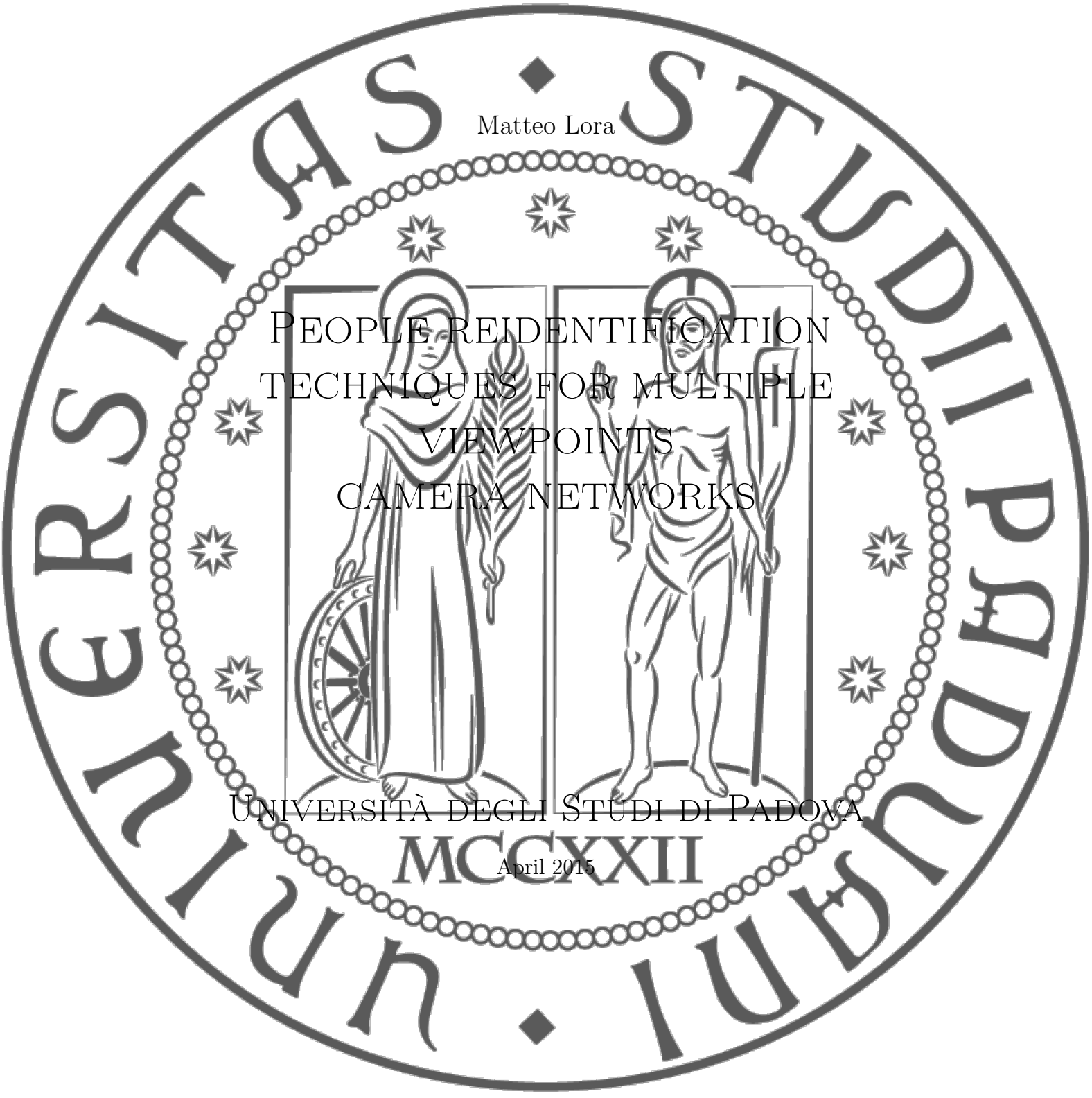
Matteo Lora



People reidentification techniques for multiple viewpoints camera networks

Università degli Studi di Padova

April 2015

Università degli Studi di Padova

Dipartimento di Ingegneria dell'Informazione

Corso di Laurea Specialistica in Ingegneria Informatica

People reidentification techniques for multiple viewpoints camera networks

Matteo Lora

Supervisor:
Prof. Emanuele Menegatti

Assistant supervisor:
Stefano Ghidoni

Anno accademico 2014-2015

Abstract

In this work we address short-term people re-identification, the task of recognizing a person that has already been observed by the system within a time frame of minutes or hours, assuming their clothing and general appearance hasn't changed. We present a novel approach for performing re-identification using local feature descriptors from joint locations of the human body without utilizing RGB-D sensors and related skeletal trackers. To obtain local keypoint coordinates we implemented a skeletal tracker built on top of a state-of-the-art pose detector whose output is augmented to work with multiple input cameras. For reidentification purposes, a single signature is extracted from each detection by combining features descriptors on keypoints and is then matched with a database to recognize a target person. We recorded two multi viewpoints dataset in order to build and test this procedure and obtain useful data.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Vision is the sense that allows humans to study the surrounding three dimensional world, to locate and recognize objects and to perceive the changes in the environment. Artificial vision (Computer Vision) is a computer science discipline that studies models and methods to allow machines to comprehend and interpret information contained in bi-dimensional or tri-dimensional pictures or videos. Computer Vision is a fast growing field thanks to the increase of both availability of high resolution sensors and computational power of modern computers. Results obtained in this field are due to intensive research activity in the last decades that, since its origin in the 80's, led to an exponential growth of scientific publications in the sector. Applications of Computer Vision are countless since the camera is a very versatile and noninvasive tool. Camera versatility is related to the amount of information that can be found in images, and the increasing capability of extracting only the salient ones in a specific application. Computer Vision is gaining popularity in industrial automation, in products quality control, in construction engineering and architecture, in military and aerospace applications. A standard model of Computer Vision is represented as one or more cameras connected to a computer which automatically interprets images of a real scene, obtaining useful information for robotic navigation and automatic detection and manipulation of objects. An inspiring Computer Vision application for autonomous robotic navigation is the visual odometry, the process of determining the position and orientation of a robot by analyzing images acquired during the motion, which has been used in a wide variety of robots, such

as on the Mars Exploration Rovers.

This thesis presents an approach to the short-term people re-identification problem in a calibrated RGB camera network system. RGB stands for Red Green Blue and it is a color model used in electronic systems to sense and represent colors. An RGB camera is a typical camera that can acquire color images. The implemented system is based on techniques that allows to estimate human poses within pictures and extract local visual descriptors that are used to effectively identify persons. Both pose estimation and people re-identification are hard problems that are not completely solved in the Computer Vision field. More formally, people re-identification is the capability of associating a new observation of a person to others made in the past. Distinguishing the different persons that are in the environment is a high-level capability that is crucial in several fields including service robotics, intelligent video surveillance systems and smart environments. People re-identification systems allows the accomplishment of tasks like automatic tracking of different moving objects or people in a given area, the automatic extraction of a sequence containing a specific person given a single frame, and others.

State-of-the-art person re-identification methods are mostly based on global clothings and body appearance since face recognition, that is potentially more effective, is often unpractical in video surveillance and robotics imagery due to low resolutions of pictures, the presence of occlusions with objects and self-occlusions and strong variations of illumination.

1.1 Related Work

The proposed re-identification method addresses two crucial problems in the Computer Vision field: human pose estimation and short term people re-identification itself. Since both of these problems are still unsolved and lots of research is going on on these subjects, state-of-the-art related work is presented for both these problems separately in this section.

1.1.1 Human pose estimation

Human pose estimation has made significant progress during the last years. Recent pose estimation methods employ complex appearance models and rely on learning algorithms to estimate model parameters from the training data.

The most used model for addressing this problem is the Pictorial Structure representation which was introduced by Fischler and Elschlager [12] in 1973. An object is modeled by a collection of parts arranged in a deformable configuration. Each part encodes local visual properties of the object, and the deformable configuration is characterized by spring-like connections between given pairs of parts. The best match of such model to an image is found by minimizing an energy function that measures both a match cost for each part and a deformation cost for each pair of connected parts. This formulation was very simple but had several shortcomings that limited its use, mainly the fact that the resulting energy minimization was hard to solve efficiently. These problems were addressed 30 years later by Felzenszwalb et al [11], who provided an efficient algorithm for the classical energy minimization problem when the connections between parts do not form cycles, which is the case of people models. They also introduced a method for learning these models automatically from training examples, which learns all the model parameters, including the structure of connections between parts. A technique for finding multiple hypotheses for the location of an object in an image was also presented in [11]. In 2009, Andriluka et al [2] revisited the Pictorial Structure concept and proved that a generalization of the human model for articulated and complex poses was possible. They showed that the right selection of components for both appearance and spatial modeling is crucial for general applicability and performance of the model. The appearance of body parts is modeled using densely sampled shape context descriptors and discriminatively trained AdaBoost classifiers. Further improvements to the models were made by Yang and Ramanan [27] who described a general, flexible mixture model for capturing contextual co-occurrence relations between parts, improving standard spring models that encoded spatial relations. Pictorial structure models became the de facto standard for 2D human pose estimation. In 2013 Sikandar et al [1] addressed the task of articulated 3D human pose

estimation from multiple calibrated cameras. This task is traditionally tackled using 3D body models and involves complex inference in a high-dimensional space of 3D body configurations. They proposed an effective method built on the success of 2D pictorial structure models that allows the formulation of the 3D inference problem as a joint inference over 2D projections in each of the camera views. This model has been further updated by Belagiannis et al in 2014 [6] to manage the case of multiple persons in the same scene. They introduced a discrete state space which allows fast inference. Pose detection methods are significantly challenged by cases outside their comfort zone, such as loose clothing and occlusions. From all other factors, pose complexity has the most profound effect on the pose estimation performance. Current methods perform best on activities with simple tight clothing, and are challenged by images with complex clothing and background clutter that are typical for many occupational and outdoor activities [3].

1.1.2 People re-identification

The people re-identification problem is a widely studied area and it can be subdivided into two sub-problems: short term and long term re-identification. Since this work addresses the short term re-identification problem, long term specific techniques are neglected. Short term re-identification is the problem of recognizing a person that has already been observed by the system within a time frame of minutes or hours, assuming their clothing and general appearance hasn't changed. In state-of-the-art works, to address the problem in static 2D images, three main characteristics are generally observed (as stated in [19]): color, texture and shape. Color is the most widely exploited feature for people re-identification. It is usually measured using local or global histograms on classical color spaces (e.g. RGB, HSV ...). This approach was proposed and refined by [21] [17] and more, who worked to reduce its main flaw: illumination changes tend to make color matching ineffective. This also happens when the picture is taken from two different cameras whose position and light conditions are different. Beside this, color based re-identification is simple and effective. More recent and sophisticated approaches are the following: Cheng et al [9] built upon the Pictorial Structures framework a re-identification model. Color

histograms of each detected part are computed independently, that are then concatenated and normalized to obtain a single feature vector for each image. Histograms are extracted in HSV space and weighted to take the different size and relevance of each part into account. Parameters are then tuned automatically using cross-validation so that they can be adapted to different visual conditions. Farenzena et al [10] proposed a model called SDALF that addresses the problem by gathering color information, spatial disposition of color into stable regions and the presence of recurrent local motifs with high entropy. The effects of pose variations are minimized by weighting body parts with respect of the vertical axis of the human body. The SDALF distance is then obtained by the sum of: histogram distance, Recurrent High Structured Patches (RHSP) distance and Maximally Stable Color Regions (MSCR) distance. Each type of feature encodes different information, namely, chromatic information, structural information through uniformly colored regions, and the nature of recurrent informative (in an entropy sense) patches. In this way, robustness to pose, viewpoint and illumination variations is achieved. Texture- and shape-based approaches usually make use of local features by exploiting 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. Bauml and Stiefelhagen [4] provided a comparison of different local features detection and extraction algorithms for re-identification. They showed that GLOH and SIFT clearly outperforms Shape Context (SC) and SURF when applied to local keypoints, while different detectors give similar results. This suggests the need to find a set keypoint locations where a given features extractor can compute an effective description of a person appearance. Munaro, Ghidoni et al [19] proposed a novel methodology based on human pose information. They compared different 2D and 3D feature descriptors that are applied on precise keypoints obtained from a third party skeletal tracker (e.g. OpenNI using Microsoft Kinect sensor). Target skeleton is obtained in 3D space using the tracking framework and is then projected onto the RGB image, thus acquiring 2D keypoints. This approach proved to be robust to illumination changes and occlusions and achieved very high performance, overcoming state-of-

the-art methods in terms of recognition accuracy and efficiency.

1.2 Our approach

The work presented is in thesis is mainly based on Munaro, Ghidoni et al [19] results and aims to provide a completely automatic method to address the short term people re-identification problem by local features comparison on skeleton joints without using RGB-D sensors. In order to obtain local keypoints, we use a 2D pose estimation algorithm based on Yang and Ramanan [27] on RGB pictures. This algorithm extracts patches from pictures corresponding to body parts, which are then reworked to obtain 2D poses candidates. Furthermore the method is generalized to support multiple cameras and has been developed to diminish the limitations of both pose detection and people re-identification problems by exploiting information from multiple points of view. Detection performance is improved in case of occlusions and false detections thanks to the information coming from multiple viewpoint. Poses obtained in each camera are then compared in order to obtain a coherent pose in the 3D space, which is then reprojected into each picture and used as keypoint for the feature descriptors extraction. A multi-viewpoint signature is obtained by combining all the descriptors, which is then used for evaluating a comparison with a model database. Feature matching is not needed because the correspondence between points is already known and different distance calculation approaches can be put in place by considering descriptors of the same person from multiple cameras. To prove the effectiveness of the method, we recorded two dataset, with respectively two and three calibrated camera viewpoints, upon which the algorithm was built and tested.

Chapter 2

Human pose estimation

As mentioned in Section 1.2, we addressed the people re-identification problem by applying a pose detection algorithm on pictures which extracts the keypoints where features are subsequently computed. Pose estimation is a tough Computer Vision problem, especially in the case of unfriendly environments. As described in Section 1.1.1, todays state-of-the-art pose detection algorithms are able to extract poses even in case of cluttered backgrounds and body parts occlusions, even though their accuracy can be unsatisfactory for certain tasks. We based our pose estimation phase of the re-identification algorithm on the Yang and Ramanan model [27]. A free and open source C++ implementation of the model is available on GitHub at [25] called Parts Based Detector (PBD). We then developed an independent 3D pose estimator built on PBD that augments its output by adding informations from multiple cameras, improving accuracy by handling bad detections and enforcing pose coherency between viewpoints.

2.1 Extracting human poses

Pose detection in 3D space is obtained with different subsequent steps, which can be summarized as follows:

1. System setup: a preliminary phase that is executed only once per system. In this phase we provide camera system configuration to the framework in order to enable operations in three dimensional space references.

2. Obtaining skeleton candidates in 2D space: the classifier is fed with input pictures from each camera to perform the initial body parts detection. Detected body parts in each camera are then transformed into a human skeleton defined by joints positions in 2D space. The number of parts is normalized to a 14 joints skeleton (see Figure 2.3).

3. Obtaining skeletons in 3D space and reprojection: detected joints are triangulated from multiple points of view to obtain a representation in 3D space. Optimization algorithms are used to extract the most promising pose. The computed pose in 3D space is then projected back into all the viewpoints, thus re-obtaining a 2D representation which is used for keypoints extraction.

These steps are explained in detail in the next sections.

2.1.1 System setup

We worked on the implementation of a generalized framework that can be used for any kind of camera setup, given that calibration informations are present. Calibration parameters are the intrinsics and extrinsic parameters of the system cameras and are described using the standard OpenCV [7] matrix representation as explained in [8]. The parameters are needed to solve a general projection problem that can be summarized in the following form:

$$x_{2D_H} = P \cdot x_{3D_H} \,. \tag{2.1}$$

Where x_{2D_H} is a point on a plane (corresponding to the plane of a camera viewpoint) in 2D coordinates, P is a 3×4 projection matrix and x_{3D_H} is a point in the 3D space. The subscript H (e.g. x_{2D_H}) is used to describe a vector in homogeneous coordinates. The parameters involved in Equation 2.1 are the following: the camera intrinsics matrix K, rotation matrix R and translation vector T.

Intrinsics matrix K

The intrinsics matrix or camera matrix K contains the parameters for the projection of the points from the 3D world coordinates into the camera coordinates, summarized by the following simple form:

$$x_{2D_H} = K \cdot x_{3D}, \quad \text{where } K = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} . \tag{2.2}$$

The matrix holds four different parameters f_x, f_y, c_x, c_y in a way that is convenient for computations. f_x and f_y represent the camera focal length in pixel units while c_x and c_y are the coordinates of the principal point in pixels. These parameters are used to describe a camera model called pinhole. [8]. The result is a relatively simple model in which a point p in the physical world, whose coordinates are (X, Y, Z), is projected onto the screen at some pixel location given by $(x_{\text{screen}}, y_{\text{screen}})$ in accordance with the following equations:

$$x_{\text{screen}} = f_x \left(\frac{X}{Z}\right) + c_x \,, \qquad y_{\text{screen}} = f_y \left(\frac{Y}{Z}\right) + c_y \,. \tag{2.3}$$

Rotation matrix R and translation vector T

Rotation matrix and translation vectors are a description of the pose of each camera relative to a common world reference. A way of describing a general rigid rotation in a three-dimensional coordinate system is a square 3×3 matrix. It can be obtained as a product of basic rotations (rotation about one of the axes of a coordinate system),

which are the following:

$$R_x(\theta) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\theta & -\sin\theta \\ 0 & \sin\theta & \cos\theta \end{bmatrix}, \tag{2.4}$$

$$R_y(\theta) = \begin{bmatrix} \cos\theta & 0 & \sin\theta \\ 0 & 1 & 0 \\ -\sin\theta & 0 & \cos\theta \end{bmatrix}, \tag{2.5}$$

$$R_z(\theta) = \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}. \tag{2.6}$$

Thus, $R = R_z(\theta), R_y(\theta), R_x(\theta)$, where R_x, R_y, R_z are rotations around x, y, z axes respectively. It is also possible to obtain R from other rotation formalisms in three dimensions, like Euler angles, Rodrigues or Quaterion. It is particularly useful to convert a quaternion Q to a rotation matrix R as many Computer Vision tools makes use of this formalism (e.g. OpenPTrack [20], see Section 4.1). A rotation matrix R can be obtained from the quaternion $Q = [q_1\ q_2\ q_3\ q_4]^T$ using the formula:

$$R = \begin{bmatrix} 1 - 2q_2^2 - 2q_3^2 & 2(q_1q_2 - q_3q_4) & 2(q_1q_3 + q_2q_4) \\ 2(q_1q_2 + q_3q_4) & 1 - 2q_1^2 - 2q_3^2 & 2(q_2q_3 - q_1q_4) \\ 2(q_1q_3 - q_2q_4) & 2(q_1q_4 + q_2q_3) & 1 - 2q_1^2 - 2q_2^2 \end{bmatrix}. \tag{2.7}$$

The translation vector is how we represent the shift from each camera and the common world reference. In other words, the translation vector is just the offset from the origin of the world coordinate system to the origin of the camera coordinate system. It is represented as a 3×1 vector T:

$$T = [\Delta_x, \Delta_y, \Delta_z]^T \tag{2.8}$$

Rotation matrix R and translation vector T are then combined to obtain a 3×4 matrix RT:

$$RT = [R\ \ T]\ . \tag{2.9}$$

Common coordinates and projection matrix P

We chose to use the camera 1 coordinates system as a system world reference. Therefore every camera coordinate system is transformed to be referred to camera 1 origin and orientation. To simplify this operation, the RT matrix is transformed into homogeneous coordinates so that its inverse can be computed.

$$RT_H = \begin{bmatrix} R & T \\ 0\,0\,0 & 1 \end{bmatrix} . \tag{2.10}$$

Camera 1 has then its RT set as a 4×4 identity matrix, while the $i-$th camera RT is computed as:

$$RT_{i_H} = RT_{1_H}^{-1} RT_{i_H} . \tag{2.11}$$

A projection matrix P, which allows to project 3D points into each camera viewpoint is then computed for each camera. P is a 3×4 matrix obtained as:

$$P_i = K_i RT_i^{-1} . \tag{2.12}$$

Where RT_i^{-1} is equal to $RT_{i_H}^{-1}$ with the last row removed. The P matrix is handy as it allows simple projections of a 3D point on each camera of the system by the simple multiplication in Equation 2.1.

2.2 Obtaining skeleton candidates in 2D space

As mentioned before, 2D pose detection is based on the Yang-Ramanan algorithm implementation available at [25]. This classifier is initialized using a pre-trained full body model which is publicly available at [26], which describes human poses as a combination of 24 constrained patches. The model is trained on a dataset called PARSE by Navneet Dalal, Mun Wai Lee, Greg Mori and Jiayong Zhang, which consists of 305 pictures representing humans in complex and articulated poses. The PARSE dataset is provided with a hand written ground truth for training and 100 of the 305 pictures have been used as a training set for the model, while the

remaining are used for testing. A picture from each camera of the setup is then given as input to the PBD classifier. Since computing poses on each picture is computationally very expensive, images are downsampled to a target resolution of 320×240, while maintaining the original image aspect ratio using the formula. This is done downscaling both height and width of the picture by the same scale factor f_{xy}, obtained as

$$f_{xy} = \max\left\{\frac{320}{w}, \frac{240}{h}\right\}, \tag{2.13}$$

where w and h are picture width and height respectively. This often also improves classification results as less textured background details, which can deceive the detector, can be extracted from the low resolution image. PBD detects a set of candidate poses sorted by confidence score, which will be later used to estimate the most coherent pose in 3D space. The model described in [27] is defined as follows: let I be

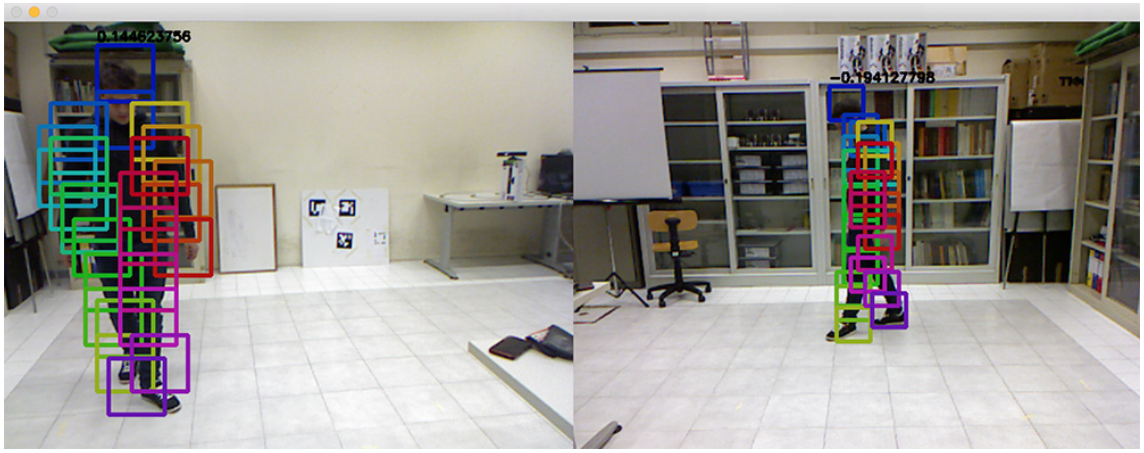


Figure 2.1: Example of detected patches from the classifier in each of the system camera pictures. Each of the patches is a square of different color, and their disposition and constraints are described in the PARSE model.

an image, $p_i = (x, y)$ for the pixel location of part i and t_i for the mixture component of part i, where $i \in 1, \ldots, K$, $p_i \in 1, \cdots, L$, $t_i \in 1, \cdots, T$. t_i is defined as the type of part i, and K, L, T are respectively the numbers of different parts, number of possible pixel locations for a part and number of part types defined in the model. As an example the type can determine the orientation of a part, but it can also span semantic classes (e.g. open versus closed hand). To score a configuration of parts,

a compatibility function for local part types is first defined:

$$S(t) = \sum_{i \in V} b_i^{t_i} + \sum_{ij \in E} b_{ij}^{t_i, t_j} \tag{2.14}$$

The parameter $b_i^{t_i}$ favors particular type assignment for part i, while the pairwise parameter $b_{ij}^{t_i,t_j}$ favors particular co-occurrences of part types. For example, if part types correspond to orientations and part i and j are on the same rigid limb, then $b_{ij}^{t_i,t_j}$ would favor consistent orientation assignments. A K-node relational graph $G = (V, E)$ edges specifies which parts are constrained to have consistent relations. The full score associated with a configuration of parts type and positions is:

$$S(I, p, t) = S(t) + \sum_{i \in V} w_i^{t_i} \cdot \phi(I, p_i) + \sum_{ij \in E}^{t_i, t_j} \cdot \psi(p_i - p_j) \tag{2.15}$$

where $\psi(I, p_i)$ is a feature vector (e.g. HOG descriptor) extracted from pixel location p_i in image I, and $\psi(p_i - p_j) = [dx \quad dx^2 \quad dy \quad dy^2]^T$, where $dx = x_i - x_j$ and $dy = y_i - y_j$, the relative location of part i with respect to j. This relative location is defined with respect to the pixel grid and not the orientation of part i. The extraction of the best model occurrence from I, or inference, corresponds to maximizing $S(x, p, t)$ over p and t. When $G = (V, E)$ is a tree, this can be done efficiently with dynamic programming. Let kids(i) be the set of children of part i in G. The message that part i passes to its parent j is computed by the following equations:

$$\text{score}_i(t_i, p_i) = b_i^{t_i} + w_{t_i}^i \cdot \psi(I, p_i) + \sum_{k \in \text{kids}(i)} m_k \in (t_i, p_i) \tag{2.16}$$

$$m_i(t_j, p_j) = \max_{t_i} b_{ij}^{t_i, t_j} + \max_{p_i} \text{score}(t_i, p_i) + w_{ij}^{t_i, t_j} \cdot \psi(p_i - p_j) \tag{2.17}$$

Equation 2.16 computes the local score of part i, at all pixel locations p_i and for all possible types t_i, by collecting messages from the children of i. Equation 2.17 computes for every location and possible type of part j, the best scoring location and type of its child part i.

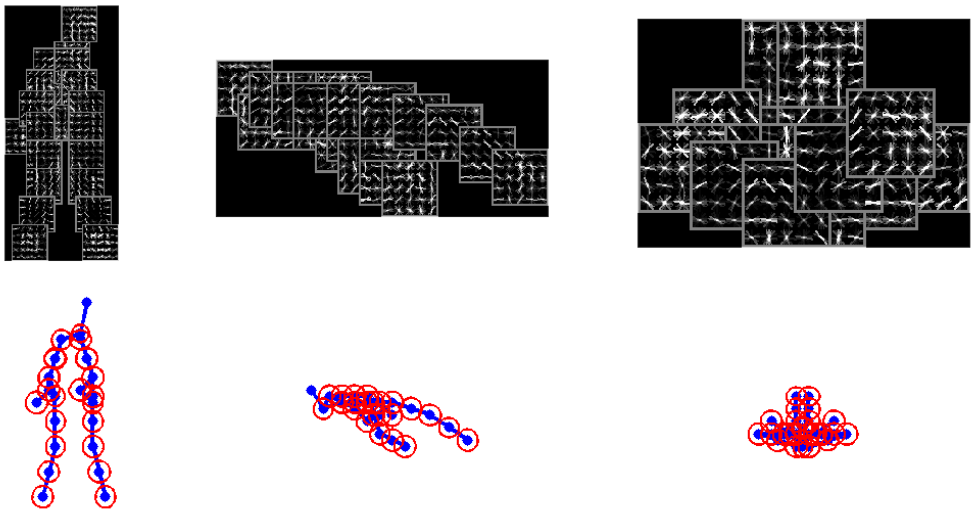


Figure 2.2: A visualization of the model trained on the Parse dataset. Local templates are shown on the figure above and the tree structure below, placing parts at their best-scoring location relative to their parent. This example shows 3 trees, even though there exists an exponential number of combinations, composing different part types. [27]

Once messages are passed to the root part ($i = 1$), $\text{score}_i\,(c_1, p_1)$ represents the best scoring configuration for each root position and type. For a more in-depth explanation of the model inference and training refer to Yang, Ramanan article [27]. Multiple detections in image I are generated by using these root scores by thresholding them and applying non-maximum suppression. Non-maxima suppression is an algorithm that removes overlapping detections whose score is not maximal, thus ideally removing multiple detections of a same pose. In this case non-maxima suppression is not applied as multiple pose candidates for each person in the picture are desired for a further optimization that leverages multi viewpoints informations. Therefore the PBD classifier outputs a set pose candidates, each of which consists of a vector of square image patches and a confidence score. Skeleton joints are then extracted from image patches by computing their center and upscaled to the original image coordinates by multiplying the center point vector by the inverse of scale factor previously used for downsampling.

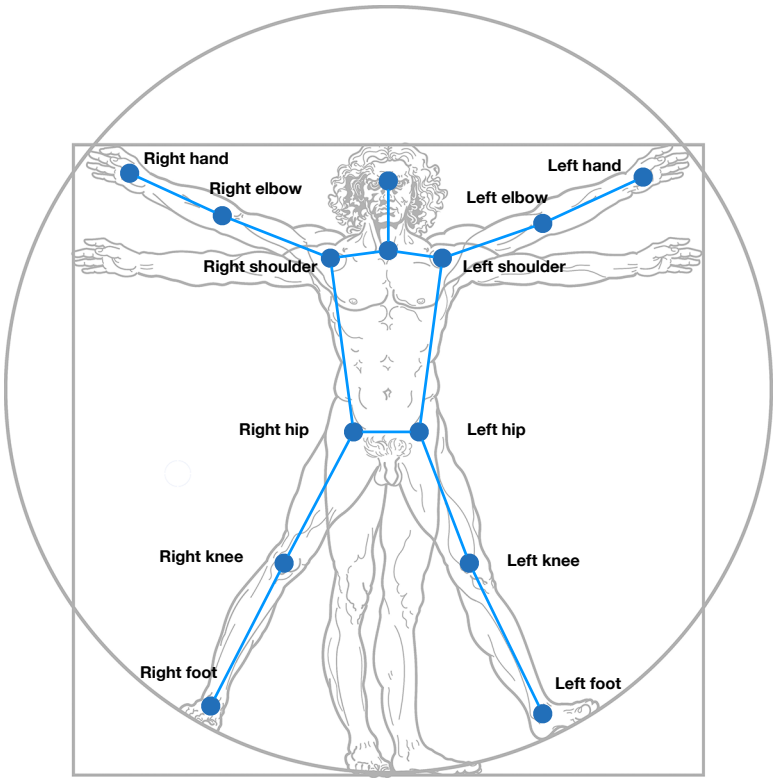


Figure 2.3: Location and name of the joints estimated by our skeletal tracker.

This operation yields a set of skeleton candidates, each of which is defined as a vector of two dimensional points and a confidence score which is inherited from the raw PBD output. Image patches are extracted by PBD in a structured way and they have a precise ordering, which is determined in the trained model. This allows to know the correspondence between image patches and skeleton joints. The tracked joints are shown in figure 2.3. Unfortunately the PBD model is unable to distinguish between frontal and posterior poses, so legs and arms orders might be inverted between viewpoints. We proposed a solution to the problem in the 3D pose estimation phase (see Section 2.3).

2.3 Obtaining skeletons in 3D space and output reprojection

Skeleton joints candidates obtained in 2D space from each camera in the system are then processed to obtain a single skeleton in 3D space. We developed different methods for merging informations obtained from all cameras to improve final detection: best detections fusion, parts clusters centroids fusion, minimization of 3D pose reprojection error. All these methods use projection and triangulation as basic operations.

- **Projection:** The projection of points from 3D space into the viewpoint of a single camera of the system. This operation is described in section 2.1.1.

- **Triangulation:** This operation uses calibration information to estimate the 3D coordinates of a points by observing its position on at least two different viewpoints. Triangulation is performed as an algebraic minimization as described in [24] [14] and is generalized for $n-$viewpoints. We derive an algebraic constraint where we note that the unit ray of an image observation can be stretched by depth α to meet the world point X for each of the n observations.

  $$\alpha_i \overline{x_i} = P_i X \,,$$

  for images $i = 1, \cdots, n$. This equation can be effectively rewritten as:

  $$\alpha_i = \overline{x_i}^T P_i X \,,$$

  which can be substituted into our original constraint such that:

  $$\overline{x_i x_i}^T P_i X = P_i X \,,$$

  $$0 = \left(P_i - \overline{x_i x_i}^T P_i\right) X \,.$$

  We can then stack this constraint for each observation, leading to the linear

least squares problem:

$$\begin{bmatrix} \left(P_1 - \overline{x_1 x_1}^T P_1\right) \\ \vdots \\ \left(P_n - \overline{x_n x_n}^T P_n\right) \end{bmatrix} X = 0\,.$$

This system of equations is of the form $AX = 0$ which can be solved by extracting the right nullspace of A. The right nullspace of A can be extracted efficiently by noting that it is equivalent to the nullspace of $A^T A$, which is a 4×4 matrix. We used the implementation of this algorithm from Theia Vision Library [24].

In all of the implemented methods, correspondence between points of different viewpoints is known a priori. For example, let x_i^k be the point corresponding to the head of the $k-$th pose detected in the picture of camera i and assume we want to find the correspondent 3D pose of the $k-$th candidates set. The point x_{3D}^k corresponding to the head is obtained as $x_{3D} = \text{triangulateNView}(x_0^k, x_1^k, \cdots, x_n^k)$ from Theia Vision Library [24]. This is allowed by the fact that, as mentioned in section 2.2, PBD distinguishes the detected parts and stores them in a structured way.

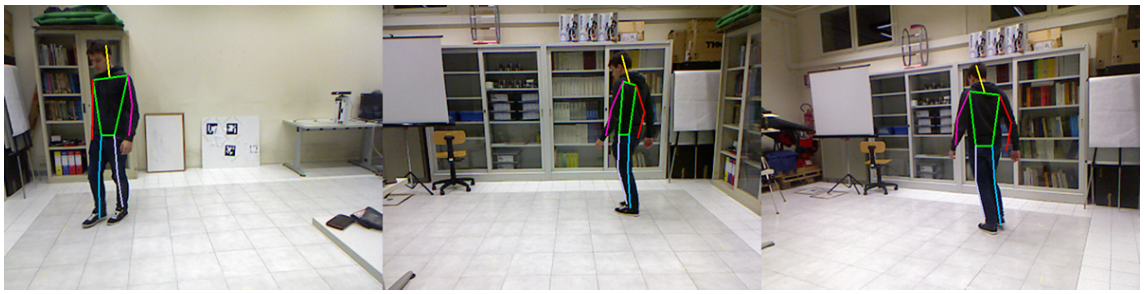


Figure 2.4: Final skeleton projected on each viewpoint picture

A visualizer has also been implemented to show the whole system setup and the skeletons in 3D space before reprojecting. It is based on the Point Cloud Library (PCL) [23] visualizer API and it allows to represent 3D world coordinates and references interactively. An example of detected 3D pose is shown in figure 2.5.

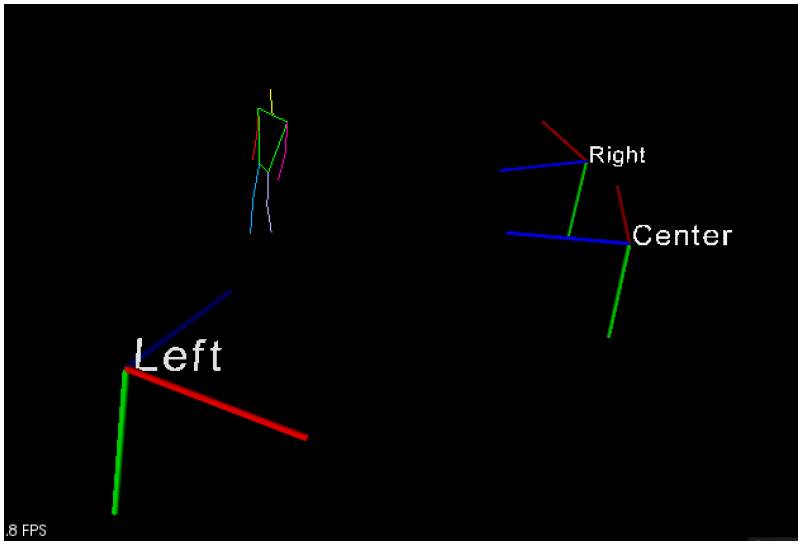


Figure 2.5: Three dimensional pose and system cameras rendered with the PCL visualizer. The system origin and references coincides with Camera 1 (Left), while Camera 2 and 3 (Center and Right) are translated and rotated from it.

Fusion of Best detections

Fusion of best detections is a simple method that exploits the capability of the PBD classifier of extracting good poses from 2D images. The best detection is extracted from every viewpoint picture using a non-maxima suppression function available with the PBD, that removes overlapping detections with non-maximal score. A 3D pose consistent with all the viewpoints is then obtained by applying point per point triangulation between the best detections of each camera of the system. The obtained pose in 3D space is then reprojected onto each viewpoint picture, thus obtaining 2D keypoints coordinates. This operation is conceptually equivalent as resolving a linear least square problem to approximate in 3D space the best observations given by the PBD. The obtained 3D skeleton is then reprojected into each viewpoints thus getting final joints coordinates.

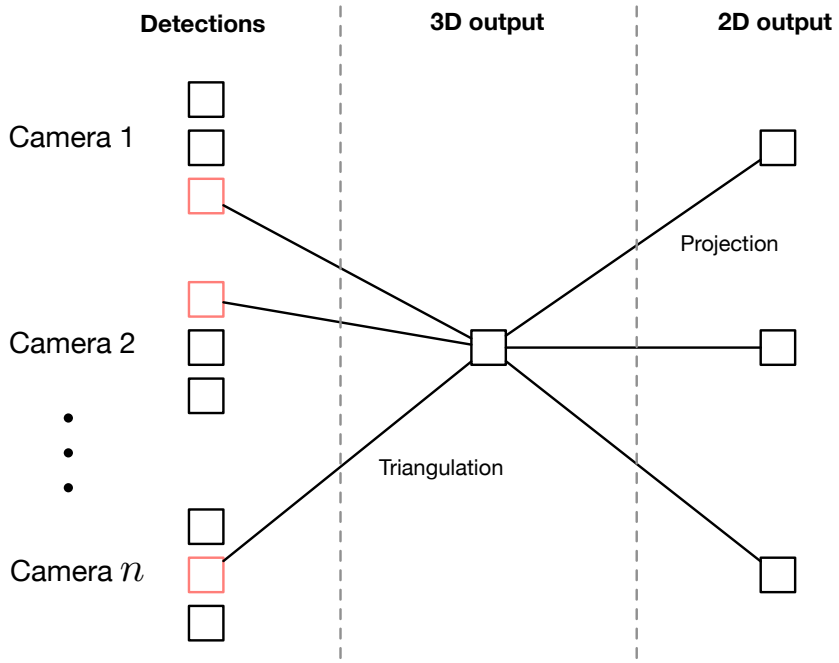


Figure 2.6: Best detections fusion algorithm DAG. Best detections for each viewpoint are marked in red and are projected to 3D space by a triangulation operation. The obtained 3D model is then reprojected into each camera viewpoint.

Fusion of Parts Clusters Centroids

Fusion of Parts Clusters Centroids is based on the idea that the detection of some parts of the skeleton is very stable and their location is very similar between all candidates in each viewpoints. This is especially true for the skeleton joints: head, neck, left and right shoulders, left and right hips (see figure 2.3). To exploit this property of detections, all detection candidates from each viewpoint are merged together by clustering each joint across all different detections. New joints are therefore obtained by computing mean coordinates for each different joint. Coordinates for $j-$th joint are computed by the formula:

$$x_j, y_j = \left(\frac{\sum_{i=0}^{n} x_{j_i}}{n}, \frac{\sum_{i=0}^{n} y_{j_i}}{n} \right) . \tag{2.18}$$

This reduces the amount of candidates to 1 per viewpoint, and the 3D pose problem can be solved in the same way as in the Fusion of best detections method by

performing joint per joint triangulation. The obtained pose in 3D space coordinates is then reprojected onto each viewpoint picture.

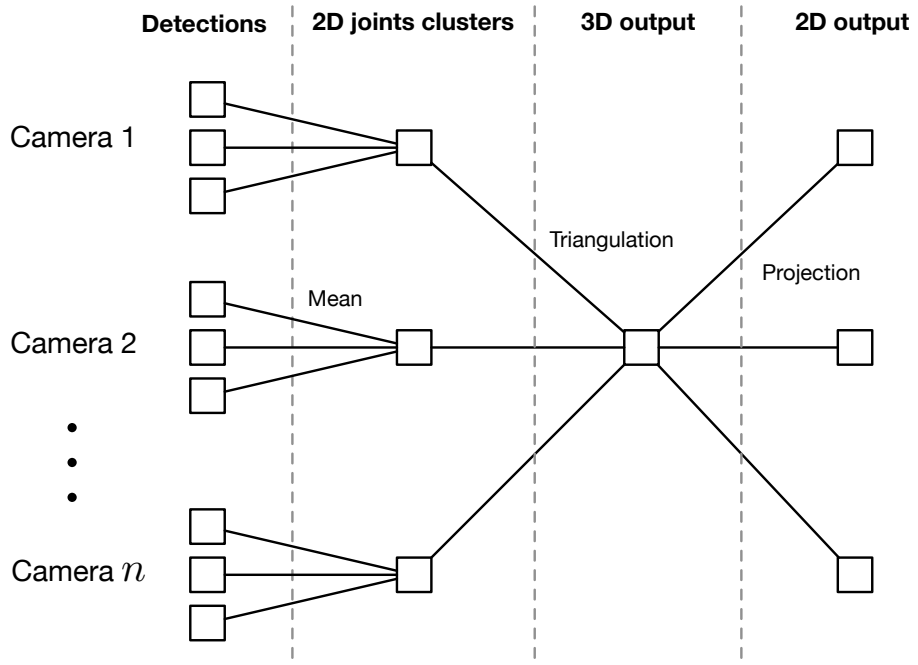


Figure 2.7: Parts clusters centroids fusion algorithm DAG. An example graph is shown for $k = 3$ candidates per viewpoints. The function perform an averaging operation on clusters of joints on the same type. The centroids obtained are projected to 3D space using $n-$point triangulation and then reprojected into each one of the camera viewpoints.

The downside of this method is that articulated poses are difficult to detect and the position of unstable joints like hands and feet is often wrong.

Minimization of Reprojection Error

Minimization of Reprojection Error algorithm (MRE) exploits the concept that since detected pose candidates are very likely distributed around the highest quality detection, there must be a $n-$plet between viewpoints (where n is the number of different viewpoints) that has a very good 3D representation which implies that their 2D appearance is also very good.

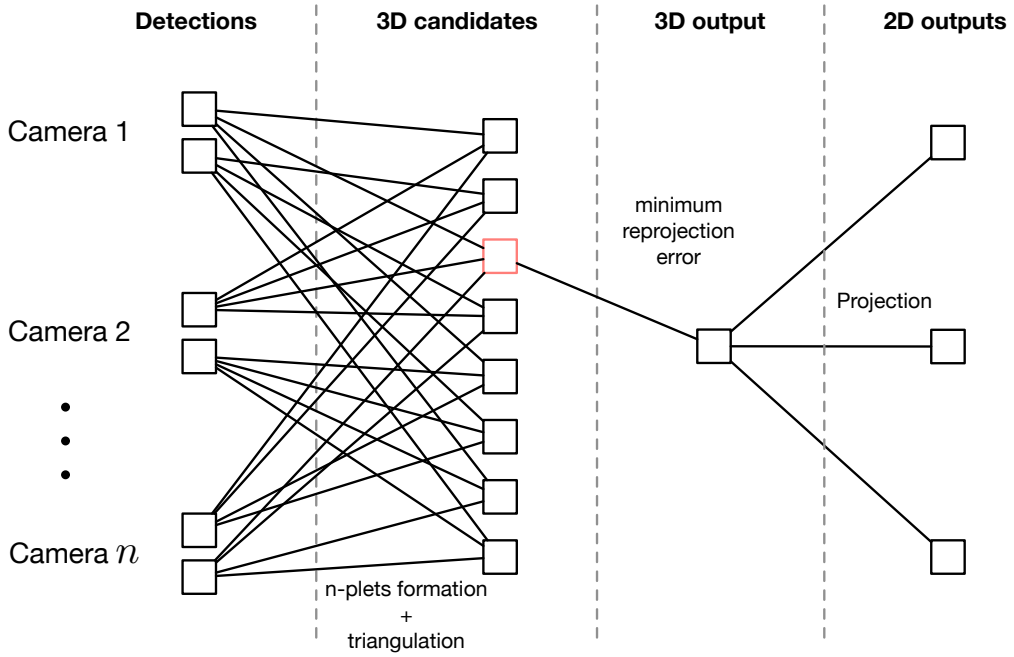


Figure 2.8: Minimization of reprojection error algorithm DAG. Example graph is shown with number of candidates $k = 2$, even though we used 5 and 10 in our tests. The 3D candidate with minimum reprojection error is marked in red and is chosen as output of the phase. The 3D output is then projected into each camera viewpoints.

This method attempts to avoid enforcing coherence between 2D poses, instead it finds the candidates set that already fits a 3D model at best. Let $C = \{c_{00}, \cdots, c_{1n}, \cdots, c_{kn}\}$ be a set of candidates for n viewpoints and k candidates per viewpoint. The function attempts to find a subset of C such that every candidate c belongs to a different viewpoint and the sum of errors on reprojected joints is minimum. The candidate set is projected to 3D spaces using triangulation operations on every set of joint types, and the obtained pose is then reprojected to each viewpoint to estimate error. The operation of finding the best $n-$plet is performed by a brute force algorithm within the search space of $n-$plets whose size is $O\left(k^n\right)$, where k is the number of detections candidates per viewpoint. The size of k can be limited, improving the performance of the algorithm and also the quality, since detections of lower quality (be the means of PBD confidence score) usually don't improve the result. The problem of exponential scaling of this method search space is in any

case negligible if compared with computational load of the PBD algorithm.

Chapter 3

Re-identification process

The reidentification method proposed is based on the concept presented in [19], which relies on computing a person signature by concatenating features descriptors extracted at body joints locations. Notation used in this chapter is also very similar to the one in [19]. Skeleton joints obtained using a tracker are very stable if compared to standard feature keypoints, and the signature obtained from the set of descriptors extracted on them have shown to be a very good description of a person appearance. Moreover the keypoint detected in this way are already labeled and can be compared without the need for a previous matching. This idea has been extended to a multi viewpoint environment, therefore the person descriptor itself and the matching process are augmented to keep track of information from multiple sources. The reidentification system can be logically divided into two steps:

1. Features extraction and Skeleton-based Person Signature (SPS) composition: a selected type feature is extracted from every joint of the skeleton detected from each viewpoint of the system. A compact signature is then computed by aggregating all the obtained descriptors into a single object called Skeleton-based Person Signature or SPS.

2. Signature matching: the SPS is matched against each person of the database or training set. Most feature types support a standard distance metric. Nevertheless, since SPS is multidimensional and contains more than a single descriptor for each joint, different matching strategies are possible. We implemented and

tested two different methods of SPS matching, which are described in Section 3.3.1.

3.1 Skeleton-based Person Signature extraction

SPS extraction is handled by computing descriptors on a set of keypoints based on human joints previously detected. Unlike [19], since SPS is multidimensional, local descriptors cannot be concatenated into a single vector. They are instead represented as a two-dimensional vector where each part can have multiple representations or, more easily, multiple descriptors extracted from different viewpoints.

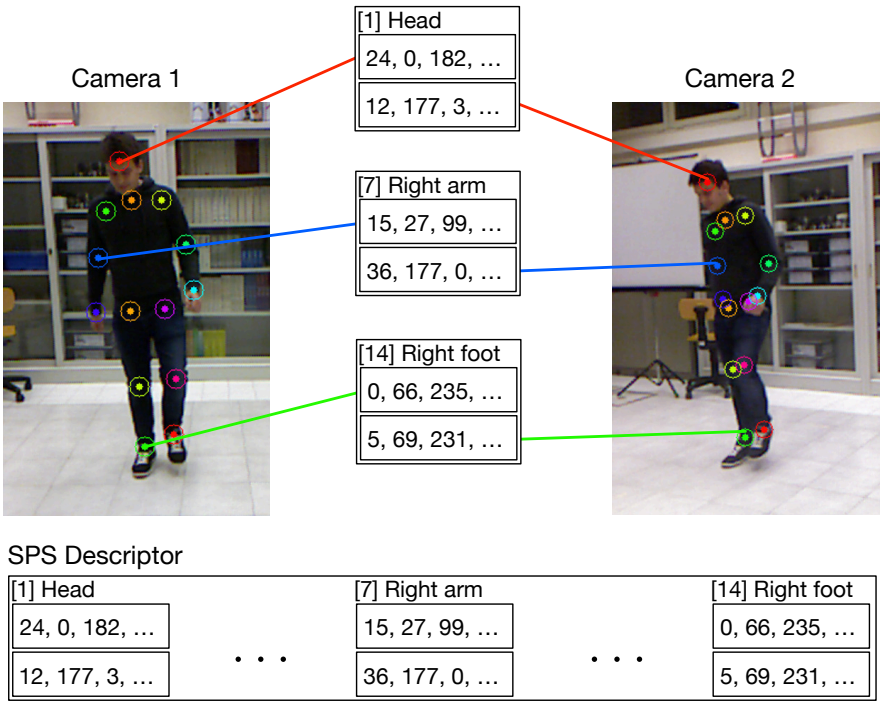


Figure 3.1: Example of multi dimensional Skeleton-based Person Signature creation from two cameras. Only joints 1 (head), 7 (right arm) and 14 (right foot) are shown. In the final descriptor, vectors corresponding to each joint are concatenated.

An example of the procedure with two cameras is shown in figure 3.1. In [19],

a feature tracking indicator I for the $i-$th joint J_i on the $k-$th target T_k is:

$$I\left(J_i, T_k\right) = \begin{cases} 1 & \text{if } i-\text{th joint of frame } k \text{ is tracked} \\ 0 & \end{cases} . \tag{3.1}$$

where $i \in [0, \cdots, N-1]$ and N is the number of joints considered. The value of I is automatically provided by Microsoft and OpenNI skeletal trackers for every joint. This is a very effective way to describe self occlusions, which unfortunately cannot be reproduced on the PBD based skeletal tracker because of its lack of accuracy if compared to 3D sensors based algorithms. To limit the error by the tracking inaccuracy of some specific joints, we introduced a vector of weight parameters θ for each group of descriptors in the SPS that allow to tune the matching algorithm so that descriptors distance relevance is not equal across all joints. The vector θ is static and must be defined manually when configuring the descriptor extractor. A reasonable way to set θ values is: $\theta_i = P\left(D(J_i)\right)$, where $P\left(D(J_i)\right)$ is probability that the joint is detected correctly by the skeleton tracker, scaled between 0 and 1. This value for each joint can be determined experimentally using a training set (see Section 4.2.1). Vector θ can also be used as a boolean mask to determine which joints descriptors are to be matched together, allowing to completely exclude some of them thus increasing method flexibility.

3.2 2D Features

We implemented a generalized local extractor that can be used with 2D features from different sources and libraries. Beside this, all the descriptors we tested are open source and available in the OpenCV library [7]. We tested and benchmarked the performance of the following descriptors: SIFT, SURF, HISTOGRAMS (a histogram based feature implementation described in Section 3.6). Each descriptor is defined with its own distance metric, and they are all described in detail in this section.

3.2.1 SIFT

Scale-Invariant Feature Transform (SIFT) is a keypoint detector and feature descriptor presented by Lowe in 1999 [15]. OpenCV library implements a version of this algorithm from the Lowe's 2004 article Distinctive Image Features from Scale-Invariant Keypoints [16]. The SIFT approach, for image feature generation, takes an image and transforms it into a "large collection of local feature vectors". Each of these feature vectors is invariant to any scaling, rotation or translation of the image. To help the extraction of these features the SIFT algorithm applies a 4-stage filtering approach: Scale-space extrema detection, keypoint localization, orientation assignment, and keypoint descriptor extraction. In this work, only step 4 of the SIFT algorithm is applied for feature matching because keypoints locations are provided by the skeletal tracker, while the orientation information is unavailable. To obtain descriptors, magnitude and orientation of the gradient of points located near the keypoint of interest. To obtain invariance to rotation, descriptor coordinates and orientation of gradient are rotated accordingly to the keypoint information. For this specific purpose, rotation is not applied due to unavailability of the keypoint rotation information. Magnitude values are then weighted according to a gaussian function with σ (standard deviation) equal to half of the descriptor window. Samples are then accumulated into histograms that summarize regions content. Every histogram bin (see Section 3.2.3) holds the sum of magnitudes of gradients whose orientation is similar to the one it represents. To avoid sudden shifts and between histograms orientation, trilinear interpolation is applied to distribute gradients values to adjacent histograms. Obtained descriptor is a vector containing the values of orientation histograms (bins heights). Lowe obtained better results with 4x4 histograms, each one with 8 bins. Therefore every descriptor is a 128 elements array ($4 \cdot 4 \cdot 8$). To make the descriptor less sensible to illumination changes, the following operations are applied: the vector is normalized to unit length (invariance to affine illumination changes); values higher than 0.2 are thresholded and the vector is normalized again (partial invariance to non linear illumination changes). The distance

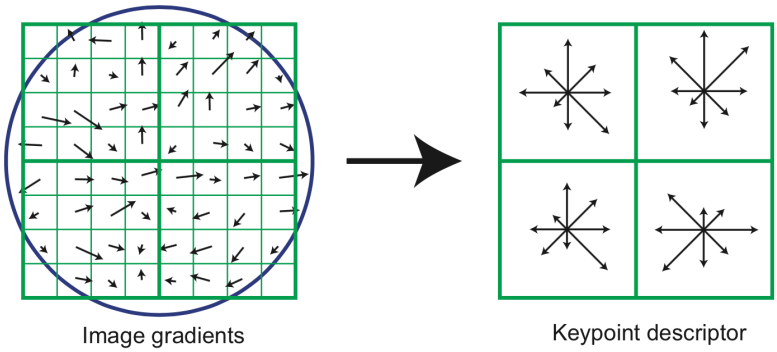


Figure 3.2: Gradient magnitude and orientation are computed at each image sample point in a region around the keypoint location, as shown on the left. These are weighted by a Gaussian window, indicated by the overlaid circle. These samples are then accumulated into orientation histograms summarizing the contents, as shown on the right. [16]

between descriptors is defined as an euclidean distance between vectors:

$$d(D_1, D_2) = \left| \sum_{i=0}^{128} D_{1_i} - D_{2_i} \right| . \tag{3.2}$$

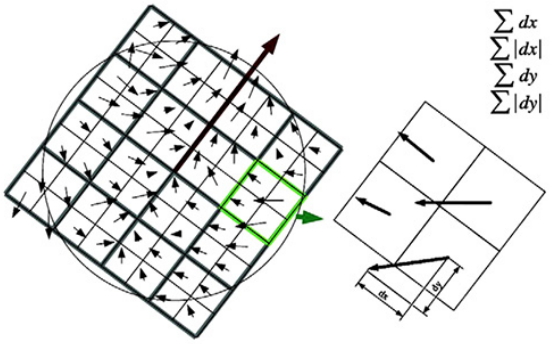
3.2.2 SURF

SURF (**S**peeded **U**p **R**obust **F**eatures) is a robust local feature detector, first presented by Herbert Bay et al [5] at the 9th ECCV in 2006. The SURF algorithm, was created for detecting and creating a descriptor in a more efficient way than SIFT. It is derived from the SIFT (see Section 3.2.1) which is known to be very robust, but also quite slow. Detection and description phases are faster on SURF and the descriptor size is decreased from 128 values to 64.

The first step in the construction of the descriptor is the definition of a square region centered on the keypoint, which in this case is given by skeleton tracker and the orientation is 0 as it is unavailable. Region is then subdivided in 4x4 subregions for each of which the response to Haar-like filters response in correspondence of a point grid is computed; horizontal response is defined as d_x, while vertical is d_y.

Each response is weighted with a Gaussian centered on the keypoint and a first set of features is obtained from weighted responses d_x and d_y in each subregion. The sum of absolute value $|d_x|$ and $|d_y|$ is also saved to keep track of intensity changes

(a) Detail of a scene showing the size of the descriptor window at different scales. [5]



(b) An oriented quadratic grid with 4×4 square sub-regions is laid over the interest point (left). For each square, the wavelet responses are computed from 5×5 samples. For each field, we collect the sums d_x, $|d_x|$, d_y, and $|d_y|$, computed relatively to the orientation of the grid (right). [5]

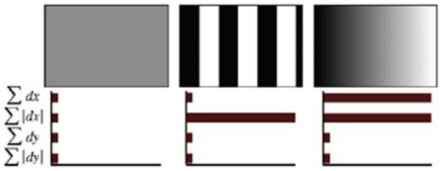


Figure 3.4: The descriptor entries of a sub-region represent the nature of the underlying intensity pattern. Left: In case of a homogeneous region, all values are relatively low. Middle: In presence of frequencies in x direction, the value of $|d_x|$ is high, but all others remain low. If the intensity is gradually increasing in x direction, both values $\sum d_x$ and $\sum |d_x|$ are high. [5]

polarity. For each subregion a descriptor is obtained as

$$v = \left(\sum d_x,\ \sum d_y,\ \sum |d_x|\,,\ \sum |d_y| \right) . \tag{3.3}$$

The concatenation of these values for each subregion yields a descriptor of length 64. The obtained descriptor is then normalized to obtain invariance to contrast changes.

There exists an alternative version of the SURF descriptor called SURF-128, in which informations are doubled in order to be more discriminative. It again uses the same sums as before, but now splits these values up further. The sums of d_x and $|d_x|$ are computed separately for $d_y < 0$ and $d_y \geq 0$. Similarly, the sums of d_y

and $|d_y|$ are split up according to the sign of d_x, thereby doubling the number of features. The descriptor is more distinctive and not much slower to compute, but slower to match due to its higher dimensionality.

3.2.3 Color Histograms

Histograms are collected counts of data organized into a set of predefined bins. The concept of data is not restricted to intensity values in an image, but can be whatever feature useful to describe an image. Color histogram is a representation of the distribution of colors in an image and can be built for any kind of color space, although the term is more often used for three-dimensional spaces like RGB or HSV. Let's see an example reported in the OpenCV documentation [7]. Imagine that a Matrix contains information of an image (i.e. intensity in the range 0-255):

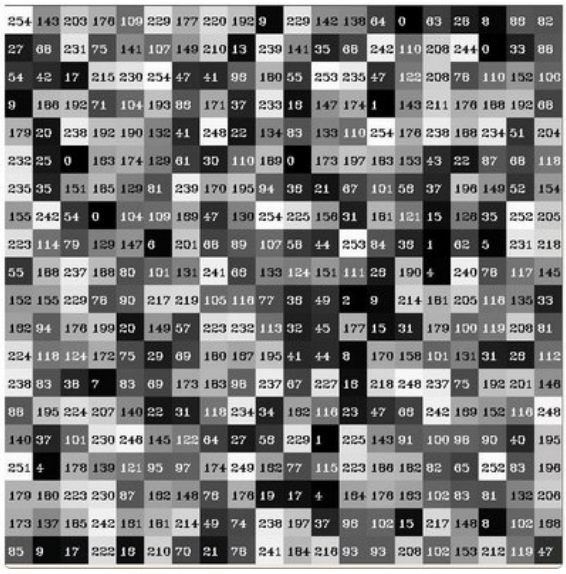


Figure 3.5: Example of a grayscale image with intensities of pixels in the range 0-255. OpenCV Library documentation, http://docs.opencv.org/

We want to *count* this data in an organized way. Since we know that the range information value for this case is 256 values, we can segment our range into subparts

called bins like:

$$[0, 255] = [0, 15] \cup [16, 31] \cup \cdots \cup [240, 255] \tag{3.4}$$

$$\text{range} = \text{bin}_1 \cup \text{bin}_2 \cup \cdots \cup \text{bin}_{n=15} \tag{3.5}$$

We can keep count of the number of pixels that fall in the range of each bin_i. Applying this to the example above we get the result in Figure 3.6 (x-axis represents the bins and $y-$axis the number of pixels in each of them).

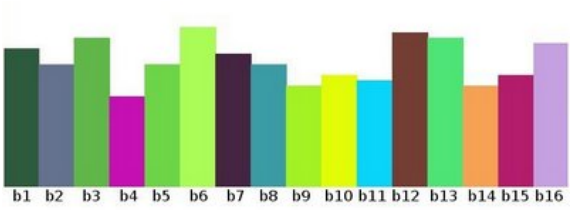


Figure 3.6: Results of histograms (with size of bin = 15) applied on intensity values of Figure 3.5. OpenCV Library documentation, http://docs.opencv.org/

Histogram descriptor parameters are:

- **dims:** The number of parameters of the collected data. In this example, dims = 1 becase we are only counting the intensity values of each pixel.
- **bins:** Number of subdivisions in each dim. In this example, bins = 16.
- **range:** Range of the values to be measured. In this case, range = [0, 255].

The histogram provides a compact summarization of the distribution of data in an image. The color histogram of an image is relatively invariant with translation and rotation about the viewing axis, and varies only slowly with the angle of view. Importantly, translation of an RGB image into the illumination invariant rg-chromaticity space allows the histogram to operate well in varying light levels (rg-chromaticity is a two dimensions space obtained from RGB space removing the intensity information by normalizing r, g, and b values on it) [18].

To compare two histograms H_1 and H_2, a distance metric $d(H_1, H_2)$ must be chosen first. Since histograms can be seen as an estimation of the probability density

function of the underlying variable, comparison between two histograms is computed using similarity metrics of discrete probability distributions. OpenCV provides the following distances for histogram comparisons:

1. **Correlation:**

$$d(H_1, H_2) = \frac{\sum_I (H_1(I) - \overline{H_1})(H_2(I) - \overline{H_2})}{\sqrt{\sum_I (H_1(I) - \overline{H_1})^2 \sum_I (H_2(I) - \overline{H_2})^2}} . \tag{3.6}$$

2. **Chi-Square:**

$$d(H_1, H_2) = \sum_I \frac{(H_1)(I) - H_2(I))^2}{H_1(I)} . \tag{3.7}$$

3. **Intersection:**

$$d(H_1, H_2) = \sum_I \min\left(H_1(I), H_2(I)\right) . \tag{3.8}$$

4. **Bhattacharyya distance:**

$$d(H_1, H_2) = \sqrt{1 - \frac{1}{\sqrt{\overline{H_1}\overline{H_2}N^2}} \sum_I \sqrt{H_1(I) \cdot H_2(I)}} . \tag{3.9}$$

These distances provides very different results when matching histograms, as seen from measured performance in Section 4.2.2.

3.3 SPS descriptors matching

The objective of the matching algorithm to match a target SPS signature extracted from a set of pictures with another one stored in the database. Since SPS signatures are multi-dimensional and each one can contain descriptors obtained from multiple viewpoints, different matching strategies are possible. We implemented two different ways of matching SPS descriptors. In both approaches, informations given to the matcher are: a SPS descriptor of a target to match and a database of SPS descriptors.

3.3.1 Matching strategies

As mentioned before we implemented two different ways of matching SPS descriptors: Full Skeletons Matching and Skeleton Parts Matching. In both approaches, a target signature is compared with those extracted from the available database and matched with the nearest by the means of the specific descriptor distance. Let S be the comparison set, T be the target person SPS, D_i be the $i-$th model from the people database, n be the number of viewpoints, k the number of joints of a skeleton model (which in this case is 14) and θ the weight vector described in Section 3.1.

Skeleton Parts Matching

In Skeleton Parts Matching, all possible combinations of descriptors for a type of joint between the target signature T and database entry D_i in order to find the minimum distance between them. The i-th entry of the database is matched with T if the distance is minimum:

$$d(T, D_i) = \sum_{j=0}^{k} \theta_j \cdot \min_{\substack{0 \leq h \leq n \\ 0 \leq m \leq n}} \|SPS_{jh}(T) - SPS_{jm}(D_i)\| . \tag{3.10}$$

SPS_{jm} and SPS_{jh} are descriptors contained in the SPS relative to $j-$th joint and $m-$th or $h-$th viewpoint respectively. To compute a joint type distance, the algorithm extracts all the candidates from SPS_j for both target T and the database entry D_i. All possible $|T_n| \cdot |D_{in}|$ candidate couples are checked to find the least distance, where $|T_n|$ is the number of viewpoints in target model and $|D_{in}|$ is the number of viewpoints in database model. The final distance is the sum of the distances of obtained from each different joint. An example is illustrated in Figure 3.7.

Full Skeleton Matching

Full Skeleton Matching compares all the possible combinations of full poses descriptors (set of local descriptors obtained from the same viewpoint) between the target signature T and database entry D_i. This approach is more similar to the original

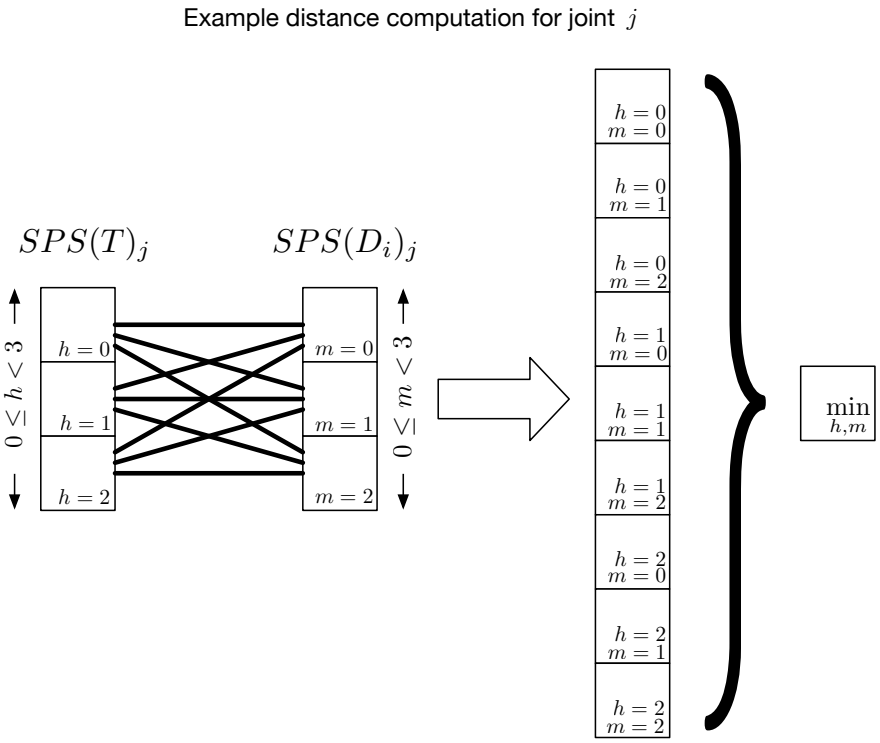


Figure 3.7: Skeleton Parts Matching strategy: example of a single joint distance computation between a target T and a database entry D_i. In this example the joint descriptor is available from 3 different viewpoints both on T and D_i. We compare the distance of each possible couple and output the minimum. In this example, each model has 3 different representation of the same joint.

work in [19] as it is in fact a comparison between full poses, even though SPS contains more than one in this work. The function to be minimized for matching target signature T with database entry D_i is:

$$d(T, D_i) = \min_{\substack{0 \leq h \leq n \\ 0 \leq m \leq n}} \sum_{j=0}^{k} \theta_j \cdot \| SPS_{jh}(T) - SPS_{jm}(D_i) \| . \tag{3.11}$$

Equation 3.11 shows that the difference with Skeleton Parts Matching is that the minimization is computed on the sum of all joints distances. Therefore h and m are the indexes of full poses from which all joints are compared. It is also possible to enforce h to be equal to m, so that only descriptors coming from the same camera are compared. This method is based on the assumption that at least one camera of the system should be able to extract a good pose with significant descriptors and

a limited number of occlusions, if any. Full Skeleton Matching is simpler and more computationally efficient than Skeleton Parts Matching, even though this advantage is negligible due to the low amount of comparisons needed.

Chapter 4

Experimental results

To build and measure the performances of proposed algorithms, we recorded two different dataset at the IAS-Lab: one two viewpoints dataset and one three-viewpoints dataset with 7 people. These datasets are targeted to both multi viewpoint articulated pose estimation and general re-identification. We performed all the testing on the three-viewpoint dataset as it is contain more articulated and significant poses to detect and more target people for re-identification. Furthermore, illumination and environmental conditions between the two datasets are identical.

4.1 Datasets recording

Datasets mentioned before were recorded with the purpose of building the algorithm upon them and testing. Both datasets are recorded inside IAS-Lab *Intelligence Autonomous System Laboratory* [1]. The system setup is based on Microsoft Kinect sensors, which are RGB-D sensors composed by a standard 640×480 RGB camera, a 640×480 30 Hz IR depth camera, a microphone and a motorized tilt. Cameras are connected to a computer in the two cameras dataset, and to two computers in the three cameras dataset. The system is based on the widely used Robot Operating System (ROS) [22] framework that allows to conveniently handle signals from the kinects and synchronization between cameras. Signals from the kinect are sent as packets through ROS topics (named buses over which ROS processes can exchange

[1] `http://robotics.dei.unipd.it`

Figure 4.1: Microsoft Kinect sensor illustration

messages[2]) and, in addition to data, they embed useful information like timestamps and intrinsic camera parameters. Timestamps are later used for extracting frames synchronized across different viewpoints from videos, while camera parameters are used to compose the K matrix described in Section 2.1.1. Extrinsic parameters of the system are computed using the calibration tool from the OpenPTrack package [20]. The tool uses a moving chessboard to estimate each camera position in real time in the world reference, and also computes positions relative to the ground plane, described as a translation vector T and a quaternion Q. Rotation matrix R is obtained from the quaternion as described in Section 2.1.1.

The system setup is illustrated in Figure 4.2 and it consists of two computers connected in a dedicated LAN using Gigabit Ethernet. Two computers are needed to overcome the limitations of USB bus on a single computer if connecting more than 2 Kinect sensors. Computer 1 is then set as master ROS node and Computer 2 connects to it in slave mode through Ethernet. Master node then runs OpenPTrack to handle all the data and perform the system calibration. When parameters are obtained, the dataset content itself is registered using rosbag, a set of tools for high performance recording from and playing back to ROS topics. Saved bag files contain all the ROS topic contents coming from the sensor, so both RGB and Depth data are saved, even though we only used RGB data in this work. Extracted frames are then synchronized using a script that matches couples or triplets of pictures obtained from different viewpoints by minimizing timestamp difference, thus obtaining a dataset of 2 synchronized frames per second.

[2]http://wiki.ros.org

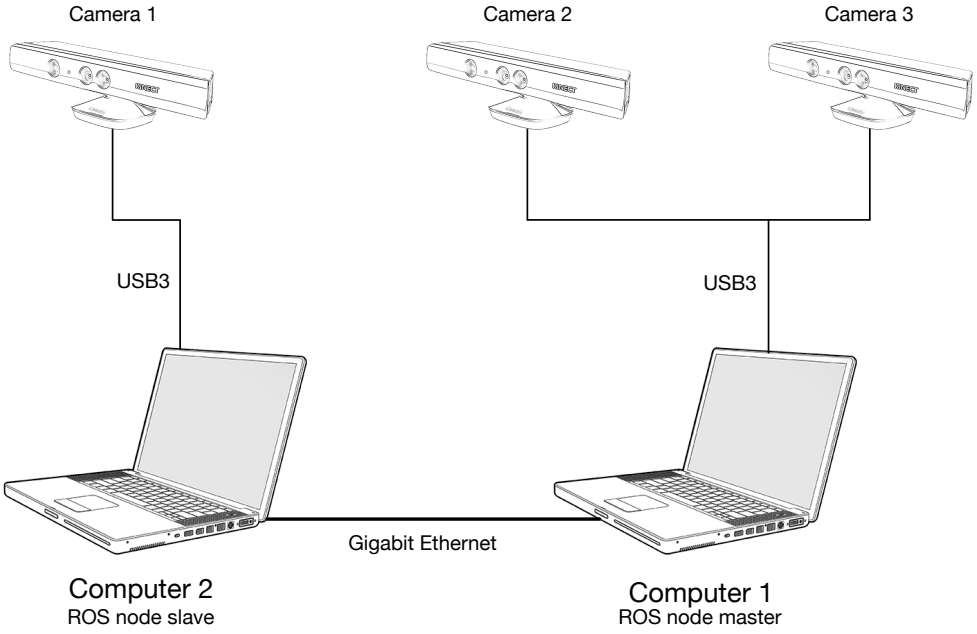


Figure 4.2: Diagram of hardware setup used to record and calibrate the three viewpoints dataset.

4.2 Tests on the Three-Viewpoints Re-identification dataset

In this section we summarize the results obtained with the proposed techniques on the recorded datasets. Since the multi view re-identification process involves many complex steps, a lot of partial results are measured, besides the re-identification itself. In particular we want to demonstrate the effectiveness of the proposed skeleton tracker by showing benefits obtained from the higher number of cameras and different triangulation and reprojection strategies that the multiple viewpoints allows. To achieve these measures, we manually marked ground truth joints on a subset of picture of the dataset, which shows the average effectiveness of the various skeleton tracking strategies. Performance is then evaluated by measuring the error between the pose extracted by the skeleton tracker and a manually marked pose. Let J_i be the $i-$th joint coordinates as extracted by the tracker and G_i be the same joint

manually marked in the ground truth, we defined the error e to be:

$$e = \sum_{j=0}^{k} \|J_i - G_i\| \ . \tag{4.1}$$

Where k is the number of joints, which is equal to 14 in the proposed skeleton model, and the obtained measure is in pixels. Therefore e is the sum of euclidean distances in pixel between the predicted pose and the ground truth pose. For the evaluation of the re-identification process, we compute the Cumulative Matching Characteristics (CMC) Curve, which is a widely used metric for measuring classification task performances and it is also used in [19]. CMC contains at the $i-$th position, the mean probability that a target is classified correctly within the first i responses given by the classifier. In this case, if a person is to be re-identified, CMC at position i holds the normalized number of time that its name appear at least within i responses from the re-identification algorithm. Typical useful parameters related to this metric are the Normalized Area Under Curve (nAUC), which is the integral function of the CMC, and the Rank-1 recognition rate.

$$\text{nAUC} = \sum_{i} \frac{\text{Rank}_i}{|\text{Samples}|} \, . \tag{4.2}$$

Rank-1 is trivially the first number in CMC and it corresponds to correct classifications, as the correct output is the first provided by the classifier.

4.2.1 Pose estimation performance

Improving classifier results by merging informations from multiple viewpoints is nontrivial. As a matter of fact, having more information also means having more potential erroneous data and noise. Enforcing coherence between poses detected by different camera systems necessarily means that if the output of the 2D pose detector on some camera is significantly wrong, that output can affect good results coming from another camera as well. This behavior has been observed in some rare cases by comparing the skeletal tracker outputs with ground truth. We also measured a parameter called Reprojection Error, which is the metric used by the MRE method

(see Section 2.3) to choose the best set of candidate poses for reprojection. In Figures 4.3 and 4.4 we show accuracy values collected on two image sequences in which MRE algorithm performs very differently. The data contained in both figures is computed as follows: pose detection is performed on every picture of the sequence using both PBD single camera skeleton detection and MRE on 3 cameras. Obtained poses on all pictures are then compared with ground truth and the total pose error is computed as the sum of euclidean distances between joints of the detected pose and the ground truth pose. Error for the MRE method is plotted in the graph with a blue line and error for single camera PBD detection is plotted in the graph with a red line on y-axis. The green line represents instead the reprojection error estimated by the MRE method. The case in Figure 4.3 represents an image sequence where the Reprojection Minimization algorithm is effective and is able to significantly reduce

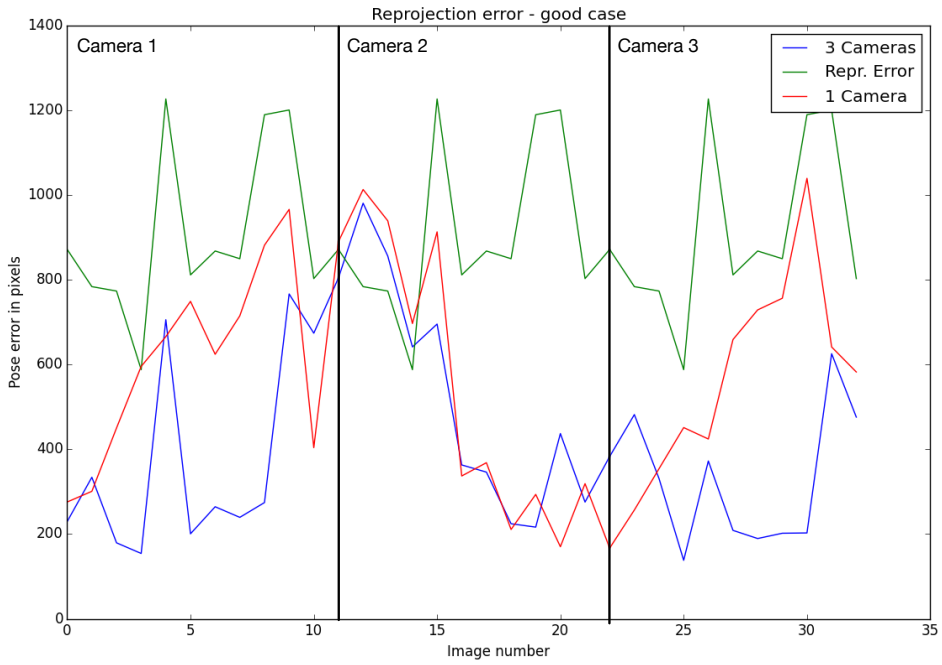


Figure 4.3: Total error per pose obtained with MRE method on a 10 images sequence per camera (30 images in total). This plot illustrates a case where MRE algorithm significantly improves the quality of the pose detection if compared to PBD. The plot is subdivided into three sequences: the first 11 pictures are obtained from camera 1, pictures n. 11-21 are obtained from camera 2 and pictures n. 21-31 are obtained from camera 3.

the mean error made by the classifier on each pose. In this case, PBD detections from the three independent cameras are quite inaccurate but the MRE algorithm is able to find better triplets of 2D poses by considering all detection candidates.

In this case, the output set of poses is significantly improved in almost all the pictures of the sequence and it can be seen by looking at the difference between the blue and the red line in Figure 4.3. On the other hand, the opposite can also happen, as shown in Figure 4.4, where detection from the cameras are generally good but the MRE algorithm is deceived by triplets that better fits its reprojection error metric and are chosen even if they are erroneous. This behavior is hardly predictable even though the reprojection accuracy can be estimated by observing reprojection error value. It can be seen experimentally that this error is related to the quality of results obtained from triangulation and reprojection and can be used in real time to evaluate whether or not the method is effective.

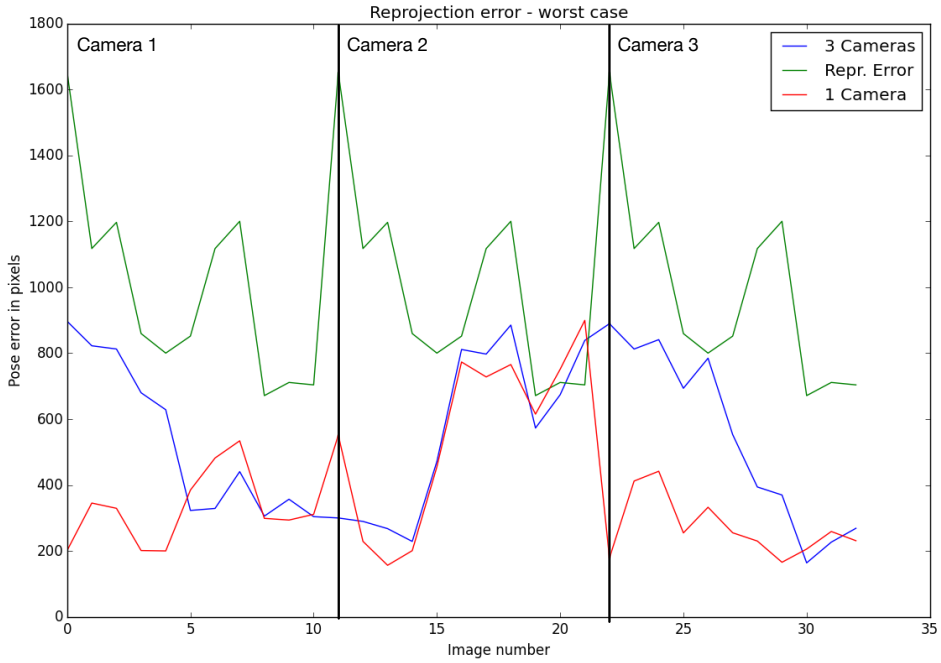


Figure 4.4: Total error per pose obtained with MRE method on a 10 images sequence per camera. This plot illustrate a case where MRE algorithm reduces the quality of the pose detection if compared to PBD. The plot is subdivided into three sequences: the first 11 pictures are obtained from camera 1, pictures n. 11-21 are obtained from camera 2 and pictures n. 21-31 are obtained from camera 3.

Algorithms average case performance are computed by comparing results obtained from the implemented methods Minimization of Reprojection Error and Fusion of Parts Clusters Centroids with PBD detections on single cameras. Fusion of Best Detections method performance are not reported in this section as its effectiveness is exactly the same as PBD, even if final results can be slightly different due to minor changes introduced by triangulation and reprojection. Results were computed by comparing detections of the pose estimators with ground truth and measuring the sum of errors for each body joint on a 100 pictures sequence. The obtained vector is then normalized to the number of joint elements added to the sum, thus computing mean error per joint whose values is shown in Figure 4.5 for each of the skeletal trackers. PBD, Parts Clusters Centroids and MRE detections are illustrated by red, green and blue bars respectively. Collected data shows that MRE is outperforming Fusion of Parts Clusters Centroids detections method and improving the quality of the PBD detections by a good margin. Results shown

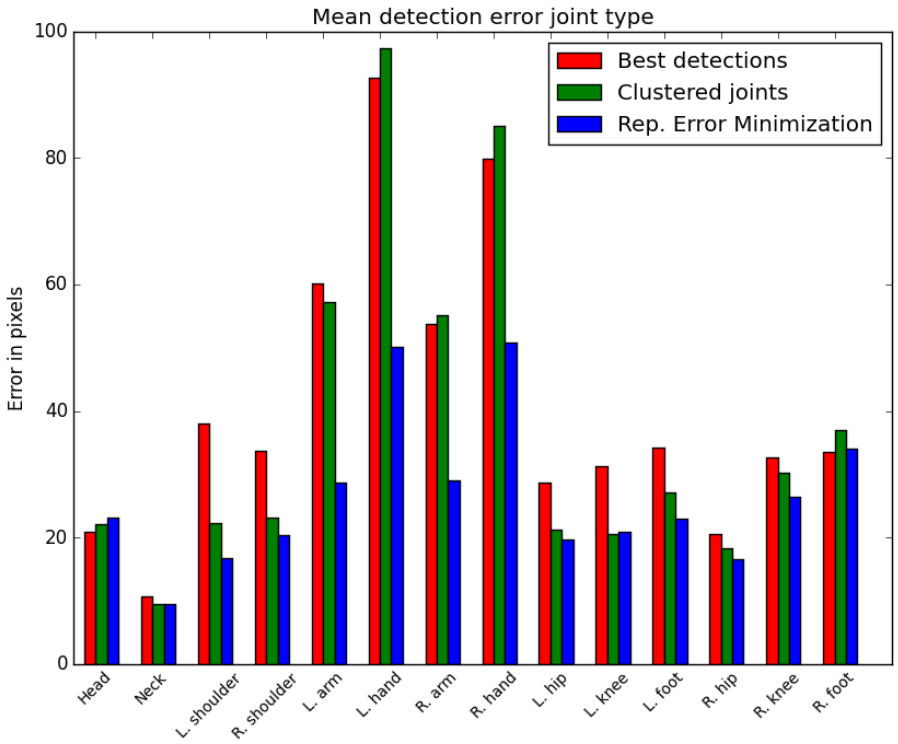


Figure 4.5: Mean detection error in pixels aggregated by different joint types. The graph shows results obtained with MRE and Parts Clusters Centroids skeletal trackers, compared with best detections on single cameras.

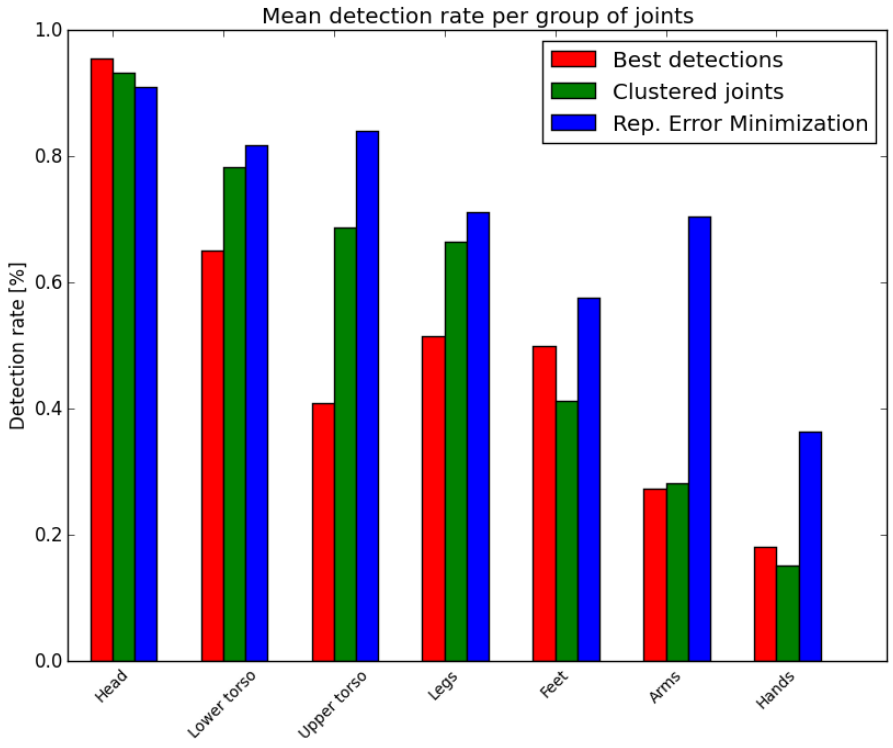
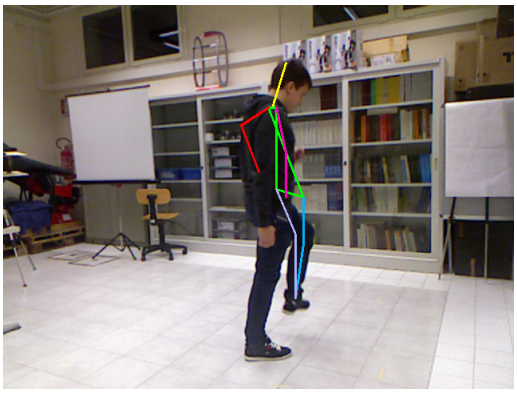


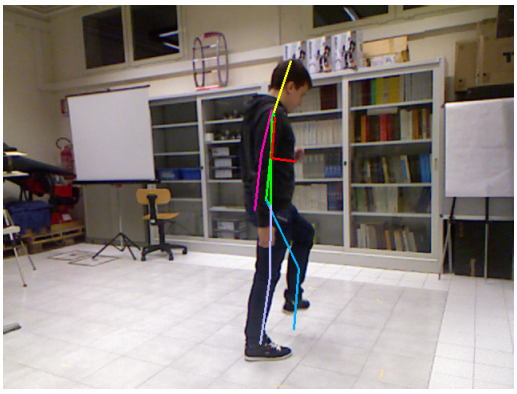
Figure 4.6: Mean detection rate per joint type group. A detection is considered correct if its keypoint is placed within 30 pixels of the ground truth corresponding point (euclidean distance).

in Figure 4.5 also clearly show that some joint types are more stable than others, having significantly lower mean error. In particular, most stable joints are: Head, L. and R. shoulder, L. and R. hips. We then aggregated joints of the same body area (e.g. L. and R. hands, L. and R. feet) and performed a more meaningful test to show implications that these inaccuracies can have on re-identification. We estimated that if a keypoint is detected farther than 30 pixels from its correct position, it can be hardly considered useful for re-identification purposes, as the extracted descriptor will not correspond to the desired joint. The SPS matching algorithm has to make up for these imprecisions, as the wrong descriptor will ideally not match any model of the database. Therefore to improve the SPS matching results, we measure detection rate for each of the joint groups and used these results (shown in Figure 4.6) to configure θ weights vector. By lowering the mean error per joints, results shows that MRE is also greatly improving the detection rate of each joints if compared to

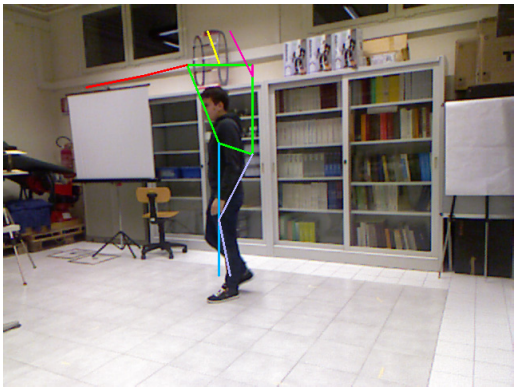
PBD, thus directly enhancing the consequent re-identification phase performance. Figure 4.7 illustrates some example cases where MRE algorithm greatly improves the performance of detections. PBD has obvious difficulties with cases of self occluded body parts, as in (a) and (c), where it can completely miss detection as confidence score attributed to candidates is unreliable in these cases, since the choice of final pose is mostly a guess. Among its candidates, however, very high quality detections can be found and extracted leveraging information from other viewpoints, as MRE manages to do in (b) and (d). Even in cases of more frontal views as in (e), (g) and (i), PBD detection can be often inaccurate and its best detection can be worse than some of its candidates: arms and legs are often positioned incorrectly and body orientation is inconsistent across different viewpoints. MRE manages to solve these problems in many cases as seen in (f), (h), (j).



(a)

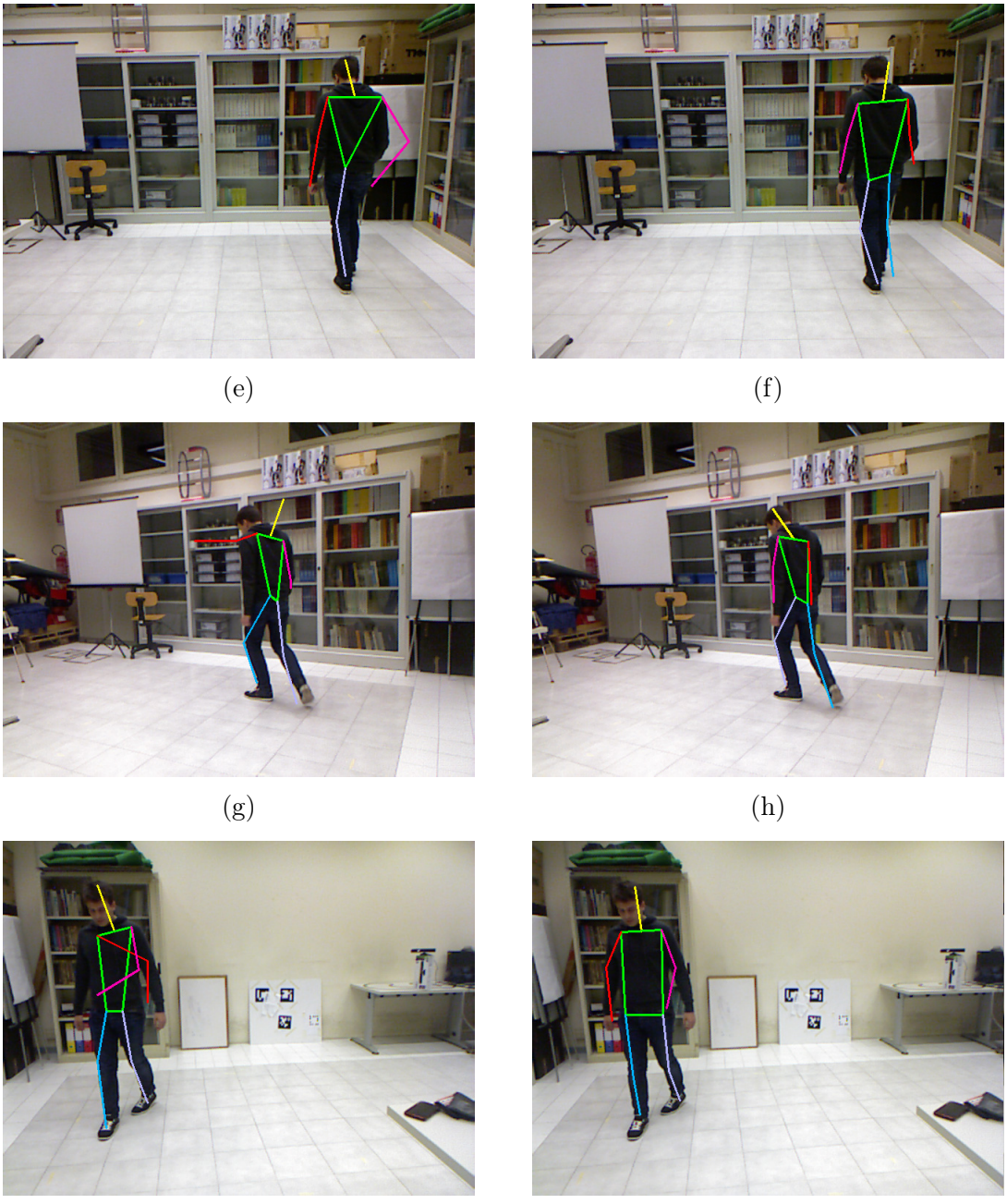


(b)



(c)

(d)



(e) (f)

(g) (h)

(i) (j)

Figure 4.7: Comparison of skeletons where MRE algorithm effectively improves detection results. Pictures on the left are obtained from a single camera using a skeleton derived directly from PBD detection, skeletons on the right are obtained using MRE algorithm from three cameras. PBD detections suffers badly from self occlusions as seen in (a) and (c) and are then improved by MRE which leverage information from multiple viewpoints in (b) and (d). (e), (g) and (i) show cases of typical detection inaccuracies from PBD which is often misplacing limbs even in frontal poses. In many cases these detections can be improved by MRE as seen in (f),(h) and (j).

4.2.2 Re-identification performance

To test the effectiveness of the re-identification algorithm, we tested the procedure on the whole dataset which has a total of 468 pictures. We compared results obtained with different features extractors: SIFT, SURF and Histograms with Correlation, Intersection and Bhattacharyya distances. Initial tests were performed with SIFT and SURF feature extractors, as they are known to provide best performance in some state-of-the-art works including [19]. Results on the proposed dataset were however much worse than expected and re-identification failed most of the time. Further analysis on this problem brought to the conclusion that the low resolution of the images combined with a typically small keypoint size, which has to fit inside the body shape, could be the cause of it. An example is illustrated in Figure 4.8 where the content of a typical keypoint is magnified. Besides the color, information contained in it is very few as the sensor is unable to capture clothing texture and folds. Figure 4.9 shows a derivative edge detection filter applied on keypoints, which is an approximation of the type of filters applied by SIFT and SURF algorithms.

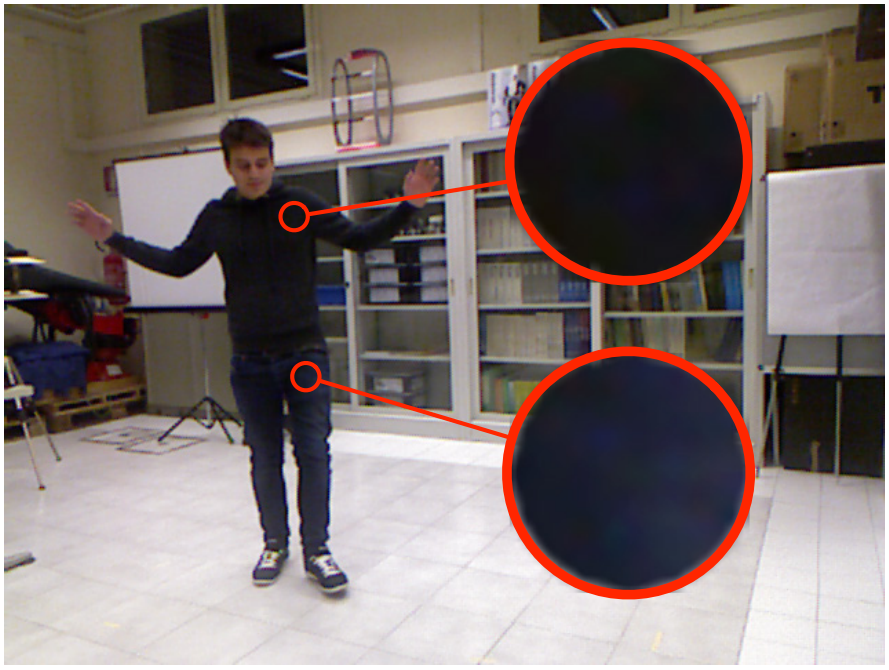


Figure 4.8: Information contained within a typical keypoint on which re-identification is based. Magnified keypoint shows that low resolution images fails to provide significant texture information for features extraction.

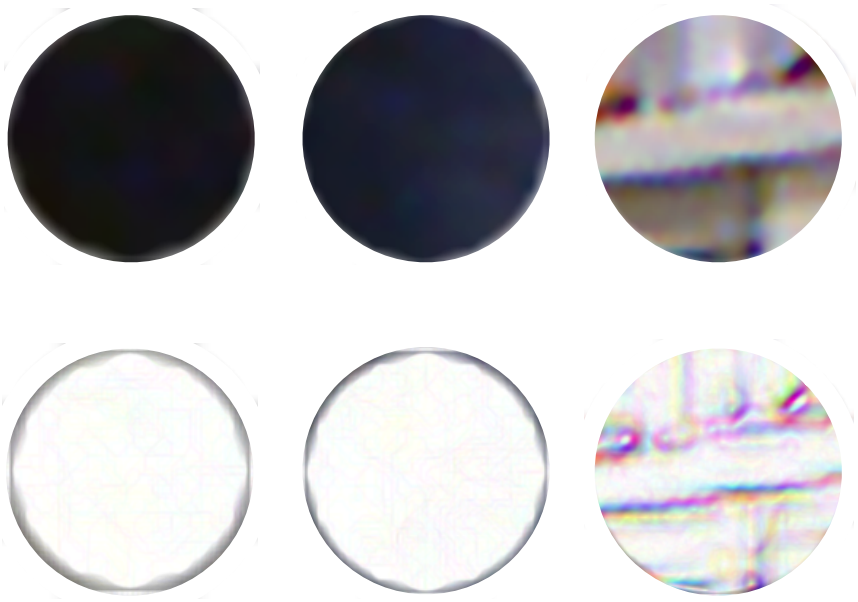


Figure 4.9: Sample keypoints extracted from body (left and center) are shown together with an typical SIFT keypoint of similar size automatically detected from the background (right). On the lower row, a derivative filter is applied to highlight the amount of information the feature extractor is able to gather from keypoints. Keypoints on the left and center are the same as in Figure 4.8.

It can be seen that a typical SIFT automatically detected keypoint of the same size, which was detected from a shelf in the background and is shown on the right circle, contains many points with a much higher color gradient amplitude. To overcome this, considering that the dataset does have a very stable illumination across different viewpoints, a custom histograms based features were tested. Tests were carried out on different color spaces: HSV, rg-chromaticity and RGB. Nevertheless, as histograms performance on RGB color space showed better performance on this datasets, final results are measured on RGB space only.

The descriptors matching phase was tested with both presented strategies: Skeleton Parts Matching and Full skeleton matching (see Section 3.3.1). Descriptors weights vector θ is set according to detection rate of the Minimization of Reprojection Error method measured with tests on pose estimation, as illustrated in Table 4.1. MRE is used as skeletal tracker for both Skeleton Parts Matching and Full Skeleton Matching strategies, thus detected keypoints locations are exactly the same between the two tests. To compute results we compared the classifier scores

Body joint	**Normalized MRE detection rate**
Head (Head and neck)	0.9
Upper torso (L. and R. shoulders)	0.93
Arms (L. and R. arms)	0.69
Hands	0.36
Lower torso	0.81
Legs	0.71
Feet	0.57

Table 4.1: List of detection rates of MRE algorithm on joints groups. These results are equivalent to those in Figure 4.6 for MRE and are used as θ weights by the matching algorithm on the performed tests.

for each person in the database with ground truth, thus obtaining CMC curves in Figures 4.10 4.11. Results for Skeleton Parts Matching, illustrated in Figure 4.10 shows that Histograms performs better then SIFT and SURF in this case. Histogram comparison with Correlation distance achieves an excellent 83.6% Rank-1 rate, while others Histograms distances yield significantly lower detection rates yet still acceptable for the task. Results obtained with Full Skeleton Matching (Figure 4.11) are very similar to those obtained with Skeleton Parts Matching. The best performing descriptors are Histograms which clearly outperform SIFT and SURF in this case as well. Again, the Correlation distance for Histogram comparisons is giving the best results with 83.9% Rank-1 detection rate, which is slightly better than the obtained result with Skeleton Parts Matching. For easier comparison of different methods, we computed the nAUC value for each of the CMC curves on both tests. Results are summarized in Table 4.2. Final results shows that Full Skeleton Matching strategy obtains slightly better performance than Skeleton Parts Matching, by having a slightly greater nAUC and Rank-1 values for almost each descriptor types. Overall results are exceptionally positive since they take into account possible skeleton tracking inaccuracies but are still comparable with state-of-the-art reference work for local descriptors-based re-identification [19].

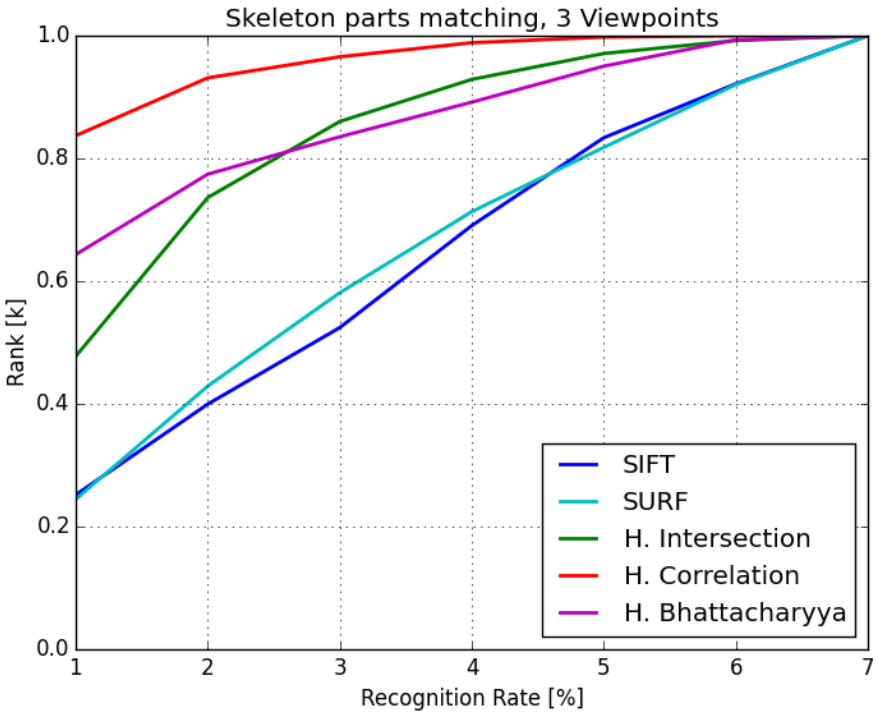


Figure 4.10: CMC curve computed by performing re-identification on the whole dataset using Skeleton Parts Matching method.

Re-identification method	**Rank-1 [%]**	**nAUC**
SPM-HIST-CORRELATION	83.7	0.888
SPM-HIST-INTERSECTION	47.7	0.780
SPM-HIST-BHATTACHARYYA	64.3	0.798
SPM-SIFT	25.1	0.588
SPM-SURF	24.3	0.600
FSM-HIST-CORRELATION	83.9	0.890
FSM-HIST-INTERSECTION	60.0	0.800
FSM-HIST-BHATTACHARYYA	64.7	0.816
FSM-SIFT	25.2	0.589
FSM-SURF	24.4	0.601

Table 4.2: Re-identification algorithms performance summary table. FSM suffix means Full Skeleton Matching and SPM means Skeleton Parts Matching, which are the two matching strategies proposed and tested.

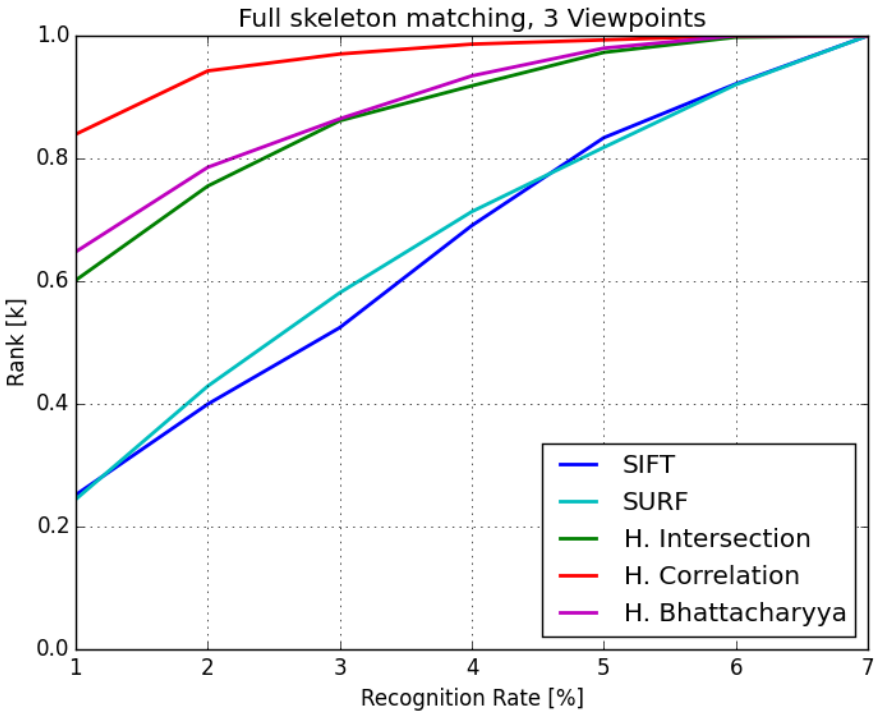


Figure 4.11: CMC curve computed by performing re-identification on the whole dataset using Full Skeleton Matching parts matching method.

Chapter 5

Conclusions

In this thesis we presented a novel approach for performing re-identification using local feature descriptors from joint locations of the human body without utilizing RGB-D sensors and related skeletal trackers. To obtain local keypoints coordinates we implemented a human pose detector built on top of a Parts Based Detector detector and augmented its output to work with multiple input cameras. For reidentification purposes, a Person Based Signature is extracted from each detection by gathering features descriptors from multiple viewpoints and structuring in a single signature called Skeleton-based Person Signature. Re-identification results are achieved by comparing training hand selected models signatures with target signatures acquired automatically from the dataset. To obtain data useful for building and testing the algorithm, we recorded two multi camera datasets using a system based on the ROS framework, composed of three Microsoft's Kinect sensors and two computers.

We tested the approach on the Three Viewpoints Dataset we recorded and proved that we can improve pose detection performances leveraging informations from multiple viewpoints and obtain generally better skeletons in 2D images. We then evaluated the quality of the re-identification approach when based on such skeletons. Re-identification tests were performed by comparing many different 2D features extractors, upon which Histogram descriptors compared with Correlation distance has shown to give better results. These results can possibly have been affected by the low resolution of used Kinect sensors, but provide a good example

of a typical video surveillance use case environment. Although the results presented here have demonstrated the effectiveness of the multi viewpoint approach, it could be further developed in a number of ways:

- Reworking of the skeleton tracker: the detector speed could be improved leveraging multithreading architectures and GPUs, thus obtaining near real time performances. Detection could also be mixed with tracking in a real time use, adding time consistency constraint between subsequent detections, which could dramatically improve accuracy.

- Enhancing features in 3D space: theoretically, if the skeleton tracker could provide higher accuracy detections, features could be extracted from keypoints in a smarter way. For example it would be possible to estimate occluded parts and remove them from signature matching. It would be also possible to determine persons orientation in 3D space and therefore to obtain a real space and orientation aware descriptor on a 3D model.

Ringraziamenti

Voglio ringraziare i miei genitori che hanno sempre avuto fiducia in me e mi hanno dato la possibilità e il sostegno per affrontare e portare a termine questo percorso.

Ringrazio il mio relatore, prof. Emanuele Menegatti, il mio correlatore Stefano Ghidoni e tutto lo IAS-Lab per la disponibilità e il supporto fornitomi in innumerevoli occasioni. Grazie a loro ho imparato moltissimo e ho maturato la mia passione per la robotica e la computer vision.

Ringrazio Giulia che ha avuto la pazienza di sopportarmi nonostante le mie nevrosi e le troppe serate passate davanti al computer.

Un grazie speciale a tutti i miei amici senza i quali non sarei la persona che sono oggi, e ai miei compagni universitari che hanno condiviso con me questa avventura rendendola più leggera e allo stesso tempo entusiasmante.

Nomenclature

CMC Cumulative Matching Characteristic Curve

DAG Directed Acyclic Graph

FSM Full Skeleton Matching

HSV Hue Saturation Value (color model)

IR Infrared

MRE Minimization of Reprojection Error

PBD Parts Based Detector

PCL Point Cloud Library

RGB Red Green Blue (color model)

RGB-D Red Green Blue + Depth

SIFT Scale-Invariant Feature Transform

SPM Skeleton Parts Matching

SPS Skeleton-based Person Signature

SURF Speeded Up Robust Features

Bibliography

[1] Sikandar Amin, Mykhaylo Andriluka, Marcus Rohrbach, and Bernt Schiele. Multi-view pictorial structures for 3D human pose estimation. In Tilo Burghardt, Dima Damen, Walterio W. Mayol-Cuevas, and Majid Mirmehdi, editors, *BMVC*. BMVA Press, 2013.

[2] M. Andriluka, S. Roth, and B. Schiele. Pictorial structures revisited: People detection and articulated pose estimation. In *CVPR*, pages 1014–1021, 2009.

[3] Mykhaylo Andriluka, Leonid Pishchulin, Peter V. Gehler, and Bernt Schiele. 2D human pose estimation: New benchmark and state of the art analysis. In *CVPR*, pages 3686–3693. IEEE, 2014.

[4] Martin Bäuml and Rainer Stiefelhagen. Evaluation of local features for person re-identification in image sequences. In *AVSS*, pages 291–296. IEEE Computer Society, 2011.

[5] Herbert Bay, Andreas Ess, Tinne Tuytelaars, and Luc J. Van Gool. Speeded-up robust features (SURF). *Computer Vision and Image Understanding*, 110(3):346–359, 2008.

[6] Vasileios Belagiannis, Sikandar Amin, Mykhaylo Andriluka, Bernt Schiele, Nassir Navab, and Slobodan Ilic. 3D pictorial structures for multiple human pose estimation. In *CVPR*, pages 1669–1676. IEEE, 2014.

[7] Gary Bradski. The OpenCV library. *Dr. Dobb's Journal of Software Tools*, 25(11):120, 122–125, November 2000.

[8] Gary R. Bradski and Adrian Kaehler. *Learning OpenCV - computer vision with the OpenCV library: software that sees.* O'Reilly, 2008.

[9] Dong Seon Cheng, Marco Cristani, Michele Stoppa, Loris Bazzani, and Vittorio Murino. Custom pictorial structures for re-identification. In Jesse Hoey, Stephen J. McKenna, and Emanuele Trucco, editors, *BMVC*, pages 1–11. BMVA Press, 2011.

[10] Michela Farenzena, Loris Bazzani, Alessandro Perina, Vittorio Murino, and Marco Cristani. Person re-identification by symmetry-driven accumulation of local features. In *CVPR*, pages 2360–2367. IEEE Computer Society, 2010.

[11] P. F. Felzenszwalb and D. P. Huttenlocher. Pictorial structures for object recognition. *International Journal of Computer Vision*, 61(1):55–79, January 2005.

[12] M. A. Fischler and R. A. Elschlager. The representation and matching of pictorial structures. *IEEE Trans. Computer*, 22(1):67–92, January 1973.

[13] D. A. Forsyth and J. Ponce. *Computer Vision: A Modern Approach*. Prentice Hall, 2003.

[14] R. I. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, June 2004.

[15] D. G. Lowe. Object recognition from local scale-invariant features. In *ICCV*, pages 1150–1157, 1999.

[16] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110, 2004.

[17] C. Madden, E. D. Cheng, and M. Piccardi. Tracking people across disjoint camera views by an illumination-tolerant appearance representation. *Machine Vision and Applications*, 18(3-4):233–247, August 2007.

[18] Birgitta Martinkauppi, Abdenour Hadid, and Matti Pietikäinen. Skin color in face analysis. In Stan Z. Li and Anil K. Jain, editors, *Handbook of Face Recognition*, pages 223–249. Springer, 2011.

[19] Matteo Munaro, Stefano Ghidoni, Deniz Tartaro Dizmen, and Emanuele Menegatti. A feature-based approach to people re-identification using skeleton keypoints. In *ICRA*, pages 5644–5651. IEEE, 2014.

[20] Matteo Munaro, Alex Horn, Randy Illum, Jeff Burke, and Radu B. Rusu. OpenPTrack: People Tracking for Heterogeneous Networks of Color-Depth Cameras. In *In IAS-13 Workshop Proceedings: 1st Intl. Workshop on 3D Robot Perception with Point Cloud Library, Padova, Italy*, pages 235–247, July 2014.

[21] C. Nakajima, M. Pontil, B. Heisele, and T. Poggio. Full-body person recognition system. *Pattern Recognition*, 36(9):1997–2006, September 2003.

[22] Conley Ken. Gerkey Brian P.. Faust Josh. Foote Tully. Leibs Jeremy. Wheeler Rob. Quigley, Morgan. and Andrew Y. Ng. ROS: an open-source robot operating system.

[23] Radu Bogdan Rusu and Steve Cousins. 3D is here: Point Cloud Library (PCL). In *IEEE International Conference on Robotics and Automation (ICRA)*, Shanghai, China, May 9-13 2011.

[24] Chris Sweeney. *Theia Multiview Geometry Library: Tutorial & Reference*. University of California Santa Barbara.

[25] wg perception. Parts based detector. `https://github.com/wg-perception/PartsBasedDetector`, 2013.

[26] D. Ramanan Y. Yang. Articulated pose estimation with flexible mixtures of parts. `http://www.ics.uci.edu/~dramanan/software/pose/`, 2011.

[27] Yi Yang and Deva Ramanan. Articulated human detection with flexible mixtures of parts. *IEEE Trans. Pattern Anal. Mach. Intell*, 35(12):2878–2890, 2013.