

Research Article

Human Posture Tracking and Classification through Stereo Vision and 3D Model Matching

Stefano Pellegrini and Luca Iocchi

Dipartimento di Informatica e Sistemistica, Università degli Studi di Roma "Sapienza," 00185 Roma, Italy

Correspondence should be addressed to Stefano Pellegrini, pellegrini@dis.uniroma1.it

Received 15 February 2007; Revised 19 July 2007; Accepted 22 November 2007

Recommended by Ioannis Pitas

The ability of detecting human postures is particularly important in several fields like ambient intelligence, surveillance, elderly care, and human-machine interaction. This problem has been studied in recent years in the computer vision community, but the proposed solutions still suffer from some limitations due to the difficulty of dealing with complex scenes (e.g., occlusions, different view points, etc.). In this article, we present a system for posture tracking and classification based on a stereo vision sensor. The system provides both a robust way to segment and track people in the scene and 3D information about tracked people. The proposed method is based on matching 3D data with a 3D human body model. Relevant points in the model are then tracked over time with temporal filters and a classification method based on hidden Markov models is used to recognize principal postures. Experimental results show the effectiveness of the system in determining human postures with different orientations of the people with respect to the stereo sensor, in presence of partial occlusions and under different environmental conditions.

Copyright © 2008 S. Pellegrini and L. Iocchi. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

1. INTRODUCTION

Human posture recognition is an important task for many applications in different fields, such as surveillance, ambient intelligence, elderly care, and human-machine interaction. Computer vision techniques for human posture recognition have been developed in the last years by using different techniques aiming at recognizing human activities (see, e.g., [1, 2]). The main problems in developing such systems arise from the difficulties of dealing with the many situations that occur when analyzing general scenes in real environments. Consequently, all the works presented in this area have limitations with respect to a general applicability of the systems.

In this article, we present an approach to human posture tracking and classification that aims at overcoming some of these limitations, thus enlarging the applicability of this technology. The contribution of this article is a method for posture tracking and classification given a set of data in the form XYZ-RGB, corresponding to the output of a stereo-vision-based people tracker. The presented method uses a 3D model of human body, performs model matching through a variant of the ICP algorithm, tracks the model

parameters over time, and then uses a hidden Markov model (HMM) to model posture transitions. The resulting system is able to reliably track human postures, overcome some of the difficulties in posture recognition, and present high robustness to partial occlusions and to different points of view. Moreover, the system does not require any off-line training phase. Indeed it just uses the first frames (about 10) in which the person is tracked to automatically learn parameters that are then used for model matching. During these training frames, we only require the person to be in the standing position (with any orientation) and that his/her head is not occluded.

The approach to human posture tracking and classification presented here is based on stereo vision segmentation. Real-time people tracking through stereo vision (e.g., [3–5]) has been successfully used for segmenting scenes in which several people move in the environment. This kind of tracker is able to provide not only information about the appearance of a person (e.g., colors) but also 3D information of each pixel belonging to the person.

In practice, a stereo-vision-based people tracker provides, for each frame, a set of data in the form XYZ-RGB containing a 2.5D model and color information of the person

being tracked. Moreover, correspondences of these data over time are also available. Therefore, when multiple people are in a scene, we have a set of XYZ-RGB data for each person. Obviously, this kind of segmentation can be affected by errors, but the experience we report in this article is that this phase is good enough to allow for implementing an effective posture classification technique. Moreover, the use of stereo-based tracking guarantees a high degree of robustness also to illumination changes, shadows, and reflections, thus making the system applicable in a wider range of situations.

The evaluation of the method has been performed on the actual output of a stereo-vision-based people tracker, thus validating in practice the chosen approach. Results show the feasibility of the approach and its robustness to partial occlusions and different view points.

The article is organized as follows. Section 2 describes some related work. Section 3 presents a brief overview of the system and describes the people tracking module upon which the posture recognition module is based. Section 4 presents a discussion about the choice of the model that has been used for representing human postures. Section 5 describes the training phase, while Section 6 introduces the algorithm used for posture classification. Then, Sections 7, 8, and 9 illustrate in detail the steps of the algorithm. Finally, Section 10 includes an experimental evaluation of the method. Conclusions and future work conclude the article.

2. RELATED WORK

The majority of the works that deal with human body perception through computer vision can be divided into two groups: those that try to achieve tracking of the pose (a set of quantitative parameters that define precisely the configuration of the articulated body) through time and those that aim at recognizing the posture (a qualitative assessment that represents a predefined configuration) at each frame.

The first category is usually more challenging since it requires a precise estimation of the parameters that define the configuration of the body. Given the inherent complexity of the articulated structure of the human body and the consequent multimodality of the observation likelihood, one might think that propagating over time the probability distribution on the state should be preferred with respect to a deterministic representation of the state. The introduction of the condensation algorithm [6] shows how this approach can lead to desirable results, however revealing at the same time that the computational resources needed for the task are unacceptable for the majority of the applications. In the following years, there have been many attempts to reduce the problem of the time elapsed, for instance by reducing the number of particles and including a local search [7] or a simulated annealing [8] in the algorithm. Even if the results remain very precise and the time elapsed decreases with these new approaches, the goal of an application that can be used in real-time scenarios is still far from being achieved due to the still inadmissible time request. Propagating a probability distribution over time yields a robust approach, because it deals effectively with the drift of the tracking error over time.

Another class of approaches address the accumulation of the error over time and the ability to recover from error by recognizing the components of the articulated body in the single image. These approaches [9–11] are characterized by the recovery in the images of potential primitives of the body (such as a leg, a head, or a torso) through template search, exploiting edge and/or appearance information, and then the search for the most likely configuration given the primitives found. While this approach easily allows for coping with occlusions, given its bottom-up nature, it still remains limited in the 2D information that it exploits and that it outputs. Other approaches try to overcome this limitation, proposing to use a well-defined 3D model of the object of interest, and then trying to match these models with the range image, either using the ICP algorithm [12] or a modified version of the gradient search [13]. These approaches are computationally convenient with respect to many others, especially the former that achieves the goal of producing real-time results, even if one can suspect that it has problems in dealing with occlusions.

The approaches in the second category, rather than recovering the pose, attempt to classify the posture assumed by the examined person in every single frame, picking up one among a predefined set of postures. Usually this means that some low-level features of the body segment of the image, such as projection histograms [14–16] or contour-based shape descriptors [16], are computed in order to achieve this classification. Otherwise, a template is obtained to represent a single class of postures and then the image is compared with the whole set of templates to find the best match, for example using Chamfer matching [17]. The main difficulty with this kind of solutions is that the sets of different defined postures are not usually disambiguated by a particular set of low-level features. Also, the templates that are used as prototypes for the different classes of postures do not contain enough information to distinguish correctly all the different postures.

Our approach tries to combine aspects of the two categories. In fact, we propose a method for posture recognition that does not discard some of the crucial information about the body configuration that we decided to track over time. With respect to methods in the first group, our approach is less time consuming, allowing us to use it in applications such as video surveillance. Indeed, though the output given by our system is not as rich as the one showed in other works [7, 8], we show that there is no need of further analysis of the image when the objective is to classify a few postures. With respect to methods in the second group, our approach is more robust, not relying on low-level features that are usually not distinctive of one single class of postures when the subject is analyzed from different points of view. In fact, we show that the amount of information we used is the right tradeoff between robustness and efficiency of the application.

3. OVERVIEW OF THE SYSTEM

The system described in this article is schematically represented in Figure 1. Two basic modules are present in this schema: PLT (people localization and tracking), which is responsible for analyzing stereo images and for segmenting

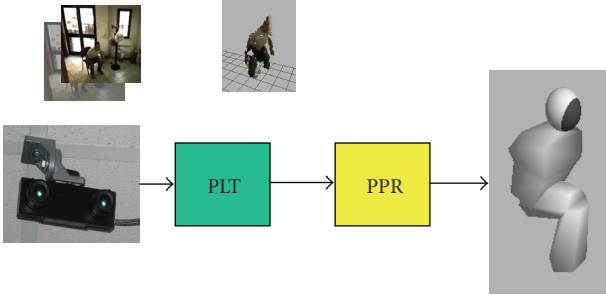


FIGURE 1: Overview of the system.

the scene by extracting 3D and color information, and PPR (person posture recognition), which is responsible for recognizing and tracking human postures.

In the rest of this section, we briefly describe these modules. Since the focus of this article is on the posture recognition module, the detailed description of its design and implementation is delayed to the next sections.

3.1. People localization and tracking

The stereo-vision-based people localization and tracking (PLT) [4, 5] is composed of three processing modules: (1) segmentation based on background subtraction, that is used to detect foreground people to be tracked; (2) plan-view analysis, that is used to refine foreground segmentation and to compute observations for tracking; (3) tracking, that tracks observations over time maintaining association between tracks and tracked people (or objects). An example of the PLT process is represented in Figure 2.

Background subtraction is performed by considering intensity and disparity components. A pixel is assigned to foreground if there is enough difference between the intensity and disparity of the pixel in the current frame and the related components in the background model. More specifically, with this background subtraction a foreground pixel must have both intensity difference and disparity difference. This allows for correctly dealing with *shadows* and *reflections* that usually produce only intensity differences, but not disparity differences. Observe also that the presence of the disparity model allows for reducing the thresholds, so that it would be possible to detect also minimal differences in intensity and thus being able to detect foreground objects that have similar colors of the background, without increasing false detection rate due to illumination changes.

Foreground analysis is used to refine the set of foreground points obtained through background subtraction. The set of foreground pixels is processed by (1) *connected components analysis*, that determines a set of *blobs* on the basis of 8-neighborhood connection; (2) *blob filtering*, that removes small blobs (due to, e.g., noise or high-frequency background motion). These processes remove typical noises occurring in background subtraction and allow for computing more accurate sets of foreground pixels for representing foreground objects. Therefore, it is adequate to be used in the subsequent background update step.

The second part of the processing is plan-view analysis. In this phase, each pixel belonging to a blob extracted in the previous step is projected in the plan-view. This is possible since stereo camera is calibrated and thus we can determine 3D location of pixels with respect to a reference system in the environment. After projection, we perform a plan-view segmentation. More specifically, for each image blob, connected components analysis is used to determine a set of blobs in the plan-view space. This further segmentation allows for determining and solving several cases of *undersegmentation*. They occur, for example, when two people are close in the image space (or partially occluded), but far in the environment. Plan-view blobs are then associated to image blobs and a set of n pairs (image blob, plan-view blob) are returned as observations for the n moving objects (people) in the scene.

Finally, tracking is performed to filter such observations over time. Our tracking method integrates information about person location and color models using a set of Kalman filters (one for each person being tracked) [4]. Data association between tracks and observations is obtained as a solution of an optimization problem (i.e., minimizing the overall distance of all the observations with respect to the current tracks) based on a distance between tracks and observations. This distance is computed by considering Euclidean distance for locations and a model matching distance for the color models, thus actually integrating the two components in data association.

Tracks in the system are also associated to finite-state automata that control their evolution. Observations without an associated track generate CANDIDATE tracks and tracks without observations are considered LOST. CANDIDATE tracks are promoted to TRACKED tracks only after a few frames. In this way we are able to discard temporary false detections. While LOST tracks remain in the system for a few frames in order to deal with temporary missing detection of people.

The output of the entire process is thus a set of tracks for each tracked person, where each track contains information about the location of the person over time, as well as XYZ-RGB data (i.e., color and 3D position) for all the pixels that the system has recognized as belonging to the person. Since external calibration of the stereo sensor is available, the reference system for 3D data XYZ is chosen with the XY plane corresponding to the ground floor and the Z axis being the height from the ground. Therefore, for each tracked person, the PLT system provides a set of data $\Omega^{\mathcal{P}} = \{\omega_t^{\mathcal{P}}, \dots, \omega_{t_0}^{\mathcal{P}}\}$ from the time t_0 in which the person is first detected to current time t . The value $\omega_t^{\mathcal{P}} = \{(X_i^t, Y_i^t, Z_i^t, R_i^t, G_i^t, B_i^t) \mid i \in \mathcal{P}\}$ is the set of XYZ-RGB data for all the pixels i identified as belonging to the person \mathcal{P} .

The PLT system produces two kinds of errors in these data: (1) *false positives*, that is, some of the pixels in \mathcal{F} do not belong to the person; (2) *false negatives*, that is, some pixels belonging to the person are not present in \mathcal{F} . Figure 3 shows two examples of nonperfect segmentation, where only the foreground pixels for which it is possible to compute 3D information are displayed. By analyzing the data produced by the tracking system we estimate that the rate of *false*

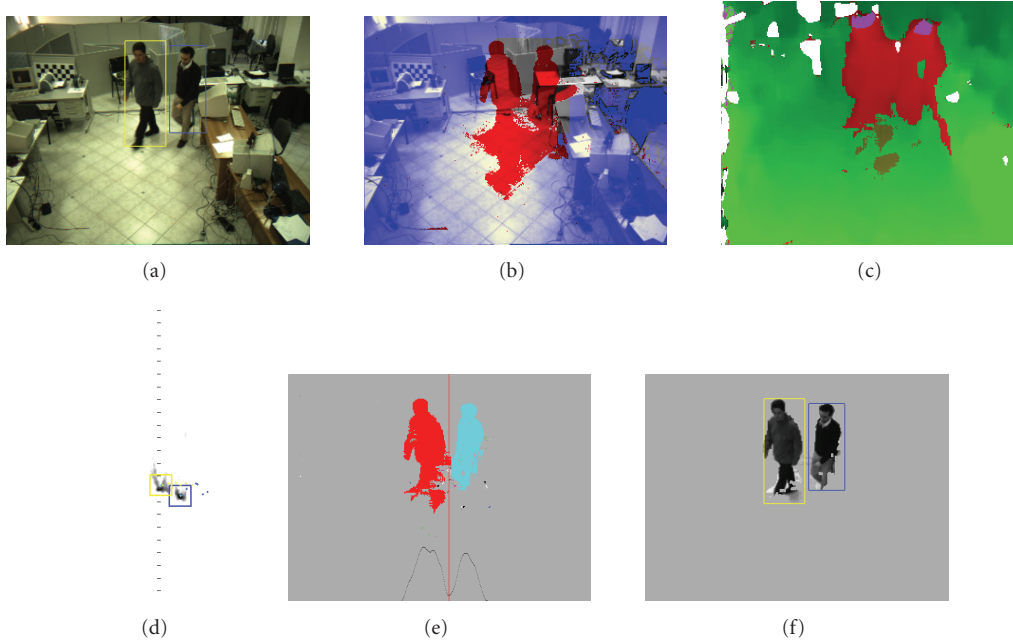


FIGURE 2: An example of the PLT process. From top-left: original image, intensity foreground, disparity foreground, plan-view, foreground segmentation, and person segmentation.

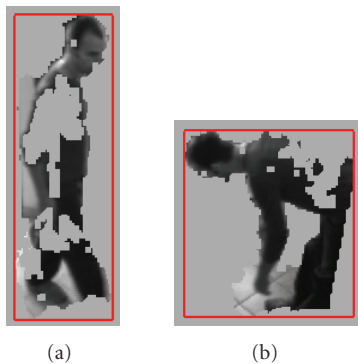


FIGURE 3: Examples of segmentation provided by the stereo tracker.

positives is about 10% and the one of *false negatives* is about 25%.

The posture classification method described in the next sections can reliably tolerate such errors, thus being robust to noise in segmentation that is typical in real world scenarios.

3.2. Person posture recognition

The person posture recognition (PPR) is responsible for the extraction of the joint parameters that describe the configuration of the body being analyzed. The final goal is to estimate a probability distribution over the set of postures $\Gamma = \{U, S, B, K, L\}$, that is, UP, SIT, BENT, ON KNEE, LAID.

The PPR module makes use of a 3D human model and operates in two phases: (1) a training phase, that allows for adapting some of the parameters of this model to the tracked

person; (2) an execution phase, that is composed by three steps: (a) model matching, (b) tracking of model principal points, (c) posture classification.

The 3D model used by the system, the training phase, and the methods used for model matching, tracking, and classification are described in the next sections.

4. A 3D MODEL FOR POSTURE REPRESENTATION

The choice of a model is critical for the effectiveness of recognition and classification, and it must be carefully taken by considering the quality of data available from the previous processing steps. Different models have been used in literature, depending on the objectives and on the input data available for the application (see [1] for a review). These models differ mainly for the quantity of information represented.

In our application, the input data are not sufficient to cope with hands and arms movement. This is because arms are often missed by the segmentation process, while noises may appear as arms. Without taking into account arms and hands in the model, it is not possible to retrieve information about hand gestures. However, it is still possible to detect most of the information that allows to distinguish among the principal postures, such as UP, SIT, BENT, ON KNEE, and LAID. Our application is mainly interested in classifying these main postures, and thus we adopted a model that does not contain explicitly arms and hands.

The 3D model used in our application is shown in Figure 4. It is composed of two sections: a head-torso block and a leg block. The head-torso block is formed by a set of 3D points that represent a 3D surface. In our current

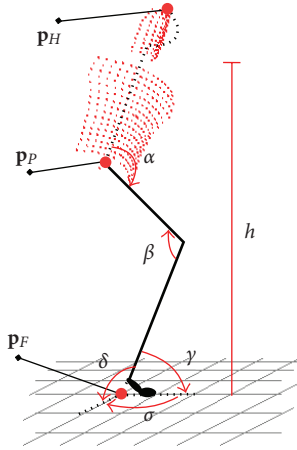


FIGURE 4: 3D human model for posture classification.

implementation, this set contains 700 points that have been obtained by a 180-degree rotation of a curve. Since we are not interested in knowing head movements, we model the head together with the torso in a unique block (without considering degrees of freedom for the head). However, the presence of the head in the model is justified by two considerations: (1) in a camera set-up in which the camera is placed high in the environment, heads of people are very unlikely to be occluded; (2) heads are easy to detect, since 3D and color information are available and modeled for tracking (it is reasonable to assume that head appearance can be modeled with a bimodal color distribution, usually corresponding to skin and hair color).

The pelvis joint is simplified to be a hinge joint, instead of a spherical one. This simplification is justified if one thinks that, most of the times, the pelvis is used to bend frontally. Also, false positives and false negatives in the segmented image and the distortion due to the stereo system make the attempt of detecting vertical torsion and lateral bending extremely difficult.

The legs are unified in one articulated body. Assuming that the legs are always in contact with the floor, a spherical joint is adopted to model this point. For the knee a single hinge joint is instead used.

The model will be built by assuming a constant ratio between the dimensions in the model parts and the height of a person, which is instead evaluated by the analysis of 3D data of the person tracked.

On this model, we define three *principal points*: the head (\mathbf{p}_H), the pelvis (\mathbf{p}_P), and the legs point of contact with floor (\mathbf{p}_F) (see Figure 4). These points are tracked over time, as shown in the next sections, and used to determine measures for classification. In particular, we define an observation vector $z = [\alpha, \beta, \gamma, \delta, h]$ (see Figure 4) that contains the estimation of the four angles α , β , γ , δ , and the normalized height h , which is the ratio between the height measured at the current frame and the height of the person measured during the training phase. Notice that σ is not included in the observation vector since it is not useful to determine human postures.

5. TRAINING PHASE

Since the human model used in PPR contains data that must be adapted to the person being analyzed, a training phase is executed for the first frames in the sequence (ten frames are normally sufficient) to measure the person's height and to estimate the head bimodal color distribution.

We assume that in this phase the person is exhibiting an erect posture with arms below the shoulder level, and with no occlusions for his/her head.

The height of the person is measured using 3D data provided by the stereo-vision-based tracker: for each frame, we consider the maximum value of Z_t^i in ω_t ; the height of the person is then determined by averaging such maximal values over all the training sequence.

Considering that a progressively correct estimation of the height (and, as a consequence, of the other body dimensions) is also available during the training phase, the points in the image whose height is within 25 cm to the top of the head (we assumed that the arms are below the shoulder level) can be considered as head points. Since the input data provide also color of each point in the image, we can estimate a bimodal color distribution by applying the k -mean algorithm on head color points, with $k = 2$. This results in two clusters of colors C_1 and C_2 that are described by the means of their centers of mass μ_{C_1} and μ_{C_2} and their respective standard deviations σ_{C_1} and σ_{C_2} .

Given the height and the head appearance of a subject, his or her model can be reconstructed and the main procedure (that will be described in the next sections) can be executed for the rest of the video sequence.

6. POSTURE CLASSIFICATION ALGORITHM

As already mentioned, the PPR module classifies postures using a three-step approach: model matching, tracking and classification. The algorithm implementing a processing step of PPR is shown in Algorithm 1

A couple of data structures are used to simplify the readability of the algorithm. The sign Π contains the three principal points of the model (\mathbf{p}_H , \mathbf{p}_P , \mathbf{p}_F); Θ contains Π , σ , and ϕ . The sign σ is the normal vector of the symmetry plane of the person. The sign ϕ defines the probability of the left part of the body to be on the positive side of the symmetry plane (i.e., where σ grows positive).

The input to the algorithm is represented by the structure Θ estimated at the previous frame of the video sequence, the probability distribution of the postures in the previous step P_y , and the current 3D point set ω coming from the PLT module. Thus the output will be the new structure Θ' together with the new probability distribution P'_y over the postures.

A few symbols need to be described in order to easily understand the algorithm: η is the model (both the shape and the color appearance); λ is the person's height learned during the training phase; z is the observation vector used for classification, as defined in Section 4.

The procedure starts by detecting if a significant difference in the person's height (with respect to the learned

```

Structures
 $\Theta = [\Pi, \sigma, \phi]$ 
 $\Pi = [\mathbf{p}_F, \mathbf{p}_P, \mathbf{p}_H]$ 

Algorithm
INPUT:  $\Theta, \omega, P_\gamma$ 
OUTPUT:  $\Theta', P'_\gamma$ 
CONST:  $\eta, \lambda, \text{CHANGE\_TH}$            #  $\eta$ : model  $\lambda$ : learned height
                                           # (these values are computed by the Training phase)

PROCEDURE:
   $H = \max\{Z \mid Z \in \omega\}$ ;
  IF  $((\lambda - H) < \text{CHANGE\_TH})\{$ 
     $\Theta' = \Theta$ ;
     $z = [0, 0, 0, 0, 1]$ ;
  }
  ELSE{
     $[\tilde{\mathbf{p}}_P, \tilde{\mathbf{p}}_H] = \text{ICP}(\eta, \omega)$ ;           #
    IF  $(! \text{leg\_occluded}(\omega, \mathbf{p}_F))$            #
       $\tilde{\mathbf{p}}_F = \text{find\_leg}(\omega, \mathbf{p}_F)$            # Detection (Section 7)
    ELSE
       $\tilde{\mathbf{p}}_F = \text{project\_on\_floor}(\tilde{\mathbf{p}}_P)$ ;           #
     $\Pi' = \text{kalman\_points}(\Pi, \tilde{\Pi})$ ;           #
     $\sigma' = \text{filter\_plane}(\sigma, \Pi')$ ;           #
     $\hat{\Pi}' = \text{project\_on\_plane}(\Pi', \sigma')$ ;           # Tracking (Section 8)
     $\rho = \text{evaluate\_left\_posture}(\hat{\Pi}', \sigma')$ ;           #
     $\phi' = \text{filter\_left\_posture}(\rho, \phi)$ ;           #
     $z = [\text{get\_angles}(\hat{\Pi}', \sigma', \phi'), H/\lambda]$ ;           #
  }
   $P'_\gamma = \text{HMM}(z, P_\gamma)$            # Classification (Section 9)

```

ALGORITHM 1: The algorithm for model matching and tracking of the principal points of the model. See text for further details.

value λ) occurred at this frame. If such a difference is below a threshold `CHANGE_TH`, that is usually set to a few (e.g., 10) centimeters, then z is set to specify that the person is standing up without further processing.

Otherwise, the algorithm first extracts the position of the three *principal points* of the model. More specifically, \mathbf{p}_H and \mathbf{p}_P (head and pelvis points) are estimated by using an ICP variant and other ad hoc methods that will be described in Section 7. While \mathbf{p}_F (feet point) is computed in two different ways depending on the presence of occlusions. The presence of occlusions of the legs is checked with the `leg_occluded` function. This function simply verifies if only a small number of points in ω_t are below half of the height of the person (the threshold is determined by experiments and it is about 20% of the total numbers of points in ω). If the legs are occluded, \mathbf{p}_F is estimated as the projection of \mathbf{p}_P on the ground, otherwise it is computed as the average of the lowest points in the data ω_t .

The second step of the algorithm consists in tracking the principal points over time. This tracking is motivated by the fact that poses (and thus principal points of the model) change smoothly over time and it allows for increased robustness to the segmentation noise. As a result of the tracking step, the observation vector z (as defined in Section 4) is computed using simple trigonometry operations (`get_angles`). The tracking step is described in detail in Section 8.

Finally, an HMM classification is used to better estimate the posture for the each frame of the video sequence (Section 9), taking into account the probability of transitions between different postures.

7. DETECTION OF THE PRINCIPAL POINTS

The principal points \mathbf{p}_H and \mathbf{p}_P are estimated using a variant of the ICP algorithm (for a review of the variants of the ICP see [18]). Given two point sets to be aligned, the ICP uses an iterative approach to estimate the transformation that aligns the model to the data. In our case, the two point sets are ω , the data, and η , the model.

The structure of the model η is shown in Figure 4. Since it represents a view of the torso-head block, it can be used only to find the position of the points \mathbf{p}_H and \mathbf{p}_P , but it cannot tell us anything about the torso direction.

ICP is used to estimate a rigid transformation to be applied to η in such a way to minimize the misalignment between η and ω . The ICP is proved [19] to optimize the function

$$E(\mathbf{R}, \mathbf{t}) = \sum_{i=1}^N \|d_i - \mathbf{R}m_i - \mathbf{t}\|^2, \quad (1)$$

where \mathbf{R} is the rotation matrix and \mathbf{t} is the translation vector that together specify a rigid transformation, d_i is a point of

ω , and m_i is a point of η . We are assuming that the points are assigned the same index if they are correspondent. Such correspondence is calculated according to the minimum Euclidean distance between points in the model and points in the data set. Formally, given a point m_j in η , d_k in ω is labeled as corresponding to m_j if

$$d_k = \arg \min_{d_u \in \omega} \text{dist}(m_j, d_u), \quad (2)$$

where the function dist is defined according to the Euclidean metric.

The ICP algorithm is applied by setting the pose of the model computed in the previous frame as initial configuration. For the first frame, a model corresponding to a standing person is used. Since postures do not change instantaneously, this initial configuration allows for quick convergence of the process. Moreover, we limit the number of steps to a predefined number (18 in our current implementation), that guarantees near real-time performance.

From the training phase, we have also computed the head color distribution, described by the centers of mass of the color clusters C_1 and C_2 and the respective standard deviations σ_{C_1} and σ_{C_2} . Consequently, the ICP has been modified to take into account these additional data. Indeed, in our implementation, the search for the correspondences of points in the head part of the model is restricted to a subset of the data set ω defined as follows:

$$\{d_k \in \omega \mid \text{dist}(\text{color}(d_k), \mu_{C_1}) < t(\sigma_{C_1}) \text{ OR} \\ \text{dist}(\text{color}(d_k), \mu_{C_2}) < t(\sigma_{C_2})\}, \quad (3)$$

where $\text{color}(d_k)$ is the value of the color associated with point d_k in the RGB color space and $t(\sigma)$ is a threshold related to the amplitude of standard deviation for each cluster.

Also, since the head correspondences exploit a greater amount of information, we have doubled their weight. This can be easily done by counting twice each correspondence in the head data set, thus increasing its contribution in determining the rigid transformation in the ICP minimization error phase. Once the best rigid transformation (\mathbf{R}, \mathbf{t}) has been extracted with the ICP, it can be applied to η in order to match ω . Since we know the relative position of \mathbf{p}_P and \mathbf{p}_H in the model η , their position on ω can be estimated.

For \mathbf{p}_F we cannot use the same technique, primarily because the lower part of the body is not always visible due to occlusions or to the greater sensibility to false negatives. Since we are interested in finding a point that represents the legs point of contact with the floor, we can simply project the lower points on the ground level, when at least part of the legs is visible. When the person legs are utterly occluded, for example if he/she is sitting behind a desk, we can anyway model the observation as a Gaussian distribution centered in the projection on the ground of \mathbf{p}_P and with variance inversely proportional to the height of the pelvis from the floor (function `project_on_floor` in the algorithm).

8. TRACKING OF PRINCIPAL POINTS

Even though the principal points are available for each image, there are still problems that need to be solved in order to have good performance in classification.

First, detection of these points is noisy given the noisy data coming from the tracker. To deal with these errors it is necessary to filter data over time and, to this end, we use three independent Kalman filters (function `kalman_points` in the algorithm) to track them. These Kalman filters represent position and velocity of the points assuming a constant velocity model in the 3D space. Second, ambiguities may arise in determining poses from three points. To solve this problem, we need to determine the *symmetry plane* of the person (that reduces ambiguities to up to two cases, considering the constraint on the knee joint) and a likelihood function that evaluates probability of different poses. The symmetry plane can be represented by a vector σ' originating at the point \mathbf{p}_F . To estimate the plane of symmetry, one might estimate the plane passing through the three principal points. However, this plane can differ from the symmetry plane due to perception and detection errors. In order to have more accurate data, we need to consider the configuration of the three points, for example colinearity of these points increases noise in detecting the symmetry plane. In our implementation, we used another Kalman filter (function `filter_plane`) on the orientation of the symmetry plane that suitably takes into account colinearity of these points. This filter provides for smooth changes of orientation of the symmetry plane. Furthermore, principal points estimated before are projected onto the filtered symmetry plane (function `project_on_plane`) and these projections are actually used in the next steps.

Given the symmetry plane, we still have two different solutions corresponding to the two opposite orientations of the person. To determine which one is correct, we use the function `evaluate_left_posture` that computes the likelihood of the orientation of the person. An example is given in Figure 5, where the two orientations in two situations are shown. We fix a reference system for the points in the symmetry plane and the orientation likelihood function measures the likelihood that the person is oriented on the left. For example, the likelihood for the situation in Figure 5(a) is 0.6 (thus slightly preferring the leftmost posture), while the one in Figure 5(b) is 0, since the leftmost pose is very unnatural. The likelihood function can be instantiated with respect to the environment in which the application runs. For example, in an office-like environment, likelihood of situation in Figure 5(a) may be increased (thus preferring more the leftmost posture).

Finally, by filtering these values uniformly through time (function `filter_left_posture`), we get a reliable estimate of the frontal orientation ϕ of the person. Considering that we already know the symmetry plane, at this point we can build a person reference system.

This step completes the tracking process and allows for computing a set of parameters that will be used for classification. These parameters are four of the five angles of the joints defined for the model (σ does not contribute

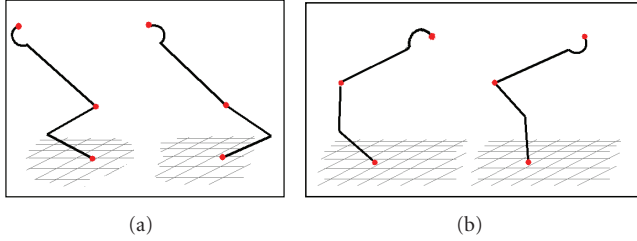


FIGURE 5: Ambiguities.

to posture detection) and the normalized height (see also Figure 4). Specifically, the function `get_angles` computes the angles of the model for the observation vector $z_t = \langle \alpha, \beta, \gamma, \delta, h \rangle$, while the normalized height h is determined by the ratio between the current height and the height learned in the training phase λ . The vector z_t is then used as input by the classification step. As shown in the next sections, this choice represents a very simple and effective coding that can be used to make posture classification.

9. POSTURE CLASSIFICATION

Our approach to posture classification is mainly characterized by the fact that it is not made upon low-level data, but on higher-level ones that are retrieved from each image as a result of the model matching and tracking processes described in the previous sections. This approach grants better results in terms of robustness and effectiveness.

We have implemented two classification procedures (that are compared in Section 10): one is based on frame by frame maximum likelihood, the other on temporal integration using hidden Markov models (HMM). As shown by experimental results, temporal integration increases robustness to the classifier, since it allows for modeling also transition between postures.

In this step, we use an observation vector $z_t = \langle \alpha, \beta, \gamma, \delta, h \rangle$, which contains the five parameters of the model and the distribution probabilities $P(z_t | \gamma)$ for each posture that needs to be classified $\gamma \in \Gamma = \{U, S, B, K, L\}$, that is, UP, SIT, BENT, ON KNEE, LAID. These distributions are acquired by analyzing sample videos or synthetic model variations. In our case, since values z_t are computed after model matching, we used synthetic model variations and manually classified a set of postures of the model to determine $P(z_t | \gamma)$ for each $\gamma \in \Gamma$. More specifically, we have generated a set of nominal poses of the model for the postures in Γ . Then, we collected, for each posture, a set of random poses generated as small variations of the nominal ones, and manually labeled the ones that can still be considered in the same posture class. This produces a distribution over the parameters of the model for each posture. In addition, due to the unimodal nature of such distributions, they have been approximated as normal distributions.

The main characteristic of our approach is that the measured components are directly connected to human postures, thus making easier the classification phase. In particular, the probability distributions of each pose in the space formed

by the five parameters extracted as described in the previous section are unimodal. Moreover, the distributions for the different postures are well separated from each other, and thus making this space very effective for classification.

The first classification procedure just considers the maximum likelihood of the current observation, that is,

$$\gamma_{\text{ML}} = \arg \max_{\gamma \in \Gamma} P(z_t | \gamma). \quad (4)$$

The second classification procedure makes use of an HMM defined by a discrete status variable assuming values in Γ . Probability distribution for the postures is thus given by

$$P(\gamma_t | z_{t:t_0}) = \eta P(z_t | \gamma_t) \sum_{\gamma' \in \Gamma} P(\gamma_t | \gamma') P(\gamma' | z_{t-1:t_0}), \quad (5)$$

$$P(\gamma | z_{t_0}) = \eta P(z_{t_0} | \gamma) P(\gamma),$$

where $z_{t:t_0}$ is the set of observations from time t_0 to time t , and η is a normalizing factor.

The transition probabilities $P(\gamma_t | \gamma')$ are used to model transitions between the postures, while $P(\gamma)$ is the a priori probability of each posture. A discussion about the choice of these distributions is reported in Section 10.

10. EXPERIMENTAL EVALUATION

In this section, we report experimental results of the presented method. Experimental evaluation has been performed by using a *standard setting* in which the stereo camera was placed indoor about 3 m high from the ground, pointing down about 30 degrees from the horizon. The people in the scene were between 3 m and 5 m from the camera, in a frontal view with respect to the camera, and without occlusions. This setting has been modified in order to explore the behavior of the system in different conditions. In particular, we have considered four other settings varying orientation of the person, presence of occlusions, different heights of the camera and outdoor scenarios.

The stereo-vision-based people tracker in [5] has been used to provide XYZ-RGB data of the tracked person in the scene. The tracker processes 640×480 images at about 10 frames per second, thus giving us high resolution and high rate data. The system described in this article has an average computation cycle of about 180 milliseconds on a 1.7 GHz CPU. This value is computed as the average process time for a cycle. However, it is necessary to observe that cycle processing time depends on the situation. When the person is recognized in a standing pose, then no processing on detection and tracking is performed allowing for a quick response. The ICP algorithm takes most of the computational time at each step, but this process is fast, since a good initial configuration is usually available and thus convergence is usually obtained in a few iterations.

The overall system (PLT + PPR) can process about 3.5 frames per second. Moreover, code optimization and more powerful CPUs will allow to use the system in real-time. The overall testing set counts 26 video sequences of about 150

frames each. Seven different people acted for the tests (subject S.P. with 15 tests, subject L.I. with 7 tests, subjects M.Z., G.L., V.A.Z., and D.C. with 1 test each). As for the postures, BENT was acted in 14 videos, KNEE was acted in 2 videos, SIT was acted in 9 videos, LAID was acted in 3 videos, and UP was acted in almost all the videos. Different lighting conditions have been encountered during the experiments that have been done in different locations and in different days, under both natural and artificial lighting with various intensities. The set of data used in the experiments is shown in <http://www.dis.uniroma1.it/~iocchi/PLT/posture.html> and they are available for comparison with other approaches.

The evaluation of the system has been obtained against a ground truth. For each video we built a ground truth by manually labeling frames with the postures assumed by the person. Moreover, since during transitions from one posture to another it is difficult to provide a ground truth (and it is also typically not interesting in the applications), we have defined transition intervals, during which there is a passage from one posture to another. During these intervals the system is not evaluated.

This section is organized as follows. First, we will show the experimental results of the system in the standard setting, then we will explore the robustness of the system with respect to different view points, occlusions, change in the height of the camera, and an outdoor scenario. In presenting these experiments, we want also to evaluate the effectiveness of the filter provided by HMM with respect to frame by frame classification.

10.1. Standard setting

The experiments have been performed by considering a set of video sequences, chosen in order to cover all the postures we are interested in. The *standard setting* described above has been used for this first set of experiments and then the results in this setting are compared with other different settings.

Both for the values in the state transition matrix and the a priori probability of the HMM, we have considered that the optimal tuning is environment dependant. Indeed, an office-like environment will very likely have different posture transition probabilities than those of a gym: in the first case, for example, it might be possible to have high values in the transition between the *sitting* and itself; in a gym the whole matrix should have similar values in all its entries, taking in this way into account that the posture changes often. The optimal values should be achieved by training on video sequences regarding the environment of interest. For simplicity purposes, in our application we have determined values that could be typical of an office-like environment. In particular, we have chosen an a priori probability of 0.8 for the *standing* position and $0.2/(|I| - 1)$ for the others. This models situations in which a person enters the scene in an initial standing position and the transition to all the other postures has the same probability. Moreover, we assume that from any posture (other than standing) it is more likely to stand (we fixed this value to 0.15) than to go to another

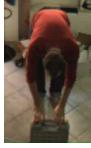



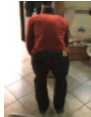
Orientation	HMM	Maximum likelihood
	91.6%	86.7%
	86%	83.1%
	91.2%	89.7%
	89.7%	89.7%
	90.5%	88.9%

FIGURE 6: Classification rates from different view points.

posture. Therefore, the transition probabilities $T_{ij} = P(\gamma_t = i | \gamma_{t-1} = j)$ have been set to

$$\begin{pmatrix} 0.800 & 0.050 & 0.050 & 0.050 & 0.050 \\ 0.150 & 0.800 & 0.016 & 0.016 & 0.016 \\ 0.150 & 0.016 & 0.800 & 0.016 & 0.016 \\ 0.150 & 0.016 & 0.016 & 0.800 & 0.016 \\ 0.150 & 0.016 & 0.016 & 0.016 & 0.800 \end{pmatrix}. \quad (6)$$

Table 1 presents the total confusion matrix of the experiments performed with this setting. The presence of no errors in the LAID posture is given by the fact that the height of the person from the ground is the most discriminant measure and this is reliably computed by stereo vision. Instead, the ON KNEE posture is very difficult because it relies on tracking the feet, which is very noisy and unreliable with the stereo tracker we have used.

The values of classification obtained by using frame by frame classification are slightly lower (see Table 2). Thus, the HMM slightly improves the performance, however maximum likelihood is still effective, since postures are well separated in the classification space defined by the parameters of the model. This confirms the effectiveness in the choice of the classification space and the ability of the system to correctly track the parameters of the human model.

10.2. Different view points

Robustness to different points of view has been tested by analyzing postures with people in different orientations with respect to the camera. Here we present the results of tracking bending postures in five different orientations with respect to the camera. For each of the five orientations we took three

TABLE 1: Overall confusion matrix with HMM.

System ground truth	UP	SIT	BENT	KNEE	LAID
UP	93.2%	0.0%	6.0%	0.0%	0.0%
SIT	0.0%	86.6%	13.4%	0.0%	0.0%
BENT	2.0%	0.5%	97.5%	0.0%	0.0%
KNEE	0.0%	22.2%	0.0%	77.8%	0.0%
LAID	0.0%	0.0%	0.0%	0.0%	100.0%

TABLE 2: Classification rates of HMM versus maximum likelihood.

	HMM	Maximum likelihood
UP	93.2%	90.7%
SIT	86.6%	80.0%
BENT	97.5%	91.6%
KNEE	77.8%	77.8%
LAID	100.0%	100.0%

TABLE 3: Classification rates without and with occlusions.

	No occlusions	Partial occlusion
UP	93.2%	91.5%
SIT	86.6%	81.6%
BENT	97.5%	93.3%
KNEE	77.8%	N/A
LAID	100.0%	100.0%

videos of about 200 frames in which the person entered the scene, bent to grab an object on the ground and then raised up exiting the scene. Figure 6 shows classification rates for each orientation. The first column presents results obtained with HMM, while the second one shows results obtained with maximum likelihood. There are very small differences between the five rows, thus showing that the approach is able to correctly deal with different orientations. Also, as already pointed out, improvement in performance due to HMM is not very high.

10.3. Partial occlusions

To prove robustness of the system to partial occlusions, we make experiments comparing situations without occlusions and situations with partial occlusions. Here we consider occlusions of the lower part of the body, while we assume the head and the upper part of the torso are visible. This is a reasonable assumption given the height (3 m) at which the camera is placed. In Figure 7, we show a few frames of two data sets used for evaluating the recognition of the *sitting* posture without and with occlusions and in Table 3 classification rates for the different postures.

It is interesting to notice that we have very similar results in the two columns. The main reason is that, when feet are not visible, they are projected on the ground from the pelvis joint p_P and this corresponds to determine correct angles

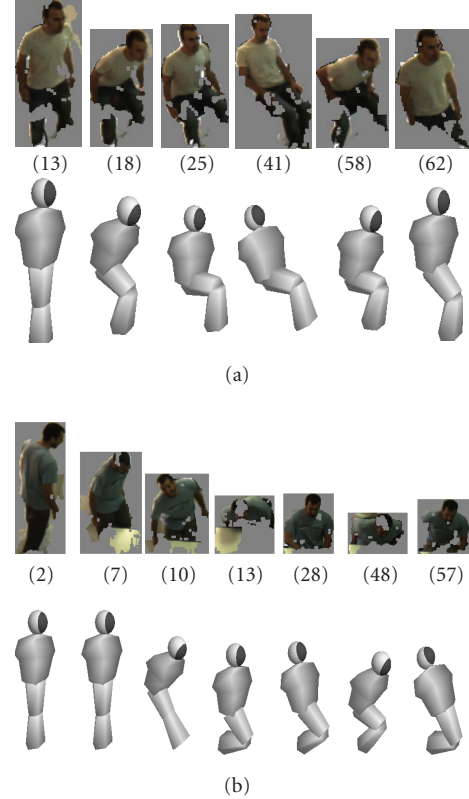


FIGURE 7: People sitting on a chair (nonoccluded versus occluded).

for the postures UP and BENT. Moreover, LAID posture is mainly determined from the height parameter that is also not affected by partial occlusions. For the posture ON KNEE we have not performed these experiments for two reasons: (i) it is difficult to recognize even without occlusions; (ii) it is not correctly identified in presence of occlusions since this posture assumes the feet to be not below the pelvis. These results thus show an overall good behavior of the system in recognizing postures in presence of partial occlusions, that are typical for example during office-like activities.

10.4. Camera at different heights

In the previous setting, the camera was placed 3 m high from the ground. However, we tested the behavior of the system also with different camera placements. In particular, we have put the camera at about 1.5 m from the ground.

TABLE 4: Classification rates with 3 m and 1.5 m camera heights.

	3 m camera height	1.5 m camera height
UP	93.2%	96.3%
SIT	86.6%	77.0%
BENT	97.5%	99.0%
KNEE	77.8%	76.9%
LAID	100.0%	90.0%

TABLE 5: Classification rates in indoor and outdoor environments (1.5 m camera height).

	Indoor	Outdoor
UP	96.3%	95.8%
SIT	77.0%	77.1%
BENT	99.9%	99.0%
KNEE	76.9%	77.0%
LAID	90.0%	90.0%

In this setting the PLT was able to reliably segment and track the people movement. The classification rates for each posture are summarized in Table 4. From the results, it is clear that there are not significant differences, except for the SIT posture that has a relatively lower score. This can be explained by the higher amount of occlusion occurring when a person sits in front of a lower camera, that makes model matching more difficult.

Given this problem with the SIT posture, we have also performed specific tests with the low camera combined with occlusions. The classification accuracy in this setting was 47.2%, thus denoting that performance are highly affected by partial occlusions when the camera is low.

10.5. Outdoor setting

Finally we have tested the system on an outdoor scenario. Since it was not possible to put a camera at a height of 3 m. In the outdoor scenario we used the 1.5 m camera height configuration. Even though the particular outdoor scenario was not very dynamic, since it is located in a private area, we were nevertheless able to test the robustness of the system against natural lights. The classification rates for this setting are summarized in Table 5. The results do not show a significant degradation of the performance with respect to the low camera height setting, showing the ability of the system to operate appropriately even in an outdoor scenario. However, this experiment highlights a higher difficulty in outdoor scenes, where usually it is not possible to place the camera in the best position for the system.

10.6. Error analysis

From the analysis of the experimental results reported above, we have highlighted situations in which errors occur. A first class of errors is due to bad segmentation: (1) when

this occurs during the initial training phase, a noncorrect initialization of the model affects model matching in the following frames, thus producing errors in the computation of the parameters that are used for classification; (2) segmentation errors in the upper part of the body (head and torso) may also be the cause of failures in the model matching performed by the ICP algorithm. These errors are generated by the underlying tracking system and in case they are not acceptable for an application, it is necessary to tune the tracker and/or to add additional processing in order to provide for better segmentation.

Errors that are more related to our approach are mostly determined by incorrect matching of the ICP algorithm, specially in situations where movements are too quick. This is a general problem for many systems based on tracking. A minor problem arises when the person does not pass through nonambiguous postures. In fact, until disambiguation is not achieved (as described in Section 8), posture recognition may be wrong.

Finally, the PPR system is quite robust to different view points, partial occlusions, and to indoor/outdoor environments. The performance is slightly worse when the camera is placed low in the environment. In particular, low camera setting shows a higher sensitivity to occlusions.

11. CONCLUSIONS

In this article, we have presented a method for human posture tracking and classification that relies on the segmentation of a stereo-vision-based people tracker. The input to our system is a set of XYZ-RGB data extracted by the tracker. The system is able to classify several main postures with high efficiency, good accuracy, and high degree of robustness to various situations. The approach is based on the computation of significant parameters for posture classification, that is performed by using an ICP algorithm for 3D model matching. 3D tracking of these points over time is then performed by using a Kalman filter in order to increase robustness to perception noise. Finally, a hidden Markov model is used to classify postures over time.

The experimental results reported here show the feasibility of the approach and its robustness to different points of view, occlusions, and different environment conditions, that makes the system applicable to a larger number of situations.

One of the problems experienced was that the people tracker module works very well when people are in standing position, while quality of data worsen when people sit, lay down, or bend. Classification errors may be reduced by providing feedback from the posture classification module to the people tracker one. In fact, given these information the tracker could adapt recognition procedure in order to provide better data.

The work described in this article can be extended to consider other activities (e.g., gestures), when an appropriate segmentation process is executed before it, providing good quality 3D information of the subject. Other activities, like running or jumping, can be recognized by analyzing directly data coming from the people tracking system, since for these cases the 3D model used in this article would be less relevant.

REFERENCES

- [1] D. M. Gavrila, "The visual analysis of human movement: a survey," *Computer Vision and Image Understanding*, vol. 73, no. 1, pp. 82–98, 1999.
- [2] T. B. Moeslund and E. Granum, "A survey of computer vision-based human motion capture," *Computer Vision and Image Understanding*, vol. 81, no. 3, pp. 231–268, 2001.
- [3] D. Beymer and K. Konolige, "Real-time tracking of multiple people using stereo," in *Proceedings of IEEE Frame Rate Workshop*, Corfu, Greece, 1999.
- [4] L. Iocchi and R. C. Bolles, "Integrating plan-view tracking and color-based person models for multiple people tracking," in *Proceedings of IEEE International Conference on Image Processing (ICIP '05)*, vol. 3, pp. 872–875, Genova, Italy, September 2005.
- [5] S. Bahadori, L. Iocchi, G. R. Leone, D. Nardi, and L. Scozzafava, "Real-time people localization and tracking through fixed stereo vision," *Applied Intelligence*, vol. 26, no. 2, pp. 83–97, 2007.
- [6] M. Isard and A. Blake, "CONDENSATION—conditional density propagation for visual tracking," *International Journal of Computer Vision*, vol. 29, no. 1, pp. 5–28, 1998.
- [7] C. Sminchisescu and B. Triggs, "Kinematic jump processes for monocular 3d human tracking," in *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR '03)*, vol. 1, pp. 69–76, Madison, Wis, USA, June 2003.
- [8] J. Deutscher, A. Blake, and I. Reid, "Articulated motion capture by annealing particle filtering," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR '00)*, vol. 2, pp. 126–133, Hilton Head Island, SC, USA, June 2000.
- [9] D. Ramanan, D. A. Forsyth, and A. Zisserman, "Strike a pose: tracking people by finding stylized poses," in *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR '05)*, vol. 1, pp. 271–278, San Diego, Calif, USA, June 2005.
- [10] S. Ioffe and D. Forsyth, "Human tracking with mixtures of trees," in *Proceedings of the 8th IEEE International Conference on Computer Vision (ICCV '01)*, vol. 1, pp. 690–695, Vancouver, BC, Canada, July 2001.
- [11] R. Navaratnam, A. Thayananthan, P. H. S. Torr, and R. Cipolla, "Hierarchical part-based human body pose," in *Proceedings of the 16th British Machine Vision Conference (BMVC '05)*, Oxford, UK, September 2005.
- [12] D. Demirdjian, T. Ko, and T. Darrel, "Constraining human body tracking," in *Proceedings of the 9th IEEE International Conference on Computer Vision (ICCV '03)*, vol. 2, pp. 1071–1078, Nice, France, October 2003.
- [13] M. Bray, E. Koller-Meier, N. N. Schraudolph, and L. Van Gool, "Fast stochastic optimization for articulated structure tracking," *Image and Vision Computing*, vol. 25, no. 3, pp. 352–364, 2007.
- [14] R. Cucchiara, C. Grana, A. Prati, and R. Vezzani, "Probabilistic posture classification for human-behavior analysis," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 35, no. 1, pp. 42–54, 2005.
- [15] B. Boulay, F. Bremond, and M. Thonnat, "Posture recognition with a 3d human model," in *Proceedings of IEE International Symposium on Imaging for Crime Detection and Prevention (ICDP '05)*, pp. 135–138, London, UK, June 2005.
- [16] L. Goldmann, M. Karaman, and T. Sikora, "Human body posture recognition using MPEG-7 descriptors," in *Visual Communications and Image Processing*, vol. 5308 of *Proceedings of SPIE*, pp. 177–188, San Jose, Calif, USA, January 2004.
- [17] M. Dimitrijevic, V. Lepetit, and P. Fua, "Human body pose recognition using spatio-temporal templates," in *Proceedings of the 10th IEEE International Conference on Computer Vision, Workshop on Modeling People and Human Interaction (ICCV '05)*, Beijing, China, October 2005.
- [18] S. Rusinkiewicz and M. Levoy, "Efficient variants of the ICP algorithm," in *Proceedings of the 3rd International Conference on 3D Digital Imaging and Modeling*, pp. 145–152, Quebec, Canada, May–June 2001.
- [19] P. J. Besl and N. McKay, "A method for registration of 3-D shapes," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 14, no. 2, pp. 239–256, 1992.