobile vision system for robust multi-person tracking. In *Computer Vision and Pattern Recognition (CVPR) 2008*, pages 1–8, 2008.

[36] A. Ess, B. Leibe, K. Schindler, and L. Van Gool. Moving obstacle detection in highly dynamic scenes. In *International Conference on Robotics and Automation (ICRA) 2009*, pages 4451–4458, 2009.

[37] M. Everingham, L. Gool, C. K. Williams, J. Winn, and A. Zisserman. The pascal visual object classes (voc) challenge. *International Journal of Computer Vision*, 88:303–338, June 2010.

[38] M. Farenzena, L. Bazzani, A. Perina, V. Murino, and M. Cristani. Person re-identification by symmetry-driven accumulation of local features. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 2360 –2367, june 2010.

[39] P. Felzenszwalb, R. Girshick, D. McAllester, and D. Ramanan. Object detection with discriminatively trained part based models. *Pattern Analysis and Machine Intelligence (PAMI)*, 32(9):1627–1645, September 2010.

[40] S. Ghidoni, S. Anzalone, M. Munaro, M. S., and E. Menegatti. A distributed perception infrastructure for robot assisted living. *To appear in Robotics and Autonomous Systems (RAS) Journal*, 2014.

[41] H. Grabner and H. Bischof. On-line boosting and vision. In *Computer Vision and Pattern Recognition (CVPR)*, volume 1, pages 260–267. IEEE Computer Society, 2006.

[42] D. Gray and H. Tao. Viewpoint invariant pedestrian recognition with an ensemble of localized features. In *European Conference on Computer Vision*, volume 5302, pages 262–275, 2008.

[43] S. S. H. Jin and A. Yezzi. Multi-view stereo reconstruction of dense shape and complex appearance. *International Journal of Computer Vision*, 63:175–189, 2005.

[44] E. Hall. *The Hidden Dimension*. Anchor books editions, 1966.

[45] M. Holte and T. Moeslund. View invariant gesture recognition using 3d motion primitives. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) 2008*, pages 797 –800, 31 2008-april 4 2008.

[46] M. Holte, T. Moeslund, N. Nikolaidis, and I. Pitas. 3d human action recognition for multi-view camera systems. In *2011 International Conference on 3D Imaging, Modeling, Processing, Visualization and Transmission (3DIMPVT)*, pages 342 –349, may 2011.

[47] L. Hu, S. Jiang, Q. Huang, and W. Gao. People re-detection using adaboost with sift and color correlogram. In *Proc. of the 15th International Conference on Image Processing (ICIP 2008)*, pages 1348–1351, 2008.

[48] S. Izadi, D. Kim, O. Hilliges, D. Molyneaux, R. Newcombe, P. Kohli, J. Shotton, S. Hodges, D. Freeman, A. Davison, et al. Kinectfusion: real-time 3d reconstruction and interaction using a moving depth camera. In *Proceedings of the 24th annual ACM symposium on User interface software and technology*, pages 559–568. ACM, 2011.

[49] A. K. Jain, S. C. Dass, and K. Nandakumar. Can soft biometric traits assist user recognition? *Proc. SPIE, Biometric Technology for Human Identification*, 5404:561–572, 2004.

[50] S. Jain. A survey of laser range finding. *National Science Foundation*, 2003.

[51] A. Janoch, S. Karayev, Y. Jia, J. Barron, M. Fritz, K. Saenko, and T. Darrell. A category-level 3-D object dataset: putting the Kinect to work. In *International Conference on Computer Vision (ICCV) Workshop on Consumer Depth Cameras in Computer Vision*, November 2011.

[52] G. Johansson. Visual perception of biological motion and a model for its analysis. *Attention, Perception, & Psychophysics*, 14:201–211, 1973. 10.3758/BF03212378.

[53] K. Jungling and M. Arens. Feature based person detection beyond the visible spectrum. In *Proceedings of the 2009 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPR Workshops 2009)*, pages 30–37, 2009.

[54] Y. Ke, R. Sukthankar, and M. Hebert. Efficient visual event detection using volumetric features. In *10th IEEE International Conference on Computer Vision (ICCV) 2005*, volume 1, pages 166–173, 2005.

[55] A. Kläser, M. Marszałek, and C. Schmid. A spatio-temporal descriptor based on 3d-gradients. In *British Machine Vision Conference*, pages 995–1004, sep 2008.

[56] P. Konstantinova, A. Udvarev, and T. Semerdjiev. A study of a target tracking algorithm using global nearest neighbor approach. In *CompSysTec 2003: e-Learning*, pages 290–295. ACM, 2003.

[57] H. S. Koppula, A. Anand, T. Joachims, and A. Saxena. Semantic labeling of 3d point clouds for indoor scenes. In *Neural Information Processing Systems (NIPS)*, pages 244–252, 2011.

[58] M. Korner and J. Denzler. Analyzing the subspaces obtained by dimensionality reduction for human action recognition from 3d data. In *IEEE 9th International Conference on Advanced Video and Signal-Based Surveillance (AVSS) 2012*, pages 130 –135, sept. 2012.

[59] K. Lai, L. Bo, X. Ren, , and D. Fox. A large-scale hierarchical multi-view rgb-d object dataset. In *International Conference on Robotics and Automation (ICRA) 2011*, pages 1817–1824, May 2011.

[60] I. Laptev, B. Caputo, C. Schüldt, and T. Lindeberg. Local velocity-adapted motion events for spatio-temporal recognition. *Computer Vision and Image Understanding*, 108:207–229, December 2007.

[61] I. Laptev and T. Lindeberg. Space-time interest points. In *9th IEEE International Computer Vision Conference (ICCV)*, pages 432–439, 2003.

[62] I. Laptev, M. Marszalek, C. Schmid, and B. Rozenfeld. Learning realistic human actions from movies. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR) 2008*, pages 1–8, 2008.

[63] J. Lei, X. Ren, and D. Fox. Fine-grained kitchen activity recognition using rgb-d. In *Proceedings of the 2012 ACM Conference on Ubiquitous Computing*, UbiComp '12, pages 208–211, New York, NY, USA, 2012. ACM.

[64] T. Leyvand, C. Meekhof, Y.-C. Wei, J. Sun, and B. Guo. Kinect identity: Technology and experience. *Computer*, 44(4):94 –96, april 2011.

[65] W. Li, Z. Zhang, and Z. Liu. Action recognition based on a bag of 3d points. In *Computer Vision and Pattern Recognition Workshops (CVPRW), 2010 IEEE Computer Society Conference on*, pages 9 –14, june 2010.

[66] J. Liu, S. Ali, and M. Shah. Recognizing human actions using multiple features. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pages 1 –8, june 2008.

[67] R. Liu, S. Z. Li, X. Yuan, and R. He. Online determination of track loss using template inverse matching. In *The Eighth International Workshop on Visual Surveillance - VS2008*, Marseille, France, 2008. Graeme Jones and Tieniu Tan and Steve Maybank and Dimitrios Makris.

[68] D. Lowe. Object recognition from local scale-invariant features. In *Proceedings of the 7th IEEE International Conference on Computer Vision (ICCV 1999)*, volume 2, pages 1150–1157, 1999.

[69] M. Luber, L. Spinello, and K. O. Arras. People tracking in rgb-d data with online boosted target models. In *International Conference On Intelligent Robots and Systems (IROS) 2011*, pages 3844–3849, 2011.

[70] B. Lukas and T. Kanade. An iterative image registration technique with an application to stereo vision. In *International Joint Conferences on Artificial Intelligence (IJCAI) 1981*, pages 674–679, 1981.

[71] C. Martin, E. Schaffernicht, A. Scheidig, and H.-M. Gross. Multi-modal sensor fusion using a probabilistic aggregation scheme for people detection and tracking. *Robotics and Autonomous Systems*, 54(9):721–728, 2006.

[72] Y. Ming, Q. Ruan, and A. Hauptmann. Activity recognition from rgb-d camera with 3d local spatio-temporal features. In *IEEE International Conference on Multimedia and Expo (ICME) 2012*, pages 344 –349, july 2012.

[73] D. Mitzel and B. Leibe. Real-time multi-person tracking with detector assisted structure propagation. In *International Conference on Computer Vision (ICCV) Workshops 2011*, pages 974–981. IEEE, 2011.

[74] O. Mozos, R. Kurazume, and T. Hasegawa. Multi-part people detection using 2d range data. *International Journal of Social Robotics*, 2:31–40, 2010.

[75] M. Munaro, G. Ballin, S. Michieletto, and E. Menegatti. 3D flow estimation for human action recognition from colored point clouds. *Journal on Biologically Inspired Cognitive Architectures*, page 4251, 2013.

[76] M. Munaro, A. Basso, A. Fossati, L. Van Gool, and E. Menegatti. 3d reconstruction of freely moving persons for re-identification with a depth sensor. In *IEEE International Conference on Robotics and Automation (ICRA), Hong Kong (China)*, June 2014.

[77] M. Munaro, F. Basso, and E. Menegatti. Tracking people within groups with rgb-d data. In *Proc. of the International Conference on Intelligent Robots and Systems (IROS)*, pages 2101–2107, Algarve, Portugal, October 2012.

[78] M. Munaro, F. Basso, S. Michieletto, E. Pagello, and E. Menegatti. A software architecture for rgb-d people tracking based on ros framework for a mobile robot. In *Frontiers of Intelligent Autonomous Systems*, volume 466, pages 53–68. Springer, 2013.

[79] M. Munaro, A. Fossati, A. Basso, E. Menegatti, and E. Van Gool. One-shot person re-identification with a consumer depth camera. In *Person Re-Identification*, pages 161–181. Springer, 2014.

[80] M. Munaro, S. Ghidoni, D. Tartaro Dizmen, and E. Menegatti. A feature-based approach to people re-identification using skeleton keypoints. In *IEEE International Conference on Robotics and Automation (ICRA), Hong Kong (China)*. Elsevier, June 2014.

[81] M. Munaro and E. Menegatti. Fast rgb-d people tracking for service robots. *To appear in Autonomous Robots Journal*, 2014.

[82] M. Munaro, S. Michieletto, and E. Menegatti. An evaluation of 3D motion flow and 3D pose estimation for human action recognition. In *RSS Workshops: RGB-D: Advanced Reasoning with Depth Cameras*, 2013.

[83] C. L. Naberezny Azevedo. The multivariate normal distribution [online]. `http://www.ime.unicamp.br/~cnaber/mvnprop.pdf`.

[84] L. E. Navarro-Serment, C. Mertz, and M. Hebert. Pedestrian detection and tracking using three-dimensional ladar data. In *The International Journal of Robotics Research, Special Issue on the Seventh International Conference on Field and Service Robots*, pages 103–112, 2009.

[85] B. Ni, G. Wang, and P. Moulin. Rgbd-hudaact: A color-depth video database for human daily activity recognition. In *IEEE International Conference on Computer Vision Workshops (ICCV Workshops), 2011*, pages 1147 –1153, nov. 2011.

[86] J. C. Niebles, H. Wang, and L. Fei-Fei. Unsupervised learning of human action categories using spatial-temporal words. *International Journal of Computer Vision*, 79:299–318, September 2008.

[87] D. Ober, S. Neugebauer, and P. Sallee. Training and feature-reduction techniques for human identification using anthropometry. In *4th IEEE International Conference on Biometrics: Theory Applications and Systems (BTAS) 2010*, pages 1 –8, sept. 2010.

[88] F. Ofli, R. Chaudhry, G. Kurillo, R. Vidal, and R. Bajcsy. Sequence of the most informative joints (smij): A new representation for human skeletal action recognition. In *Computer Vision and Pattern Recognition Workshops (CVPRW), 2012 IEEE Computer Society Conference on*, pages 8 –13, june 2012.

[89] F. Ofli, R. Chaudhry, G. Kurillo, R. Vidal, and R. Bajcsy. Berkeley mhad: A comprehensive multimodal human action database. In *Proc. of the IEEE Workshop on Applications on Computer Vision*, 2013.

[90] J. Oliver, A. Albiol, and A. Albiol. 3d descriptor for people re-identification. In *Proceedings of the 21st IEEE International Conference on Pattern Recognition (ICPR 2012)*, pages 1395–1398, 2012.

[91] C. Pantofaru. The moving people, moving platform dataset. `http://bags.willowgarage.com/downloads/people_dataset/`.

[92] N. Pears, Y. Liu, and P. Bunting. *3D imaging, analysis and applications*. Springer, 2012.

[93] M. Popa, A. Koc, L. Rothkrantz, C. Shan, and P. Wiggers. Kinect sensing of shopping related actions. In undefined, K. Van Laerhoven, and J. Gelissen, editors, *Constructing Ambient Intelligence: AmI 2011 Workshops*, Amsterdam, Netherlands, 11 2011.

[94] R. Poppe. A survey on vision-based human action recognition. *Image and Vision Computing*, 28(6):976–990, June 2010.

[95] J. Preis, M. Kessel, M. Werner, and C. Linnhoff-Popien. Gait recognition with kinect. In *Proceedings of the First Workshop on Kinect in Pervasive Computing*, 2012.

[96] M. Quigley, B. Gerkey, K. Conley, J. Faust, T. Foote, J. Leibs, E. Berger, R. Wheeler, and A. Ng. Ros: an open-source robot operating system. In *International Conference on Robotics and Automation (ICRA)*, 2009.

[97] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski. Orb: An efficient alternative to sift or surf. In *Proc. of the 2011 IEEE International Conference on Computer Vision (ICCV 2011)*, pages 2564–2571, 2011.

[98] R. Rusu, N. Blodow, and M. Beetz. Fast point feature histograms (fpfh) for 3d registration. In *Proc. of the 2009 International Conference on Robotics and Automation (ICRA 2009)*, pages 3212–3217, 2009.

[99] R. Rusu, N. Blodow, Z. Marton, and M. Beetz. Aligning point cloud views using persistent feature histograms. In *Proc. of the 2008 International Conference on Intelligent Robots and Systems (IROS 2008)*, pages 3384–3391, 2008.

[100] R. B. Rusu. Semantic 3d object maps for everyday manipulation in human living environments, 2010.

[101] R. B. Rusu, J. Bandouch, F. Meier, I. A. Essa, and M. Beetz. Human action recognition using global point feature histograms and action shapes. *Advanced Robotics*, 23(14):1873–1908, 2009.

[102] R. B. Rusu and S. Cousins. 3D is here: Point Cloud Library (PCL). In *International Conference on Robotics and Automation (ICRA) 2011*, pages 1–4, Shanghai, China, May 9-13 2011.

[103] J. Satake and J. Miura. Robust stereo-based person detection and tracking for a person following robot. In *Workshop on People Detection and Tracking (ICRA 2009)*, 2009.

[104] R. Satta, F. Pala, G. Fumera, and F. Roli. Real-time appearance-based person re-identification over multiple Kinect cameras. In *International Conference on Computer Vision and Applications (VisApp)*, 2013.

[105] C. Schuldt, I. Laptev, and B. Caputo. Recognizing human actions: a local svm approach. In *17th International Conference on Pattern Recognition (ICPR) 2004*, volume 3, pages 32–36, 2004.

[106] P. Scovanner, S. Ali, and M. Shah. A 3-dimensional sift descriptor and its application to action recognition. In *Proceedings of the 15th international conference on Multimedia*, MULTIMEDIA '07, pages 357–360, New York, NY, USA, 2007. ACM.

[107] J. Shotton, A. W. Fitzgibbon, M. Cook, T. Sharp, M. Finocchio, R. Moore, A. Kipman, and A. Blake. Real-time human pose recognition in parts from single depth images. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1297–1304, 2011.

[108] N. Silberman and R. Fergus. Indoor scene segmentation using a structured light sensor. In *ICCV 2011 - Workshop on 3D Representation and Recognition*, pages 601–608, 2011.

[109] L. Spinello and K. O. Arras. People detection in rgb-d data. In *International Conference On Intelligent Robots and Systems (IROS) 2011*, pages 3838–3843, 2011.

[110] L. Spinello, K. O. Arras, R. Triebel, and R. Siegwart. A layered approach to people detection in 3d range data. In *Conference on Artificial Intelligence AAAI'10, PGAI Track*, Atlanta, USA, 2010.

[111] L. Spinello, M. Luber, and K. O. Arras. Tracking people in 3d using a bottom-up top-down people detector. In *International Conference on Robotics and Automation (ICRA) 2011*, pages 1304–1310, Shanghai, 2011.

[112] J. Sturm, N. Engelhard, F. Endres, W. Burgard, and D. Cremers. A benchmark for the evaluation of rgb-d slam systems. In *International Conference On Intelligent Robots and Systems (IROS) 2012*, pages 573–580, Oct. 2012.

[113] J. Sung, C. Ponce, B. Selman, and A. Saxena. Human activity detection from rgbd images. In *Plan, Activity, and Intent Recognition*, 2011.

[114] J. Sung, C. Ponce, B. Selman, and A. Saxena. Unstructured human activity detection from rgbd images. In *International Conference on Robotics and Automation, ICRA*, pages 842–849, May 2012.

[115] H. L. U. Thuc, P. V. Tuan, and J.-N. Hwang. An effective 3d geometric relational feature descriptor for human action recognition. In *IEEE RIVF International Conference on Computing and Communication Technologies, Research, Innovation, and Vision for the Future (RIVF), 2012*, pages 1 –6, 27 2012-march 1 2012.

[116] F. Tombari, S. Salti, and L. Di Stefano. Unique signatures of histograms for local surface description. In *Proc. of the 2010 European Conference on Computer Vision (ECCV 2010)*, pages 356–369. Springer, 2010.

[117] F. Tombari, S. Salti, and L. Di Stefano. A combined texture-shape descriptor for enhanced 3d feature matching. In *Proceedings of the 18th IEEE International Conference on Image Processing (ICIP 2011)*, pages 809–812. IEEE, 2011.

[118] C. Velardo and J.-L. Dugelay. Improving identification by pruning: A case study on face recognition and body soft biometric. In *International Workshop on Image and Audio Analysis for Multimedia Interactive Services*, pages 1–4, 2012.

[119] P. Viola and M. Jones. Rapid object detection using a boosted cascade of simple features. In *Computer Vision and Pattern Recognition (CVPR) 2001*, volume 1, pages 511–518, 2001.

[120] J. Wang, Z. Liu, Y. Wu, and J. Yuan. Mining actionlet ensemble for action recognition with depth cameras. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2012), Providence, Rhode Island*, pages 1290–1297, June 2012.

[121] Z. L. Wanqing Li, Zhengyou Zhang. Action recognition based on a bag of 3d points. In *IEEE International Workshop on CVPR for Human Communicative*

Behavior Analysis (in conjunction with CVPR 2010), San Francisco, CA, June 2010.

[122] D. Weinland, R. Ronfard, and E. Boyer. Free viewpoint action recognition using motion history volumes. *Computer Vision and Image Understanding*, 104(2):249–257, Nov. 2006.

[123] A. Weiss, D. Hirshberg, and M. Black. Home 3d body scans from noisy image and range data. In *IEEE International Conference on Computer Vision (ICCV)*, pages 1951–1958, 2011.

[124] C. Wolf, J. Mille, E. Lombardi, O. Celiktutan, M. Jiu, M. Baccouche, E. Dellandra, C.-E. Bichot, C. Garcia, and B. Sankur. The LIRIS Human activities dataset and the ICPR 2012 human activities recognition and localization competition. Technical report, LIRIS Laboratory, 2012.

[125] L. Xia, C.-C. Chen, and J. Aggarwal. View invariant human action recognition using histograms of 3d joints. In *Computer Vision and Pattern Recognition Workshops (CVPRW), 2012 IEEE Computer Society Conference on*, pages 20 –27, june 2012.

[126] J. Xing, H. Ai, and S. Lao. Multi-object tracking through occlusions by local tracklets filtering and global tracklets association with detection responses. In *Computer Vision and Pattern Recognition (CVPR)*, pages 1200–1207, 2009.

[127] Y. Yacoob and M. Black. Parameterized modeling and recognition of activities. In *6th International Conference on Computer Vision (ICCV), 1998*, pages 120 –127, jan 1998.

[128] X. Yang and Y. Tian. Eigenjoints-based action recognition using naive-bayes-nearest-neighbor. In *IEEE Workshop on CVPR for Human Activity Understanding from 3D Data*, 2012.

[129] A. Yilmaz and M. Shah. Actions sketch: a novel action representation. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR), 2005*, volume 1, pages 984 – 989 vol. 1, June 2005.

[130] K. Yoon, D. Harwood, and L. Davis. Appearance-based person recognition using color/path-length profile. *Journal of Visual Communication and Image Representation (JVCIR 2006)*, 17(3):605–622, 2006.

[131] H. Zhang and L. E. Parker. 4-dimensional local spatio-temporal features for human activity recognition. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 2044 –2049, September 2011.

[132] L. Zhang, Y. Li, and N. R. Global data association for multi-object tracking using network flows. In *Computer Vision and Pattern Recognition (CVPR)*, pages 1–8, 2008.

[133] Y. Zhao, Z. Liu, L. Yang, and H. Cheng. Combining rgb and depth map features for human activity recognition. In *Signal Information Processing Association Annual Summit and Conference (APSIPA ASC), 2012 Asia-Pacific*, pages 1 –4, dec. 2012.