UNIVERSITÉ DE NICE - SOPHIA ANTIPOLIS
UFR Sciences

École Doctorale STIC

Thèse

pour obtenir le titre de
Docteur en Sciences
de l'Université de Nice - Sophia Antipolis

Spécialité : Informatique

présentée et soutenue par

Bernard Boulay

Human Posture Recognition for Behaviour Understanding

Thèse dirigée par Monique Thonnat
Équipe d'accueil : ORION – INRIA Sophia-Antipolis

Soutenue publiquement le 23 Janvier 2007
devant le jury composé de :

M.	Michel	Barlaud	Pr. UNSA, France	Président
M.	James	Crowley	Pr. INP Grenoble, France	Rapporteur
M.	Gian Luca	Foresti	Pr. University of Udine, Italy	Rapporteur
M.	François	Brémond	CR, INRIA Sophia Antipolis, France	Examinateur
M.	Philippe	Joly	MC, IRIT, France	Examinateur
M.	Lionel	Martin	STMicroelectronics, France	Examinateur
Mme	Monique	Thonnat	DR, INRIA Sophia Antipolis, France	Directrice

UNIVERSITÉ DE NICE - SOPHIA ANTIPOLIS
UFR Sciences

École Doctorale STIC

Thèse

pour obtenir le titre de
Docteur en Sciences
de l'Université de Nice - Sophia Antipolis

Spécialité : Informatique

présentée et soutenue par

Bernard Boulay

Reconnaissance de Postures pour l'Interprétation d'Activité Humaine

Thèse dirigée par Monique Thonnat
Équipe d'accueil : ORION – INRIA Sophia-Antipolis

Soutenue publiquement le 23 Janvier 2007
devant le jury composé de :

M.	Michel	Barlaud	Pr. UNSA, France	Président
M.	James	Crowley	Pr. INP Grenoble, France	Rapporteur
M.	Gian Luca	Foresti	Pr. University of Udine, Italy	Rapporteur
M.	François	Brémond	CR, INRIA Sophia Antipolis, France	Examinateur
M.	Philippe	Joly	MC, IRIT, France	Examinateur
M.	Lionel	Martin	STMicroelectronics, France	Examinateur
Mme	Monique	Thonnat	DR, INRIA Sophia Antipolis, France	Directrice

Dedicated to
Elisabeth, Serge, my parents,
Célia, Quentin, Bastien, the MIM's babies.

Acknowledgments

I would like to thank Pr. James Crowley and Pr. Gian Luca Foresti for accepting to review this manuscript. I want to thanks them for their very pertinent advices and remarks. A special thanks to Pr. James Crowley for the hand annotated manuscript. I will now switch to French.

Merci à Philippe Joly d'avoir accepté d'être membre du jury de thèse. Merci à Lionel Martin pour sa participation au jury et son aide tout au long de ma thèse. Enfin, je tiens tout particulièrement à remercier Michel Barlaud pour avoir accepté de présider ce jury.

Merci à Monique Thonnat de m'avoir donné la chance de faire cette thèse dans son équipe. Merci à François Brémond pour les discussions scientifiques et diverses que nous avons eues. Merci à tous les deux pour avoir su m'encadrer tout en me laissant autonome dans mes choix.

Un grand merci à Catherine Martin, pour toute l'aide administrative qu'elle m'a apporté tout au long de ces trois années et pour sa grande gentillesse.

Merci à Étienne, premier lecteur de ce manuscrit, à Valery et Marcos pour leurs lectures très constructives.

Un remerciement tout spécial à mes co-bureaux passés: Florent et Tu, pour toute l'aide et l'animation qu'ils ont pu mettre dans le bureau. Un énorme remerciement à mon co-bureau actuel qui m'a supporté pendant la phase de rédaction de ce manuscrit. Et pour toutes ces discussions hautement philosophiques que nous avons pu avoir.

Un merci particulier à Valery, Nadia et Vincent pour leur soutien durant ces années et pendant la phase de rédaction et pour leur amitié.

Un remerciement plus général à tous les membres de l'équipe Orion: Lan, Van-Thinh, Ruihua, Anh-Tuan, Luis, Etienne, Nadia, Naoufel, Marcos, Mohammed, Jean-Paul, Annie, Sabine, Patrice, Nicolas, Celine, Christophe, Magalie, Alberto, Benoit, Julien, Jihene, Florence, Gabriele, pour avoir su y faire régner une

ambiance propice au travail et à la détente.

Un grand remerciement à toutes les personnes ayant fait office de chauffeur et qui m'ont bien simplifié la vie dont je ne citerais pas les noms car la liste est bien longue.

Une petite dédicace aux MIMs: Laure et Nicolas (MIM par alliance), Marie, Alexandra, Carmelo, Fabien pour toutes ces sorties toujours pleines de bonne humeur.

Merci à Valery, Thomas, Muriel et Ariane, de m'avoir fait découvrir les vertes (et moins vertes) plaines d'Azeroth. Un grand merci aux Exaltés, Opportunity et Pacte de sang pour toutes ces excellentes soirées que j'ai passées avec vous.

Enfin un remerciement à ma famille, et plus particulièrement à mes parents pour leurs soutiens et encouragements.

Finalement, merci et mes excuses à toutes les personnes que j'aurais oubliées de nommer précedemment.

Abstract

During this thesis, we have proposed a real-time, generic, and operational approach to recognising human posture with one static camera. The approach is fully automatic and independent from the view point of the camera.
Human posture recognition from a video sequence is a difficult task. This task is part of the more general problem of video sequence interpretation. The proposed approach takes as input information provided by vision algorithms such as the silhouette of the observed person (a binary image representing the person and the background), or her/his position in the scene.
The first contribution is the modeling of a 3D posture avatar. This avatar is composed of a human model (defining the relations between the different body parts), a set of parameters (defining the position of the body parts) and a set of body primitives (defining the visual aspect of the body parts).
The second contribution is the proposed hybrid approach to recognise human posture. This approach combines the use of 3D posture avatar and 2D techniques. The 3D avatars are used in the recognition process to acquire a certain independence from the camera view point. The 2D techniques represent the silhouettes of the observed person to provide a real-time processing. The proposed approach is composed of two main parts: the posture detection which recognises the posture of the detected person by using information computed on the studied frame, and the posture temporal filtering which filters the posture by using information about the posture of the person on the previous frames
A third contribution is the comparison of different 2D silhouette representations. The comparison is made in terms of computation time and dependence on the silhouette quality. Four representations have been chosen: geometric features, Hu moments, skeletonisation, and the horizontal and vertical projections.
A fourth contribution is the characterisation of ambiguous postures. Ambiguities can happen by using only one camera. An ambiguous posture is defined as a posture which has visually similar silhouettes rather an other posture. Synthetic data are generated to evaluate the proposed approach for different point of view. The approach has also been evaluated on real data by proposing a ground truth model adapted to the posture recognition purpose.
A fifth contribution has been proposed by applying the results of the recognition to human action detection. A method based on a finite state machine has been proposed to recognise self-action (action where only one person acts). Each state

of the machine is composed of one or several postures. This method has been successfully applied to detect falling and walking actions.
The human posture recognition approach gives good results. However, the approach has some limitation. The main limitation, is that we are limited in terms of postures of interest for computation time and discrimination reasons. The second limitation is the computation time of the 3D posture avatar generation. By using information about the movement of the observed person in the scene, the approach is able to treat 5-6 frames by second. Some improvement can be done to solve these limitations. In particular, the set of interest postures can be adapted automatically at each frame by considering the previously recognised postures to decrease the number of 3D posture silhouette to extract.

keywords:human posture, 3D human model, geometric features, Hu moments, skeletonisation, Horizontal and vertical projections.

Résumé

Durant cette thèse nous avons proposé une approche temps réel, générique et fonctionnelle pour reconnaître la posture des personnes filmées par une caméra statique. Notre approche est conçue pour être complètement automatique et indépendante du point de vue de la caméra.
La reconnaissance de posture à partir de séquence vidéo est un problème difficile. Ce problème s'inscrit dans le champ de recherche plus général de l'interprétation de séquence vidéo. L'approche proposée prend en entrée des informations provenant d'algorithmes de vision telles que la silhouette de la personne observée (une image binaire où une couleur représente la personne et l'autre le fond) ou sa position dans la scène.
La première contribution est la modélisation d'un avatar 3D de posture. Un avatar 3D de posture est composé d'un modèle 3D humain (définissant les relations entre les différentes parties du corps), d'un ensemble de paramètre (définissant les positions des différentes parties du corps) et d'un ensemble de primitive (définissant l'aspect visuel des parties du corps).
La seconde contribution est la proposition d'une approche hybride combinant l'utilisation de modèles 3D et de techniques 2D. Les avatars 3D de postures sont utilisés dans le processus de reconnaissance pour avoir une certaine indépendance du point de vue de la caméra. Les techniques 2D représentent les silhouettes des personnes détectées pour garder un temps réel de calcul. Cette thèse montre comment les avatars 3D peuvent être utilisés pour obtenir une approche générique et fonctionnelle pour reconnaître les postures. Cette approche est composée de deux parties : la détection de postures qui reconnaît la posture de la personne détectée en utilisant seulement l'information calculée sur l'image considérée, et le filtrage temporel de posture qui reconnaît la posture en utilisant l'information provenant des images précédentes. Une troisième contribution a été faite en comparant différentes représentations 2D des silhouettes au niveau du temps de calcul nécessaire et de leur dépendance à la qualité de la silhouette. Quatre représentations ont été retenues : une représentation combinant différentes valeurs géométriques, les moment de Hu, la skeletonisation et les projections horizontale et verticale.
Une quatrième contribution est la caractérisation des cas ambigus. Des ambiguïtés au niveau de la reconnaissance peuvent se produire en utilisant seulement une caméra statique. Une posture ambiguë est définie par plusieurs postures

qui ont des silhouettes visuellement similaires. Des données de synthèse sont générées pour évaluer l'approche proposée pour différents points de vue. Ainsi, les postures ambiguës sont identifiées en considérant la posture et son orientation. L'approche est aussi évaluée pour des données réelles en proposant un modèle de vérité terrain pour la reconnaissance de posture.
Une cinquième contribution a été proposée en appliquant le résultat de notre approche à la reconnaissance d'action. Une méthode utilisant des machines à états finis a ainsi été proposée pour reconnaître des actions faisant intervenir une seule personne. Chaque état de la machine est composé d'une ou plusieurs postures. Cette méthode est appliquée avec succès pour détecter les chutes et la marche.
Bien que notre approche donne de très bon taux de reconnaissance, il subsiste quelques limitations. La principale limitation de l'approche est qu'elle est limitée en nombre de postures d'intérêt pour des raisons de temps de calcul et de discrimination entre les postures considérées. La seconde limitation est le temps nécessaire à la génération des silhouettes des avatars 3D de posture. En utilisant l'information sur le déplacement de la personne dans la scène, l'algorithme de reconnaissance de posture traite entre 5 et 6 images par seconde. Des améliorations peuvent être faites pour résoudre ces limitations. En particulier, nous pourrions adapter automatiquement l'ensemble des postures d'intérêt au cas considéré, en utilisant par exemple la posture reconnue précédemment pour restreindre les postures 3D dont nous voulons extraire les silhouettes.

Mots-Clés:posture de personne, modèle 3D de personne, caractéristiques géométriques, les moments de Hu, la skeletonisation, les projections horizontale et verticale.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

Human posture recognition is a difficult and challenging problem due to the huge quantity of possible cases. The number of postures depends on the degree of freedom of the human body (i.e. the articulations such as shoulders or knees). Moreover, the morphology of the person (height, corpulence, etc...) influences the perception of the posture. Finally, clothes can also give different types of appearances for the same posture.
The following sections describe the motivations, the context and objectives of this thesis in human posture recognition. This chapter concludes with the manuscript structure.

1.1 Motivations

Human posture recognition is an important part of human behaviour understanding because it provides accurate information about the studied person. The human posture recognition task is involved in three major kinds of applications:

- **Surveillance** applications can be defined like the tracking of one or several people over time to analyse their behaviour. Video surveillance or aware house are typical examples where people are tracked to analyse their activities.

- **Control** applications use information about the posture of the person as a control functionality. For example, the person can interact with a computer according to an intelligent human computer interface (HCI).

- **Analysis** applications need an accurate information about the posture. It is typically used in medical applications (for instance orthopedic purpose), sport monitoring or virtual animation.

In this work the proposed approach aims at recognising human posture for surveillance and control applications. We believe that analysis applications need specific treatment to obtain the desired accuracy in the measurement of the different body parts (size, localisation in space, orientation).

Each of these three types of application must respect certain properties classified in three categories:

- **Number of constraints** needed by an application. For example, a constraint can be to have a static camera, no occlusion, the people in front of the camera, a constant lighting, etc... Surveillance applications need to have less constraints than other application types since they have to work automatically and for a long period of time in various environments. The control and analysis applications have more constraints than surveillance applications since they are generally designed to work on a short period of time in a constrained space. For instance, an user can be front of the camera in an intelligent human computer interface.

- **Accuracy** can be measured by the similarity of the recognised posture with the one performed by the person evolving in the video. A great accuracy is not necessary for surveillance application, whereas it is an important cue for control and analysis applications. Indeed analysis applications need accurate measures on the different body parts.

- Processing **speed** can be classified as real-time or off-line. A real-time computation is commonly defined as a computation that returns the results within a fixed delay. This delay is different according to the purpose of the application. Surveillance and control application may need a high processing speed to detect some behaviour at time. For instance, when a person interacts with a computer, the results must be immediate. On contrary, analysis applications can be processed off-line.

1.2 Context of the Study

It is necessary to place the human posture recognition task in the complete treatment chain of video understanding. Different surveys on video understanding (also called human motion analysis) have been proposed throughout the last twenty years:

- In [Cedras and Shas, 1995], the authors present an overview of methods for motion extraction prior 1995. Human motion is described as action recognition, recognition of body parts and body configuration estimation.

- In [Aggarwal and Cai, 1999], human motion is interpreted as three tasks which are the same as in [Cedras and Shas, 1995] but with different names: motion analysis involving human body parts, human tracking with a single or multiple cameras and human activity recognition.

- In [Gavrila, 1999], the authors described the major works on human motion analysis prior 1998. They describe different methodologies classified into 2D approaches with or without explicit shape models and 3D approaches.

- In [Moeslund and Granum, 2001], the authors give an overview of human motion prior 2000 and completed in [Moeslund, 2003] prior 2002. A human motion analysis system is constituted of four tasks: initialisation, tracking, posture estimation and recognition. An initialisation of the data is necessary e.g. an appropriate model of the subject must be established. The tracking task computes the relations over the time of the object detected by the segmentation task by finding correspondences in consecutive frames. Then the posture estimation of the subject is made. The final task analyses the posture and other parameters to recognise the actions performed by the subject.

- In [Wang et al., 2003], prior works on human motion analysis are described up to 2001. The proposed taxonomy is composed of five tasks: motion segmentation, object classification, human tracking, action recognition and semantic description. The purpose of semantic description of human behaviour is to "reasonably choose a group of motion words or short expressions to report the behaviors of the moving objects in natural scenes".

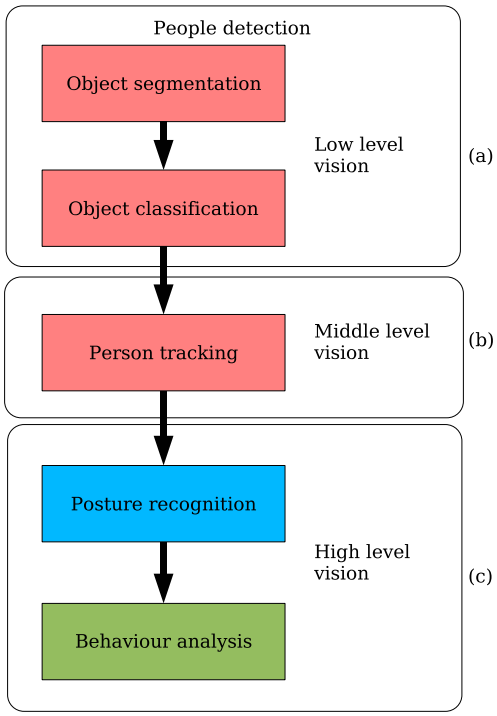

Figure 1.1: A general video understanding task framework. The task is composed of: (a) a low level vision task which detects people evolving in the scene, (b) a middle level vision task which tracks detected people and (c) a high level vision task to recognise posture and analyse behaviour according to information previously computed.

This work has been conducted in the Orion team located at INRIA Sophia-Antipolis. Orion is a multi-disciplinary team at the frontier of computer vision, artificial intelligence and software engineering. The team has accumulated a strong expertise in these areas throughout the years. One topic of particular interest is knowledge-based image and video understanding. The proposed work takes place in this context. Like in [Wang et al., 2003], general framework of a video understanding task can be described from low level vision to high level vision (figure 1.1):

- object segmentation.
- object classification.
- human tracking.
- human posture recognition.
- human behaviour analysis.

The first step to such an approach is to detect people evolving in a video sequence. The people detection task is important for the the next tasks such as human tracking or behaviour analysis. Detecting people is generally achieved by a segmentation and a classification task. Humans are then tracked throughout the video sequence. Finally human behaviour analysis is performed using the information computed during the previous tasks. The localisation of the posture recognition task in the treatment chain is discussed in the chapter 3. In particular, why this task needs temporal information provided by the human tracking task is presented. Moreover the solution of video understanding problem proposed in the team is described in detail in appendix A.

1.3 Objectives

The goal of this work is to propose a generic approach to recognise the global human body posture in video sequences. The approach must be generic to be adapted to most of the situations.

The approach takes place in a more general task of video understanding, fed by a people detection task and computes posture information to a behaviour understanding task. The people detection task gives information about the people evolving in the scene such as its positions and dimensions. The people are generally represented by a binary silhouette. Since the approach has to use this silhouette to determine the posture, it must be efficient with different types of silhouette (perfect and erroneous ones).

Driving by the fact that the targeted applications are surveillance and control ones, the proposed approach must respect the previously listed properties (i.e.

number of constraints, accuracy and speed).
As seen in section 1.1, the number of constraints needed by the approach is relevant to propose a generic human posture recognition approach. First, the type of video camera needed is important. By using only one static camera, the approach can be directly applied to existing systems or easily applied for new systems of video interpretation. Second, a certain independence from the point of view is an important cue to propose an operational approach. Indeed, for instance if a person may face the camera for control application, it is generally not possible to ask to people evolving in a scene to look at the camera in surveillance applications.
The accuracy needed by a surveillance and a control application is not the same. Surveillance needs more general information about the posture than a control one. The speed of surveillance and control applications is a very important property. For instance, the application must be able to raise an alarm when a person is falling (or even before) and not 10 minutes after.
Our work aims at solving these problems by the main following contributions:

- The advances made in the computer graphics research field is used to propose a 3D human model adapted to the human posture recognition purpose. An independence from the camera viewpoint is acquired by using a 3D human model.

- The proposed hybrid approach to recognise human posture combines 2D silhouette representations and the use of a 3D human model. The 2D representations maintain a real-time processing and are adapted to the different types of silhouette.

A hierarchical taxonomy of interesting postures are identified and the 3D model parameters are defined to represent these postures of interest. Once a person is detected in the scene, the 3D models are placed in the same position of the detected person thanks to the calibration matrix. The 2D silhouettes for each posture of interest and each possible orientation are then generated. These generated silhouettes are compared with the detected silhouette to choose the most similar one and determine the posture of the person evolving in the scene. Temporal coherency of the posture is used to remove sporadic recognition errors. This approach is successfully evaluated on both synthetic and real data. Moreover the proposed approach is tested for behaviour analysis to recognise actions such as the fall or the walk. These contributions are presented in the next chapters of the manuscript as described in the next section.

1.4 Manuscript Structure

This manuscript is structured in six main chapters.

Chapter 2 introduces the reader to the previous works on human posture recognition. Different techniques are presented for both physiological and

mechanical sensors and video sensors. Physiological sensors, such as MEMS (Micro Electro Mechanical System), are designed for cooperative person whereas video sensors are involved for non-cooperative person. A focus is made on human posture recognition by describing in particular body markers and video sensors techniques. The video sensors techniques are classified in 2D and 3D techniques. Both of them have strengths and weaknesses. The goal of this thesis is to propose an approach which combines their strengths by minimising their weaknesses.

Chapter 3 presents our objectives and gives an overview of the proposed approach to recognise human posture. As explained in section 1.3, the targeted applications are surveillance and control ones. Thus several constraints for the approach have been identified: real-time, independence on the view-point, an automated approach and the use of one monocular static camera. An hybrid approach is then proposed by combining 2D techniques and the use of 3D posture avatar, to respect these constraints. Moreover a contextual knowledge base is used to drive the posture recognition task by giving information about the scene.

Chapter 4 describes the proposed 3D posture avatar which is a combination of a 3D human body model and a set of parameters corresponding to a particular posture. The chapter shows how the different body parts of the 3D avatars are animated according to the parameters. A set of postures of interest is then identified and modeled. These postures are classified in a hierarchical way from general to detailed postures.

Chapter 5 introduces the proposed hybrid approach which is composed of two main tasks:

- the first task computes the posture of the detected person with only information of the current frame and the 3D models. The 3D candidate posture avatar silhouettes are generated by projecting the 3D posture avatars on the image plane by defining a virtual camera with the same characteristics than the real one. Each 3D posture avatar is placed in the 3D scene according to the people detection task, then all possible avatars are oriented with respect to different angles to generate all possible silhouettes. The detected and generated silhouettes are modeled and compared with 2D representations to obtain the posture.

- the second task uses information about the posture from the previous frames. The recognised postures from the previous frames are used to verify the temporal coherency of the posture in order to provide the most probable posture.

The different 2D silhouette representations involved in the approach are also described in this chapter.

In **chapter 6**, the proposed approach is investigated and optimised. A ground-truth model is proposed to evaluate the proposed human posture recognition algorithm. Synthetic data are generated from many viewpoints to compare the different 2D representations and the influence of the parameters on the proposed human posture recognition approach. The approach is tested on several real video sequences and for different types of silhouettes.

Chapter 7 shows how the posture can be used to recognise some actions involving only one person. An action is represented with a finite state machine. Each state is represented with one or several postures. The method has been tested for different actions such as the fall (important action for medical purpose) or walking.

Finally, **chapter 8** concludes this works, by summarising the contributions of this thesis, and by presenting short-term and long-term perspectives.

Chapter 2

State of the Art

As seen in the previous chapter, human posture recognition is one step in the human behaviour analysis task. In this chapter, the previous work on human posture recognition is described. According to the type of sensor used by the human posture recognition technique, the existing approaches can be categorised in the main families using:

- physiological and mechanical sensors,
- video sensor.

Physiological and mechanical sensors are used for applications where the patient is cooperative such as in health-care applications. Video cameras (but not only) are generally used for applications where people are not cooperatives such as in video-surveillance applications. Techniques using physiological and mechanical sensors are described in the section 2.1 focusing on the body markers. Then a description of techniques using video sensors is given in section 2.2. The strengths and weaknesses of both techniques are discussed in section 2.3. This chapter is concluded in section 2.4 by briefly describing the proposed approach to recognise human posture.

2.1 Physiological and Mechanical Sensors

Physiological and mechanical sensors are designed for cooperative people. Typically, they are used for applications involving elderly people (e.g. elderly people care at home). The sensors can be used for health purposes (e.g. by monitoring cardiac rhythm) or fitness/sport applications (e.g. by monitoring cardiac rhythm, weight, etc...). Existing techniques can be classified in terms of their constraints for the patient: the invasive and the non-invasive techniques.
Invasive techniques are described in section 2.1.1, and the non-invasive techniques are presented in section 2.1.2, focusing on body markers in section 2.1.3.

2.1.1 Invasive Techniques

The invasive techniques use sensors worn by the patient. The sensors must respect some constraints:

- not-constraining for the patient in her/his daily activity,
- easy to use,
- non-health dangerous.

The sensors measure specific data of the patient, and can interpret her/his activity. They are often linked to a distant tele-operator who receives the alarm provided by the sensor. The sensors detect some specific motions such as walking, sitting or standing. They are also often used to detect unexpected motion, in particular the fall. In [Williams et al., 1998], the authors measure the impact associated with the fall of a person and used a mercure based sensor to detect lying posture. Kemp et al. determine the 3D orientation of body parts with 3 accelerometers and 3 magnetometers [Kemp et al., 1998]. Fall is then detected with the 3D orientations. In [Wu, 2000], the author measures the horizontal and vertical velocities at various locations of the trunk to detect the fall signature from normal activities (walking, rising from a chair, sitting down, descending stairs, picking up an object from the floor, lying down on a bed). Kelly et al. evaluate a non intrusive sensor to reduce falls of nursing home patients [Kelly et al., 2002]. The sensor is a patch (size of a credit card) that can be worn directly on the skin (on the thigh) or incorporated into clothing. The patch sends an alarm when the patient approaches weight-bearing angle (see figure 2.1). In [Noury et al., 2004], the authors propose a sensor constituted of three accelerometers to determine the leaning (i.e. orientation) of the trunk. Since these sensors are worn by the patient, they are not always well accepted by the patient.

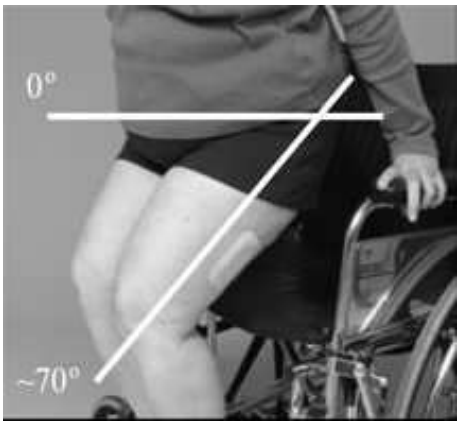

Figure 2.1: Patch sends alarm when patient approaches weight-bearing angle [Kelly et al., 2002].

2.1.2 Non-Invasive Techniques

For invasive techniques, the sensors are worn by the person herself/himself. Whereas, for non invasive techniques, the sensors are installed in the environment. For instance, the sensors can be a video camera (in the next section, a detailed description is performed), a pressure sensor on a chair or an armchair, a sensor for detecting opened/closed door, window or cupboard. The non-invasive techniques are used to monitor daily patient behaviour. For instance in [Center, 2006], room for non-invasively monitoring the human respiratory system is described. The room consists of sensorised furniture: a ceiling dome microphone, a pressure sensor bed and a wash-stand display. The ceiling dome microphone can detect both normal and abnormal breathing sounds. The pressure sensor bed can monitor body movement and posture. Finally, the wash-stand display gives information on the daily life of the patient. These techniques are better accepted by the patient than invasive ones because the patient do not have to wear the sensors. But these techniques are up to now limited to some specific behaviour understanding.

2.1.3 Body Markers

Body markers are widely used for motion capture applications. Motion capture is used to simulate realistic motion of synthetic objects in a virtual space for applications such as animation, medical simulation, biomechanics, virtual reality, simulation and training. A motion capture system is composed of markers and receivers. The markers are usually placed at the different articulations of the person. The markers are tracked throughout the time by the receivers to determine their position and orientation. This information is analysed to determine the posture of the performer.
Several systems are commercialised. These systems can be classified in three categories:

- Inertial based systems,
- Magnetic systems,
- Opto-electronic systems.

The systems can be a combination of these different techniques.

Inertial based systems measure positions and angles of different devices such as accelerometers or gyroscopes. *Intersense* [Intersense, 2006] proposes a system of 6 MEMS (Micro Electro Mechanical Systems) inertial sensors with an integrated system (figure 2.2) which provides position and orientation of the sensors.

Magnetic systems calculate the position and angle of a marker by measuring the relative magnetic flux between the marker and the receiver. *MotionStar* and *MotionStar Wireless 2* (figure 2.3) are two products of *Ascension* [Ascension, 2006] society, which use magnetic tracker to determine the motion of the performer.

Figure 2.2: Wireless inertiaCube 3: the receiver and the marker [Intersense, 2006].

Other systems do not use a dedicated magnetic source. The system TRIDENT developed by LETI is a wearable system to capture the movement in 3D. It contains 6 MEMS (Micro Electro Mechanical Systems) sensors (3 accelerometers and 3 magnetometers) that may be worn on the trunk of a person. 3D rotation angles are determined from the earth gravitational and magnetic fields, it does not need any external source. It can be used to determine if a person is standing, sitting or lying by studying the orientation of the upper body. However, this kind of system is sensitive to exterior magnetic sources such as computers and electricity cables.

The opto-electronic systems use reflective markers illuminated from strobes on the camera and triangulate each marker from its relative location on a 2D map. There exist several commercial systems. For instance, *CODA* (Cartesian Opto-electronic Dynamic Anthropometer) of *Codamotion* [Codamotion, 2006] is composed of a receiver with three sensors (two sensors measure the horizontal movements and the other one measures the vertical movements) and markers. The markers are small infrared LED (Light Emiting Diodes). The LEDs are powered by batteries placed on the performer. *Elite* system of *BTS Bioengineering* [Bioengineering, 2006] has been developed for gait analysis. *Vicon* [Vicon, 2006] system is composed of a series of high resolution cameras (figure 2.4) with special strobe lights to capture the position of the markers. The markers are small spheres painted with a retro-reflective substance (figure 2.5). The main drawback of the opto-electronic systems is that they cannot be used in outdoor environment because the reflective markers can be misdetected. Moreover, another point is that the receiver must be multiplied to avoid the non-detection of markers due to occlusion.

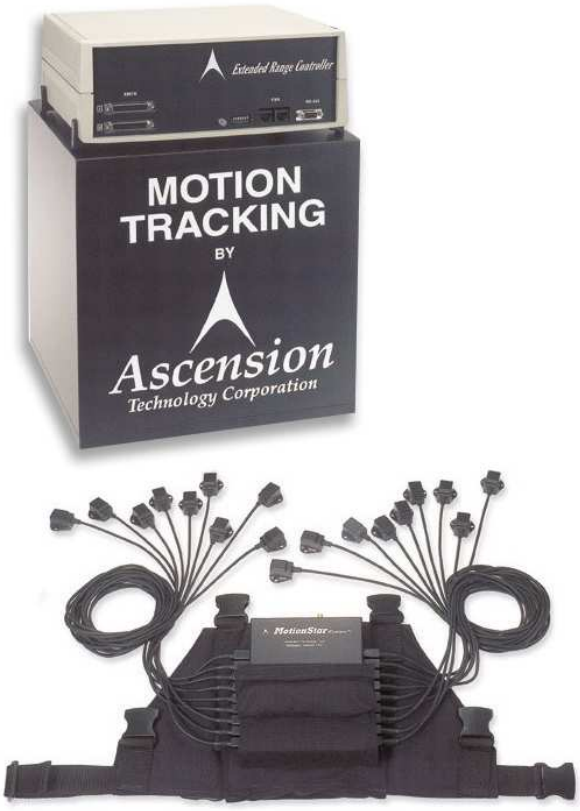

Figure 2.3: MotionStar Wireless 2: the extended range transmitter and controller, and performer mounted electronics sensors and RF (Radio Frequency) transmitter [Ascension, 2006].

These systems are accurate and are able to give measures with a very good precision (errors are often less than 0.1 mm). However, they are limited to a constrained space: the performer must be in a predefined location. Also. the material is often expensive, in particular for opto-electronic systems which need high frequency receivers to correctly capture the motion of the performer.
Moreover, some vision techniques can train their algorithms to recognise human body postures by using data computed with motion capture systems. Synthetic data can also be generated with such motion data for performance evaluation purpose.

2.2 Vision Techniques

The vision techniques to determine human posture can be classified by considering different taxonomies: the type of model used (stick figure, statistical, volumetric),

Figure 2.4: The MX3 camera for a Vicon system [Vicon, 2006].

the dimensionality of the work space (2D or 3D), the sensor type (infra-red, visible light), the quantity of sensors (mono or multi-cameras), static or moving camera. Similar as [Gavrila, 1999], we have classified previous work based on non-intrusive vision techniques by considering the type of model used and the dimensionality of the work space:

- 2D approaches with explicit models,
- 2D approaches with statistical models
- and 3D approaches.

2.2.1 2D Approaches with Explicit Models

The 2D approaches with explicit models need a 2D model and a priori knowledge on how people appears on the image. They compute the different body parts of the detected person to determine the posture. The different body parts are generally the extremities of the human body (the two hands, the two feet and the head) and the limbs of the body (the two legs and the two arms). The 2D models can be stick figures wrapped with ribbons like in the cardboard model [Ju et al., 1996] see figure 2.6.

In [Haritaoglu et al., 1998a] and [Haritaoglu et al., 1998b], the authors describe their *Ghost* system. This system determines the location of different body parts for recognising general postures. It first determines the general posture (standing, sitting, bending and lying) and the orientation (front or back view, right or left view) of a person by representing postures based on the average horizontal and vertical projections of the silhouette. The information on posture and orientation allows the system to analyse the contour of the silhouette in order to determine the different body parts.
In [Park et al., 2000], the authors propose an approach to recognise human postures from a single image. Each body part is considered as a 3D cylinder and its projection to the image plan is a 2D ribbon. Ribbons that correspond to body

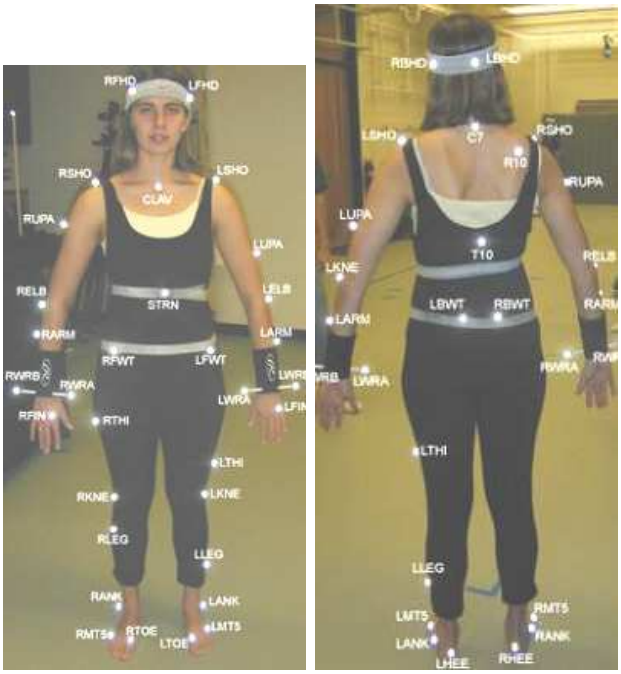

Figure 2.5: 41 markers for Vicon system placed strategically on the body [Al-Zahawi, 2006].

parts are estimated from the boundary contours. The image is segmented with a watershed algorithm to fuse the homogeneous regions. The curve segments are then extracted and a new region fusion is made by studying the regions attached on a curve segment. If the regions can be a part of the human body then they are fused into a single region. A skin color region detection is also applied whenever it is possible. The 2D ribbons are then estimated from the curve contours that enclose candidate human body parts. Finally, the 2D ribbons are matched with a human body model.
In [Wren et al., 1997], the different body parts are determined directly during the segmentation step of the video. Each pixel of the background is represented with a mean color value and a distribution about that mean. These values are updated in time to take into account the changes in the background. Each pixel is then classified into a background or a foreground pixel by using a multi-class statistical model of colour and shape. The result is a 2D representation of the different homogeneous parts of the body (figure 2.7).

These approaches need to detect correctly all the body parts to achieve good posture recognition. They are generally very sensitive to segmentation errors. The 2D approaches with explicit models gives good result when the segmentation is correct. Moreover, since a 2D model is used, the approaches are dependent on the point of view of the camera.

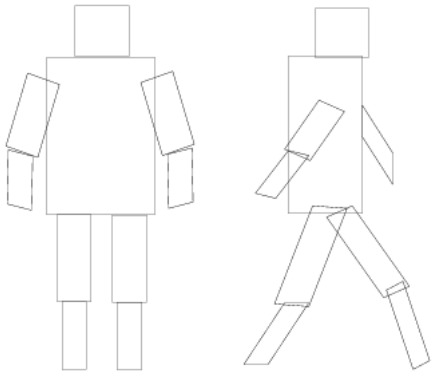

Figure 2.6: The cardboard person model. The limbs of a person is represented by planar patches which are different depending on the orientation of the model [Ju et al., 1996].

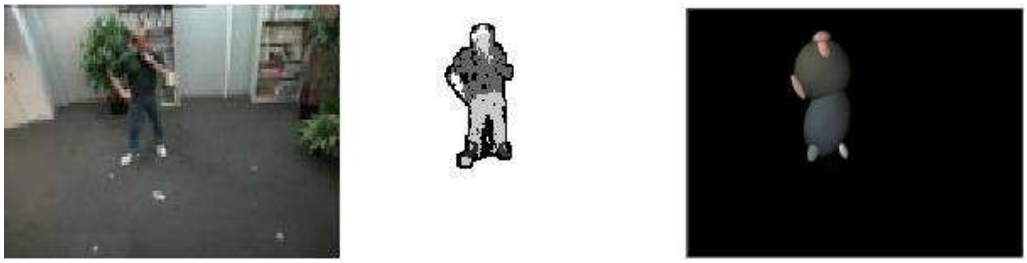

Figure 2.7: Video input, people segmentation and a 2D representation of the homogeneous parts of the body [Wren et al., 1997].

2.2.2 2D Approaches with Statistical Models

To solve the problem of segmentation errors, the 2D approaches with statistical models recognise postures without having to detect the different body parts. The postures are statistically modeled during a training phase. Statistical terms are generally derived from the silhouette of the person.
In [Baumberg and Hogg, 1995], the authors analyse statistically the 2D contours of the silhouette. The contour is represented by the point distribution model (PDM). The PDM is based on a set of example shapes of a person. Each shape is described with a set of points which correspond to the characteristic of the shape (i.e. extremities of the body). The authors propose a method to recognise walking persons.
Rosales and Scarloff propose a non linear supervised learning technique: the specialised mapping architecture (SMA). The SMA is composed of several mapping functions (from input data to output data) and a matching function automatically estimated from the data. Each mapping function is defined by a part of the input data [Rosales and Sclaroff, 2000b], [Rosales and Sclaroff, 2000a].
[Ardizzone et al., 2000] propose an approach to recognise human arm posture.

The eigen values of the covariance matrix associated with the arm silhouette are computed. A support vector machine (SVM) is trained with the eigen vectors to recognise the different arm postures.
In [Fujiyoshi et al., 2004], the authors use skeletonisation to represent a person. The skeleton is computed on the silhouette by extracting the points of the contour which maximise the distance to the centroid. The posture of the person is determined by using a metric based on the skeleton. The body inclination is computed to achieve this task.
[Panini and Cucchiara, 2003] model postures with 2D probabilistic maps by using horizontal and vertical projections of the silhouette. A training set of T images referred to the standing posture is considered. The 2D horizontal probabilistic map $\mathcal{H}$ is computed as follow:

$$\mathcal{H}(x,y) = \frac{1}{T}\sum_{t} g\left(H^t\right) \tag{2.1}$$

where

$$g\left(H^t\right)(x,y) = \begin{cases} 1 & \text{if } y = H^t(x) \\ 0 & \text{elsewhere} \end{cases}$$

and H^t is the horizontal projection of the t^{th} silhouette example. The analogous computation is done for the vertical projection. The recognition is achieved by comparing the horizontal projection H of the silhouette with the pre-computed 2D probabilistic map $\mathcal{H}$:

$$\frac{1}{width(H)} \sum_{x=1}^{width(H)} \mathcal{H}(x, H(x)) \tag{2.2}$$

An example of standing posture probabilistic map is shown in figure 2.8. The authors are interested in detecting four postures: standing, crawling, sitting and lying.

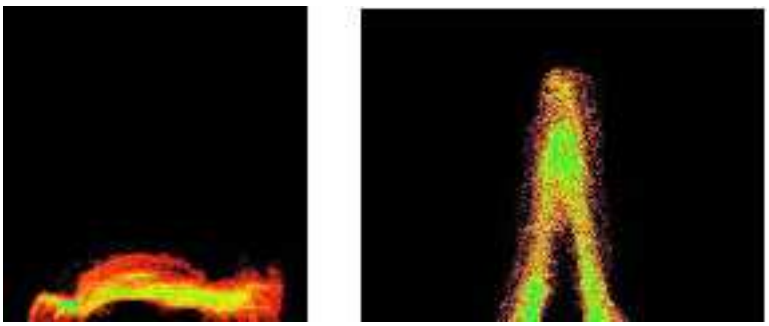

Figure 2.8: Horizontal and vertical 2D probability maps for standing posture. Green points have a higher probability than the red ones to belong to a standing posture [Panini and Cucchiara, 2003].

In [Bradski and Davis, 2002], the authors propose an approach to recognise T-shape, Y-shape and ⊢-shape postures. The silhouettes are represented by the seven higher order Hu moments. Since these moments are of different orders, the Mahalanobis distance metric is used as a matching criteria based on a statistical measure of closeness to training samples.
In [Tanikawa and Takahashi, 2003], the authors train an artificial neural network (ANN) to determine significant points of human body (head, hands, feet, shoulder joints, elbow joints and knee joints). The input feature vector of the ANN is extracted from a human silhouette image, and the output of the ANN indicates the 2D coordinates of the significant points. Three types of feature vectors are extracted: the raw pixel data, the coordinates of a sample of contour centered on the centroid location, or the distance of the sampled pixels to the centroid. In their work, the camera is assumed facing the front of the human.

2.2.3 3D Approaches

Proposed approaches for human posture recognition in 3D space are described. The general approach consists in finding the parameters of a 3D model such as the projection of the model on the image plane to fit the silhouette of the detected person. Previous works can be classified according to the quantity of video cameras needed by the approaches.

2.2.3.1 Mono Camera

Some works have been proposed in the recognition of hand posture, in particular for sign language recognition applications. The hand is represented with an articulated model and the approach can be applied to the whole human body.
In [Shimada et al., 2001] and [Athitsos and Sclaroff, 2001], the estimation of the hand postures is based on 2D image retrieval. A large amount of possible hand appearances are generated from a given 3D hand model by rotating the model joints and for different view points. Appearances and associated joints parameters are stored in a data-base. The hand posture of an input image is determined by retrieving in the data-base the most similar appearance. In [Shimada et al., 2001], the appearance is the boundary of the hand silhouette. In order to achieve a real-time processing, the data-base is defined as an adjacency map which groups the hand postures with similar joints and point of view. The adjacency map of 16000 possible hand appearances is implemented on a cluster of six computers. In [Athitsos and Sclaroff, 2001], the considered appearances are the edges of the hand. The 107328 appearances are compared with the input image using a Chamfer distance on the edges.

Work on human body posture recognition with one camera is now described. Work in this area can be classified as model-based or learning-based.

Model-based approaches use an articulated 3D body model. They consist in computing the parameters of the 3D model, such as the model projection on the image plane fits with the input image (often the silhouette). Some approaches compare the contour of the input silhouette with one of the projected model.

In [Kameda et al., 1993], a model-matching method to estimate the pose of an articulated object is proposed. The model and the algorithm are clearly separated. An articulated object is defined as a set of several solid parts arranged in a tree graph structure. Each part in the model is taken up one by one and its rotation angles are determined based on the overlap relationship between the contour of the silhouette and that of the projected part on the image plane.
An alternative is to directly compare the two silhouettes.
In [Moeslund and Granum, 2000], the authors represent the human model in a phase space spanned by the degree of freedom of the model. They use the analysis by synthesis approach to match the phase space model with the real image and thereby estimating the posture. Several constraints are used to decrease the dimension of the phase space. The dimensionality of the phase space is set according to the application (if only the head posture is needed then only the degree of freedom associated with the head is considered). Kinematic constraints of the human motor system are considered (the leg cannot bend forward to the knee). Collision constraints are also considered (two body parts cannot occupy the same space at the same time). This approach focuses on the arm posture to allow real-time processing. The comparison of the image silhouette and the synthesised model depends on the complexity of the model. If a complex model is similar to the subject (in term of clothe deformations simulation) a XOR operation can be used. If the model corresponds to a stick figure model, an AND operation compares the silhouette. Moreover, the approach needs an initialisation phase, where the actor places her/his left arm stretched out and parallel to the image plane. In [Sminchisescu and Telea, 2002], the authors use a 3D human body model (figure 2.9) which consists in an articulated skeleton covered by flesh built from superquadric ellipsoids. They assume a reasonable initialisation of the 3D model and focus on a likelihood model composed of an attraction term and an area overlap term. Both terms are based on distance map (minimal distance of pixel silhouette to the boundary of the silhouette) extracted from the silhouette. The surface of the model is discretised as mesh and each node is projected on the image plane. During parameter estimation, likelihood is computed and minimised for each projected node.

A third 3D model-based technique is based on the articulations of the human body (extracted manually) and anthropometric information. Usually around 15 articulations and body parts are manually annotated. Given the set of articulation points, the body posture is estimated. [Barron and Kakadiaris, 2003] propose a method based on the geometric relations between the different body parts, and apply it to a single image. [Zhao et al., 2004] propose a method to

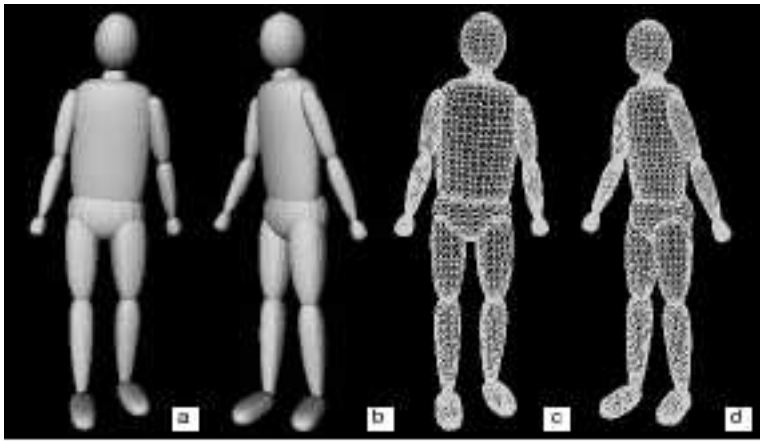

Figure 2.9: Human model: flat shaded (a,b) and discretisation (c,d) [Sminchisescu and Telea, 2002].

reconstruct human posture from un-calibrated monocular image sequences. The human body articulations are extracted and annotated manually on the first image of a video sequence, then image processing techniques (such as linear prediction or least square matching) are used to extract articulations from the other frames. In order to minimise an energy function, translations and rotations are proposed to adjust the flexible 3D human model with encoded biomechanical constraints.

The learning-based approaches avoid the need of an explicit 3D human body model. These approaches store in a data-base images with annotated 3D postures. To recognise the posture of an input image, the most similar annotated image is taken as reference to 3D posture. [Mori and Malik, 2002] localise the position of 14 articulations on the image to estimate the posture in the 3D space. The approach consists in storing a number of 2D view examples of the human body in a variety of different configurations and viewpoints with respect to the camera. On each of these stored views, the locations of the body joints are manually marked and labeled. The test shape is then matched to each stored view, using the technique of shape context matching (a histogram is associated to a sample of contour points). Assuming there is a similar stored view, the body joints are transfered on the shape. Given the joint locations, the 3D posture is then estimated by assuming that the relative lengths of the body parts are known. In [Shakhnarovich et al., 2003], the authors use hashing-based search technique to rapidly find relevant examples in a large image data-base, and to estimate the parameters for the input using a local model learnt from these examples. Images are represented by multi-scale edge direction histograms. Edges are detected with the Sobel operator and each pixel is classified into one of predefined direction. The histogram of the direction is then computed within

square windows of different sizes.
In [Agarwal and Triggs, 2006], the authors propose a learning-based method for recovering 3D human body posture from single images and monocular image sequences. Instead of explicitly storing and searching for similar image in a data-base, they use a non-linear regression to distill a large data-base into a compact model. The silhouettes of the training data-base are encoded with shape context descriptor. Their method recovers postures by direct non-linear regression against shape context descriptor extracted from an input silhouette. Different regression methods are evaluated: ridge regression, relevance vector machine (RVM) regression, and support vector machine (SVM) regression. To handle the problem of ambiguity due to monocular point of view, the method is embedded in a regressive tracking framework using dynamics from the previous estimated state. Mean angular errors of 4-6 degrees are obtained for a variety of standing postures involved in walking motion.

2.2.3.2 Multiple Cameras

To improve the accuracy of the 3D measures and to solve self-occlusion ambiguities, some approaches involve more than one camera in the human posture recognition process. The same taxonomy that has been used for mono-camera can be used in this case and the existing approaches can be classified as model-based and learning-based approaches.

The model-based approaches search the parameters of the 3D human model such as its projections on the different image planes fit with the input images.
In [Delamarre and Faugeras, 2001], the authors propose a 3D human model constituted of truncated cones, spheres and parallelepipeds to fit with a person observed by three cameras. The model has 22 degrees of freedom corresponding to the articulations of the body model. Their algorithm computes the force necessary to match the contour of the projected 3D model on the image plane with the contour of the detected person (figure 2.10). The posture at time $t-1$ initialises the pose at time t. Moreover the authors assume that the initial pose is known.

In [Mittal et al., 2003], the authors describe a system to estimate human postures from multiple views. The silhouettes of the persons are extracted and body part primitives are computed based on the study of the curvature of the boundary: the silhouette is cut according to a short-cut rule. The obtained 2D body parts are then matched across views using epipolar geometry to yield 3D body parts. A 3D model composed of cylinders is then computed according to the determined 3D body parts.

The learning-based approaches learn characteristics about the human postures.
In [Iwasawa et al., 1999], the joints (elbows and knees) locations are represented by a linear combination of the centroid, head, and hands/feet positions. Training

Figure 2.10: Continuous white lines are the contour of the silhouette of the projected 3D model. Dotted white lines are those of the real silhouette. Red lines are the forces necessary to match the contours [Delamarre and Faugeras, 2001].

images are manually annotated to be used in a genetic algorithm. The authors estimate the human posture with three cameras. The cameras are optimally placed: one observing the person from the front, another from the side and the last from the top. The orientation of the upper body (above the waist) is computed based on the statistical moments. According to this orientation a heuristic analysis of the contour is made to determine salient points of the body: head, hands, feet and the different joints in each image based on a genetic algorithm. Using the camera parameters and the geometrical relationships between the three cameras the 3D coordinates of the silhouette salient points are obtained by selecting two views. This method needs an initialisation step where the subject keeps the T-shape posture: the centroid and the salient points are stored as reference.
In [Rosales et al., 2001], the authors introduce an approach to estimate 3D body posture using three uncalibrated cameras. The approach consists in training a specialised mapping architecture (SMA) which takes as input visual features of the silhouette (the seven Hu moments) and gives as output several body posture hypotheses (the 2D locations of the body joints). The training is made with images obtained with concentric virtual cameras (intrinsic parameters are then known), where each principal axis of the cameras pass through the circle center. An expectation maximisation (EM) algorithm is used to find a self-consistent combination of hypotheses to provide the estimation of the 3D body postures and estimation of the camera parameters.
In contrast to other approaches which treat the different views as images, in [Cohen and Li, 2003], the authors work on 3D shape. They reconstruct the 3D visual hull of the detected person from the silhouettes of four synchronous views. The 3D visual hull is placed in a 3D reference form (a sphere or a cylinder, figure 2.11). The reference form is divided in several bins and the distribution of

visual hull points in each bin describes the 3D shape. This shape representation is used to train a support vector machine (SVM) allowing the characterisation of human postures from the computed visual hull.

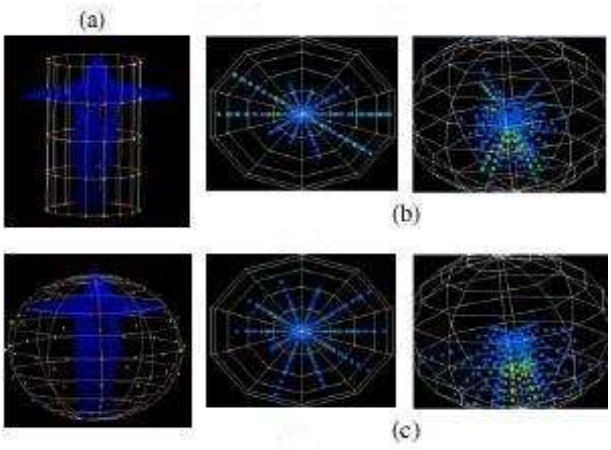

Figure 2.11: Example of shape representation associated to a cylinder and a sphere as reference form (a). In (b) and (c), the visual hull and the spherical shape representation viewed from above and the side are displayed [Cohen and Li, 2003].

2.3 Discussion

Different approaches have been presented to recognise human postures in the previous sections according to the cooperation of the studied people:

- Mechanical and physiological sensors, in particular body markers are designed for cooperative people. The techniques using such markers give accurate measures of the location and orientation of the body joints, but they are limited by the space where the actor evolves. Results obtained with motion capture can be used to tune or train the algorithms involved in the vision techniques described below.

- Vision techniques are designed for non-cooperative people. A taxonomy based on the human model used by the approach is presented:

  - 2D approaches with explicit models. These determine postures by finding the different body parts of the detected person. They are very sensitive to segmentation errors (if a body part is misdetected or not detected, the posture is not correctly recognised). Moreover, since these approaches use a 2D model, they are dependent on the camera view point.

- 2D approaches with statistical models. These approaches are designed to solve the previous problem of sensitivity to segmentation errors. A 2D model of a posture is learned with annotated data. They are well adapted for real-time processing. As with the previous approaches, they are dependent on the camera view point (they depend on the viewpoint of the learning phase).
- 3D approaches. These can be classified as model-based and learning-based approaches. Model-based approaches determine the 3D coordinates and orientation of the different body joints of a given 3D human model such as the projection of the 3D model on the image plane fit with the input image. They require the tuning of a large number of parameters (around 20 parameters depending on the quality of the 3D human model involved in the recognition process) which must respect biomechanical constraints. These parameters model the degrees of freedom of the model, in particular the articulations of the human body. Moreover, many approaches need an initialisation phase where the observed subject performs a predefined posture or suppose that the posture is known on the first frame. Learning-based approaches need to annotate manually training images, in particular the location of the different articulations. Since, these approaches work in 3D space they are partially independent from the camera view point. "Partially independent" because the problem of self-occlusion can happen (for instance one arm can be in front of the body). In order to be totally independent from the camera view point, some of these approaches use several cameras to solve ambiguity and to estimate accurately the depth (3D coordinates) of the 2D images points.

2.4 Conclusion

In this chapter, previous work on human posture recognition has been presented. The accuracy of the mechanical and physiological sensors has been shown but they are limited for applications where people are cooperative. In contrary, vision techniques are well adapted for non-cooperative persons and they are more generic (less constraint, cheaper, in a large type of applications) than approaches using non-video sensors.
As previously introduced, our objective is to propose an approach to recognise the entire human body posture by using only one static camera and in real-time. Few works address this objective. In [Zhao and Nevatia, 2004], the authors determine the postures of a walking or a running person using an articulated dynamic human model. Running and walking motion is decomposed in cycles based on several 3D motion capture data sequences. 16 cycles are identified for each of the motion. Their approach compares the 32 predicted leg motion template (models) with the detected leg using a block matching based on a optical flow algorithm (figure 2.12). In this work, only standing postures are studied.

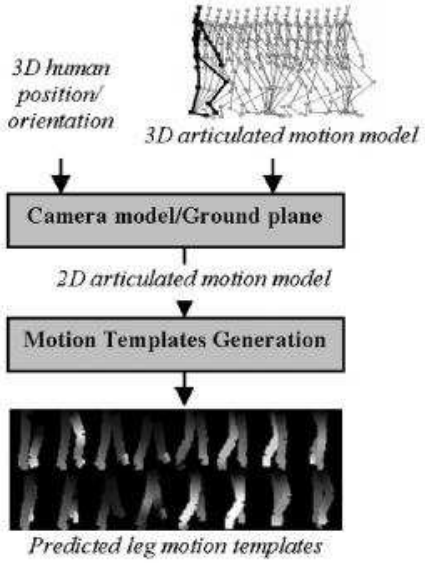

Figure 2.12: Computation of the predicted leg of walking motion based on camera model [Zhao and Nevatia, 2004].

We propose an approach inspired by this work, by combining the advantages of the 2D and 3D approaches to recognise the entire human body postures in real-time. We are interested in a set of posture which does not only contain the standing ones. The proposed approach is based on a 3D human model for achieving independency from the point of view of the camera and employs silhouette modeling from 2D approaches to provide a real-time processing. In the next chapter, an overview of the proposed approach is given.

Chapter 3

Human Posture Recognition Approach Overview

The goal of the human posture recognition approach is to provide accurate information about the people evolving in the scene to the behaviour analysis task. As seen in chapter 2, the posture recognition problem has been treated with 2D approaches and 3D approaches. Our goal is to propose a framework that takes the best of each approach. In particular, we aim at combining the computation speed of 2D approaches and the independence from the viewpoint of the 3D approaches. The objectives are presented in section 3.1 and an overview of the proposed approach is described in section 3.2. A discussion is made in section 3.3.

3.1 Objectives

3.1.1 An Approach to Recognise Human Postures in Video Sequences

The goal of this work is to propose an approach to recognise human postures in video sequences (figure 3.1) whether the person is cooperative or not (chapter 2). This approach aims at helping the behaviour analysis task in order to refine the analysed behaviour. This approach follows the spatio-temporal analysis task in the treatment chain. Indeed, the filtering posture task needs information about the previous postures of the recognised person. This information is given by the tracking task. The filtered postures are then provided to the behaviour analysis task.

3.1.2 Constraints on the Posture Recognition Approach

To propose a generic approach, which can be used in different applications such as video surveillance or aware house (also called home-care), several constraints have been identified.

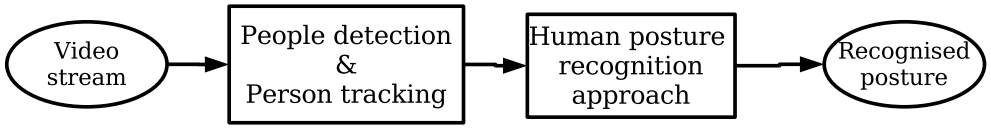

Figure 3.1: The posture recognition approach provides information about the postures of people evolving in a video stream.

Real-time

One of the most important constraint is the real-time processing. A video understanding system must be able to interpret human behaviour in real-time in order to raise an alarm (or other action) as soon as an event is detected. A frame-rate of 10 images by second is usually sufficient to efficiently monitor daily elderly people activities in home-care applications. This frame-rate is usually sufficient to raise an alarm with an acceptable response time. To be efficient in term of delay, the human posture recognition algorithm must provide in real-time the posture of the people evolving in the scene to the human behaviour analysis task.

Independence of the proposed approach from the view-point

The camera view-point defines how a person appears on the image plane depending on the camera position and on the orientation of the person. Therefore depending on the position of the camera and the orientation of the person, the same posture can appear differently on the image plane. The proposed approach must recognise posture from any position of the camera, and for any orientation of the person.

Automated Approach

Another important point of the approach is the need of an automated process. A video understanding system generally aims at computing pertinent information for an operator. The operator should not have to interact with the recognition process which should work automatically. Moreover, the approach must be able to recognise postures of non cooperative people. For instance, the people should not be assumed to be observed from an optimal point of view, looking at the camera.

One Monocular Static Camera

The approach depends on the information provided by the object detection task. Our approach has to be able to provide good results whatever type of camera the system uses. A single camera is used in this work to propose a generic approach. The approach can be adapted to existing system with already installed video

cameras.

The four previous constraints are defined to justify our generic and operational approach described in the next section.

3.2 Proposed Approach for Human Posture Recognition

Given the fact that the recognition of human postures is a stage of a video understanding process, the posture recognition task has to work in collaboration with the other tasks. The input of the posture recognition task is the results of the people detection task (detection and classification of physical objects of interest), and of the spatio-temporal analysis of the people evolving in the scene:

- Results of people detection. The people detection task gives information about people evolving in the scene such as position or size. Generally, a person is represented with her/his silhouette defined as a binary image where the foreground pixels belong to the person.
- Results of the spatio-temporal analysis task. The spatio-analysis task gives the link between frames and people. A single identifier is associated to each person during a video sequence.

Moreover, a contextual knowledge base is necessary to interpret a scene. A contextual knowledge base may contain information about the context of the scene such as:

- The position of the contextual objects (furniture such as chair or desk).
- The localisation of the zones of interest (forbidden zone, safe zone, etc...).
- The characteristics of the camera (the calibration matrix and the position of the camera).
- The semantic associated to each contextual object to be used in particular by behaviour analysis to infer high level scenario.

The proposed human posture recognition approach only need information about the camera, in particular the calibration matrix and its position in the scene.

As shown in figure 3.2, the approach can be described in two inter-connected tasks:

- Task 1: The posture detector.
  This task recognises the posture of a person isolated in the scene. The task combines the 2D techniques and the use of 3D posture models to generate silhouettes. These generated silhouettes are then compared with the silhouette of the detected object based on 2D techniques, to determine the

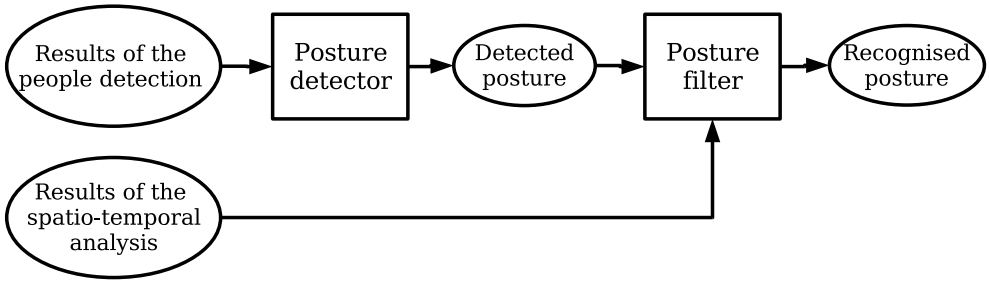

Figure 3.2: The proposed posture recognition approach is composed of two inter-connected tasks.

posture. This task uses a contextual knowledge base to generate a virtual camera (by using camera calibration information) and uses information computed by the people detection task to generate silhouettes. Thus the virtual scene is observed by a virtual camera which has the same characteristics than the real one. 3D postures are used to acquire a certain independence from the point of view, whereas the 2D techniques helps to maintain low processing time.

- Task 2: The posture filter.
  The temporal coherence of posture is exploited in this task to repair posture recognition errors from task 1. The identifier of the recognised person is used to retrieve the previous detected postures. These postures are then used to compute the filtered posture (i.e. the main posture) by searching the most frequent posture for a certain period of time. This task provides stable recognised postures to analyse the actions of the people observed in the scene.
  The filtered postures are then provided to the human behaviour analysis task.

3.2.1 3D Posture Avatar

In order to provide a posture recognition algorithm independent from the camera view point, a 3D posture avatar has been introduced. It is a 3D virtual avatar representing a given posture. The 3D posture avatar is composed of a 3D human model, a set of joint parameters and body primitives (figure 3.3).

The 3D human body model is represented with body parts and joints (the articulations of the body). The 3D human body model defines the relation between the different body parts (e.g. the left forearm is connected to the left upper arm by the left elbow articulation).
The realism of the 3D body model depends on the refinement of the body parts and the quantity of joints. The quality of a body part is defined in terms of its representation choices. For example, the forearm can be represented by a cylinder

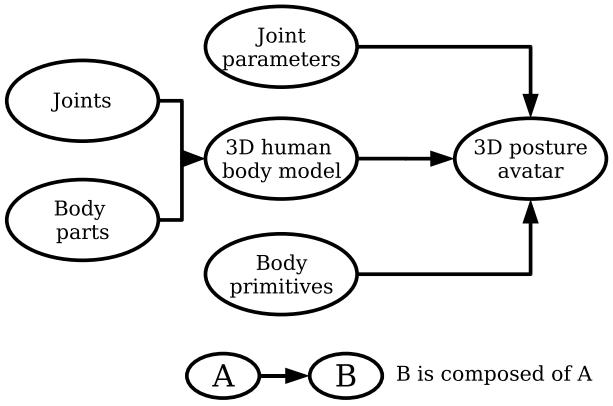

Figure 3.3: A 3D posture avatar is composed of a 3D human body model, joint parameters and body primitives.

or by a set of polygons. The number of joints defines the degree of freedom of the 3D human model. Selection of the number of joint is determined as a compromise between accuracy of the 3D model and computational time.
A set of postures of interest is chosen to cover the possible applications. These postures are hierarchically classified from general postures to detailed postures. A 3D posture avatar is defined for each of the postures of interest by associating a predefined set of joint parameters to the 3D human model.

3.2.2 The Proposed Hybrid Approach

We proposed a hybrid approach combining 2D techniques with the use of 3D posture avatars. Human posture recognition algorithm determines the posture of the person using the corresponding 2D moving regions (the silhouettes) and its 3D positions. This algorithm is composed of three main steps:

- Silhouette generation (figure 3.4): the 3D posture avatar silhouettes are generated by projecting the corresponding 3D posture avatar on the image plane. The 3D avatars are observed with a virtual camera defined with information of the contextual knowledge base (the camera parameters). Each 3D posture avatar is placed in the 3D scene according to the object detected by the people detection task. The 3D position of the detected person is computed with the calibration matrix and the silhouette. Then the avatar is oriented for different angles to generate different possible silhouettes.

- The silhouette comparison (figure 3.5): the detected silhouette and generated silhouettes are compared to obtain an estimation of the posture of the detected person. The comparison is made with classical 2D techniques (geometric representation, Hu moments, skeletonisation, horizontal and vertical projections). The choice of the 2D technique depend on the quality of the silhouette and of the objectives of the application.

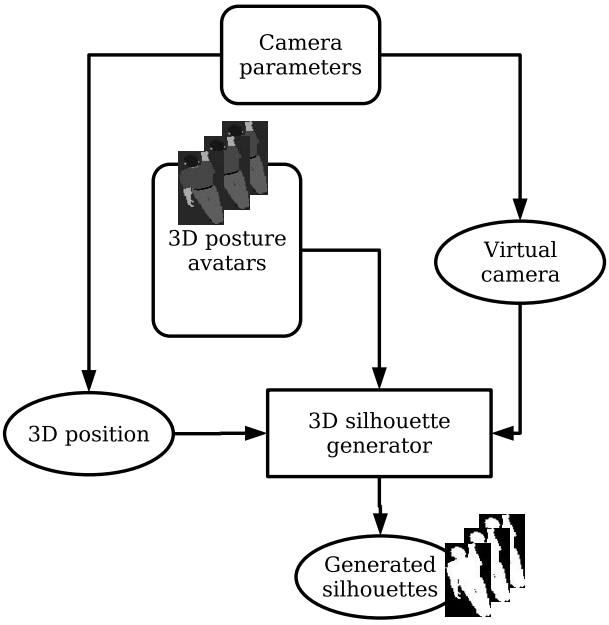

Figure 3.4: 3D posture avatar silhouettes generation depending on the detected person.

- The temporal coherency (figure 3.6): the recognised posture is then compared with the previously recognised postures to verify the temporal coherency, and corrections are made if necessary to obtain a filtered posture. The filtered postures are the input of the human behaviour analysis task.

The 3D posture avatars are involved in the recognition process to acquire a certain independence from the point of view. 2D techniques consist mainly in detecting the moving regions corresponding to detected people and match the silhouette generated from 3D avatar. These techniques enable the global posture recognition process in real-time.

3.3 Discussion

We have presented in this chapter an overview of the proposed approach to recognise the posture of whole human body. Human posture recognition is a step of a video interpretation process as seen in section 1.2. In particular, the recognition process needs information provided by the people detection and tracking tasks. This process also provides the people posture to the behaviour analysis task.
The approach must comply with several constraints: real-time processing, independence from the camera view point, fully automated approach and the use of only one static camera (section 3.1).
The proposed approach is designed to take into account these different constraints and proposes a generic framework to design an operational component

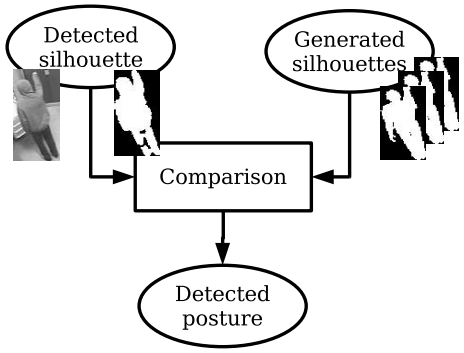

Figure 3.5: Comparison of the detected silhouette with the generated silhouettes.

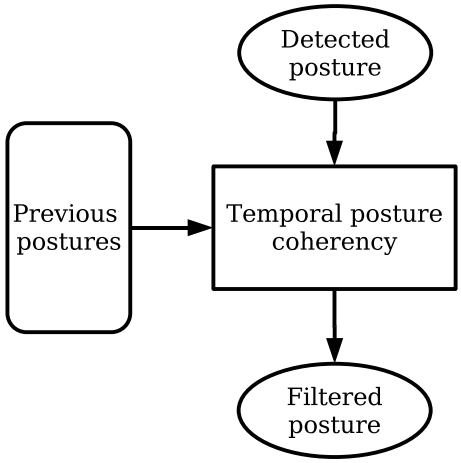

Figure 3.6: Detected posture is compared with previous detected postures to verify the temporal coherency.

(section 3.2):

- The real-time processing is achieved by proposing an hybrid approach which combines 2D techniques and the use of the 3D posture models. The silhouettes are compared with computationally fast 2D techniques.
- The 3D posture models are introduced to acquire a certain independence from the camera point of view. By using 3D posture avatars, a silhouette can always be obtained for any type of posture, any person position and orientation, and any camera position.
- A contextual knowledge base is used in the recognition process to define a virtual camera and to compute the 3D position of a person in the scene. This contextual knowledge base contains properties of the real camera and calibration information. The 3D posture avatars are then placed according

to the object detected by the people detection task in the virtual scene and observed from the point of view of the real camera. Thus the proposed approach is fully automated.

- The approach is able to work with a single static camera thanks to camera information (calibration matrix) provided in the contextual knowledge base.

The overview of the proposed approach is shown in figure 3.7.
In the next chapters, the proposed approach for human posture recognition is described in details. In chapter 4, the 3D posture avatars are defined and the 3D human body model is described in details. The body primitives involved in the body parts modeling is studied and their implementation is described. The articulation of the body parts is examinated. The implementation of the 3D posture avatars is explained.
In chapter 5, the proposed hybrid approach to recognise posture of the whole human body is presented. We will see how the 3D posture avatars can be performed in a real-time recognition process. 2D techniques used to represent the silhouettes are studied, and the temporal coherency of the postures to improve the recognition is presented.
The approach is evaluated in chapter 6.

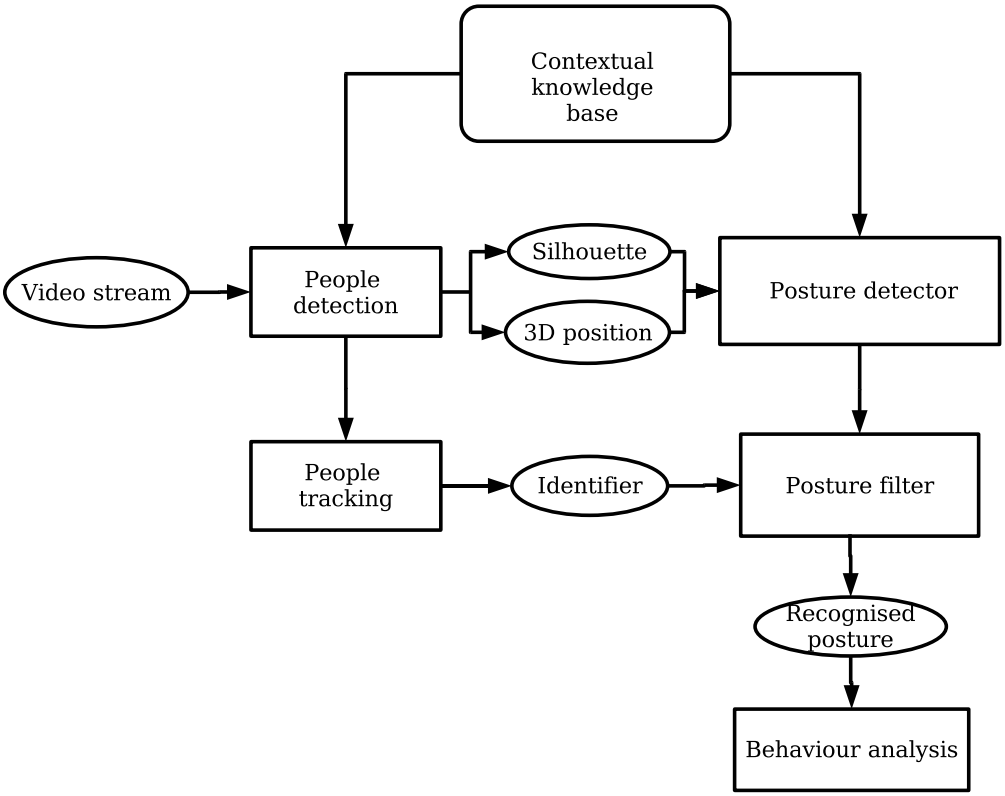

Figure 3.7: Overview of the proposed human posture recognition approach

Chapter 4

3D Posture Avatar

4.1 Introduction

As seen in the overview (section 3.1), the proposed human posture recognition approach involves the use of a 3D posture avatar. The 3D posture avatar is built through a 3D human body model. The 3D human body modeling has been improved during the past decade in part due to the increase of computer power as well as application needs. 3D human body models are mainly used in 3D human animations. An overview of the existing 3D human body models is given below: 3D human animations appear in different applications:

- Film industry. Virtual character (i.e. avatar) are widely used in film industry (Final Fantasy-Advent Children, Fantastic Four and a many other films)
- Real-time applications which need a real-time interaction between the user and the virtual character such as computer games or surgery applications.
- Simulation. Virtual characters are used in simulation applications for ergonomic study in the automotive industry as well as sports applications.

The 3D human body models previously proposed can be classified as one of four categories:

- stick figure models,
- surfacic models,
- volumetric models,
- multi-layered models.

The first proposed 3D human model was based on **stick** representation. Such models are represented by a set of hierarchical sticks connected by joints, as shown in figure 4.1. The sticks roughly represent the main bones of a human body. This kind of model is not realistic since it does not take into account the deformation

of the body during animation.

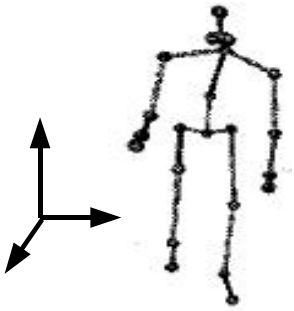

Figure 4.1: Stick figure model used in [Barron and Kakadiaris, 2003] to estimate posture.

Surfacic model improves the stick model by proposing a new layer to manage the deformations of the body: the skin layer. The skin surrounds the previous sticks. The skin models the body deformation due to the animation of the sticks. The skin can be represented by points and lines, polygons and curved surface (Bezier, B-spline). The model is realistic but there is a problem of surface deformation at joints of the body. Indeed, these models do not take into account the surface deformation due to a configuration change in the body joints. An example of such a model can be found in figure 4.2 where the body primitives are polygons composed by a set of facets.

Figure 4.2: Surfacic model developed in this work and example of the set of facets representing the chest and the right collar body parts.

In a **volumetric** model, simple geometric primitives are used to model the different body parts: cylinders, spheres, truncated cones (figure 4.3). This model is less realistic than the previous one since geometric primitives are less accurate to represent body parts than the surfacic model. Volumetric model is well adapted

for real-time processing. It is thus generally used for computer vision applications.

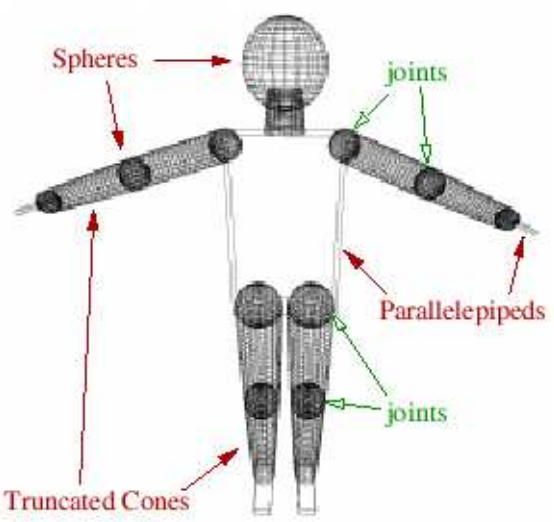

Figure 4.3: 3D model involved in [Delamarre and Faugeras, 2001] to track people in several views.

The **multi-layered** model is generally composed of three layers: the skeleton, the muscles and the skin. The skeleton gives the relation between the different body parts and animates the model. The muscle layer and the skin layer model the deformation of the body due to the animation. An example of such a model is given in figure 4.4 based on metaball representation. A physical engine is generally associated to the model to handle the deformation of the skeleton and its impact on the muscle and skin layers. This model has a lot of parameters difficult to control which are hardware dependent.

The choice of the 3D human body model depends on the realism and the purpose of the application. Surfacic and volumetric models are generally used in computer vision applications. The realism of such a 3D human body model depends on two principal characteristics:

- the realism of the body primitives (visual realism),
- and the number of joints (animation realism).

Body parts have been widely represented with 2D ribbons such as in the Cardboard model [Ju et al., 1996]. But, now the most used representation is based on 3D volumetric models which can either be geometric or surface-based (polygons). Geometric representation can be based on sticks [Barron and Kakadiaris, 2000], polyhedron [Yamamoto et al., 1998], cylinders [Cohen et al., 2001] or super quadrics [Gavrila and Davis, 1996]. In [Delamarre and Faugeras, 2001], the

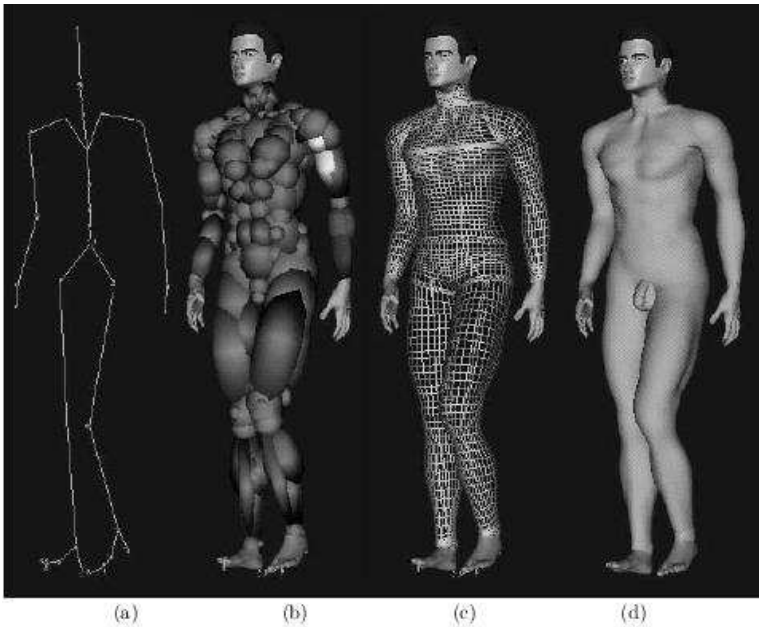

Figure 4.4: The layered model used in [D'Apuzzo et al., 1999]: (a) the skeleton, (b) ellipsoidal metaballs used to simulate muscles and fat tissues, (c) polygonal surface representation of the skin, (d) the shadered rendering.

authors use truncated cones to track individuals in multi-views. The body primitives represented by polygons are more realistic than ones with the geometric representation.
In our case, we want to recognise human postures. The most adequate model is then a surfacic model which is the best compromise between realism and computational speed. This model can have realistic body parts since the body parts are modeled with facets. Moreover, surfacic model only needs a classical computer with no dedicated hardware to be drawn.

Another important consideration in 3D human body modeling is the number of rotation parameters associated with the joints. This number defines the degrees of freedom (DOF) of the 3D human body. The degrees of freedom are related to the realism of the 3D human body animation. A computer vision application often needs less than 30 DOF. In [Delamarre and Faugeras, 2001] or [Gavrila and Davis, 1996], the authors use only 22 DOF, by considering only a subset of the articulations of the total human body. On the contrary, a computer graphics application may require more than 50 DOF. Aubel et al. [Aubel et al., 2000] use 68 DOF corresponding more to the real number of human joints, plus a few global mobility nodes that are used to orient and locate the virtual human in the world.

The rotation parameters of the joints are generally represented by the Euler angles. According to the *Euler's rotation theorem*: an arbitrary rotation may be described by only three parameters which are three angles along the axis of a coordinate system. This representation can create singularities. Indeed, Euler angles lead to three problems in the case of 3 DOF joints:

- This representation does not reflect reality. The rotation with 3 Euler angles corresponds to three successive rotations around the classical axes (x, y, z), whereas in reality the rotation is achieved directly and not sequentially.
- Euler angles are mathematically flawed. The Gimbal lock singularity can happen when Euler angles are used in the case of 3 degrees of freedom. Since thhe rotations in the Euler representation are done with respect to the global axis, a rotation in one axis can be confused with another axis. Then a degree of freedom is lose. If the rotation in the Y axis rotates a vector (parallel to the X axis) then the rotated vector is parallel to the Z axis. Any rotation in the Z axis would have no effect on the vector: this is called the Gimbal lock problem.
- Several Euler angle representations can be associated to a single 3D rotation.

When necessary, the rotation parameters can be represented with quaternions to solve these problems. A quaternion can be defined as a rotation in a 4D world, represented by four values: three define rotation axes and one defines a rotation angle. A conversion is possible between Euler angles representation and quaternion representation (cf. appendix C). Euler angles are widely used because they are much easier to read and conceptualise than a quaternion. Euler angle representation is sufficient to represent static postures since no animation is needed.

A 3D human body model is characterised by its body primitives and its degrees of freedom. These characteristics depend on the application purpose. A computer vision applications often require high computational speed and thus use few joints. On the other hand, a computer graphics application may use many joints to obtain a more realistic 3D human body model.
The next section describes the 3D human body model involved in our human posture recognition approach.

4.2 3D Human Body Model

4.2.1 Standards on 3D Human Body Model Representation

The human body has been strongly studied in the last centuries. Each body part, articulation, as well as many other small parts have medical terms. With the increasing interest in 3D graphics over the past decade, there has also been an important emergence of character modeling software to create and animate

3D human body. The lack of a skeletal structure often forces animation companies and motion capture studios to develop their own proprietary solutions. H-anim (Humanoïd Animation specification) [H-Anim, 2006] proposes a VRML-based specification to represent a 3D human body model.
VRML (Virtual Reality Modeling Language), [VRML, 2006] is a 3D graphics language to represent 3D virtual worlds. It is not a programming language since (similarly to HTML language) a VRML file contains information to visualise the different elements of the scene (shape, light, 3D position, texture, sound, etc...).
H-anim design goals are:

- compatibility: humanoïds should work with any VRML browser
- flexibility: no assumptions are made about the types of applications that will use humanoïd
- simplicity: the specification contains only the necessary information to model and animate a 3D human body model.

Up to now, H-anim has proposed three specifications based on the advanced of the VRML language and the introduction of new features.
H-anim 1.0 specification is based on VRML 2.0. A 3D human body model is represented with a set of hierarchical nodes. Each node contains several features:

- the rotation center of the joint
- other joint nodes linked to this joint
- a stick node which is the body part associated to that joint (3D geometry, color, texture)
- hints for inverse kinematics systems (upper/lower joint limits, orientation of the joint limits, stiffness/resistance values)

H-anim 1.1 formalism extends the previous specification to take into account the deformations of the model during the animation. Site nodes are added to define specific locations relative to the body primitive. Displacer nodes are also defined to specify which vertices within the link corresponds to a particular configuration. H-anim 200x makes small changes to best support deformation engines and animation tools. In the MPEG-4 standard, the face/body definitions are based on the H-anim specifications.

4.2.2 Proposed 3D Human Body Model

We propose a 3D human body model inspired by the H-anim specification. Below, the body parts and the joints of our 3D human body model are described. We then explain how to compute a 3D posture avatar.

The joint nodes of our 3D human model are composed of:

- *body_parts*: the two body parts associated to the joint
- *default_pos*: the default position of the associated body part
- *rot*: the rotation parameter of the associated body part
- *rot_min*: the lower joint limit
- *rot_max*: the upper joint limit

We define 9 joint nodes: an abdomen joint, left elbow, right elbow, left knee, right knee, left shoulder, right shoulder, left hip, right hip plus a special joint: the pelvis. The function of the pelvis joint is to position the 3D human body model in the 3D space. The pelvis has the same characteristics as other joint nodes plus the *pos* parameter to translate the 3D human body model. We do not use all the possible joints because we define a 3D human body model for a computer vision application. Because the application is constrained by real-time processing, a tradeoff must be chosen between realism and processing time. However, the chosen nodes are sufficient in quantity to represent all the postures we have planed to recognise. Our 3D human body model is composed of 20 body parts: hair, waist, left thigh, right thigh, left upper arm, right upper arm, left shin, right shin, left hand, right hand, left forearm, right forearm, left foot, right foot, left collar, right collar, neck, head, chest, and abdomen. Some body parts that are directly connected without a joint node are defined such as head and hair. This cue gives the ability to change the body primitive which models the body part. A body primitive is required to visualise the body parts of our 3D human model which is shown in figure 4.5. For instance, the hair body part can be short hair as well as long hair. A polygon-based representation is chosen for two main reasons. The first is that the processing time for polygon-based primitives is similar to the processing time for cylinder or other classical geometric primitives with a classical computer. The second reason is that we plan to use this realistic 3D human model to generate synthetic data close to the real human being. Since the proposed human posture recognition approach is based on the comparison of silhouettes as explained in chapter 3, realistic synthetic silhouettes have to be generated.
Each body part is composed of vertices (2D facets which live in 3D space). These facets can either be a triangle (composed of three 3D points) or a quadrilateral (composed of four 3D points). A 3D point is defined by:

- the 3D space coordinates of the point: $[v_x, v_y, v_z]^T$,
- the color associated to the point: $[v_r, v_g, v_b]^T$ corresponding to the red, green and blue values
- the normal vector: $[n_x, n_y, n_z]^T$ with $n_x^2 + n_y^2 + n_z^2 = 1$. This vector gives information to display light depending on its direction with light sources.

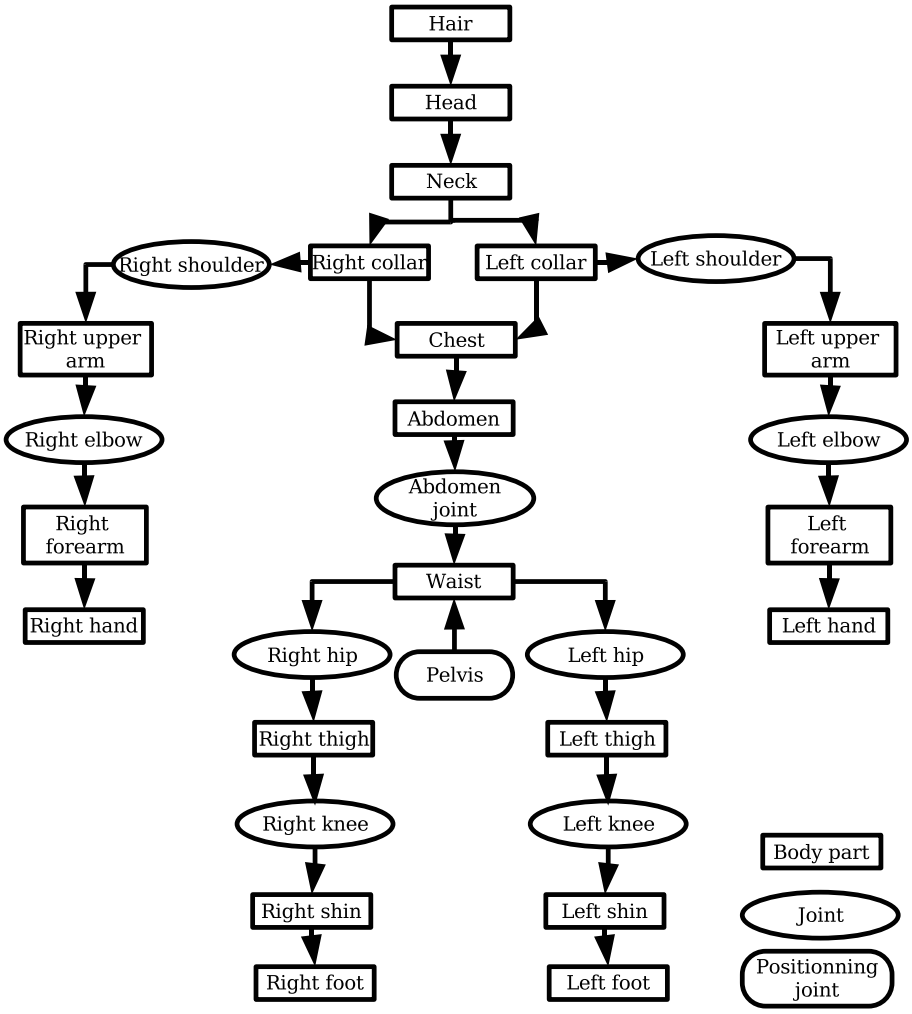

Figure 4.5: Body parts and joints of our 3D human model

Color and normal vector are not important cues for the silhouette extraction, they are only used for a displaying purpose.

Our 3D human model is defined by 10 joints and 20 body parts. The 10 joints are sufficient to model the postures of interest and move it in the virtual scene. The body primitives which represent the different body parts are polygon-based to obtain realistic synthetic data.

4.3 Posture Avatar Generation

Our 3D human body model has been defined by proposing a set of joints and body parts. Now, we use this 3D human body model to generate our 3D posture avatar.

A 3D posture avatar corresponds to the 3D human body plus a set of joint parameters and body primitives. The 3D human body is animated with the joint parameters and visualised with the body primitives. We have defined a 3D engine to animate a 3D human body. Moreover, a tool has been developed to animate the 3D human body model. Each of the articulations can be selected, and the body primitive associated to this articulation can be rotated (around the articulation). The parameters can be saved to obtain the joint parameters corresponding to a 3D posture avatar.

The parameters of each joint are the three Euler angles α, β and γ. Since some articulations have only one degree of freedom (the knees), the 3D posture avatar is represented by a set of 23 parameters. The articulation must respect biomechanical constraints (see table 4.1).

	α $\alpha_{min}/\alpha_{max}$	β β_{min}/β_{max}	γ $\gamma_{min}/\gamma_{max}$
Abdomen	-15/90	-15/15	-30/30
Left shoulder	-45/45	-160/15	-90/90
Left knee	0/120	0/0	0/0
Left elbow	-100/0	0/135	-100/5
Left hip	-90/30	-90/90	-30/90
Right shoulder	-45/45	-15/160	-90/90
Right knee	0/120	0/0	0/0
Right elbow	0/100	-135/0	-5/100
Right hip	-90/30	-90/90	-90/30

Table 4.1: Biomechanical constraints of our 3D human model: minimum and maximum Euler angles of each articulation (in degrees).

The proposed 3D engine relies on the fact that when a body part is moved all the subparts are also moved. For instance, if the left upper arm is moved, then the left forearm and the left hand must follow the corresponding movement.
As seen previously, the different body primitives are composed of facets, and thus moving a body part is equivalent to move each facets which composed the body part. Rotation and translation transformations are applied to points constituting the different facets. These transformations are represented by 4x4 matrices for homogeneous coordinates. The rotation around the X axis for an angle α is given by the following matrix:

$$M_X(\alpha) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos(\alpha) & -\sin(\alpha) & 0 \\ 0 & \sin(\alpha) & \cos(\alpha) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

The rotation around the Y axis for an angle β is given by the following matrix:

$$M_Y(\beta) = \begin{bmatrix} \cos(\beta) & 0 & \sin(\beta) & 0 \\ 0 & 1 & 0 & 0 \\ -\sin(\beta) & 0 & \cos(\beta) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

The rotation around the Z axis for an angle γ is given by the following matrix:

$$M_Z(\gamma) = \begin{bmatrix} \cos(\gamma) & -\sin(\gamma) & 0 & 0 \\ \sin(\gamma) & \cos(\gamma) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

The translation by a vector $[x, y, z]^T$ is represented with the following matrix:

$$M_T\left([x, y, z]^T\right) = \begin{bmatrix} 1 & 0 & 0 & x \\ 0 & 1 & 0 & y \\ 0 & 0 & 1 & z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

The order of the computation of the transformations is important: the transformation $M_X M_T$ is different from the transformation $M_T M_X$. In the case of human body animation, each body part has to be rotated according to the corresponding joint parameters. Considering a given body primitive B, three information are available:

1. the set of facets and the set of points which composed the body parts: $F = \{F_i\} = \{\{P_j\}\}$
2. the default position of the body part $default\_pos = [x, y, z]^T$, according to the origin of the world reference
3. the joint parameters, $rot = [\alpha, \beta, \gamma]^T$ defining how the body part has to be rotated

Each point P_j are rotated in three steps:

- first the point is translated to the origin with the matrix $M_T(default\_pos)$
- second the point is rotated around the X axis, then around the Y axis and finally the Z axis with the matrix $M_Z(\gamma)M_Y(\beta)M_X(\alpha)$
- third the point is translated to its original location with the matrix $M_T(-default\_pos)$

Each point P of the different body primitives are thus moving in 3D space by applying the transformation $P' = M_T(-default\_pos)M_Z(\gamma)M_Y(\beta)M_X(\alpha)M_T(default\_pos)P$ to obtain the

new coordinates P' of the considered point.
Since the human body is articulated, body parts must take into account the movement of their parents. We call parent of a given body part B, the body parts which influence B. Then when a body part is moved, the transformation due to the parents and characterised by a 4x4 matrix M is also applied as shown in algorithm 1.

Algorithm 1 $move(B, M)$

for all $F_i \in F$ **do**
  for all $P_j \in F_i$ **do**
    $P'_i = M_T(-B.default\_pos)M_Z(B.\gamma)M_Y(B.\beta)M_X(B.\alpha)M_T(B.default\_pos)MP_i$
  end for
end for

Moreover, the algorithm which moves an entire human model is given in algorithm 2.

Algorithm 2 $moveWholeBody()$

$move(waist, M_T(pelvis.pos))$ {the translation of vector *pelvis.pos* position the 3D avatar in the virtual scene}
$moveUpperBody(M_T(pelvis.pos))$ {described in algorithm 3}
$moveLeftLeg(M_T(pelvis.pos))$ {described in algorithm 4}
$moveRightLeg(M_T(pelvis.pos))$

Algorithm 3 $moveUpperBody(M)$

$move(abdomen, M)$
$M_1 = M_Z(abdomen\_joint.rot.\gamma)M_Y(abdomen\_joint.rot.\beta)M_X(abdomen\_joint.rot.\alpha)$
$move(hair, M_1M)$
$move(head, M_1M)$
$move(neckM_1M)$
$move(chest, M_1M)$
$move(rightcollar, M_1M)$
$move(leftcollar, M_1M)$
$moveLeftArm(M_1M)$ {described in algorithm 5}
$moveRightArm(M_1M)$

The implementation has been made with the Mesa Library [Mesa, 2006]. Mesa is a 3D graphics library with an API (Application Programming Interface) which is very similar to OpenGL library [OpenGL, 2006]. We used Mesa because it is based on C language and well adapted to real time tasks. Details on the implementation are given in appendix A.
We have adapted the body primitives defined in SimHuman [Vosinakis and Panayiotopoulos, 2001] to our human body avatar to model

Algorithm 4 $moveLeftLeg(M)$

$move(leftthigh, M)$
$M_1 = M_Z(left\_knee.rot.\gamma)M_Y(left\_knee.rot.\beta)M_X(left\_knee.rot.\alpha)$
$move(leftshin, M_1M)$
$move(leftfoot, M_1M)$

Algorithm 5 $moveLeftArm(M)$

$move(leftupperarm, M)$
$M_1 = M_Z(left\_shoulder.rot.\gamma)M_Y(left\_shoulder.rot.\beta)M_X(left\_shoulder.rot.\alpha)$
$move(leftforearm, M_1M)$
$M_2 = M_Z(left\_elbow.rot.\gamma)M_Y(left\_elbow.rot.\beta)M_X(left\_elbow.rot.\alpha)$
$move(lefthand, M_2M_1M)$

our body primitives.
The case where all the joint parameters are null corresponds to the T-shape posture: the person is standing with the two arms up.
The 3D posture avatar is defined by a set of 23 parameters, which are the Euler angles of the joints of the 3D human body model.

4.4 Postures of Interest

We have defined a generic 3D posture avatar. Now, we are describing which postures we want to recognise. There is almost an infinity of postures due to the complexity of the human body.
In the literature, the main postures used are standing, sitting and lying postures [Panini and Cucchiara, 2003] [Haritaoglu et al., 1998a] which usually are sufficient to interpret the behaviour of persons in a video sequence. A granularity in our postures of interest is introduced. This granularity depends on the accuracy of the recognised posture needed by the application. The general postures and the detailed postures are then defined. Detailed postures are subclasses of the corresponding general posture. We define four general postures: standing, sitting, bending and lying, and eight detailed postures are associated: standing with one arm up, standing with arms along the body, T-shape posture, sitting on a chair, sitting on the floor, bending posture, lying with spread legs and lying with curled up legs. The parameters of the posture model are defined to represent each of these postures. We can see, for example, the parameters of the posture model corresponding to sitting on the floor posture in table 4.2. The associated posture corresponding to the 3D man model can be seen in figure 4.6. This set of postures of interest can be modified by adding or removing postures according to the need of the application.

The parameters which characterise the posture models are independent of the 3D human body parts. The joint parameters can be used with different body

joints	α	β	γ
abdomen	0	0	0
left_elbow	0	-88	-14
left_knee	110	0	0
left_shoulder	0	0	-82
left_hip	-144	-6	0
right_elbow	0	86	16
right_knee	112	0	0
right_shoulder	0	0	82
right_hip	-144	0	0

Table 4.2: Euler angles (in degree) for the different joints of the posture model for sitting on the floor posture

Figure 4.6: 3D model of sitting on the floor posture

primitives (different size, scale) to represent the same 3D posture avatar. For example, only the rotation of the left shoulder is sufficient to represent the standing posture with left arm up. These postures correspond to the main postures concerning targeted applications. Some postures are dependent on the 3D human avatar, in particular on the size of the body primitives. For example, the posture touching the nose with the left hand is dependent on the size of the arm primitives: the rotation angles of the different articulations (shoulder and elbow) will be different in function of the lengths of the forearm and upper arm. Moreover, the proposed 3D human body model cannot represent facial expression such as smile, and more generally postures which need complex deformation for given body primitives. A complex deformation need to introduce new vertices to model it. Our human body model can deal with scale transformations of the body primitives by using the scale transformation matrix:

$$M_S\left([s_x, s_y, s_z]^T\right) = \begin{bmatrix} s_x & 0 & 0 & 0 \\ 0 & s_y & 0 & 0 \\ 0 & 0 & s_z & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

For instance a 3D man model with different corpulences and heights is given in figure 4.7.

A hierarchical representation of the postures of interest is shown in figure 4.8.

Figure 4.7: 3D model with different corpulences and heights

4.5 Conclusion

In this chapter, possible techniques for modeling a 3D human body have been presented: the stick figure model, the surfacic model, the volumetric model and the multi-layered model. Since the purpose of our application is to recognise human posture and not animation, we propose to use a surfacic human body model (as explained in section 4.2.2). Our model is composed of ten joints (the major body articulations) and twenty body parts
The 3D posture avatars have been designed to model human postures and it is composed of:

- a 3D human body model constituting of body parts and joints
- a set of joint parameters
- a set of body primitives

Euler angles are chosen as joint parameters. This is sufficient to represent the eight postures of interest we have planed to recognise. The body primitives of the model are polygons and thus the model is enough realistic to generate synthetic data, consisting of silhouettes, close to real world .
The Mesa library is used to generate a posture avatar. The generation is based the composition of translation and rotation operations. A hierarchical classification of the postures of interest has been introduced. We want to recognise four general postures and eight detailed postures.
The 3D posture avatars are generic and can deal with different types of body primitives (polygons, cylinders, etc...). The body primitives can be interpreted as a data-base containing many body parts representations chosen according to the need of the applications (in term of realism and visual representation). The 3D posture avatar cannot handle complex deformations of the body primitives to represent specific expression such as smile. But it can deal with global transformation such as scale of the body primitives.

The next chapter shows how these posture avatars are embedded in the human posture recognition task for video sequence analysis.

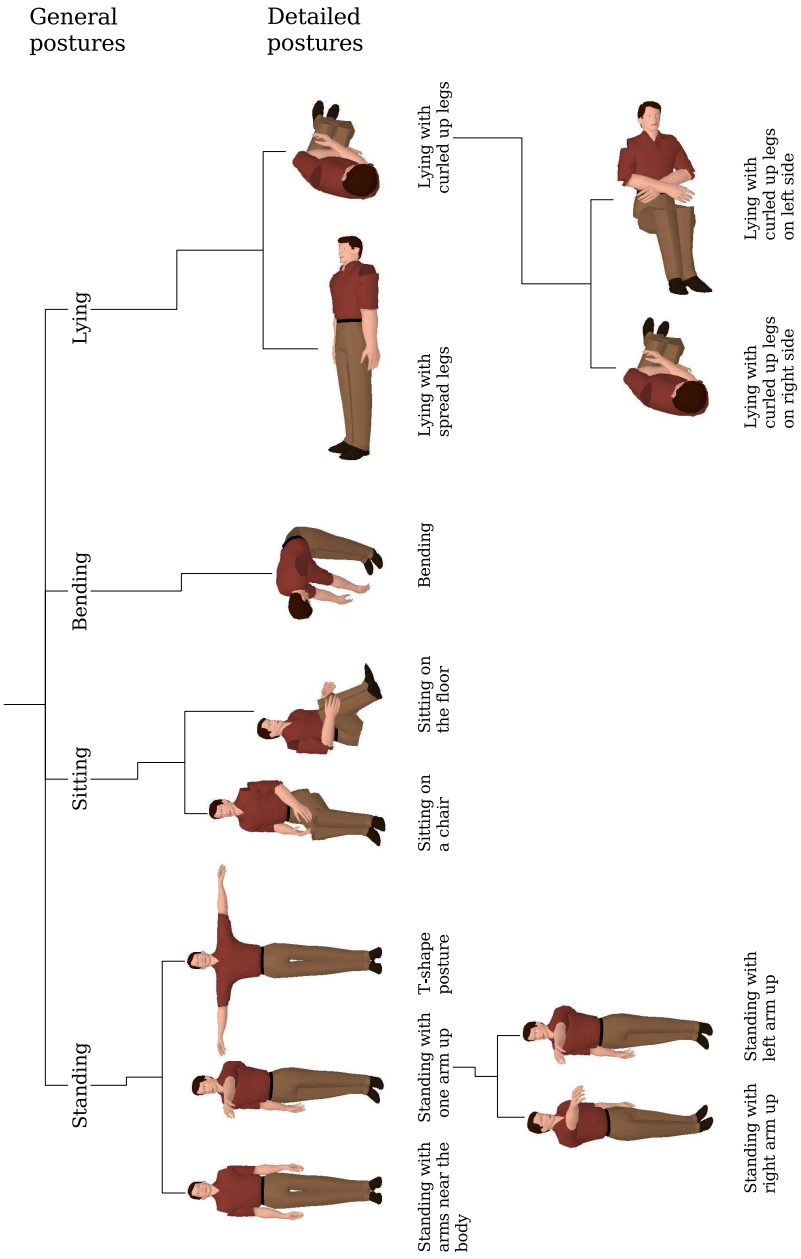

Figure 4.8: Hierarchical representation of the postures of interest

Chapter 5

The Proposed Hybrid Approach

5.1 Introduction

The previous chapter has shown how the 3D posture avatar is generated from a 3D human body model. The goal of this chapter is to show how the 3D posture avatar is embedded in the human posture recognition approach which takes advantage of 3D techniques and 2D techniques. The 3D techniques are independent from the camera point of view and the 2D techniques are well adapted for real-time processing. The approach consists in defining a data-base which contains the 3D posture avatars to be recognised. The 3D position of the detected person, the data-base of posture avatars and a virtual camera are used to generate reliable silhouettes. Then the generated silhouettes are compared with the detected silhouette (section 3.2) to determine the posture of the observed person. Finally, the detected posture is filtered throughout time to enforce temporal coherency on the postures.
Section 5.2 describes the generation of silhouettes from the 3D posture avatar through three steps: (1) a virtual camera is generated, (2) the posture model is positioned in the scene and (3) the silhouettes are generated. Section 5.3 describes different techniques to represent and compare person silhouette and section 5.4 details the temporal posture coherency mentioned above.

5.2 Silhouette Generation

In this section, the mechanism to generate silhouettes from a posture avatar is described. The posture avatar is placed in a 3D virtual scene according to a position and an orientation. The avatar is visualised with a virtual camera which gives the same point of view than the real one by projecting the 3D scene on the image plane.
Section 5.2.1 describes the creation of a virtual camera designed to have a similar point of view than the real one. Section 5.2.2 explains how the 3D posture avatars are positioned in the virtual scene and how the silhouettes are generated.

5.2.1 Virtual Camera

A virtual camera is created to visualise a virtual scene with the same point of view than the real camera. The virtual camera is defined by two different sets of parameters. The first extrinsic set defines how the virtual scene is observed: the camera transform. The second intrinsic one defines how the objects of the virtual scene are projected into the image plane: the perspective transform.

5.2.1.1 The camera transform

The extrinsic set of parameters defines the transformation from the world reference to the camera reference. This set is define by three vectors:

- $eye = [eye.x, eye.y, eye.z]^T$ is the coordinate vector of the position of the camera in the virtual world coordinate system,
- $center = [center.x, center.y, center.z]^T$ is the coordinate of a point on the axis view (usually if it is possible, it is the intersection point of the axis view with the ground plane corresponding to the point where the camera look at),
- $up = [up.x, up.y, up.z]^T$ is the direction of the up vector of the camera (the vector perpendicular to the view-axis of the camera).

The transformation from the world reference to the camera reference is characterised with the 4x4 matrix M_{CT} (the camera transform matrix):

$$M_{CT}(\beta) = \begin{bmatrix} s[0] & s[1] & s[2] & 0 \\ u[0] & u[1] & u[2] & 0 \\ -f[0] & -f[1] & -f[2] & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} M_T(-eye)$$

where M_T is the translation transformation matrix, and

$$F = \begin{bmatrix} center.x - eye.x \\ center.y - eye.y \\ center.z - eye.z \end{bmatrix}$$

$$f = \frac{F}{||F||} \tag{5.1}$$

$$up = \begin{bmatrix} up.x \\ up.y \\ up.z \end{bmatrix}$$

$$up' = \frac{up}{||up||} \tag{5.2}$$

$$s = f.up' \tag{5.3}$$

and

$$u = s.f \tag{5.4}$$

This transformation aligns the Z-axis with the view axis.

5.2.1.2 The perspective transform

The second intrinsic set of parameters define the transformation to model the distortion of the camera. this set is composed of four parameters:

- *fovy* corresponds to the angle of the field of view of the real camera as shown in figure 5.1. The Z-axis is perpendicular to the image plane with values increasing toward the viewer.
- *aspect* is the ratio between the width and the height of the image acquired by the real camera.
- *znear* defines the clipping plane ($Z = znear$) near the observer (in a virtual scene all the objects cannot be drawn, clipping planes are then defined to describe a virtual area: only the objects between the *znear* and $zfar$ planes are displayed).
- *zfar* defines the clipping plane ($Z = zfar$) far from the observer.

The perspective transformation is represented with a 4x4 matrix defined by:

$$M_{PT} \begin{bmatrix} \frac{f}{aspect} & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & \frac{zfar+znear}{znear-zfar} & \frac{2*zfar*znear}{znear-zfar} \\ 0 & 0 & -1 & 0 \end{bmatrix}$$

where $f = atan\left(\frac{fovy}{2}\right)$.

A given point P defined by

$$P = \begin{bmatrix} x \\ y \\ z \\ w \end{bmatrix}$$

is then transformed by:

$$P' = \begin{bmatrix} x' \\ y' \\ z' \\ w' \end{bmatrix} = M_{PT} M_{CT} P$$

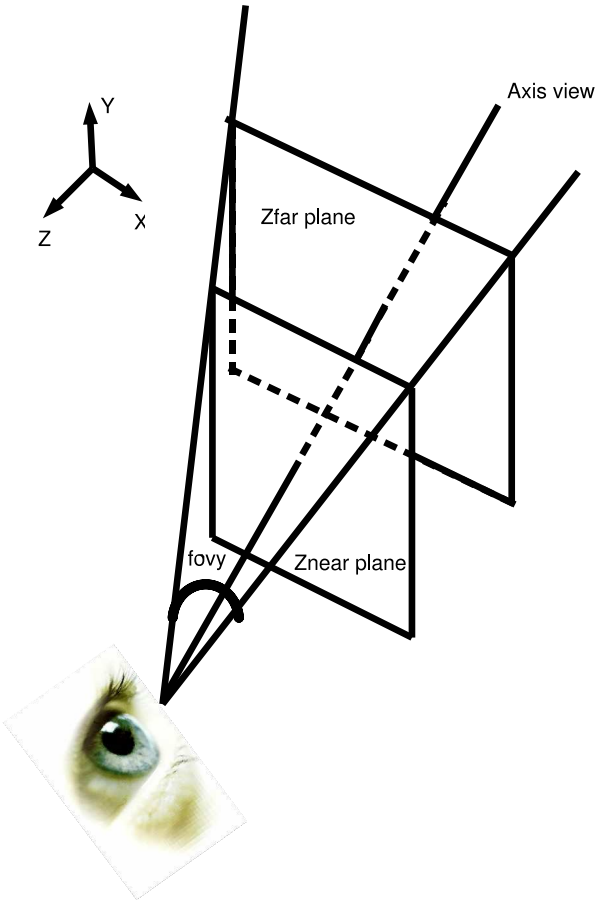

Figure 5.1: A virtual camera and its associated *znear* and $zfar$ planes, and its field of view $fovy$. Only the objects localised between the two planes are displayed.

The obtained point $\left[\frac{x'}{w'}, \frac{y'}{w'}\right]^T$ are the coordinates of the projected point P in the image plane where $[-1,-1]^T$ is the bottom left corner of the image, and $[1,1]^T$ is the top right corner of the image. $\frac{z'}{w'}$ is an important value related to the depth of the point: if $-1 \leq \frac{z'}{w'} \leq 1$ then the point P is between the clipping plane. Moreover this value is used for the Z-buffer technique as described in section 5.2.2.3 to extract the silhouette of the 3D avatar. Some details on the implementation of these transformation with the Mesa library are given in appendix A.
Once the virtual camera is designed, the 3D posture avatars are positioned in the virtual scene, as explained in the next section.

5.2.2 3D Posture Avatar Positioning

An important key of the proposed human posture recognition approach is to determine how the posture model is positioned and oriented in the virtual scene which depends on the posture avatar type. The avatar is positioned in the virtual scene by using the estimated position of the detected person. The orientation of the avatar is based on an angle estimated by trying all possible values based on a rotation step. The extraction of the avatar silhouette is based on a Z-buffer technique.

5.2.2.1 Posture Avatar Position

The 3D position of the detected person can be estimated from the detected blob and the calibration matrix associated with the video camera. The calibration matrix represents the entire transformation from the world to the image coordinates. The matrix can be determined by the internal parameters of the camera (image center, focal length and distortion coefficients) and the external parameters (position and orientation relatively to a world coordinate system). The Tsai calibration method is used to calibrate the real camera using known 2D/3D points correspondences [Tsai, 1986]. The transformation can be described by a 4x3 matrix P by considering homogeneous coordinates. The coordinates of a 3D point $[U, V, W]^T$ in the world coordinate system and its corresponding image coordinates $[u, v]^T$ (in pixel coordinate system) are related by:

$$s \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = P \begin{bmatrix} U \\ V \\ W \\ 1 \end{bmatrix} \tag{5.5}$$

with s an arbitrary scale coefficient.
Because homogeneous coordinates are considered, only 11 of the 12 matrix elements are independent. The same terms are obtained if every elements are multiplied by the same constant. Here the twelfth element (P_{12}) of the matrix P is assumed to be equal to 1.

$$P = \begin{bmatrix} P_1 & P_2 & P_3 & P_4 \\ P_5 & P_6 & P_7 & P_8 \\ P_9 & P_{10} & P_{11} & 1 \end{bmatrix} \tag{5.6}$$

The matrix P can be decomposed into $P = A\,[R|t]$ where A is a 3x3 matrix, mapping the normalized image coordinates to the retinal image coordinates. $[R|t]$ is the 3D transformation from the world coordinate system to the camera coordinate system (where R is the $3x3$ rotation matrix and t the translation vector).
By developing equations 5.5 and 5.6, the following system is obtained:

$$(P_9U + P_{10}V + P_{11}W)\,u = P_1U + P_2V + P_3W + P_4 \tag{5.7}$$
$$(P_9U + P_{10}V + P_{11}W)\,v = P_5U + P_6V + P_7W + P_8 \tag{5.8}$$

and

$$(P_9u - P_1)\,U + (P_{10}u - P_2)\,V + (P_1u - P_3)\,W = P_4 - u \quad (5.9)$$
$$(P_9v - P_5)\,U + (P_{10}v - P_6)\,V + (P_1v - P_7)\,W = P_8 - v \quad (5.10)$$

The computation of the 3D world coordinates of a point in the image is performed under the assumption that the world point belongs to a particular plane. In our case, we are interested by the position of the detected person on the ground plane ($W = 0$). Equations 5.9 and 5.10 becomes 5.11 and 5.12 respectively when W is set to 0,

$$(P_9u - P_1)\,U + (P_{10}u - P_2)\,V = P_4 - u \quad (5.11)$$
$$(P_9v - P_5)\,U + (P_{10}v - P_6)\,V = P_8 - v \quad (5.12)$$

By eliminating U in equations 5.11 and 5.12:

$$V = \frac{(P_4 - u)\,(P_9v - P_5) - (P_8 - v)\,(P_9u - P_1)}{(P_{10}u - P_2)\,(P_9v - P_5) - (P_{10}v - P_6)\,(P_9u - P_1)} \quad (5.13)$$

Then, by replacing the V values in equation 5.11:

$$U = \frac{P_4 - u}{P_9 - P_1} - \frac{P_{10}u - P_2}{P_9u - P_1}V \quad (5.14)$$

For all points $[u, v]^T$ in the image, the corresponding 3D coordinates $[U, V, W]^T$ on the ground plane ($W = 0$) can be computed according to equations 5.13 and 5.14. Depending on the type of the posture avatar to be positioned, two distinct points on the blob are considered: the middle point of the bottom of the bounding box and the silhouette centre of gravity (figure 5.2). The middle point of the bottom of the bounding box is used to position standing, bending and sitting posture avatars. It approximates the position of the detected person feet. The centre of gravity is used to position lying posture model i.e. it approximates the abdomen position of the lying person.

5.2.2.2 Posture Avatar Orientation

As previously seen, the posture recognition approach rotates the different posture avatars around one rotation axis with respect to the camera point of view in order to generate silhouettes with different orientation angles. The rotation axis is defined in function of the type of posture avatars and depends on how the avatars are positioned. The rotation axis of standing, bending and sitting posture avatar is the vertical axis passing through the feet of the person. The rotation axis of the lying postures is the vertical axis passing through the abdomen of the person.
The 0 degree orientation is chosen when a person is facing the camera from any

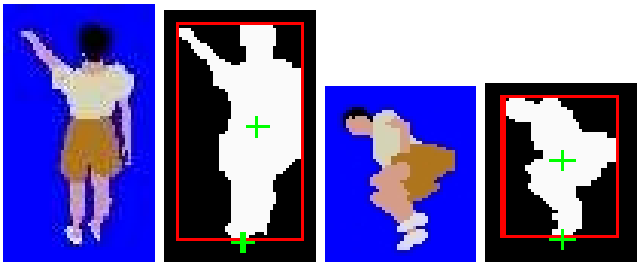

Figure 5.2: Example of 2D point location to position the 3D posture avatars. In the two first images (a standing posture avatar), the considered point is the bottom cross (middle of the bottom of the bounding box). In the two last images (a lying posture avatar) the considered point is the top cross (the centre of gravity of the silhouette).

3D position. The default orientation can be computed with the position of the person $[U, V, W]^T$ and the position of the camera $[U_C, V_C, W_C]^T$. The default orientation α is then computed by applying the Pythagore theorem as follows (figure 5.3):

$$\alpha = acos\left(\frac{U - U_C}{\sqrt{(U - U_C)^2 + (V - V_C)^2}}\right) \tag{5.15}$$

In the case where the denominator of the ration in equation 5.15 $(U - U_C)^2 + (V - V_C)^2 = 0$, that is to say the person is located at the vertical of the camera, we decide that the default orientation is equal to 0.

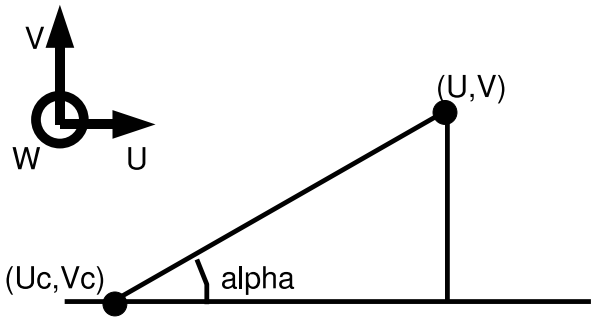

Figure 5.3: Computation of the default orientation α of a person where 0 degree correspond to a person looking at the camera. The figure represents the projection of the camera position $[U_C, V_C]^T$ (respectively the position of the person $[U, V]^T$) on the ground plane.

Thus, according to the type of the considered posture avatar, it can be correctly positioned and oriented in the scene by computing its 3D position and its default orientation. The approach rotates the posture avatars around

themselves with a given rotation step ($rotation\_step$). The algorithm positions each posture avatar at location $[U_{bb}, V_{bb}, 0]$ in the virtual scene corresponding to the middle point of the bottom of the bounding box in the image plane (or the centre of gravity of the silhouette $[U_{cog}, V_{cog}, 0]^T$ depending on the type of posture avatar). The algorithm increasly rotates the avatar by incrementing its orientation by a given rotation step until the avatar makes a complete turn. The virtual scene is observed with the previously defined virtual camera, then the silhouette is extracted for each orientation (algorithm 6).

The silhouette extraction is described in detail in the next section.

Algorithm 6 $computeAllGeneratedSilhouettes(U_{bb}, V_{bb}, U_{cog}, V_{cog}, U_C, V_C, rotation\_step)$

$all\_silhouettes \leftarrow NULL$
$\alpha_1 \leftarrow defaultOrientation(U_{bb}, V_{bb}, U_C, V_C)$ {compute the default orientation in function of the camera and avatar positions}
$\alpha_2 \leftarrow defaultOrientation(U_{cog}, V_{cog}, U_C, V_C)$
for all $P_i \in postures\_of\_interest$ **do**
  $\beta_{init} \leftarrow choose\_default\_orientation(P_i, \alpha_1, \alpha_2)$ {choose the default orientation depending on the considered posture}
  $U, V \leftarrow choose\_default\_position(P_i, U_{bb}, V_{bb}, U_{cog}, V_{cog})$ {choose the default position depending on the considered posture}
  $\beta \leftarrow 0$
  while $\beta < 360$ **do**
    $Rotate(P_i, \beta + \beta_{init})$ {rotate the i^{th} avatar}
    $Translate(P_i, U, V)$ {translate the i^{th} avatar}
    $all\_silhouettes \leftarrow all\_silhouettes, SilhouetteExtraction()$
    $\beta \leftarrow \beta + rotation\_step$
  end while
end for
$return\ all\_silhouettes$

5.2.2.3 Silhouette Extraction

The silhouette extraction algorithm is based on the Z-buffer technique. The basic idea of the Z-buffer is to store in an array the maximum Z coordinates of any feature plotted at a given location $[u, v]^T$ on the image plane. The Z-axis is perpendicular to the image plane with values increasing toward the viewer so that any point where Z coordinate is less than the corresponding Z-buffer value will be hidden behind some features which have already been plotted. So in our case, the Z-buffer is used to know if a pixel on the image plane belongs to the silhouette or to the background. For all pixels of the image, the Z-buffer value is determined with the transformation describes in section 5.2.1. The value $\frac{z'}{w'}$ is computed for each pixel of the image: if this value respects $-1 \leq \frac{z'}{w'} \leq 1$ the pixel belongs to

the silhouette of the avatar otherwise, the pixel is classified as a background pixel.

To optimise the computation time and to avoid problem of misdrawing, a double buffering technique is used. The silhouette must be extracted only when all the body parts are drawn. Indeed, if the silhouette is extracted when all the drawing operations are not done then the silhouette will be false. To respect this constraint, two buffers are used:

- the operation buffer: all the drawing operations are done in this buffer. The different body parts are sequentially drawn in this buffer according to the desired 3D position and orientation.

- the current buffer: this buffer contains the considered 3D posture avatar with all the different body parts. The silhouette is extracted from this buffer.

Experimentations have shown that the computation time needed to ensure that all the drawing operations are done is about 0.001 second for one silhouette. When all the drawing operations are done, i.e. when all the body parts are drawn, the second buffer becomes the current one and the silhouette of the considered 3D posture avatar can be extracted with the Z-buffer technique described above. The silhouettes are then obtained from the posture avatars. Now these silhouettes must be compared with the detected blob to determine the posture of the observed person.

5.3 Silhouette Representation and Comparison

A silhouette representation must be chosen to compress and to model the silhouette data. An associated comparison method must also be provided to measure the similarity between the silhouettes. The silhouette representations must respect two issues:

- Computation time. In our approach several silhouettes are modeled and compared. For instance, if a rotation step of 36 degrees and 10 postures of interest are considered, then 100 silhouettes are generated. The silhouette representation must model and compare these silhouettes in a little time.

- Dependence on the silhouette quality. Since the comparison is based on the silhouettes, the representation must be able to treat noisy silhouettes.

In section 5.3.1, several silhouette representations are described and their robustness to the two previous issues are discussed. In section 5.3.2, a focus is made on the chosen representations.

5.3.1 Silhouette Comparison

Comparing two silhouettes is a problem of shape similarity which is inherently ill-defined because the significance of "similar" is application dependent. A brief survey of techniques used in silhouette representation is given in next section. More complete surveys on shape matching can be found in [Veltkamp and Hagedoorn, 2001] or in [Loncaric, 1998].
The existing approaches to represent a silhouette may be classified into three categories:

1. feature-based
2. boundary-based
3. structural-based

1. **Feature-based** approaches determine a feature vector for a given silhouette. Two operations need to be defined: a mapping of the silhouette into the feature representation and a similarity measure of feature vectors. The simplest features are represented by geometric values:

- *Area*: the quantity of pixels which belong to the silhouette.
- *Perimeter*. The quantity of pixels which belong to the boundary of the silhouette.
- *Centroid*. The centre of gravity of the silhouette.
- *Compactness*. This value determines how round is a silhouette.
- *Eccentricity* or *Elongation*. It represents the ratio of the short axis length to the long axis length of the best fitting ellipse of the silhouette. This ratio is similar to the ratio of the height and width of a rotated minimal bounding box which contains the silhouette.
- *Rectangularity*. It defines "how" rectangular the silhouette is by computing the ratio of the area of the silhouette and the area of the bounding box. The bounding box is the minimal rectangle which encloses the silhouette. This feature has a value of 1 for a rectangular silhouette and decreases to 0 for a cross shape (cross shape minimises the area of the shape and maximise the area of its bounding box).
- *Orientation*. the overall orientation of the 2D silhouette on the image plane.

More sophisticated features may be used. In particular, statistical moments are applied. Based on these moments, many variations have been proposed, so that they remain invariant under certain transformations such as translation, scaling or rotation. The most commonly used moments are the seven Hu moments: [Bobick and Davis, 2001], [Rosales, 1998]. Another widely used person

representation is based on horizontal and vertical projections of the silhouette [Haritaoglu et al., 1998b], [Haritaoglu et al., 1998a], [Cucchiara et al., 2003]. Usually, a combination of these features is used to represent the silhouette. Either the Euclidean distance or a weighted vector distance is used to measure the similarity between silhouettes S_1 and S_2 :

$$S\left(\vec{S_1}, \vec{S_2}\right) = \sum_{i=1}^{m} \alpha_i \Psi_i \left(\vec{S_{1i}}, \vec{S_{2i}}\right) \tag{5.16}$$

where $\Psi_i \left(\vec{S_{1i}}, \vec{S_{2i}}\right)$ is the distance between the feature vectors $\vec{S_{ji}}$, $j \in \{1, 2\}$, associated with the i^{th} feature of the silhouette S_j. These representations are not time consuming (about 0.04 second to represent and compare 100 silhouettes). The geometric features and Hu moments representations have a certain dependence on the quality of the silhouette since they are computed on the whole silhouette, an error in the silhouette is in all the terms of the representation. The horizontal and vertical projections representation is less sensitive to the quality of the silhouette. Its smoothing power tends to treat errors in the silhouette.

2. **Boundary-based** approaches represent a silhouette by only its boundary. The approaches can consider a sample of points or all the points of the boundary. Fujiyoshi et al. ([Fujiyoshi and Lipton, 1998] and [Fujiyoshi et al., 2004]) extract salient points on the boundary of the silhouette by studying the distance between the boundary and the centre of gravity of the silhouette. The authors call this operation the "skeletonisation": the skeleton of the silhouette is obtained by linking the salient points with the centre of gravity. In [Dedeoglu et al., 2006], the authors use "skeletonisation" to classify a moving object evolving in a video into classes such as human, human group or vehicle; and human actions are classified into predefined classes such as walking, boxing or kicking.
Belongie et al. ([Belongie et al., 2002]) propose to match shapes and to recognise objects with a technique called shape context. A sample of uniformly spaced points on the contour is extracted. For a given sampled point, a shape context descriptor is defined by determining the set of vectors from this point to all other sampled points on the shape. Specifically, the shape context for a point is a log-polar histogram that sorts all vectors for a given point by a relative distance and an angular orientation. The histogram is computed according to a log-polar target and can be interpreted as a series of concentric circles enclosing a number of bins (figure 5.4). The density of sampled points is computed for each bin indexed by θ and $log\ r$: darker is the histogram, greater is the density. In [Mori and Malik, 2002], human body is estimated with shape context matching.

The similarity between two silhouettes can also be measured with their Chamfer distance [Barrow et al., 1977]. Given the two sets of points, $P = \{p_i\}_{i=1}^{n}$ and $Q = \{q_j\}_{j=1}^{m}$ which belong to the boundaries, the Chamfer distance is computed

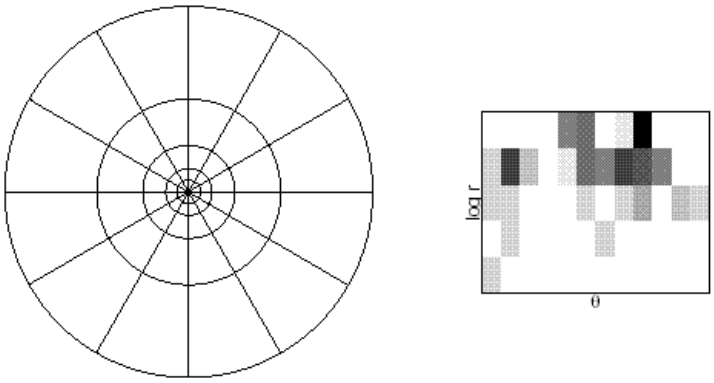

Figure 5.4: Log-polar target used in shape from context representation and a corresponding histogram where intensity is relative to density.

as the mean of the distances between each point belonging to P and its closest point in Q:

$$d_{cham}(P,Q) = \frac{1}{n}\sum_{p_i \in P} \min_{q_j \in Q} \parallel p_i - q_j \parallel \tag{5.17}$$

The symmetric Chamfer distance is obtained by adding $d_{cham}(Q,P)$. A comparison of shape context and Chamfer distance is given in [Thayananthan et al., 2003] for object localisation in cluttered scene. The authors claim that Chamfer distance is more robust in cluttered scenes than shape context matching by testing the measures for hand localisation.
These representations are based on the boundary of the silhouettes, so they have a certain dependence on the quality of the silhouette. The shape from context need a huge computation time according to the number of considered bins and points on the boundary (about 3.5 seconds for 100 silhouettes by considering 100 points on the boundary and 18 bins). The skeletonisation representation treats 100 silhouettes in 0.04 second.

3. **Structural-based** approaches usually represent the silhouette by a graph. A skeleton is computed with a distance transform. A distance transform D, computes a map in which each point corresponds to the distance of the pixel to the closest pixel of the object boundary. Once the skeleton is computed, its different "branches" are described, usually in polar coordinate, by their orientation and position. In [Sminchisescu and Telea, 2002], the distance transform D is approximated by solving the Eikonal equation:

$$|\nabla D| = 1 \tag{5.18}$$

This equation models the displacement in a perpendicular direction of a curve at a constant speed. D is initialised to 0 on the boundary. The solution of equa-

tion 5.18 has the property that its level sets are at equal distance from each other in the 2D space. Thus D is a good approximation of the distance transform map. The silhouette skeleton is then extracted from this distance map. The principal drawback of these techniques is that the skeleton of a noisy silhouette may be completely different from the one of a sharp silhouette.
In [Aslan and Tari, 2005], the authors propose a new axis-based silhouette representation by defining the relative spatial arrangement of local symmetry axes and their properties in a shape centered coordinate frame. The symmetry points are extracted from the evolving curves roughly mimic the motion by curvature ρ. The curve is evolving according to the equation 5.19 by initialising D by 1 on the boundary:

$$\nabla^2 D - \frac{D}{\rho^2} = 0 \tag{5.19}$$

where ρ is the curvature. In this representation, the branches are not necessary connected.
In [Erdem et al., 2006], the skeleton is extracted using the equation 5.19. The authors argue that this representation does not distinguish a likely articulation from an unlikely one. They propose an "articulation space" in which similar articulations yield closer coordinates.
The main drawback of this technique to compute the skeleton is the iterative process. Another technique has been implemented, based on the propagation of local distances in a two passes algorithm over the image, known as "the lawn mowing algorithm" [Rosenfeld and Kak, 1976]. The distance map is initialised to the infinity, and the pixels of the boundary to 0. During the first pass, forward pass, the image is processed from left to right and from top to bottom. During the second pass, backward pass, the image is processed from right to left, bottom to top. The pixel under consideration, is given the minimum value of itself and the values of its already visited neighbors each increased by their respective local step weights. This process of propagating information over the image using local step weights is often referred as chamfering, and weighted distance transforms (WDTs) [Borgefors, 1986], are therefore sometimes called Chamfer distance transforms or simply distance transform.
The principal drawback of the distance transform is that it is strongly dependent on the quality of the silhouette. An hole in the considered silhouette gives a distance transform different from the one of the same silhouette without hole. Some examples are given in figure 5.5.

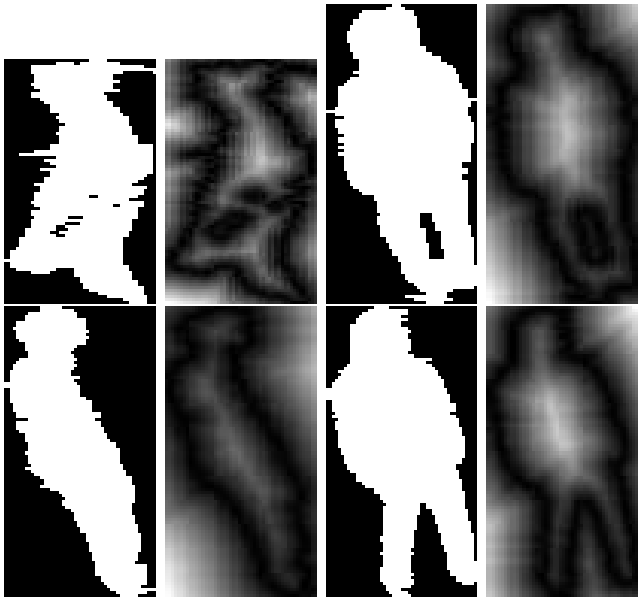

Figure 5.5: Silhouettes and associated distance maps. A darker pixel implies a nearest pixel to the boundary of the silhouette.

The table 5.1 summarises the different silhouette representations cited above by evaluating their computation time need and their dependence on the silhouette quality. The number of + and − gives an approximated idea about the two cited properties. The 2D method is well adapted to the considered property if there is several −. Inversely, the 2D method is not adapted to the considered property if there is several +.

2D methods	Computation time	Silhouette quality dependence
Geometric features	- -	++
Hu moments	- -	++
H. & V. projections	- -	+
Skeletonisation	-	++
Shape from context	+++	++
Distance transform	-	+++

Table 5.1: Classification of different 2D methods to represent silhouette according to their computation times and their dependence on the quality of the silhouette.

Four different representations have been chosen according to the table 5.1:

- a combination of geometric features of the silhouette,
- the seven Hu moments,
- the skeletonisation

- and the horizontal and vertical projections.

In the next section, the four silhouette representations which composed the proposed hybrid approach are described in details.

5.3.2 Silhouette Representation

5.3.2.1 Hu Moments

Shape representation by statistical moments is a classical technique in the literature [Bobick and Davis, 2001]. We use the definition described in [Bobick and Davis, 2001]. These moments are based on 2D polynomial moments:

$$m_{pq} = \Sigma_x \Sigma_y x^p y^q \rho(x, y)$$

where ρ is equal to 1 for pixels belonging to the silhouette and 0 for the background. In order to make moments invariant to translations, the moments are centered :

$$\mu_{pq} = \Sigma_x \Sigma_y (x - \bar{x})^p (y - \bar{y})^q \rho(x, y)$$

where $\bar{x} = \frac{m_{10}}{m_{00}}$ and $\bar{y} = \frac{m_{01}}{m_{00}}$. Furthermore, the following moments are computed to be invariant to scale changes by dividing the centered moments by the area of the silhouette:

$$\eta_{pq} = \frac{\mu_{pq}}{\mu_{00}^{\frac{p+q}{2}+1}}$$

where $p + q \geq 2$. Finally for these moments to be invariant to rotations, the following seven Hu moments are computed:

$$\begin{aligned}
H_1 &= \eta_{20} + \eta_{02} \\
H_2 &= (\eta_{20} - \eta_{02})(\eta_{20} - \eta_{02}) + 4\eta_{11}\eta_{11} \\
H_3 &= (\eta_{30} - 3\eta_{12})(\eta_{30} - 3\eta_{12}) + (\eta_{03} - 3\eta_{21})(\eta_{03} - 3\eta_{21}) \\
H_4 &= (\eta_{30} + \eta_{12})(\eta_{30} + \eta_{12}) + (\eta_{03} + \eta_{21})(\eta_{03} + \eta_{21}) \\
H_5 &= (\eta_{30} - 3\eta_{12})(\eta_{30} + \eta_{12})[(\eta_{30} + \eta_{12})(\eta_{30} + \eta_{12}) - 3(\eta_{03} + \eta_{21})(\eta_{03} + \eta_{21})] \\
&\quad + (3\eta_{21} - \eta_{03})(\eta_{03} + \eta_{21})[3(\eta_{30} + \eta_{12})(\eta_{30} + \eta_{12}) - (\eta_{03} + \eta_{21})(\eta_{03} + \eta_{21})] \\
H_6 &= (\eta_{20} - \eta_{02})[(\eta_{30} + \eta_{12})(\eta_{30} + \eta_{12}) - (\eta_{03} + \eta_{21})(\eta_{03} + \eta_{21}) \\
&\quad + 4\eta_{11}(\eta_{30} + \eta_{12})(\eta_{03} + \eta_{21})] \\
H_7 &= (3\eta_{21} - \eta_{03})(\eta_{30} + \eta_{12})[(\eta_{30} + \eta_{12})(\eta_{30} + \eta_{12}) - 3(\eta_{21} + \eta_{03})(\eta_{21} + \eta_{03})] \\
&\quad - (\eta_{30} - 3\eta_{12})(\eta_{21} + \eta_{02})[3(\eta_{30} + \eta_{12})(\eta_{30} + \eta_{12}) - (\eta_{21} + \eta_{03})(\eta_{21} + \eta_{03})]
\end{aligned} \tag{5.20}$$

The detected blob and the generated silhouettes are represented with these seven Hu moments defined in equations 5.20. The comparison between two sets of Hu moments is performed using an Euclidean distance.

5.3.2.2 Geometric Features

In this section, a combination of different geometric features is studied to represent the silhouette: area, centroid, orientation, eccentricity and compactness. Each of these features are described below. Most of these measures make reference to the classical moments m_{ij} and to the centered moments μ_{ij}. A definition of these moments are given in section 5.3.2.1 and they are computed on the whole silhouette.

Area

The area A_S of the silhouette S is computed by counting the quantity of pixels p which belong to the silhouette:

$$A_S = Card\{p \in S\} = m_{00} \tag{5.21}$$

where # is the cardinal operator and m_{00} the zero order moment.

Centroid

The centroid of the silhouette is computed using the classical moment m_{00}, m_{01} and m_{10}:

$$[\bar{x}, \bar{y}]^T = \left[\frac{m_{10}}{m_{00}}, \frac{m_{01}}{m_{00}}\right]^T \tag{5.22}$$

Orientation

The 2D orientation of the silhouette is determined using the second order centered moments μ_{11}, μ_{20} and μ_{02}. By considering the covariance matrix of the image:

$$Cov(I) = \begin{bmatrix} \mu'_{20} & \mu'_{11} \\ \mu'_{11} & \mu'_{02} \end{bmatrix} \tag{5.23}$$

where $\mu'_{ij} = \frac{\mu_{ij}}{\mu_{00}}$. The orientation θ of the silhouette corresponds to the angle of the eigenvector associated to the largest eigenvalue with the vertical axis and can be computed as:

$$\theta = \frac{1}{2} atan\left(\frac{2\mu'_{11}}{\mu'_{20} - \mu'_{02}}\right) \tag{5.24}$$

The value belongs to the range $]-90, 90]$ degrees and gives the angle with the vertical axis (figure 5.6).

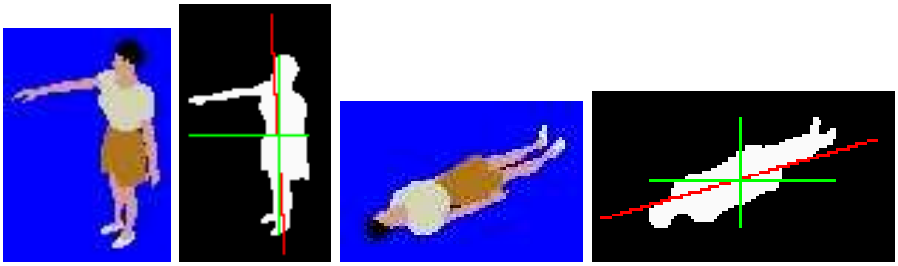

Figure 5.6: Example of orientation for two different generated silhouettes. The coordinate system is represented in green, and the principal axis is drawn in red. The orientation on the two first (resp. last) images is of 3.3 (resp. -73.9) degrees

Eccentricity

The eccentricity represents the ratio of the short axis length to the long axis length of the best fitting ellipse of the shape. The eccentricity is computed from second order central moments of the shape. The two eigenvalues λ_i of the matrix defined in equation 5.23 can be calculated with:

$$\lambda_i = \frac{\mu'_{20} + \mu'_{02}}{2} \pm \frac{\sqrt{(\mu'_{20} - \mu'_{02})^2 + 4\mu'^2_{11}}}{2} \tag{5.25}$$

λ_i are proportional to the squared length of the eigenvectors and the ratio of the eigenvalues gives the eccentricity of the silhouette:

$$Ecc = \frac{\mu'_{20} + \mu'_{02} - \sqrt{(\mu'_{20} - \mu'_{02})^2 + 4\mu'^2_{11}}}{\mu'_{20} + \mu'_{02} + \sqrt{(\mu'_{20} - \mu'_{02})^2 + 4\mu'^2_{11}}} \tag{5.26}$$

This value belongs to the range $[0, 1]$. It defines if the shape approximates more a circle ($Ecc = 0$) than a segment ($Ecc = 1$). In figure 5.6 the eccentricity values are for the left and right images respectively 0.5 and 0.8.

Compactness

The compactness value determines how round is the silhouette.

$$Com = \frac{4\Pi * A_S}{P_S^2} = \frac{4\Pi m_{00}}{P_S^2} \tag{5.27}$$

with A_S the area of the silhouette (equation 5.21), and P_S the quantity of pixels which belong to the silhouette boundary. The compactness value is maximum for a circle silhouette ($Com = \frac{4\Pi\Pi r^2}{(2\Pi r)^2} = 1$) and is less for other silhouettes. In figure 5.6, the compactness values are 0.18 and 0.29 for the left and right images respectively.

Combination of the geometric features

In order to compare two silhouettes, S_1 and S_2, the previous described features are computed and combined by a similarly measure S:

$$\mathrm{S}\left(S_1, S_2\right) = \sum_{i=1}^{m} \alpha_i \Psi_i \left(S_1, S_2\right) \tag{5.28}$$

with Ψ_i the measure associated to the i^{th} feature.
For each feature, a distance measure is proposed and the results always belongs to the interval $[0, 1]$ (0 for identical features and 1 for totally different):

- Area.

$$\Psi_A \left(S_1, S_2\right) = \frac{|A_1 - A_2|}{A_1 + A_2} \tag{5.29}$$

  where A_i is the area of the silhouette S_i.

- Centroid.

$$\Psi_x \left(S_1, S_2\right) = \frac{\sqrt{\left(\bar{x_1} - \bar{x_2}\right)^2 + \left(\bar{y_1} - \bar{y_2}\right)^2}}{\sqrt{max\left(h_1, h_2\right)^2 + max\left(w_1, w_2\right)^2}} \tag{5.30}$$

  where h_i and w_i are the height and width of the bounding box of the silhouette S_i and $[\bar{x_i}, \bar{y_i}]^T$ is its centroid.

- Orientation.

$$\Psi_\theta \left(S_1, S_2\right) = \begin{cases} \frac{|\theta_1 - \theta_2|}{90} & \text{if } |\theta_1 - \theta_2| < 90 \\ \frac{180 - |\theta_1 - \theta_2|}{90} & \text{else} \end{cases} \tag{5.31}$$

  where θ_i is the orientation of the silhouette S_i

- Eccentricity.

$$\Psi_{Ecc} \left(S_1, S_2\right) = |Ecc_1 - Ecc_2| \tag{5.32}$$

  where Ecc_i is the eccentricity of the silhouette S_i.

- Compactness.

$$\Psi_{Com} \left(S_1, S_2\right) = |Com_1 - Com_2| \tag{5.33}$$

  where Com_i is the compactness of the silhouette S_i.

5.3.2.3 Skeletonisation

A silhouette can be represented by its boundary. One way to extract salient points of the boundary is by skeletonising the silhouette. There are many techniques to compute the silhouette skeleton such as thinning or distance transformation (section 5.3). These techniques are computationally expensive. The method we use here is similar to the one proposed in [Fujiyoshi et al., 2004] and is described below.
The silhouette is dilated twice to remove small holes. Then an erosion is applied to smooth out any anomalies. The boundary is obtained by using a border following algorithm. The centroid of the silhouette is determined based on statistical moments. The distances from the centroid to the boundary points are calculated as Euclidean distances. Finally, the obtained distance curve is smoothed by using a smoothing filter before local maxima extraction. The local distance maxima correspond to the salient points of the boundary. The skeleton is then formed by connecting these maxima to the centroid.
A mean window algorithm is chosen to smooth the curve: the smoothed value of the curve is equal to the mean of the distances of the neighbor boundary points within the window. A larger window (in size) allows the detection of a smaller number of salient points 5.7.

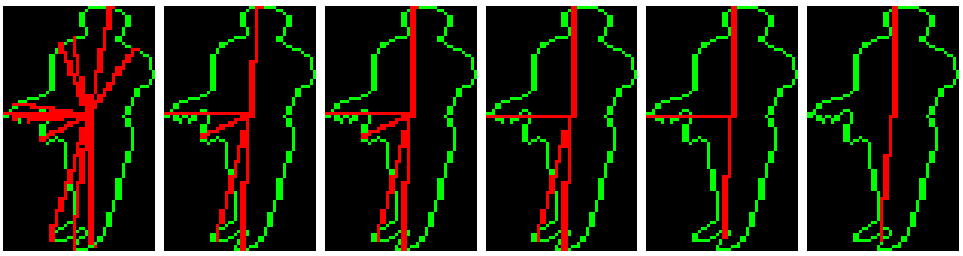

Figure 5.7: Examples of skeleton obtained for different window size: 0, 7, 9, 11, 21, and 41. The boundary of the silhouette is shown in green, and the skeleton is drawn in red. More the size of the window is big, less salient points on the boundary are found.

A measure based on the distance between salient points is proposed to evaluate the similarity between two silhouettes. The skeleton points are centered around the centroid of the silhouette. Let us define SD a set which contains the skeleton points of the detected silhouette, and SA_i a set which contains the skeleton points of the avatar silhouette of the i^{th} posture. The measure between the two skeletons characterised by SD and SA_i is given by:

$$M_i = \sum_{d \in SD} \min_{a \in SA_i} (\|d - a\|) \tag{5.34}$$

where $\|.\|$ is the Euclidean distance. The posture that minimises this measure is chosen as the solution.

5.3.2.4 Horizontal and Vertical Projections

A silhouette can be represented by its horizontal and vertical projections [Haritaoglu et al., 1998a], [Panini and Cucchiara, 2003], [Boulay et al., 2005].
The horizontal (resp. vertical) projection onto the reference axis is obtained by counting the number of moving pixels corresponding to the detected person at each image row (resp. column) denoted by H (and V respectively).
The 3D avatar is projected onto an image for each reference posture which are generated for all possible orientations. Then the horizontal and vertical projections of these silhouettes are compared with those of the detected person silhouette.

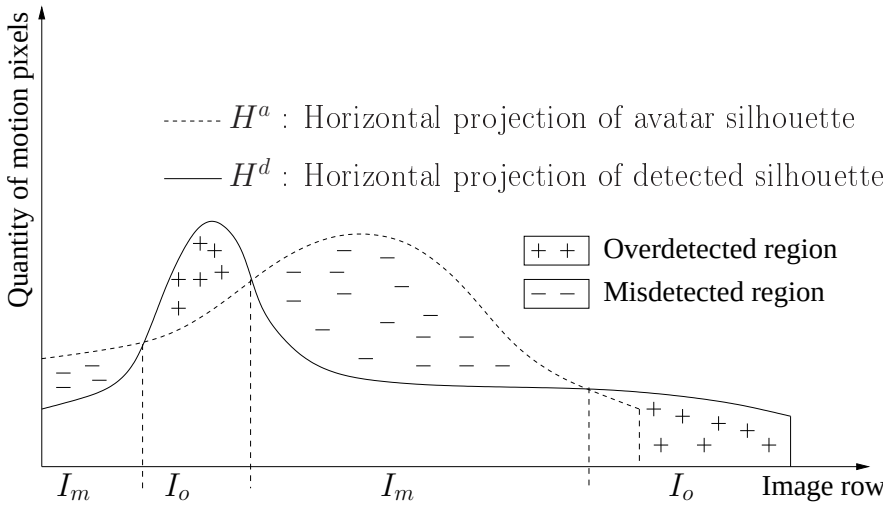

Figure 5.8: The "overdetected regions" I_o correspond to the regions where the horizontal projection of the detected silhouette is greater than the horizontal projection of the avatar silhouette, and inversely for the "misdetected regions" I_m.

Usually projections are compared with a classical SSD (Sum of Squared Differences) but tests have shown its limitation to handle noisy silhouette and difference between the 3D avatar and the observed person. Thus, we propose a comparison between projections based on the non-overlapping areas defined by equations 5.35, 5.36 and 5.37, and an illustration is given in figure 5.8.
Let us define two ratios $R_o(H)$ and $R_m(H)$ as follows:

$$R_o(H) = \frac{\sum_{i \in I_o} \left(H_i^d - H_i^a\right)^2}{\sum_i \left(H_i^d\right)^2} \tag{5.35}$$

$$R_m(H) = \frac{\sum_{i \in I_m} \left(H_i^d - H_i^a\right)^2}{\sum_i \left(H_i^a\right)^2} \tag{5.36}$$

The first ratio $R_o(H)$ represents the sum of squared differences of the projections computed on the interval I_o, normalised by the sum of squared values of the

horizontal projection of detected person H^d. The second ratio $R_m(H)$ represents the sum of squared differences of the projections computed on the interval I_m, normalised by the sum of squared values of the horizontal projection of generated avatar H^a. The same computation is performed along the vertical axis to obtain the ratios $R_o(V)$ and $R_m(V)$
The distance between the detected silhouette S_d and the avatar silhouette S_a is given by the mean of the four ratios $R_o(H)$, $R_m(H)$, $R_o(V)$ and $R_m(V)$ in equation 5.37:

$$dist(S_a, S_d) = \frac{1}{4}(R_o(H) + R_m(H) + R_o(V) + R_m(V)) \tag{5.37}$$

This distance belongs to the range $[0, 1]$ whereby 0 corresponds to similar silhouettes. Before computing these measures, the silhouettes are aligning on their centroid. The posture model which gives the minimum distance is chosen as the posture of the observed person.

5.4 Temporal Posture Coherency

The posture of an observed person is recognised in each frame independently from each other. However the postures of a person from one frame to another frame are correlated with each other. This dependence defines the posture stability principle described in the next section.

5.4.1 Posture Stability Principle

The posture stability principle states that for a high enough frame-rate the posture changes gradually. The use of this principle relies on the fact that the previously detected postures are known. The tracking information given by the people tracking task (the identifier) provides the list of the previously detected postures. The stability principle is then applied to a window of successive postures of a person where the most probable posture is chosen as the filtered posture of the person (algorithm 7) within a time interval.

Algorithm 7 *postureStability*(*detectedPosture*, *windowSize*, *weightList*, *t*)

postureList ← *NULL* {The list which contains the quantity of occurence of the postures}
for $i = -windowSize$ *to* *windowSize* **do**
  $postureList[detectedPosture[t+i]] += weightList[i]$
end for
return $getIndexOfTheMaximum(postureList)$ {return the posture which occurs the most frequently as the filtered posture at time *t*.}

The weight list, $weightList[i]$, determines how probable the *ith* posture occurs in the window of size 2**windowSize*+1. This smoothing algorithm reduces

posture misdetection and allows action recognition to benefit from reliable filtered postures. Different tests are realised for several set of weight. In particular, by pondering more or less the posture at time t. The experimentations have shown that a weight of 1 for each posture gives the best results.

5.4.2 Time Processing Control

Up to now, the posture avatar data-base is generated for each frame in function of the position of the detected person and the virtual camera. The computation of the data-base is expensive as described in chapter 6 (1.28 second to generate 100 silhouettes). To decrease the processing time, the data-base is generated when necessary depending on the position of the person: if the detected person does not move, the posture avatar data-base remains the same as previously and it does not need to be updated. The data-base is only updated when the detected person moves relatively far enough from the position corresponding to the last data-base update. This cue allow the proposed hybrid approach to treat 5-6 frames by second.

5.5 Conclusion

The proposed human posture recognition approach has been presented in this chapter. The approach combines 2D techniques and the use of 3D posture avatar to have a certain independence from the camera point of view and to minimise processing time as explained in section 3.3.
The approach uses the posture avatars defined in chapter 4, a virtual camera and the estimated position of the detected person to generate silhouettes of the postures of interest. The posture avatars are positioned in the scene depending on the type of the posture and avatars are rotated with a given rotation step. Finally, a Z-buffer technique is used to extract the silhouettes as described in section 5.2.2.3. Four different 2D techniques widely used to represent person silhouette have been chosen according to their reliability in terms of computation time and silhouette quality dependence. One of these techniques involves a combination of geometric features: area, centroid, orientation, eccentricity and compactness. Another technique is based on the silhouette region characterised by the seven Hu moments. The third technique studies the boundary of the silhouette to extract salient points: this technique is referred to the skeletonisation. The last technique involves horizontal and vertical projections of the silhouette.
The choice of the silhouette representation depends directly on the segmentation quality. For example, silhouette with holes must not used Hu moments representation since this approach mis-computes the moment terms. Also a problem of boundary detection will occur if the silhouette is defined by several blobs. Moreover, the silhouette representation must be chosen according to the goal of the

application. For instance, if the application requires locating salient points of the detected person (head or feet), skeletonisation is more appropriate than horizontal and vertical projections representation since these points are features for the skeletonisation. Finally, temporal information is used by applying the posture stability principle described in section 5.4. The next chapter experimentally computes the performance of these techniques using both synthetic and real data.

Chapter 6

Experimental Performance Evaluation

The goal of this chapter is to experimentally compare the techniques described in chapter 5. Section 6.1 presents the ground truth associated with the test sequences and describes how the video sequences are annotated and which attributes are considered. A tool is presented to easily acquire ground truth. Finally, the method which compares the obtained results with the ground-truth data is explained.
The experimental protocol is presented in section 6.2. Section 6.3 describes the results obtained with synthetic data. As our 3D posture avatars are realistic enough to generate realistic silhouettes, they are used to generate input video data. The great advantage of synthetic data is that all the video input data characteristics are controlled and a large amount of data from any view point can be easily generated. Indeed the virtual camera can observe the scene from any place in the virtual scene. Moreover, the segmentation can be more or less perfect.
Section 6.4 describes the results obtained with real data to evaluate the proposed human posture recognition approach. The robustness of the recognition for different segmentation types (over-segmentation and under-segmentation are described in section 6.4.1).
The conclusion of this chapter is given in section 6.5 which explains the robustness of the approach to over/under silhouette segmentations and the generecity of the approach by adding/removing postures of interest according to the type of application.

6.1 Ground Truth

The usual way to evaluate a vision algorithm is to compare its results with ground-truth. The ground truth is defined by its attributes that correspond to some properties of the video sequences. Once these attributes are defined, the remaining problem consists in acquiring these attributes. Finally, when the ground truth is acquired, it is compared with the results data obtained with the algorithm.

6.1.1 Ground Truth Attributes

The attributes of the ground truth to be annotated depend on the task to be performed by the vision algorithm to evaluate. In our case, we are interesting in evaluating the ability of our approach to recognise the posture of the persons in video sequences. The information needed by the ground-truth is:

- the information to locate the different people evolving in the scene. At the time of the comparison of the ground-truth with the obtained results data, the comparison algorithm must associate the detected person with a person in the ground-truth (or it may be able to say that the detected person does not exist).

- the posture of the person. The comparison algorithm must have information about the posture of the person to evaluate the results obtained by the human posture recognition algorithm.

To locate the different people evolving in the scene, two attributes are proposed. The first attribute is a single **identifier** associated to each of the person who appears in the video sequence. The people are then tracked in the entire sequence with their single identifier. The second attribute is the **bounding box** around the person. The person evolving in the sequence is localised with this bounding box which is represented by the coordinates of the upper left box corner and by the *height* and the *width* of the box.

The **posture** of a person is defined by an identifier and an approximation of its orientation. The posture is represented by an identifier associated with the detailed postures: standing with the left arm up (0), standing with the right arm up (1), standing with arms near the body (2), T-shape (3), sitting on a chair (4), sitting on the floor (5), bending (6), lying with spread legs (7), lying on the right side with curled up legs (8) and lying on the left side with curled up legs (9). Ground truth posture is manually chosen among the previous list which visually matches the observed posture. The next attribute represents the orientation of the person which is approximated by choosing one of the eight intervals: $[0, 45[$, $[45, 90[$, $[90, 135[$, $[135, 180[$,$[180, 225[$, $[225, 270[$, $[270, 315[$ and $[315, 360[$. A person who looks at the camera has a 0 orientation.

A last attribute represents the **occlusion type** of the person. The person can either be partially occluded or not occluded by an object or a person.

Thus in the ground truth, each person of the video sequence is represented by a single identifier, and for each frame, the person is described by its bounding box, its posture, its orientation and its occlusion type. The location of a person is given by an identifier and a bounding box. Ideally, since the proposed human posture recognition approach has to be evaluated, the quality of the silhouette should also be annotated. In practice, it is not possible, since the ground truth

data would be associated to a given segmentation algorithm and not to a video sequence. The segmentation task and the posture recognition tasks are thus evaluated together. To take into account the impact of the silhouette quality on the human posture recognition, we propose an evaluation based on a bounding box criteria as described in section 6.1.3.

6.1.2 Ground Truth Acquisition

Ground-truth acquisition is a tedious and long task. Fortunately, there is a helpful graphical tool for annotation: the Viper software from University of Maryland (VIdeo Performance Evaluation Resource) [Mariano et al., 2002], [Doermann and Mihalcik, 2000], [Viper, 2006]. This tool (figure 6.1) makes possible to easily draw bounding boxes and to assign user defined information to each person evolving in the scene (posture, occlusion, ...). The Viper software saves the ground truth in a Viper XML format file. The evaluation of the approach consists in comparing the Viper XML file with the data obtained with the posture recognition algorithm.

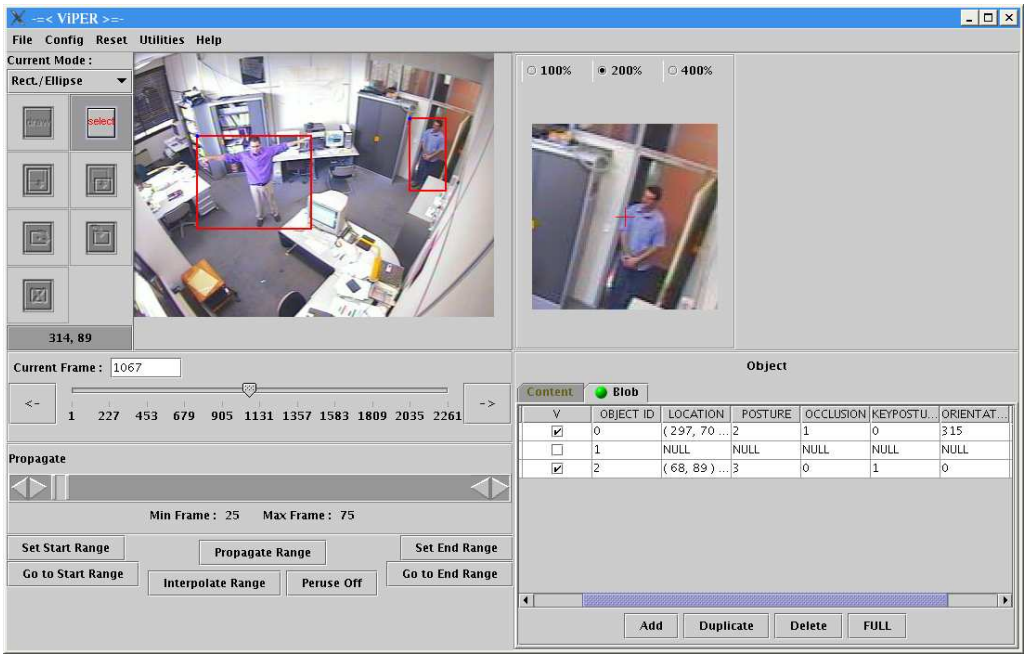

Figure 6.1: The Viper graphical tool to annotate a video sequence.

We have defined some rules to homogenise the ground truth throughout the experimentation:

- The bounding box is drawn around the entire person even for the occluded parts. A part of the person must be visible on the image.

- A single identifier must be associated to the same person of the entire video sequence, even if the person temporally disappears.
- The bounding box is not drawn if the person is completely occluded.

6.1.3 Evaluation Method

The proposed human posture recognition approach provides a file which describes the obtained results. This output file contains three attributes for each frame and each person:

- the number of the frame
- and for each detected person in each frame:
  - its bounding box to compare with the bounding box in the ground truth file
  - the best recognised posture, its orientation and an associated error (the error measures the similarity between the detected silhouette and the chosen generated one).
  - the other recognised postures, classified from best recognised posture to worst recognised posture, to check how far the best recognised posture is from the other recognised postures.

The evaluation of the approach is based on the comparison of the data contained in this file with the ground truth data associated with the video sequence.

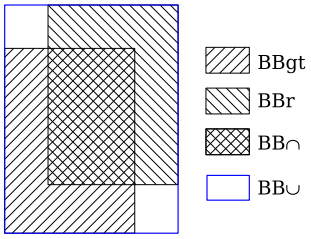

Figure 6.2: Illustration of two overlapping bounding boxes, BB_{gt}: ground truth bounding box and BB_r: bounding box computed by the people detection task. $BB_{\bigcap}$ (respectively $BB_{\bigcup}$) denotes their intersection (resp. union).

Each detected person in the result file is searched in the ground-truth file by comparing their bounding boxes according to the frame number. The bounding box overlapping rate is computed as follows:

$$\frac{\#\left(BB_{\bigcap}\right)}{\#\left(BB_{\bigcup}\right)} \tag{6.1}$$

where #() is the cardinal operator and $BB_{\bigcap}$ (respectively $BB_{\bigcup}$) is the intersection (resp. union) of the ground truth bounding box BB_{gt} (figure 6.2). The values of the overlapping rate varies from 0 (disconnected bounding boxes) to 1 (perfectly matched bounding boxes).
Since the posture recognition algorithm is evaluated and not the segmentation algorithm (neither the people detection algorithm), the case where a person is detected but does not exist in the ground truth file is not taken into account. Moreover the case where a person is not detected is not taken into account. A threshold on the overlapping bounding boxes criteria (equation 6.1) is used to take into account the quality of the silhouette. Once the person is identified, the postures are compared: if the postures are the same, the recognition is correct and if not the recognition is wrong.
The three classical evaluation rates are computed for the posture types P_i as follow ($i \in \{1 \cdots ng\}$, where ng is the number of general postures and $i \in \{1 \cdots nd\}$, where nd is the number of detailed postures):

- true positive (TP): the posture P_i is correctly detected according to the ground truth.

$$TP(P_i) = \frac{\#\{P_i \text{ correctly detected}\}}{\#\{P_i \text{ in the ground-truth}\}} \tag{6.2}$$

- false positive (FP): the posture P_i is wrongly detected according to the ground truth.

$$FP(P_i) = \frac{\#\{P \text{ wrongly detected as } P_i\}}{\#\{P \text{ in the ground-truth}\}} \tag{6.3}$$

- false negative (FN) gives the rate of wrong recognition of posture type P_i according to the ground truth.

$$FN(P_i) = 1 - TP(P_i) \tag{6.4}$$

The results are given with two levels of detail by considering the general postures and the detailed postures:

- the general posture recognition rate: $GPRR$ corresponds to the TP associated to the number ng of general postures Pg_i:

$$GPRR = \frac{1}{n_T} \sum_{i=1}^{ng} ng_i * TP(Pg_i) \tag{6.5}$$

where ng_i is the number of case where the posture is Pg_i and n_T is the total number of considered cases.

- the detailed posture recognition rate: $DPRR$ corresponds to the TP associated to the number nd of detailed postures Pd_i:

$$DPRR = \frac{1}{n_T} \sum_{i=1}^{nd} nd_i * TP(Pd_i) \quad (6.6)$$

where nd_i is the number of case where the posture is Pd_i and n_T is the total number of considered cases.

6.2 Experimental Protocol

The tests were performed on a classical PC under the Linux operating system:

- processor: Intel Xeon 3.06GHz
- memory: 1 Go of RAM
- Graphic card: NVidia Quadro 280NVS, AGP 8X, 64 Mo.

6.3 Synthetic Data

As explained in section 4.2.2, the body parts of our 3D human model have been designed to obtain a realistic model in order to generate synthetic data. Synthetic data have several advantages:

- The data can be generated easily for any view point and for any position of the avatars in the virtual scene.
- The posture recognition approach can be studied according to different problems: segmentation quality, intermediate postures, ambiguous postures and variability between the observed person and the 3D avatar.
- The ground truth generation is completely automatic. Indeed, during the synthetic data generation process, all the parameters are controlled, therefore all the information needed by the ground truth is available at any time (posture, 3D position, orientation, etc.)

The main drawback of using synthetic data is that it is difficult to realisticly simulate some noise such as real sensor one. In particular, we must be careful on choosing the best silhouette representation, which also depend on the quality of the silhouette obtained through the segmentation task.

6.3.1 Synthetic Data Generation

In this section, synthetic data are generated from two different ways. A first way to generate synthetic data is based on a virtual trajectory method (figure 6.3). A graphical interface displays a scene visualised from the top. The user clicks

on the interface to draw the desired trajectory. At each salient point of the trajectory a posture chosen among the selected ones is associated. The images of the 3D posture avatar which moves on the trajectory are then computed. The intermediate postures between two postures of interest are not computed. This experimentation is interesting to have a quick overview of the recognition rate of the algorithm. The second generation is exhaustively done. The different 3D

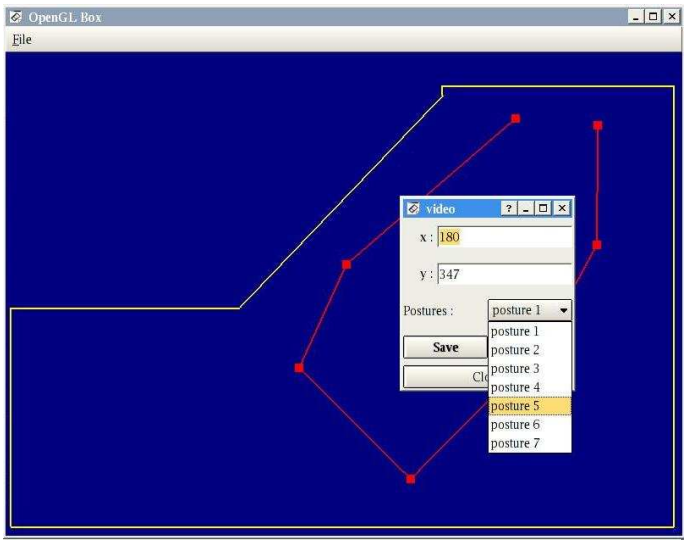

Figure 6.3: Graphical tool to easily generate data based on trajectory.

posture avatars are positioned in the virtual scene and rotated around the axe w for any given rotation angle on the ground α_g. A virtual camera is positioned on a circle trajectory at every five degrees (β_c) as shown in figure 6.4. The exhaustive data generation is simple to use to evaluate the proposed human posture recognition. In the next section, the different silhouette representations are evaluated with this experimentation.

6.3.2 Silhouette Representation Evaluation

Synthetic data are used to evaluate the different silhouette representations. A data-base is computed according to the exhaustive technique described above. Ten posture are used: standing with left arm up, standing with right arm up, standing with arms along the body, T-shape posture, sitting on a chair, sitting on the floor, bending posture, lying with spread legs, lying on the left side with curled up legs and lying on the right side with curled up legs. 19 different points of view are considered by moving the virtual camera at every 5 degrees in a circle around the avatars as shown in figure 6.5 for the T-shape posture. The data-base is then composed of 68400 frames (10 avatars * 360 orientations * 19 viewpoints). Moreover, the 3D posture avatar model involved in the generation of data is different from the one used for the posture recognition process (figure 6.6).

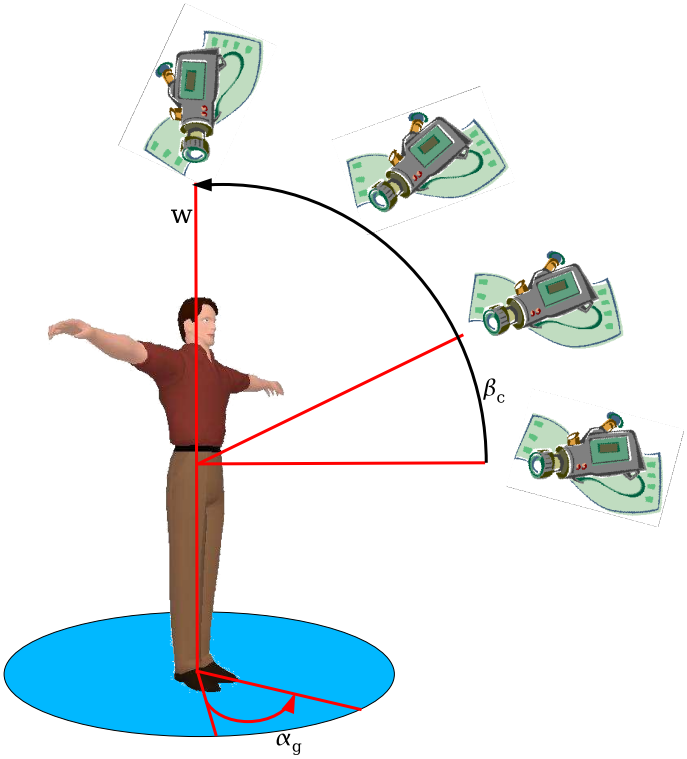

Figure 6.4: Generation of synthetic data for different points of view.

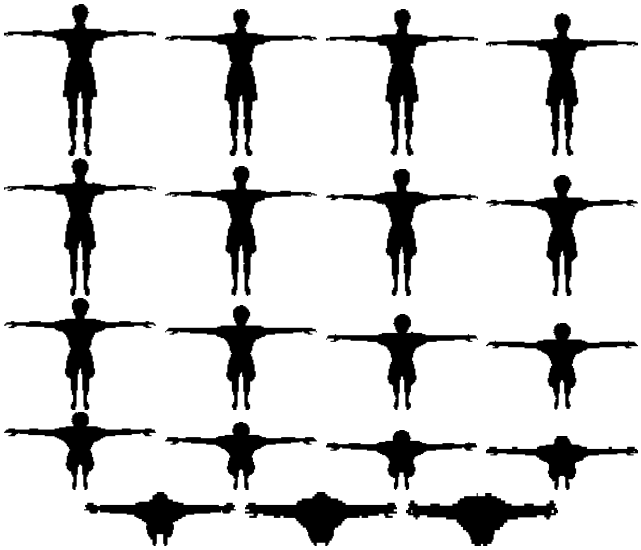

Figure 6.5: Silhouettes obtained with the woman model for the different considered points of view: $\beta_c = 0, 5, 10, 15, 20, 25, 30, 35, 40, 45, 50, 55, 60, 65, 70, 75, 80, 85, 90$ degrees.

Figure 6.6: 3D posture avatar involved in data generation (testing with woman model) and in the posture recognition process (recognition done with man model).

Geometric features

We have performed some experimentations to select which features should be chosen for geometric features silhouette representation. The features involved in the geometric features representation are composed of the 2D orientation, the eccentricity and the compactness. Their values are displayed in figures 6.7, 6.8, and 6.9 respectively for different 3D avatar orientations and a given point of view ($\beta_c = 0$). The experimentation consists in rotating the different 3D posture avatars and computing the different geometric features for each degrees.
A symmetry can be observed on each graphic according to the abscisse point 180 (the back of the avatar facing the camera), due to the symmetry of the human body. We can see in figure 6.7 that the orientations of the four standing postures (the four top curves in dark blue on the figure) are near 0 degree and are very similar. More generally, the postures which belong to the same general posture have a similar orientation feature. Moreover, the orientation features are different for the different general postures excepted for few 3D avatar orientations ($\alpha_g =$ 180 degrees). Thus, the orientation feature seems to be a good discriminant for the general postures. Other features should be used to discriminate the detailed postures. Eccentricity feature is studied in the following. The eccentricity feature value defined in section 5.3.2.2 represents if the silhouette approximates more a circle (eccentricity equal to 0) than a segment (eccentricity equal to 1). The figure 6.8 shows also the symmetry of the eccentricity value according to an avatar orientation $\alpha_g = 180$ degrees. The eccentricity feature separates the detailed postures except for the two top curves (respectively the two bottom curves) which represents the eccentricity values for standing with left arm up and standing with right arm up postures (respectively lying on the right side and lying on the left side). These four precited postures are visually ambiguous (see figure 6.10 the first and second rows, and the last but one and last rows) for many avatar orientations

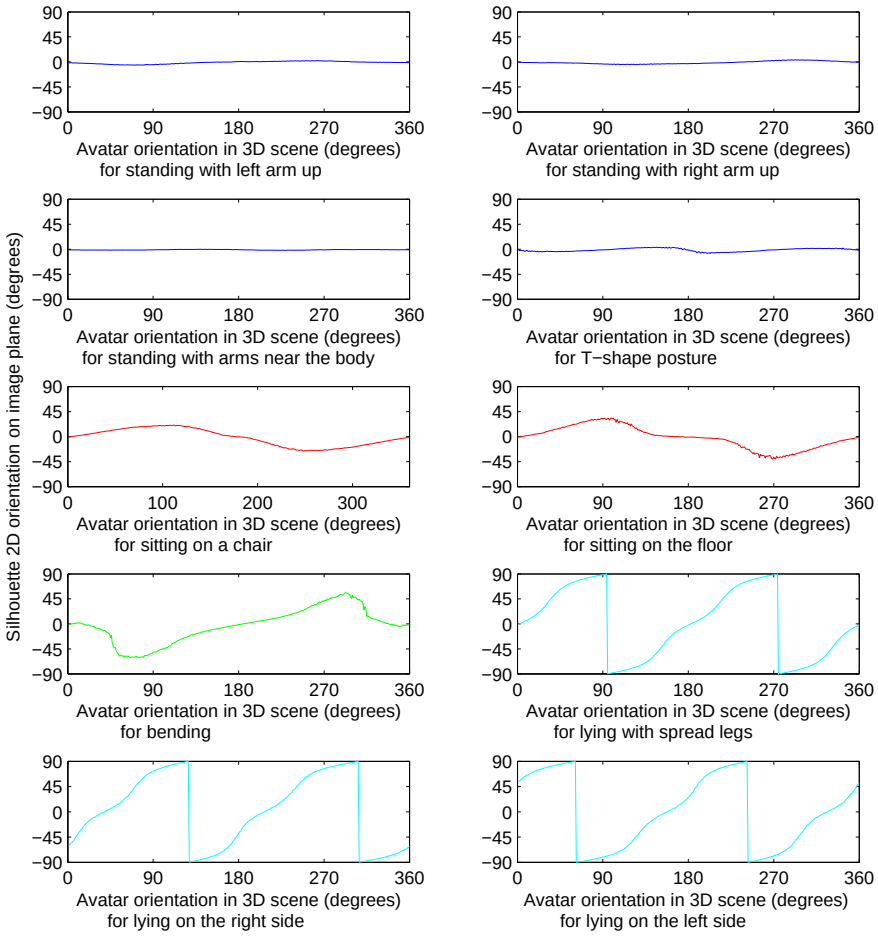

Figure 6.7: Orientation of the silhouette in function of the orientation of the 3D posture avatar.

(α_g). In the next, we consider that these postures correspond to only two postures of interest: the standing with one arm up posture and the lying with curled up legs.

The compactness feature value defined in section 5.3.2.2 represents how round is the silhouette. In figure 6.9, we can see that the compactness feature value is less than 0.6 for all the postures of interest because the compactness value is equal to 1 for a circle silhouette. The different curves show that the compactness values are similar for postures which belong to the same general posture. The compactness feature is a good discriminant for the general postures. Therefore, the combination of these different features is necessary to recognise correctly the

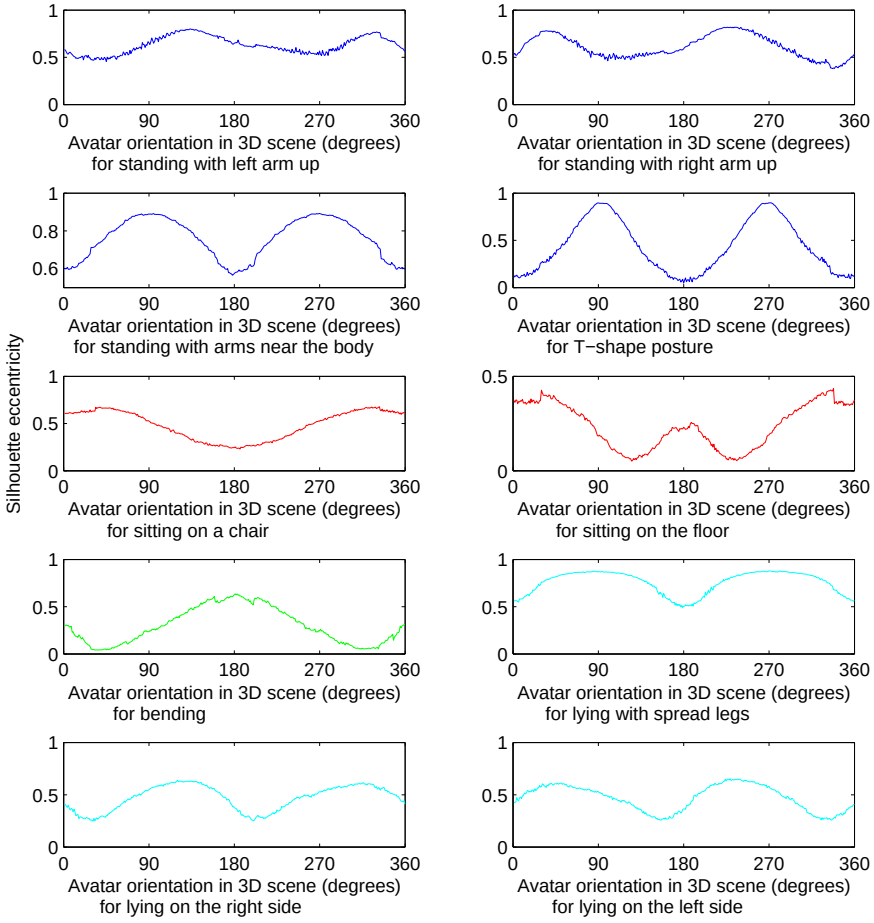

Figure 6.8: Eccentricity of the silhouette in function of the orientation of the 3D posture avatar.

general and the detailed postures.

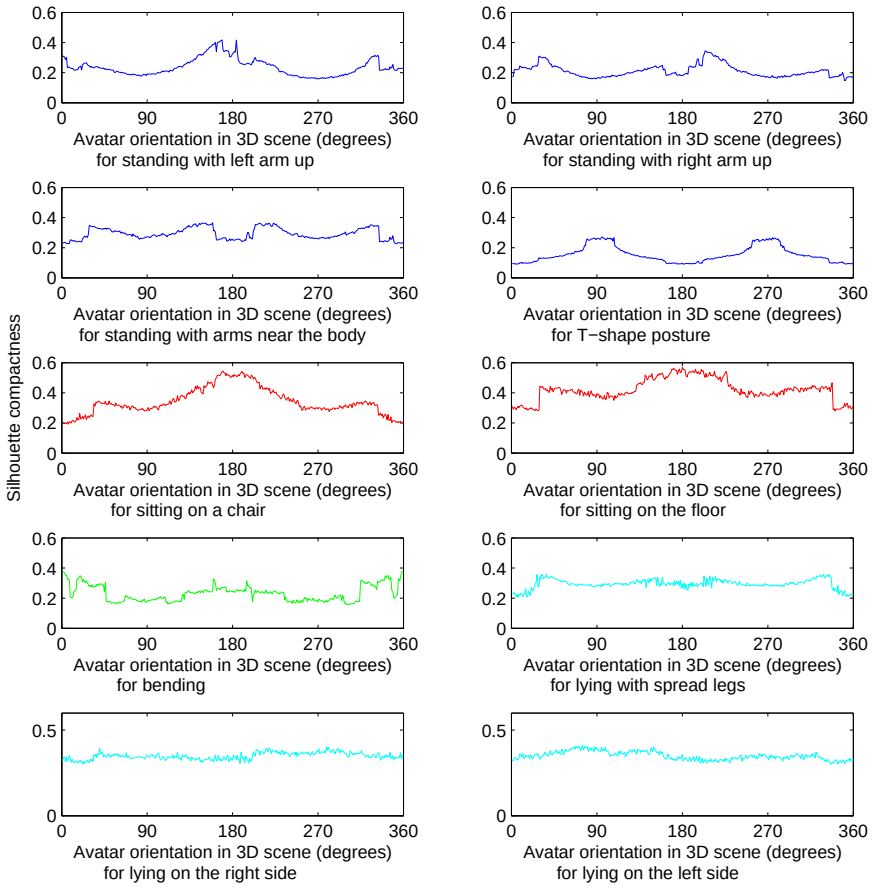

Figure 6.9: Compactness of the silhouette in function of the orientation of the 3D posture avatar.

Silhouette Representation Evaluation

The approach is evaluated for different rotation steps which is one of the main parameter of the proposed human posture recognition approach (see figure 6.10).

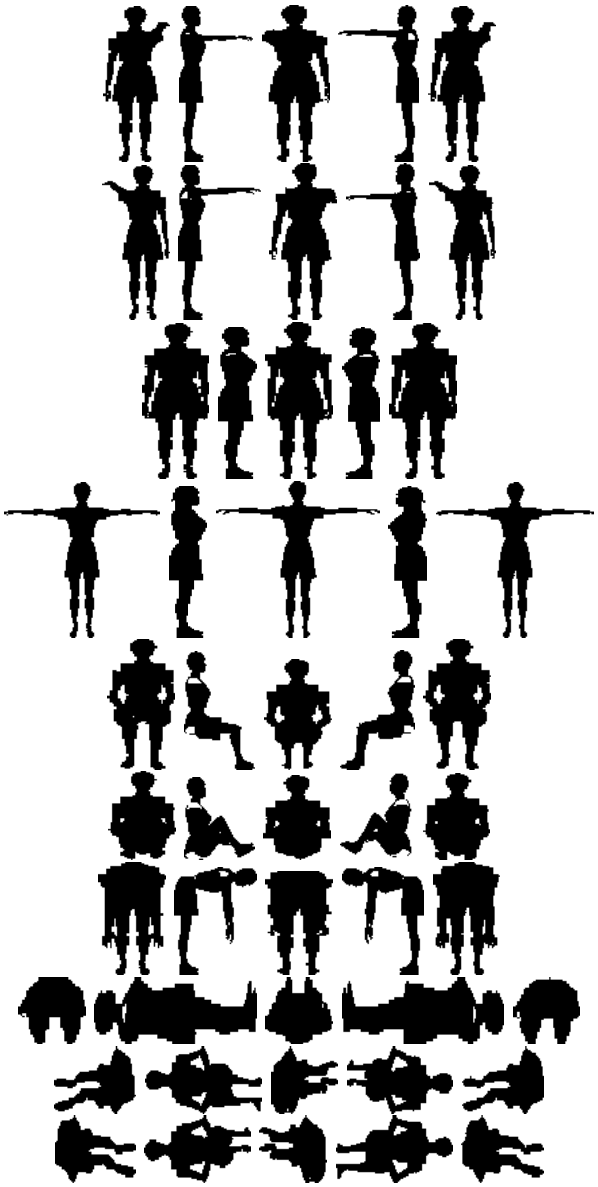

Figure 6.10: Silhouettes obtained with the woman model for the ten postures of interest for several avatar orientations ($\alpha_g = 0, 90, 180, 270, 359$), for the point of view $\beta_c = 0$.

The processing time is also evaluated which is characterised by three times:

- the silhouette generation time tg which represents the necessary time to generate the silhouettes of the 3D posture avatars. It depends on the rotation

step which defines the number of generated silhouettes.

- the silhouette representation time tr which represents the necessary time to model each generated silhouette. It depends on the considered 2D silhouette representation approach,

- the silhouette comparison time tc which represents the necessary time to compare the generated silhouettes with the detected one. It also depends on the considered representation.

The posture recognition rates are given in table 6.1 for the four chosen silhouette representations and for different rotation steps. The woman model is used to generate data and the man model is used to the recognition. The table 6.1 is made of four parts, each one is associated with a studied silhouette representation. Each part describes the general posture recognition rate (GPRR), the detailed posture recognition rate (DPRR) and gives a computation time approximation for a given rotation step. The table shows that the rotation step is an important cue for recognising posture. The results can be interpreted in term of posture recognition rate and in term of computational time:

- The general postures are better recognised than the detailed ones for all the silhouette representations since the GPRR is always greater than the DPRR. There are fewer visual ambiguities with general postures than with detailed postures. The horizontal and vertical projections of the silhouette representation gives the best recognition rates for both the general and detailed postures as shown in table 6.1 by comparing the GPRR and DPRR for each silhouette representation. The geometric features representation gives better recognition than skeletonisation and Hu moments representations. The posture recognition rates increase when the rotation step decreases (i.e. when more silhouettes are generated) for the H. & V. projections and the Hu moments representations. The geometric features and skeletonisation representations are less discriminant for a rotation step below 20 degrees because the discrimination power of these approaches are not sufficient to correctly discriminate the different silhouettes (180 silhouettes for a rotation step of 20 degrees up to 3600 silhouettes for a rotation step of 1 degree).

- The computation time depends on the rotation step value: the computational time decreases when the rotation step increases since the number of generated silhouette decreases. The most consuming step is the silhouette generation. To obtain a real time processing, a trade-off must be chosen between recognition and computation time. A rotation step of 36 degrees was chosen as the optimal rotation step the proposed human posture recognition approach. This rotation step corresponds to the generation of 100 silhouettes corresponding to 10 postures of interest and 10 orientations per posture. As shown, in section 6.4, for a rotation step of 36 degrees, the approach treats 5 to 6 frames per second.

Geometric Features							
Rotation step (degrees)	1	5	10	20	36	45	90
GPRR (%)	88	88	91	92	89	88	69
DPRR (%)	79	78	82	81	75	72	52
tg (s/frame)	40.6	8.2	4.12	2.12	1.28	1.04	0.52
tr (s/frame)	1.39	0.28	0.14	0.07	0.04	0.03	0.02
tc (s/frame)	0.49	0.02	0.005	0.0013	0.00039	0.00025	0.00006
Hu Moments							
Rotation step (degrees)	1	5	10	20	36	45	90
GPRR (%)	72	72	72	72	69	68	59
DPRR (%)	64	62	62	59	57	54	43
tg (s/frame)	40.6	8.2	4.12	2.12	1.28	1.04	0.52
tr (s/frame)	1.35	0.27	0.14	0.07	0.04	0.03	0.01
tc (s/frame)	0.46	0.02	0.005	0.0012	0.0004	0.0003	0.00004
Skeletonisation							
Rotation step (degrees)	1	5	10	20	36	45	90
GPRR (%)	86	87	89	89	84	82	71
DPRR (%)	74	76	77	75	68	63	47
tg (s/frame)	40.6	8.2	4.12	2.12	1.28	1.04	0.52
tr (s/frame)	1.5	0.29	0.14	0.07	0.04	0.03	0.01
tc (s/frame)	0.47	0.02	0.005	0.0015	0.0006	0.0004	0.0001
Horizontal and Vertical Projections							
Rotation step (degrees)	1	5	10	20	36	45	90
GPRR (%)	99	99	98	95	90	89	75
DPRR (%)	95	94	92	87	76	72	54
tg (s/frame)	40.6	8.2	4.12	2.12	1.28	1.04	0.52
tr (s/frame)	1.34	0.27	0.13	0.06	0.04	0.03	0.02
tc (s/frame)	0.71	0.06	0.03	0.012	0.006	0.005	0.003

Table 6.1: General (GPRR) and detailed posture recognition rate (DPRR), and different processing times obtained: silhouette generation time (tg), silhouette representation time (tr) and silhouette comparison time (tc) according to the different silhouette representations.

6.3.3 Variability in the synthetic data

To analyse the behaviour of the posture recognition algorithm on intermediate postures, a second set of synthetic data has been generated by randomly modifying the joint parameter angles. The random added angles are in the range $[-15; 15]$ in degree. The different recognition rates are given in table 6.2 according to the silhouette representation approach.

Geometric Features					
Rotation step (degrees)	10	20	36	45	90
GPRR (%)	84	84	81	82	81
DPRR (%)	58	61	53	58	53
Hu Moments					
Rotation step (degrees)	10	20	36	45	90
GPRR (%)	51	51	54	45	47
DPRR (%)	37	36	35	34	29
Skeletonisation					
Rotation step (degrees)	10	20	36	45	90
GPRR (%)	65	66	70	66	73
DPRR (%)	44	44	47	42	42
Horizontal and Vertical Projections					
Rotation step (degrees)	10	20	36	45	90
GPRR (%)	73	74	73	75	63
DPRR (%)	51	54	54	54	42

Table 6.2: General (GPRR) and detailed posture recognition rate (DPRR) obtained according to the different silhouette representations for joint angles variation.

Geometric features and H. & V. projections representations are less sensitive to the variability in 3D posture avatars than other representations. In addition, the gesture: "left arm in motion" is studied in detail. A third synthetic sequence is obtained by modifying the left shoulder angle parameters as shown in figure 6.11: 90 degrees corresponds to the arm at the up vertical, -90 degrees corresponds to the arm down. We expect that the approach recognise first the standing posture wit arms near the body (the most visually similar posture of interest), then standing with one arm up and finally standing wit arms near the body. The result of the posture recognition algorithm is shown in figure 6.12 wit hand without temporal filtering. First, we can notice the the temporal filtering (second column), removes the "noisy" recognised postures by smoothing the recognition. Second, the H. & V. projections representation (curves on the second row in red on the figure 6.12) gives the best results by recognising clearly the three successive precited postures. The approach with the Hu moment representation recognise more soon the standing with one arm up posture, but recognise also on few frames the lying posture. The H. & V. projections representation is less sensitive to differ-

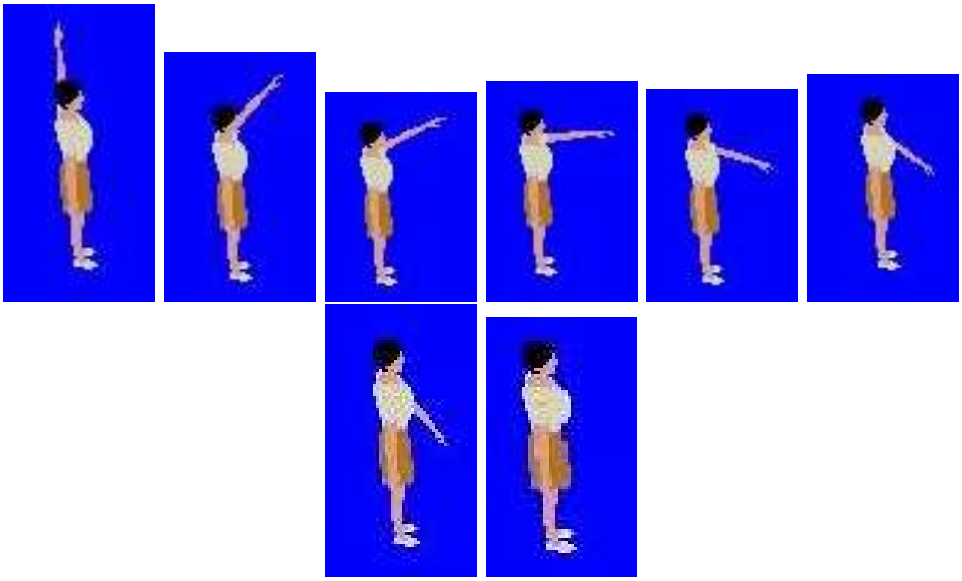

Figure 6.11: Different images of the sequence: "left arm in motion" for a given orientation in degree of the left shoulder: -90,-45,-22,0,25,45,68,90. The fourth image corresponds to the posture of interest: standing with left arm up (0 degree), and the last image corresponds to the posture of interest standing with arms near the body (90 degrees).

ence of the intermediate postures from the postures of interest by smoothing the silhouettes.

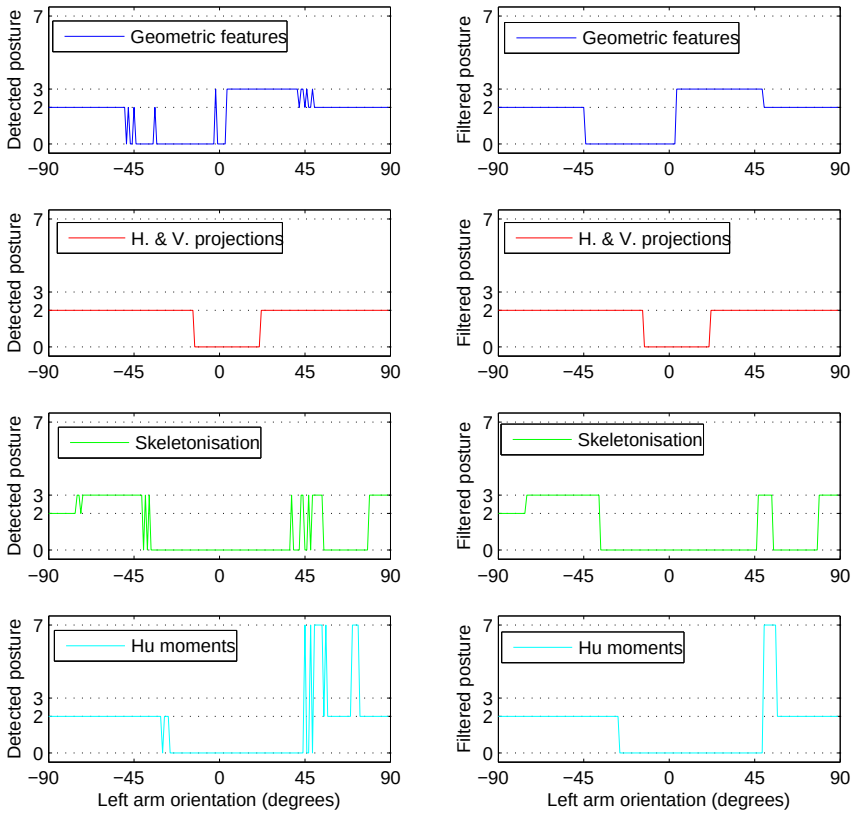

Figure 6.12: Recognised postures for the "left arm in motion" sequence, according to the different silhouette representations with (left column) and without (right column) temporal posture filtering. The detected postures are : 0-standing with one arm up, 2-standing with arms near the body, 3-T-shape, 7-lying posture.

6.3.4 Ambiguous Cases

Silhouettes representative of different postures can have the same projection on the image for a certain point of view generating ambiguous cases. These ambiguities are due to ambiguous view points and person self-occlusion. The quality of recognition for these cases depends on the silhouette representations and the comparison measure. Synthetic data can be used to identify ambiguous cases according to the point of view, the posture and the orientation. Confusion matrices for each silhouette representation and each point of view (each value of β_c) are given in appendix B.

	0	1	2	3	4	5	6	7	8	9
0	185	49	18	42	0	0	0	0	0	0
1	113	249	48	60	0	0	0	0	0	0
2	27	33	294	53	0	0	0	0	0	0
3	35	29	0	205	0	0	0	0	0	0
4	0	0	0	0	338	2	0	0	0	0
5	0	0	0	0	22	357	37	0	0	0
6	0	0	0	0	0	0	323	9	0	0
7	0	0	0	0	0	0	0	283	23	0
8	0	0	0	0	0	1	0	18	284	121
9	0	0	0	0	0	0	0	50	53	239

Table 6.3: Confusion matrix for detailed postures recognition for H. & V. projections for synthetic data according to the 0th point of view. (0: standing with left arm up, 1: standing with right arm up, 2: standing with arms near the body, 3: T-shape posture, 4: sitting on a chair, 5: sitting on the floor, 6: bending, 7: lying with spread legs, 8: lying with curled up legs on right side, 9: lying with curled up legs on left side)

For example the table 6.3.4 shows the confusion matrix corresponding to the point of view ($\beta_c = 0$): the rows correspond to the recognised postures and the column correspond to the ground-truth ones. By analysing the table, we see that the detailed postures are confused with their general postures. In detail, the posture sitting on the chair is recognised correctly 338 times on 360 cases: the confident value of this posture is then $\frac{338}{360}$. Moreover, accuracy can be added by analysing results according to the orientation angle of the 3D avatar (α_g). Figure 6.13 illustrates the ambiguity problem for standing with one arm up posture for the (H. & V.) projections. This posture is similar to standing with arms near the body posture for many orientations. The graph can provide confidence value for the recognition in function of the recognised posture and orientation. For example during the interval [50,125] and [200,250], standing with one arm up posture can be recognised without ambiguity.

This a priori knowledge can be exploited in the recognition process to associate a confident value to how the postures are recognised.

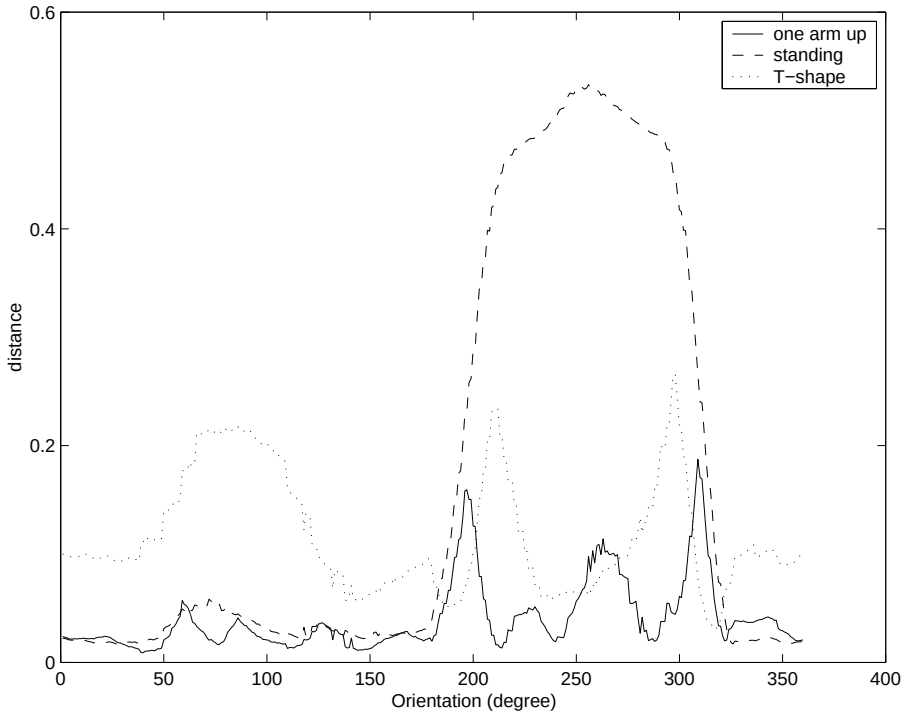

Figure 6.13: The graphics shows distance obtained by comparing standing postures with one arm up (3D woman model) with all the standing postures (3D man model). (H. & V.) projections representation is used for different avatar orientation.

6.4 Real Data

Evaluating a vision algorithm on real data is an important step to validate an algorithm. Indeed, real data are more complex than synthetic ones, and introduce some difficulties in recognising postures. In our case, the principal difficulty occurs during the segmentation of the people evolving in the scene which introduces errors in the silhouette. In real data, the segmentation is noisy due to many reasons such as light changes or bad image contrast. Since the posture recognition is based on the silhouette, segmentation errors influence the recognition. Moreover, in the case of our algorithm, real data introduce variability in the posture: the intermediate postures between the postures of interest can vary significantly. Finally, real data create variability of the morphology of the people evolved in the video. This suggests the use of several 3D avatars models corresponding to different morphologies or clothes. In section 6.4.1, the three segmentation algorithms used in this work are described. The different video sequences and results are described in section 6.4.2.

6.4.1 People Detection

Detecting people is the first step in a human posture recognition system. This step is crucial because it has a strong impact on the quality of the recognised postures and it is the main reason to explain the limitation of the posture recognition. To achieve a good detection, the mobile objects of the scene can be segmented using different segmentation algorithms. A common technique is based on reference image subtraction, where the reference image is an image of the scene, without any mobile object (like human in our case). During this thesis work, we have used three different segmentation algorithms: one developed by the ORION team (*VSIP algorithm*), an other proposed by the CMM (Centre de Morphologie Mathématique) of Ecole des Mines de Paris (*watershed algorithm*), and a last used in gait analysis [Sarkar et al., 2005].

The VSIP segmentation algorithm is based on the subtraction of the current image with the reference image in different color spaces. The difference image is thresholded with several criteria based on pixel intensity. Moreover, for real-time issues, only some pixels are analysed. Pixels are sampled at regular interval and then analysed to determine if they belong to the background scene or to the foreground. We call pixel of interest a pixel of the foreground. The neighbour pixels of a pixel of interest are also analysed since they are likely to be of interest too.
To categorise the type of a pixel, four consecutive criteria are applied in different color spaces:

- The difference between the red, green and blue values of the pixel in the current image and in the reference image are computed. If these differences are less than a threshold, the pixel is considered as background else the next criteria is used.

- The intensity value Y of the pixel is considered in the YUV color space. Two cases can happen: the dark case ($Y_{ref} < 100$) and the clear one ($Y_{ref} \geq 100$) where Y_{ref} is the Y value intensity in the reference image. The difference of the Y values in the reference image and in the current image are compared to the corresponding threshold. If the difference is less than the threshold the pixel is considered as a pixel of interest else the next criteria is applied.

- The color values U,V of the pixel are considered in the YUV color space. Two cases can also happen: depending on the previous dark and clear cases. The absolute value of the difference of the U and V values between the reference image and the current one are compared to suitable threshold. If the two differences or the sum of the differences are more than the threshold, the pixel is classified as a pixel of interest else the next criteria is applied.

- The criteria is based on the HSV color space. This stage aims at removing shadow based on the chrominance of the moving pixel. Two cases are considered: the pixels of the current and reference images are colored ($S \geq 0.2$) or only one pixel of the two is colored. The difference of the H and S values are compared to a threshold accordingly. If one of the differences are less than a threshold, the pixel is considered as a pixel of interest.

- If none of these criteria labels the pixel as of interest then it is considered as a background pixel.

The resulting binary image is compressed in the *RLE* (Run Length Encoding) format. This format codes the repetition of a same pixel: the repeated pixels are stored as a single data value (the value of the pixel) and a counter (the quantity of consecutive appearance of the pixel). We can also notice that all the parameters of this segmentation algorithm have been tuned manually to obtain the best segmentation for human posture recognition process.

The CMM (Centre de Morphologie Mathématique) approach is based on several successive steps to refine progressively the following information [Lerallut, 2006]:

- The difference between the current image and the reference image is processed in the RGB space. A set of pixels of interest is then obtained. This set gives the position and an approximation of the contour of the person. This set may contain a lot of noise due to similarity in the texture between the object of interest and background or because the object is not contrasted enough compared to the background.

- To take into account the well known problem of shadow, there is a step in *watershed algorithm* which removes projected shadows in the scene. The operation is based on the fact that a projected shadow does not modify the chrominance of the region but it decreases the intensity reflected by this surface [Cavallaro et al., 2004].

- The color gradient of the images are computed to detect edges in the image. The edge of the current image is combined with the edge of the same image without shadow to obtain robust contours. An erosion is made to compute interior markers, and a dilatation is made to compute the exterior markers. A watershed algorithm is then applied with the markers to obtain the best contour between the markers.

These both segmentations provide different types of silhouette. The *watershed algorithm* gives compact silhouettes but tends to over-segment people evolving in the scene. In contrast, the *VSIP algorithm* detects pixels which belong to the person but the silhouette is less compact than *watershed* one with some holes. The two algorithms have been manually tuned to obtain good silhouettes with the test sequences. Examples of segmented silhouettes obtained with the both two algorithms are shown in figure 6.14. The first row shows a leak with the *watershed* silhouette due to the property of the watershed algorithm. The following two rows show under-segmented *VSIP* silhouettes with some holes. The both algorithms treat about 25 frames by second for color images 388x284 pixels (without taking into account the reading/writting of the images).

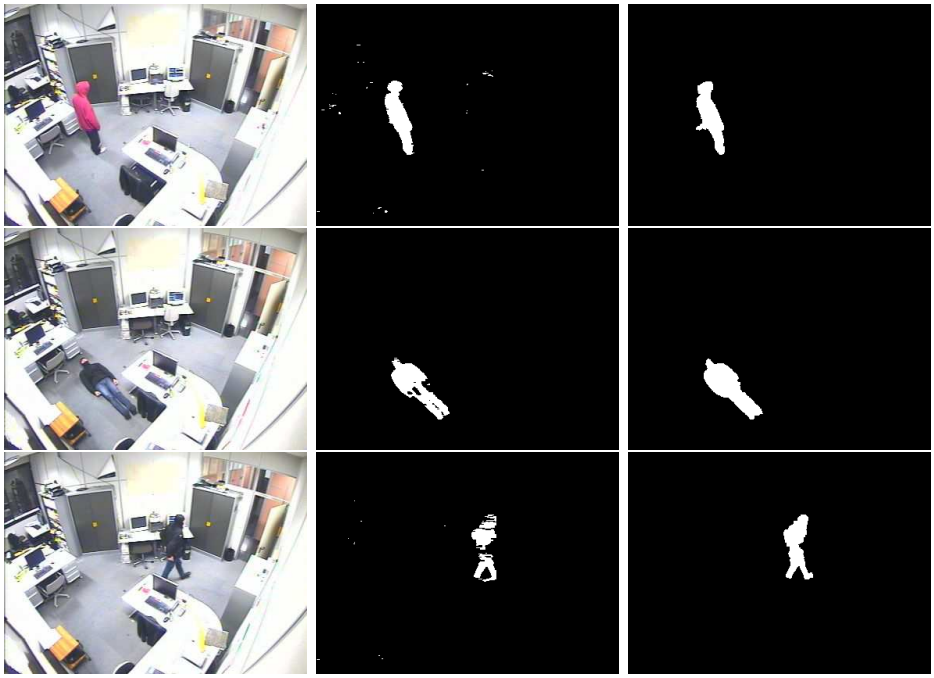

Figure 6.14: Segmentations of the image in the first column obtained according to the *VSIP algorithm* (second column) and the *watershed algorithm* (third column).

The third segmentation algorithm has been used for gait analysis purpose: *gait segmentation*. A semi-automatic procedure is used to detect the bounding boxes which contain the people evolving in the scene. The bounding boxes are manually

outlined in the starting, middle and ending frames. The bounding boxes of the intermediate frames are linearly interpolated from these manual ones, using the upper left and bottom right corners of the boxes. This method works well for cases where there is fronto-parallel and constant velocity motion. A background model of the scene is built by computing statistics of the RGB values at each pixel outside the manually defined bounding boxes. The mean and the covariance of the RGB values at each pixel are computed.
For pixel in the bounding boxes, the Mahanalobis distance is computed in RGB space for the pixel value from the estimated mean background value. Based on this distance the pixel is classified into foreground (the person) or background pixel. The decision is taken by computing the foreground and background likelihood distributions on each frame by using the iterative expectation maximisation (EM) procedure. Examples of obtained silhouettes are given in figure 6.15. These silhouettes are very noisy due in particular to the detection of the shadow of the person.

Figure 6.15: Several silhouettes obtained with the segmentation algorithm used in gait analysis.

The properties of the three segmentation algorithms allow to test the robustness of the proposed human posture recognition according to different types of segmentation: noisy over segmented silhouettes (*watershed segmentation*), under segmented silhouettes (*VSIP segmentation*) and very noisy over segmented silhouettes (*gait segmentation*).

6.4.2 Acquisition Context

The proposed human posture recognition has been tested with different image sequences:

- Different image sequences have been acquired from a non optimal camera viewpoint (the people evolving in the scene are not facing the camera). Four different people act the postures of interest by rotating around themselves to have many points of view.

- The approach has also been tested on sequences acquired for gait analysis purpose. The "walking" posture, shown in figure 6.16, has been added to the postures of interest set in order to adapt the recognition to this application.

Figure 6.16: The "walking" posture avatar from different points of view.

6.4.2.1 Own Sequences

Different sequences have been acquired in our laboratory. Four people act the different postures of interest by rotating around themselves to acquire many points of view. A ground truth is associated to each image sequence as described in 6.1 for more than 1000 frames. A sample of test images is given in figure 6.17.
All the results are given for the optimal rotation step of 36 degrees. Table 6.4 shows the recognition rates of general postures for the different silhouette representations with the *watershed segmentation* algorithm (similar recognition is obtained with the *VSIP segmentation* algorithm) which are equivalent to the rates obtained with synthetic data. The H. & V. projections representation gives the best results as shown on the last row of the table 6.4 and it is studied in more depth in the following. The ability of the H. & V. projections representation to smooth silhouette explains these results. This representation takes into account the variability of the 3D avatar from the observed person and silhouette misdetection.

Ground Truth / Recognition	Standing	Sitting	Bending	Lying
Geometric features	94	82	77	83
Hu moments	68	73	27	35
Skeletonisation	93	68	82	65
H. & V. projections	100	89	78	93

Table 6.4: General postures recognition rates for the different silhouette representations with watershed segmentation.

Table 6.5 and table 6.6 show the confusion matrices for the recognition of the general postures according to the segmentation approach. The obtained results are satisfactory (the rate of correct recognition is above 80%) and show the robustness of recognition of general postures in all possible orientations.

Ground Truth / Recognition	Standing	Sitting	Bending	Lying
Standing	**271**	25	2	0
Sitting	0	**196**	13	36
Bending	0	0	**54**	2
Lying	0	0	0	**484**
Detected/total	271/271	196/221	54/69	484/522
Success percentage	100	89	78	93

Table 6.5: Confusion matrix for general postures recognition for H. & V. projections with watershed segmentation.

Ground Truth / Recognition	Standing	Sitting	Bending	Lying
Standing	**271**	46	6	0
Sitting	0	**167**	0	40
Bending	0	8	**63**	2
Lying	0	0	0	**471**
Detected/total	271/271	167/221	63/69	471/513
Success percentage	100	75	91	92

Table 6.6: Confusion matrix for general postures recognition for H. & V. projections with VSIP segmentation.

The confusion matrices for the recognition of detailed postures are given in table 6.7 and in table 6.8 and gives similar results for the different segmentation approaches. For instance, the lying with curled up posture is recognised with 63% for *watershed segmentation algorithm*, and with 60% for *VSIP segmentation algorithm*. This posture is more mis-recognised with the lying with spread legs posture (81 cases for *watershed segmentation* and 86 cases for *VSIP segmentation*), than the sitting on the floor posture (25/28 cases) and sitting on a chair posture (11/12 cases) for the both segmentations. Postures are often mis-classified with another posture of the same category (e.g. sitting on the floor and sitting on a chair) due to the ambiguous cases previously described. Even if the (H.& V.) projections silhouette representation manages the differences between the 3D posture avatar and the posture acted by a person there is some mis-recognition in extreme cases. Few errors occur because the 3D posture avatars represent specific postures and do not take into account the variability of these postures

(e.g. the arm can vary when raised for the standing posture with one arm up). The processing time is about 5-6 frames by second by taking into account the following tasks: reading task, segmentation task, classification task, tracking task and posture recognition task.

Recognition \ Ground Truth	1	2	3	4	5	6	7	8
Standing with one arm up (1)	**79**	21	13	3	0	0	0	0
Standing (2)	5	**67**	1	22	0	2	0	0
T-shape (3)	27	3	**55**	0	0	0	0	0
Sitting on a chair (4)	0	0	0	**51**	44	7	0	11
Sitting on the floor (5)	0	0	0	1	**100**	6	0	25
Bending (6)	0	0	0	0	0	**54**	2	0
Lying with spread legs (7)	0	0	0	0	0	0	**162**	81
Lying with curled up legs (8)	0	0	0	0	0	0	45	**196**
Detected/total	79/111	67/91	55/69	51/77	100/144	54/69	162/209	196/313
Success percentage	71	74	80	66	69	78	77	63

Table 6.7: Confusion matrix for detailed postures recognition with (H. & V.) projections approach obtained with watershed segmentation.

Recognition \ Ground Truth	1	2	3	4	5	6	7	8
Standing with one arm up (1)	**79**	21	13	0	0	5	0	0
Standing (2)	5	**67**	1	36	10	1	0	0
T-shape (3)	27	3	**55**	0	0	0	0	0
Sitting on a chair (4)	0	0	0	**40**	20	0	0	12
Sitting on the floor (5)	0	0	0	1	**106**	0	0	28
Bending (6)	0	0	0	0	8	**63**	2	0
Lying with spread legs (7)	0	0	0	0	0	0	**158**	86
Lying with curled up legs (8)	0	0	0	0	0	0	35	**192**
Detected/total	79/111	67/91	55/69	40/77	106/144	63/69	158/195	192/318
Success percentage	71	74	80	52	74	91	81	60

Table 6.8: Confusion matrix for detailed postures recognition with (H. & V.) projections approach obtained with VSIP segmentation.

Figure 6.17: Sample of the image sequences used in the tests.

Segmentation Errors

The results are given up to now for a bounding box overlapping rate equal at least 70%. By changing this value different cases can be considered. A value near 100 means segmentation is perfect whereas a lower value corresponds to cases of mis-segmentations. Figure 6.18 gives the general and detailed posture recognition rates for standing postures with different levels of bounding box overlapping. The histogram are computed for several overlapping rates: $100 \pm 2.5\%$, $95 \pm 2.5\%$, $90 \pm 2.5\%$, $85 \pm 2.5\%$, $80 \pm 2.5\%$, $75 \pm 2.5\%$, $70 \pm 2.5\%$, $65 \pm 2.5\%$, $60 \pm 2.5\%$. The curve represents the number of considered cases. Under 70% of overlapping only 1 or 2 cases are considered, then their values are not concluding. It is the reason why the results are previously given for an overlapping rate superior to 70%. By analysing the histograms, both general and detailed postures recognitions are correct (above 65%) for all tested situations (with at least 70% of overlapping). Then the proposed human posture recognition approach is able to recognise detailed postures even if the segmentation is not perfect as shown on the figure 6.18.

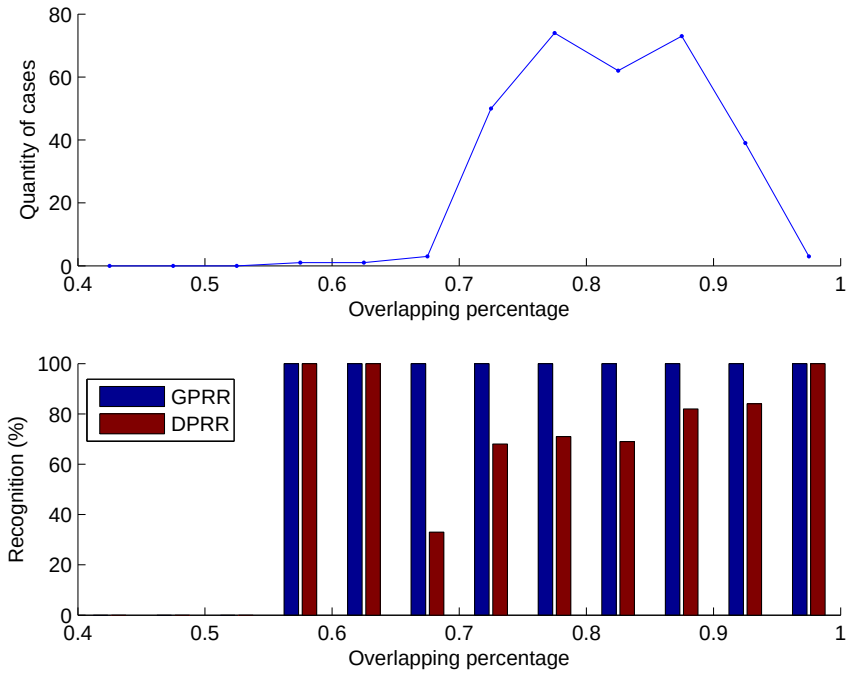

Figure 6.18: General (GPRR) and detailed (DPRR) recognition rates for standing postures with different overlapping percentages with the watershed algorithm. The number of considered cases is also given.

6.4.2.2 Walking Sequences

The proposed approach has also been tested using different image sequences acquired for human gait analysis purpose. The walking posture (figure 6.16) has been added to the set of posture of interest in order to better fit with the gait postures. This posture has been modeled very easily (in 5 minutes) by animating a 3D avatar body part by body part to determine the set of parameters (the articulation angles) corresponding to the walking posture.
The first sequence has been used in [Sidenbladh et al., 2002] for human motion tracking. The sequence is composed of one person walking from right to left: see figure 6.19. Due to the contrast between the person and the background the segmentation is perfect (the results are obtained with the *watershed algorithm*). A manually ground truth has been made by annotated for each frame the posture of the person: standing with arms near the body or walking posture. The recognition results are displayed in the graph of figure 6.20. 78 postures are correctly detected on the total of 81 postures. Moreover, the walking periodicity (the repetition of the standing posture followed by walking posture) is clearly identified. The four walking cycles are correctly detected.

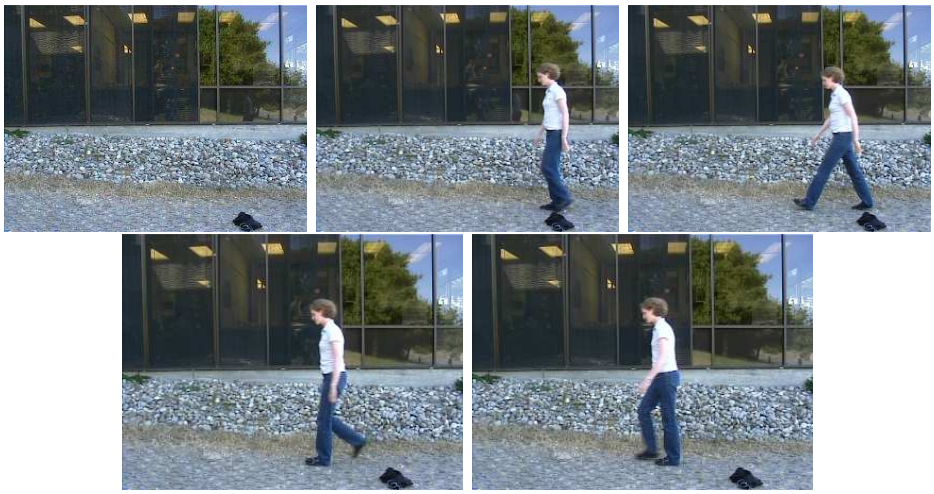

Figure 6.19: A person walking straight from the right to the left

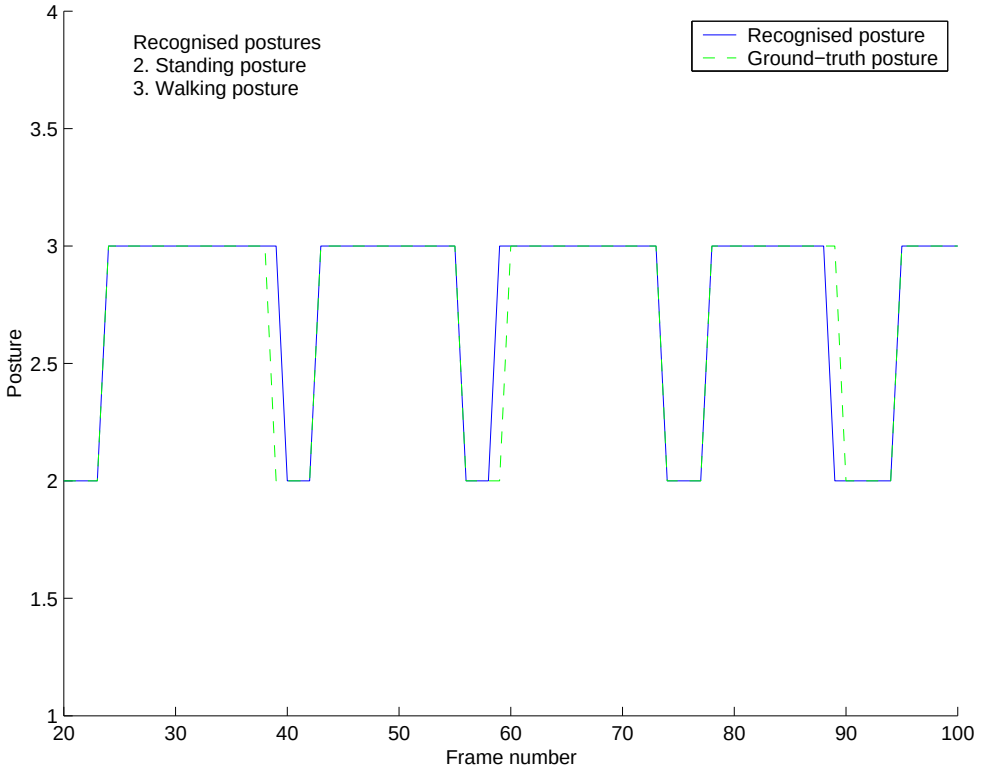

Figure 6.20: Results obtained on walking sequence with H. & V. projections representation and ground truth.

The other sequences are extracted from the HumanID gait challenge problem data set ([Sarkar et al., 2005]). This data set has been designed to evaluate the state of the art in automatic gait recognition and to characterise the effects of different environmental and walking conditions: concrete ground/grass ground, different shoes (sneakers, sandal, ...). The people evolving in the scene walk along a demi-ellipse observed by two cameras. The segmented silhouettes are also available as shown in figure 6.21 and they are obtained with the *gait segmentation algorithm*. The proposed approach is tested on these silhouettes. The original data-base contains 122 image sequences. We test our human posture recognition on five sequences acquired with two different points of view and different grounds (concrete and grass). This corresponds to 65 walking cycles and 911 frames. The result of the recognition for one of the tested sequence is given in figure 6.22. 162 postures are correctly detected for 186 considered cases. The walking periodicity is also clearly identified even if some one are not completely detected such as the ninth cycle. By considering all the five sequences, 711 postures are recognised among the 911 total cases.

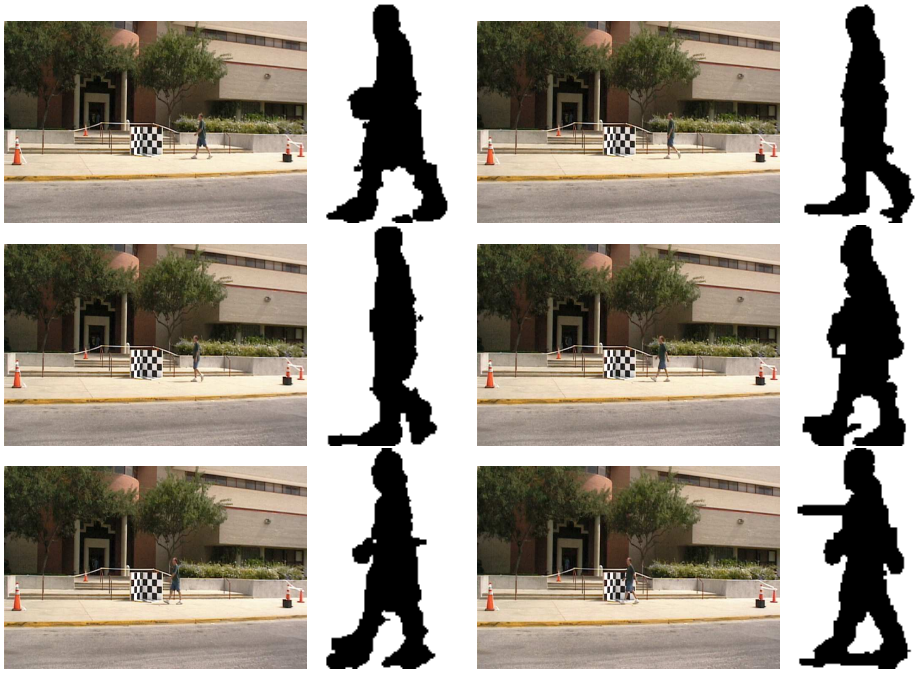

Figure 6.21: A person walking along a demi-ellipse and its corresponding silhouettes.

By adding the walking posture to the posture of interest set, the proposed human posture recognition approach shows its adaptability to the need of a given application. Moreover, the approach combining the 3D posture avatar and the horizontal and vertical projections has shown its robustness to the erroneous silhouettes by testing it with noisy silhouettes.

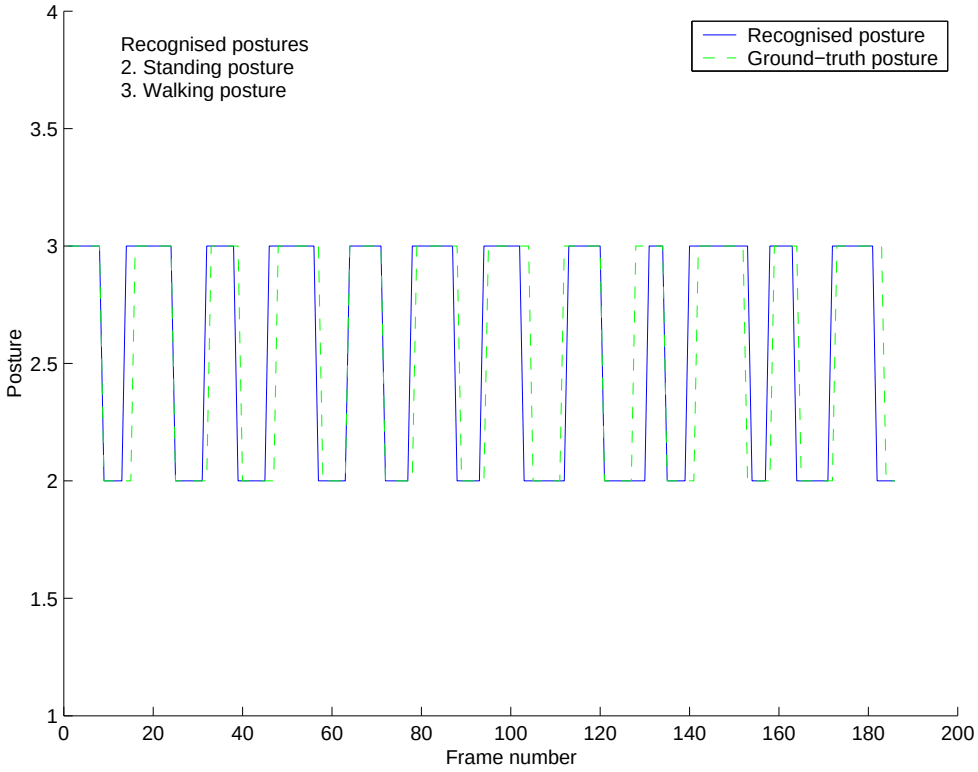

Figure 6.22: Results obtained on walking binary sequence with H. & V. projections representation.

6.5 Conclusion

The proposed human posture recognition approach has been successfully tested using both synthetic and real data. A ground-truth is defined to evaluate the proposed approach composed for each person in the video sequence of:

- a single identifier,
- a bounding box which localises a person in an image,
- the posture which evaluates the result of the recognition.

The bounding box is used to test the quality of the segmented silhouette by computing the overlapping rate of the bounding boxes defined in the ground-truth and detected by the vision algorithms.
Synthetic data were used to evaluate the proposed approach. The advantages of using such data is that many viewpoints can be considered and ground-truth data are automatically generated. We have used synthetic data to compare the different silhouette representations and to determine the optimal rotation step defined in section 6.3.2. The evaluation is performed in terms of posture recognition and computational time efficiency. The horizontal and vertical projections representation defined in 5.3.2.4, gives the best recognition. Moreover a rotation step of 36 degrees is chosen as optimal since it gives the best compromise between high recognition and low computation time.
The posture recognition algorithm has also been tested with different real data. Real data introduces some difficulties like segmentation errors (noisy silhouettes) or the problem of intermediate postures (postures can be slightly different according to the postures of interest). A data base has been acquired to evaluate the approach. It is composed of more than 1000 manually annotated frames for four people and the eight postures of interest. The horizontal and vertical projections representation gives the best results since it takes into account little holes which can appear in the segmentation or small changes in the posture. The other representations are more sensitive to the segmentation errors. In particular, since Hu moments are computed on the whole silhouette, an error on the silhouette implies erroneous terms for the Hu moments representation. Moreover, skeletonisation is based on the contour of the silhouette then it is sensitive to segmentation errors. Hu moments representation has also an other problem. Since the Hu moments are independent from the scale and orientation, standing posture and lying posture can be confused.
The approach has also been tested on data set acquired for a gait analysis purpose. A new posture of interest, the walking posture, has been added in order to better fit with the gait postures. The approach clearly identifies the gait period. The approach has been successfully tested on outside/inside sequences and for over/under segmentations. It has shown its robustness with missing body parts such as extremities (head, feet). It can be easily adapted to any view points thanks to the virtual camera. Moreover posture avatars can be added/removed

according to the specificity of the application.
The approach has been tested under three different segmentation algorithms: *VSIP segmentation*, *watershed segmentation* and *gait segmentation*. We have tuned the two first algorithms to obtain pretty silhouettes, whereas we have just use the available silhouettes obtained with the third algorithm. These algorithms provide different type of silhouette: noisy over segmented silhouettes (*watershed segmentation*), under segmented silhouettes (*VSIP segmentation*) and very noisy over segmented silhouettes (*gait segmentation*). The posture recognition is similar for the two first algorithms by representing the silhouettes with the (H. & V.) projections. The segmentation problems associated to these segmentation algorithms (hole in the silhouette or over segmentation) are taking into account with the smoothing power of the (H. & V.) projections. Finally the proposed approach gives good recognition rate for the *gait segmentation*, by considering the "simple" case of the walk (i.e. the different standing postures).
As said in chapter 2, the proposed approach extends works described in [Zhao and Nevatia, 2004] by considering four general postures (standing, sitting, bending and lying). Our proposed hybrid approach takes advantages of the techniques described in chapter 2, by combining the 2D techniques (real-time computation) and the use of 3D posture avatar (independence from the view point). The fourth constraints given in chapter 3 have been respected. The approach works in real time (5-6 frames by second). The generation of the 3D avatar silhouettes is the most time consuming step (1,28 second to generate 100 silhouettes). The approach is independent from the camera viewpoint by using a virtual camera which has the same characteristics than the real one. The approach is fully automated by adapting the height of the 3D posture avatar to the observed person height. Finally, the approach uses only one static camera by taking advantage of a knowledge base, in particular the calibration matrix of the camera. However, the approach is limited in terms of quantity of interest postures. The computing time increases when more postures are considered. Moreover, when more postures are considered the number of ambiguity cases increases. Finally, we have made the strong hypothesis that the observed person is isolated. The approach does not take into account the problem of occluded person.
The next chapter, focuses on how the postures recognised by our posture recognition approach can be used in action recognition.

Chapter 7

Action Recognition using Postures

7.1 Introduction

In the previous chapters, the proposed human posture recognition approach has been presented which combines 2D techniques and the use of 3D posture avatars. The results of the approach (the filtered postures) can be exploited in the human behaviour analysis.
More generally, actions can be classified in three categories:

- the self-actions where only the concerned person acts: walking, running, sitting, falling, pointing, jumping, etc...
- the actions involving a person with contextual objects: drinking, eating, writing on a whiteboard, typing (using computer keyboard), taking an object, putting an object, etc...
- the actions where several individuals interact with each other: shaking hand, kissing, fighting, etc...

In this chapter, a focus is made on how self actions can be recognised thanks to the filtered postures because the main problem to recognise the other types of action is the object detection which is not the scope of this work.
Existing techniques to recognise actions are briefly presented in the first section. The modeling of self-actions is presented in the second section and the recognition in the third section. Finally some recognition results are given for the falling and walking examples.

7.2 Existing techniques

The action recognition task can be considered as a time-varying data matching problem. Classical methods to solve this kind of problem may involves: dynamic

time warping (DTW), hidden Markov model (HMM) and neural network. Two different approaches can be used to analyse human action.

Probabilistic approach

Probabilistic approaches need a learning phase to make a probabilistic model of the desired behaviour. An image sequence of a given action is converted into a static template which represents the action. Dynamic time warping algorithm is then used to measure similarity between template and detected sequence which may vary in time or speed. In [Bobick and Davis, 2001], the action is represented with the motion energy image (MEI) and the motion history image (MHI). The MEI is obtained by cumulating binary motion image: each pixel corresponding to a motion in the image is considered like a pixel of interest. The MHI is obtained similarly by indicating the quantity of movement for each pixel. These images are then represented with the seven Hu moments and compared to stored samples by Mahanalobis distance.
In [Chen et al., 2006], the authors propose a HMM-based methodology for action recognition using skeletonisation as a representative descriptor of human posture. Each action is described by representative skeletons. Then a HMM is optimised to represent the desired action.
The probabilistic approaches are easy to implement because based on well known learning tool such as neural network. But the main drawback is that it is difficult to know how these approaches work: what does represent a neuron of a given neural network?

Deterministic approach

In contrast to the previous approaches, the deterministic ones do not need a learning phase. An expert decides explicitly the rules to represent a given behaviour. In [Cucchiara et al., 2003], a finite state machine is presented to identify a fall: each state is represented by a posture and an information about the movement of the person (static or moving).
Approaches based on constraint resolution are able to recognise complex event with several actors. In [Vu et al., 2002], the authors present an approach to optimise the temporal constraint resolution by ordering in time the sub-scenarios of the scenario to be recognised. An efficient algorithm of this approach takes advantage of a pre-compiled stage of scenario models to recognise in real time complex scenarios with a large number of objects is described in [Vu et al., 2003].
The deterministic approaches are easily understandable because they are based on expert rules. They do not take into account the dimension of the data and we must take care of combinatory explosion during implementation.

7.3 Action Representation

Each action of interest is represented by a series of postures and are modeled using a finite state machine as shown in figure 7.1. We choose this kind of representation because it can easily model the self actions with very simple rules. Each state of the finite machine is characterised by one or several postures denoted P, and their minimal (noted min) and maximal (noted max), authorised occurring number of successive postures. These thresholds are used to estimate the duration of each posture. A state is defined with several types of postures to take into account uncertainty in the recognition of posture.
The postures can either be general or detailed postures depending on the action to model.

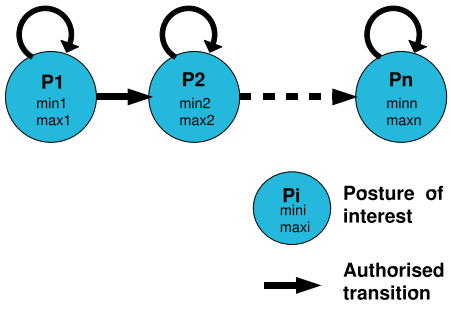

Figure 7.1: Finite state machine modeling an action with n states.

To optimise the discrimination of the desired actions, we define the postures characterising the states and the associated threshold values by running sufficient enough experimentations. These threshold values are dependent on the framerate of the studied video sequences.
The desired actions are recognised through the associated finite state machines and processed with video sequences.

7.4 Action Recognition

During the posture recognition process, a stack of filtered postures is associated with each person detected in the scene. Any new recognised postures are pushed into the stack, and the number of occurrence is increased by one. This processing enables to recognise in efficient way the sequence of postures modeled by finite state machines.

The action recognition is performed by comparing the different finite state machines which represent the self actions we want to recognise, with the stack associated to each detected person. When the finite state machine associated to the desired action is recognised, the action is associated to the corresponding

person.
This preliminary recognition method has some limitations. Noise and misrecognition of posture on few frames may imply that the actions are not recognised. This issue is partially solved by using the filtered postures and by representing a state by one or several postures.
In the next section, the example of the falling and walking/running actions are studied.

7.5 Example: the people falling action

The automatic detection of people falling is of a great interest for medical and home-care applications. It is important to detect this kind of actions to trigger an appropriate alarm to warn medical staff for example. Falling is an action based on general postures and it is considered as a passage from a standing posture to a lying posture. The final state machine of the falling action is represented in figure 7.2. The second state e is defined as a combination of bending and sitting postures. This state is introduced to model the intermediate postures of the falling action. Indeed, the postures between standing and lying ones can either be detected as sitting or bending postures according to the type of fall.

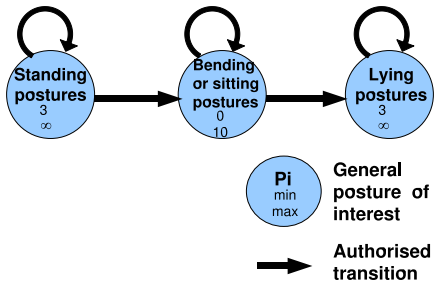

Figure 7.2: Finite state machine which represents the falling action.

The maximum threshold value of the first state is fixed to ∞. Moreover, the minimum threshold of the final state is fixed to 3, to robustly detect the falling action.
Validation tests have been performed on different acted sequences for several types of fall as shown in figures 7.4, 7.5, 7.6 and 7.7:

- falling ahead (performed 4 times),
- falling behind (performed 3 times),
- and sinking down (performed 3 times).

The ten falling actions have been correctly recognised as shown table 7.1. The video sequences are challenging because the persons fall and stand up immediately.

	TP	FP	FN
Recognised falling action	10	0	0

Table 7.1: True positive (TP), false positive (FP), and false negative (FN) recognition of the falling action.

7.6 Example: the walking action

The walking action is modeled through detailed postures. It is defined as a succession of standing posture with arm near the body followed by the walking posture (figure 7.3).

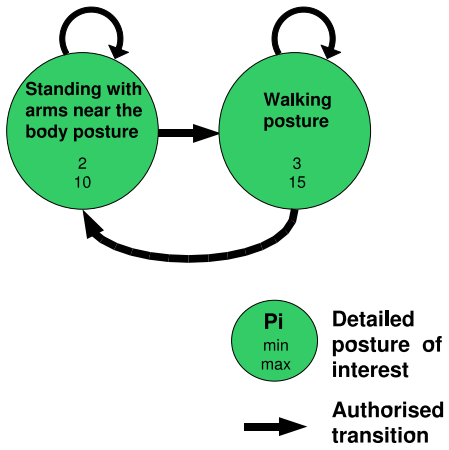

Figure 7.3: Finite state machine which represents the walking action.

The validation tests have been realised on the sequences acquired for gait analysis purpose and described in section 6.4.2.2. The walking action has been successfully recognised among five sequences corresponding to 65 cycles of the walking action (succession of standing and walking postures) as shown in table 7.2. 62 cycles are correctly detected, and 3 cycles are mis-recognised (the cycles are confused with another one).

	TP	FP	FN
Recognised walking action	62	0	3

Table 7.2: True positive (TP), false positive (FP), and false negative (FN) recognition of the walking action.

The walking action is based on detailed postures which multiplies the confusion.

7.7 Conclusion

In this chapter, a method to model and recognise self actions based on posture has been presented. These actions are modeled using general or detailed postures depending on the accuracy needed to represent the actions. Actions are modeled with a finite states machine where each state consists of one or several postures with minimal and maximal authorised occurrence value of successive postures. The approach has been successfully tested for the falling action (based on general postures) and walking action (based on detailed postures) actions.
These preliminary results are encouraging but some problems remain to be solved. Particularly, the uncertainty due to posture mis-detection should be modeled. With the proposed algorithm, a finite state machine must be completely detected to recognise an action. The action recognition algorithm can be improved by adding to the state information about the movement of the person: the person is moving or not. Moreover, the transition between the states can be represented by probability density functions to finally obtain a hidden markov model (HMM). This representation allows to be independent from the frame rate of the video.

Figure 7.4: Example of the fall action.

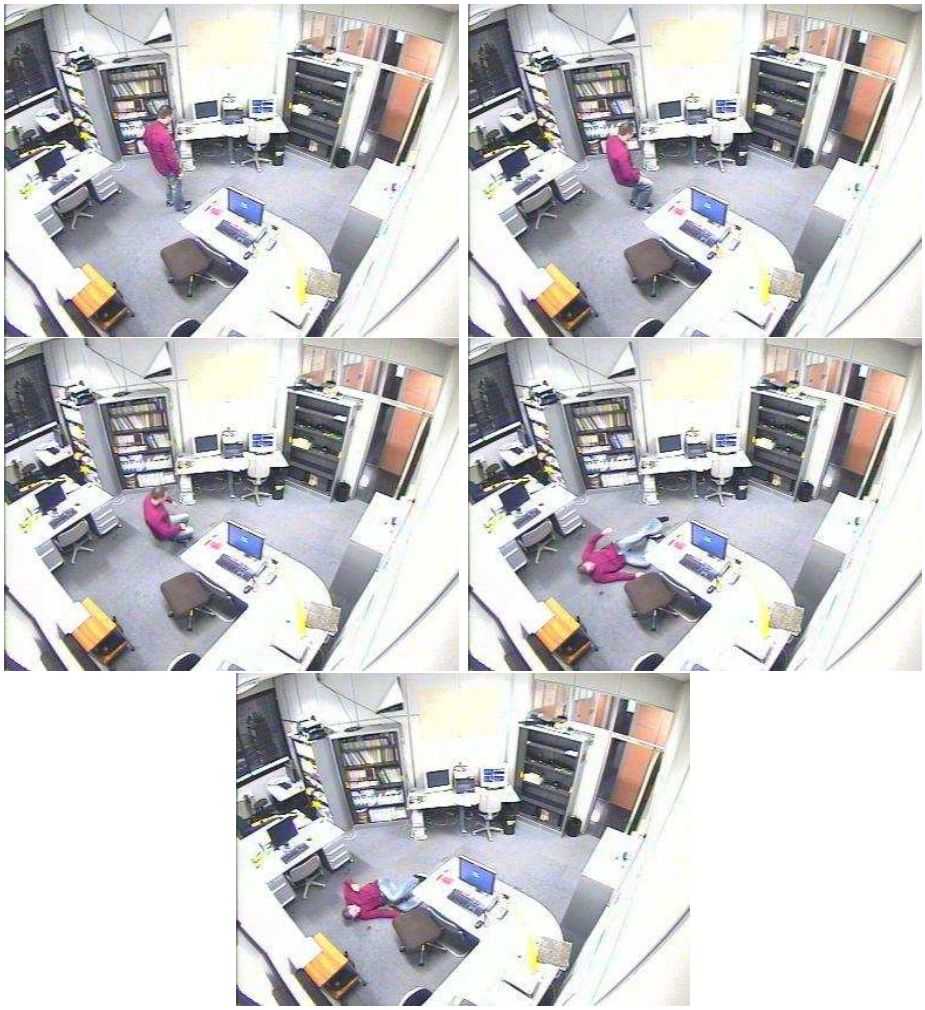

Figure 7.5: Example of the fall action.

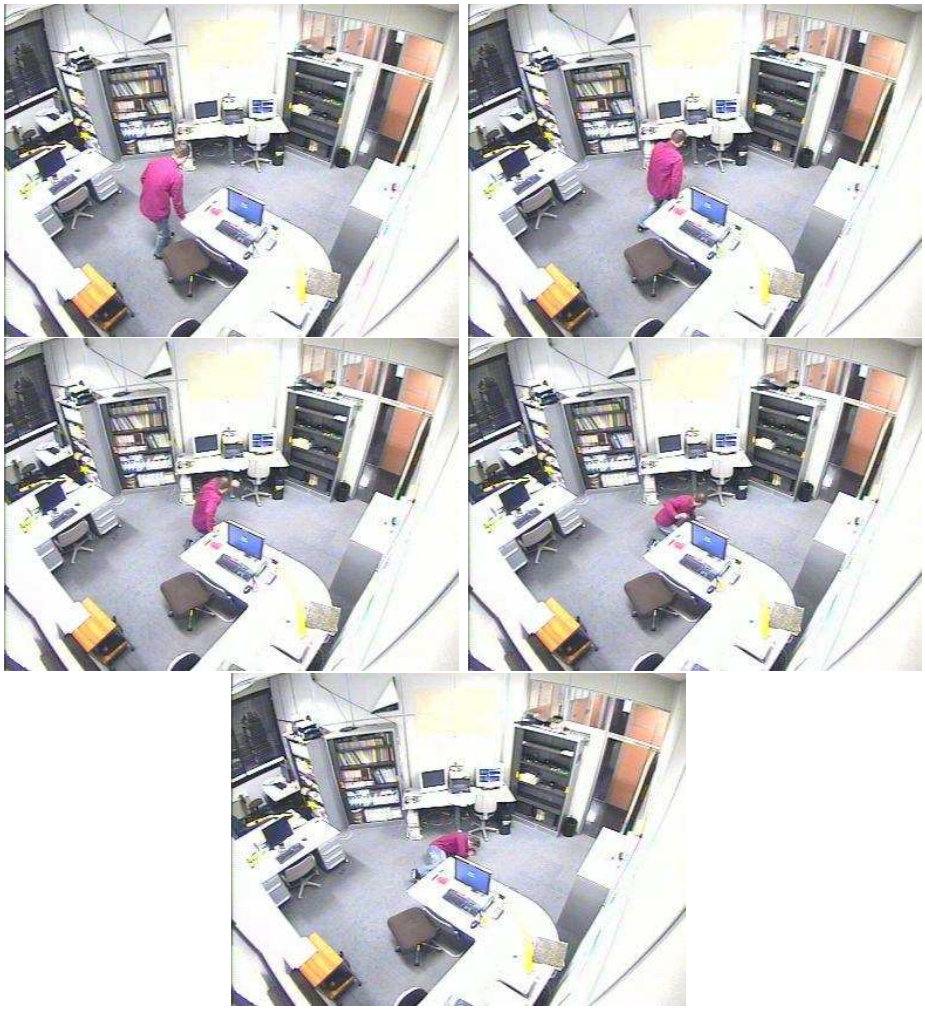

Figure 7.6: Example of the fall action.

Figure 7.7: Example of the fall action.

Chapter 8

Conclusion

In this thesis we have proposed a new approach to recognise human posture. This approach combines 2D techniques and the use of the 3D posture avatars. The 2D techniques are used to keep a real-time processing whereas the 3D posture avatar allows some independence from the point of view.
The proposed approach takes as input the detected 2D moving silhouette region corresponding to the detected person and the estimated 3D position. The approach is composed of four tasks:

- The 3D posture avatar silhouettes are generated by projecting the avatar on the image plane by using a virtual camera. The 3D avatar is placed in the virtual scene where the observed person is detected, then the avatar is oriented for different angles to generate all possible silhouettes with respect to the pre-defined postures.

- The silhouettes are represented and compared according to four 2D techniques: geometric representations, Hu moments, skeletonisation and horizontal and vertical projections.

- The posture of the detected person is determined by keeping the most similar generated silhouette according to the previous task.

- The posture filtering task filters out the erroneous postures detected in the previous task by exploiting their temporal coherence. The posture stability principle states that for a high enough frame-rate the posture remains similar within few successive frames. The tracking information of the recognised person is used to retrieve the previously detected postures. These postures are then used to compute the filtered posture (i.e. the main posture) by searching the most frequently appearing posture during a short period of time.

An overview of the contributions of this work is given in the next section. Then a discussion is made in particular to show the limitations of the proposed approach. Finally, future work are proposed to improve the approach in the last section.

8.1 Overview of the Contributions

- The **3D posture avatars** have been introduced to model the human postures to be recognised as described in chapter 4. It is inspired by the previous work in computer graphics and has been adapted by proposing a simplified model to posture recognition purposes. A 3D posture avatar is composed of a 3D human model, which defines the relation between the 20 body parts, a set of 23 parameters, which defines the position the different body parts and a set of body primitives which defines the visual aspect of the body parts. The proposed 3D human model contains also ten joints which are the major body articulations and the twenty body parts. The articulations are represented with Euler angles to represent the eight postures of interest: standing with one arm up, standing with arms along the body, T-shape posture, sitting on a chair, sitting on the floor, bending posture, lying with spread legs and lying with curled up legs. The body primitives are represented with polygons mesh to obtain a realistic 3D human model in order to generate synthetic silhouette close to the real world. Such body primitives enhance the recognition quality. The proposed 3D posture avatars are independent from the body primitives used to represent the body parts. Indeed, different primitives can be used to visualise different types of avatars adapted to the observed person. The parameters of the articulations can be modified to model intermediate postures. Moreover, the 3D posture avatars are classified in a hierarchical way from general to detailed postures.

- A novel **hybrid approach** is proposed in chapter 5 to recognise human postures in video sequence. The approach is based on the characterisation of the detected person silhouette. The approach combines 2D techniques and the use of the 3D posture avatars previously described. The 2D techniques are used to keep a real-time processing whereas the 3D posture avatars allow to have a certain independence from the view point of the camera. Several 2D techniques have been tested to represent the silhouettes:

  - the first one is based on the combination of different geometric features such as area, centroid, orientation, eccentricity and compactness,
  - the second one is based on the seven Hu moments,
  - the third one, referred as skeletonisation, uses salient points on the contour,
  - and the last one is based on the horizontal and vertical projections of the silhouette.

  The 2D techniques are selected according to the segmentation quality and computation time. Experimentations have shown that the silhouette representation based on the horizontal and vertical projections works the best.

- A **silhouette representation evaluation** has been performed in section 6. Since the 3D posture avatars are realistic enough, they are used to generate

synthetic data for several point of views. The 3D posture avatar involved in the data generation is different from the 3D posture avatar involved in the recognition process. There are three main advantages in using synthetic data:

  - First, lots of data can be easily generated for any point of view and the virtual avatar can be placed at any place.
  - Second, the approach can be studied according to different problems: segmentation quality, intermediate postures, ambiguous postures and variability between the observed person and the 3D avatar.
  - Finally, a ground-truth file can be automatically associated at each generation step.

  The synthetic posture data-base has been generated for 19 points of view, for 10 postures and for 360 orientations separated by one degree. The points of view are localised on a quarter of circle around the person for each five degree from 0 to 90 degrees populating the data-base with 68400 silhouettes. The proposed approach has been validated on both synthetic and real data. The horizontal and vertical projections representation gives better posture recognition than the geometric features, the skeletonisation and the Hu moments representations because this representation is more robust to noisy silhouettes and intermediate postures.

- An exhaustive **characterisation of ambiguous postures** are also studied with the synthetic data-base as shown in chapter 6. Ambiguity cases happen when silhouettes representative of different postures have the same projections on the image for a given point of view. The ambiguity is then characterised by a posture and an orientation according to a given point of view. It is dependent on the silhouette representation. This a priori knowledge can be exploited in the recognition process to associate a confidence value with the recognised postures.

- The results of the proposed approach, the recognised postures, have been used for **action recognition** in chapter 7. The targeted actions are self-actions, that is to say actions where only the considered person is involved. The actions are modeled with a finite state machine where each state consists of a posture and a minimal/maximal occurrence number of consecutive appearing postures. The approach has been successfully tested for the falling action and the walking action. The falling action is based on general postures whereas the walking action is based on the detailed ones. A new posture, the walking posture, has been easily added to the set of interest postures and shows the adaptability of the proposed approach.

- Moreover, during this work several tools have been developed:

  - The first tool consists of a **3D engine** able to visualise and manipulate a 3D posture avatar by moving the different body parts and to extract

the silhouette according to a virtual camera. The engine is based on the Mesa library facilities, combining orientation and translation transformations to animate the 3D posture avatar. This engine is a component for the tools described below.

- The second tool **animates the 3D posture avatar** and defines the parameters associated with a given 3D posture avatar. Each body part of the 3D model can be selected, and the parameters of the corresponding articulations can be modified to obtain the desired 3D avatar. The parameters can be saved and used with the previously described 3D engine.
- The third tool **generates exhaustively synthetic data** by defining different points of view and different orientations of the 3D posture avatars.
- The fourth tool **generates synthetic silhouette based on a trajectory**. A virtual scene is observed from the top (in vertical direction), the user draws a trajectory and selects the posture at the salient points of the trajectory. This tool is useful for demonstration purpose.
- The last tool is a **posture recognition prototype** which integrates the complete treatment of a video sequence from acquisition to posture recognition. It visualises the results of the recognition approach. A description of the prototype is given in appendix A.

8.2 Discussion

In the section 3.1.2, several constraints have been identified to propose a generic approach: real-time, independence from the point of view, fully automated approach and one monocular static camera. We detail below how these constraints have been solved:

- Real-time. The proposed algorithm is able to treat about 5 to 6 frames per second using real video stream. It has been shown to be efficient in recognising some actions such as the falling or the walking ones. This framerate is possible thanks to the use of the 2D representation of the silhouette.

- Independence from the point of view. In section 6.3.2 the approach has shown its independence from the point of view. The virtual camera allows the generation of the 3D posture avatar silhouettes using the same point of view than the real camera. Thus, a virtual camera can be associated to a real one for any position of the real camera.

- Automated approach. The approach is completely automatic and can be easily adapted to any video sequences. Moreover, this approach can be adapted to different types of application by modifying the set of postures of interest. A new posture of interest can be defined by determining a specific

set of parameters (the joint parameters) to represent the desired posture such as the walking posture.

- One monocular static camera. The approach works with only one monocular static camera by using a contextual knowledge base associated to the scene. In particular, the calibration matrix allows the computation of the 3D position of the people evolving in the scene and the initialisation of the virtual camera.

Since the proposed approach combines the 2D techniques and the use of 3D posture avatars, it takes the strengths of these techniques. In particular, the approach is independent from the view point of the camera by using the 3D posture avatars. The approach is relatively fast, and it is able to treat between 5 and 6 frames by second. Experimentations have shown that the approach gives good posture recognition rates for real data (above 80% for the general posture and around 75% for the detailed postures). The approach is robust to different types of segmentation. The approach has been tested with the *watershed segmentation* algorithm (which provides a noisy over-segmented silhouette), with the *VSIP segmentation* algorithm (which provides an under-segmented silhouette with some holes) and with the segmentation associated with the gait sequences (which provides very noisy over-segmented silhouette). The approach can be adapted to a specific application purpose, in particular the set of posture of interest can be modified to solve a particular problem. For instance, the "walking" posture is added to the set of posture of interest to recognise the "walk" action. However the proposed approach has some limitations. The main drawback of the approach is that its limitation in terms of the quantity of postures of interest. The first reason of this limitation is the time processing. The computing time increases when more postures are considered limiting the number of postures of interest. The second reason is the discriminating power. When more posture avatars are considered, the number of ambiguity cases increases. The second and related problem is the computation time. The silhouette generation of the 3D posture avatars represents the most expensive phase in terms of computation time. The generation time of 100 silhouettes corresponding to 10 postures avatars and a rotation step of 36 degrees is about 1.28 second. By only computing the generated silhouette, when the detected person has a sufficient displacement in the scene, the frame-rate is about 5 to 6 frames per second. To decrease this computation time, some improvements are necessary. For example, the set of posture of interest can be adapted by taking into account the recognised posture in the previous frame. Moreover, we have made the strong hypothesis that the observed person is isolated. But the observed person can be occluded by objects of the context, or she/he can interact with other people. Finally, in the approach the 3D posture avatar is adapted to the studied person by only taking into account the height of the person. But more information, can be use, the type of the clothes or the corpulence of the studied person.
The approach has been applied in a video understanding system. But this ap-

proach is quite generic and can be applied in other type of application which need the same requirements: real-time processing, viewpoint independence, automated approach and one monocular static camera. In particular, this approach can be used for a new form of human computer interface.

8.3 Future Works

This work can be improved in different ways classified in short and long term perspectives.

8.3.1 Short-Term Perspectives

Occlusion

The virtual scene can be used to handle the problem of occlusion. A 3D model of the scene can be displayed with the 3D posture avatars. By positioning correctly the 3D posture avatar in the scene, an occluded silhouette can be extracted and compared with the detected one. In this case the Z-buffer technique, described in section 5.2.2.3, cannot be used to extract silhouette since not only the posture avatar is in the virtual scene but also the contextual objects. A simple color segmentation can be envisaged to solve this problem by coloring the contextual objects with the background color. An example of an occluded silhouette is given in figure 8.1.

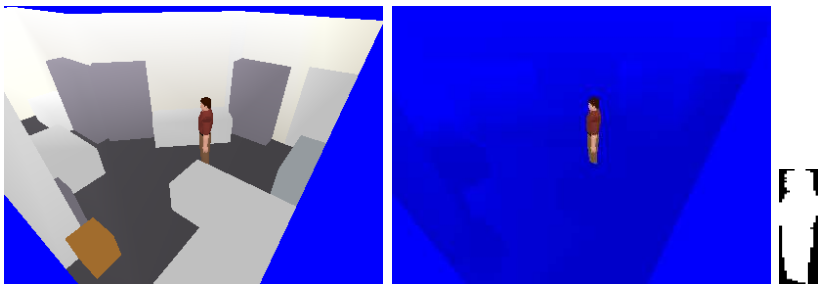

Figure 8.1: The objects of the scene are displayed in the virtual scene together with the observed person. These objects are colored in blue to make a simple color segmentation and to obtain an occluded silhouette.

Deformation with the virtual camera

During this work, some tests were achieved with images acquired with a CMOS sensor equipped with an objective with a large field of view (figure 8.2). Using such sensor implies geometrical deformations of the image. The virtual camera model can be improved to take into account the deformations of the real camera

in order to obtain deformed silhouettes of the 3D posture avatars. The deformed silhouettes will then be directly compared with the silhouette of the detected person. The linear model used in the calibration approach is not valid for this kind of image and another method of calibration must be used to handle these deformations.

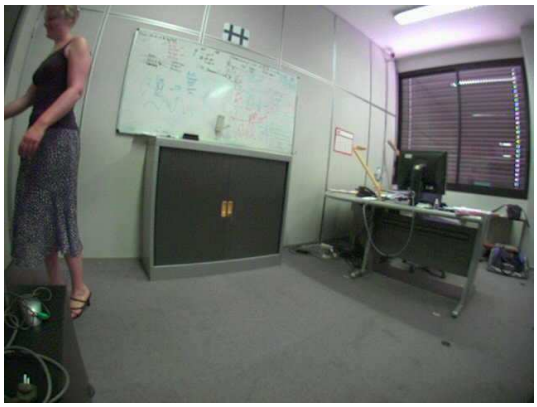

Figure 8.2: Deformations on the image due to an objective with a large field of view.

8.3.2 Long-Term Perspectives

Dynamic body primitive adaptability

During this work, the 3D model is automatically adapted to the studied person by only considering her/his height. More information about the person can be computed to initialise her/his 3D posture avatar. This information can either be the corpulence or the clothes worn by the person for instance. It would generate more accurate silhouettes and thereafter improve the recognition performance of our approach. Information about the corpulence can be handled by the proposed 3D posture avatar. A solution, for integrating information about the clothes is to have several 3D body primitives associated to different types of clothes. This can be simply achieved by defining body primitives which represents the different body parts for a given cloth. For instance, a body primitive can be designed to display a head with a hat. The proposed 3D engine displaying the 3D posture avatars would display a more complex head in terms of geometry.

3D posture avatar variability or gesture recognition

The proposed approach is based on predefined static 3D posture avatars and can induce wrong recognition with intermediate postures as observed in section 6.3.3. When a 3D posture avatar is recognised, the parameters of the 3D posture

avatar can be varied to better match the silhouette of the detected person. This variability may be driven by the type of the recognised posture. This improvement would allow the recognition of gesture. A thesis based on the subject "gesture recognition" has begun in the ORION team.
Another point concerning gesture recognition is the generation of synthetic data. As seen in section 6.3.1, synthetic data can be used to easily evaluate a posture recognition algorithm. The analogy can be done with the gesture recognition algorithms. An improvement must be done directly with the representation of the rotation parameters of the 3D posture avatar. Indeed, the representation is based on the Euler angles which is not adapted to animation purposes. Quaternion could be used to represent rotations, as described in appendix C to avoid this problem.

2D silhouette representation choice

The 2D silhouette representation is dependent on the quality of the silhouette. We have shown that the horizontal and vertical projections representation gives the best results for different types of segmentation in section 6.4. An interesting task is the ability to automatically evaluate the quality of the silhouette in order to choose the most appropriate 2D silhouette representation.

Processing time improvement

The main limitation of the proposed approach is the time processing of the 3D posture avatar silhouette generation. One way to decrease this time is to compute less generated silhouettes. An automata can be used to represent the authorised posture transitions. The recognition of the posture of the detected person should be used as an information to guide in the next frames which 3D posture avatar to consider. The set of postures of interest should be adapted automatically by only considering the authorised postures. This cue should reduce the processing time. Moreover, information on the orientation of the person could be used to only generate 3D posture avatar silhouettes for the correct orientation. An approach describes in [Zuniga et al., 2006] proposes to classify object by determining 3D parallelepiped which contains this object. In particular, the orientation of the parallelepiped is given and can be used as an approximation of the orientation of the person.

Hierarchical segmentation

The proposed human posture recognition is based on the study of the detected binary silhouette. An improvement can be done by not only considering one region but a set of region. By using a hierarchical segmentation, different regions

can be considered inside the silhouette and help to localise the different body parts in order to initialise the 3D posture avatar.

Segmentation Improvement

The recognised posture should be used to ameliorate the segmentation task by helping as feedback the parametrisation of the segmentation algorithm. The recognised silhouette could determine which body parts are missing or which pixels do not correspond to the detected person are in the segmented image.

Appendix A

Human Posture Recognition Implementation

In this appendix, details are given on the different implementation made during this thesis. First, the video understanding platform developed within the ORION team is described. Then, the implementation of the 3D posture avatar is presented, in particular how using the Mesa library to make this task. The virtual camera implementation is also given. Finally, a prototype using the proposed human posture recognition approach is presented.

A.1 Video Understanding Platform

This section describes the video understanding platform developed within the ORION team. The description of the global model of general framework for automatic video understanding, VSIP (Video Surveillance Intelligent Platform) can be found in [Bremond, 1997] and a more technical description can be found in [Avanzi et al., 2005]. VSIP has been successfully tested for several applications:

- During two European projects PASSWORDS and ADVISOR, the VSIP platform has been used to recognise some scenarios as fighting or blocking situation in metro surveillance ([Cupillard et al., 2002]),
- During the French project CASSIOPEE the platform has been used to monitor bank agencies ([Georis et al., 2006]),
- During the French project SAMSIT, the platform has been used to recognise scenarios of vandalism in train ([Vu et al., 2006]),
- During the European project AVITRACK, VSIP has been used to monitor activities on an airport apron [Borg et al., 2006].

The software architecture of VSIP is described in [Avanzi et al., 2005]. VSIP is composed of three tasks (cf. figure A.1):

- Detection and classification of physical objects of interest.
  The task of detection and classification of physical objects of interest is to provide a list of labeled physical objects. Several steps are necessary to achieve this task. First, the input images, either in colour or in black and white mode are acquired from one or several cameras. The reference image is then generated according to one of the following methods: the reference image can either be the first captured image, a given image or an image averaged from several frames. The segmentation algorithm detects the moving regions by subtracting the current image from the reference image. The difference image is thresholded with several criteria based on pixel intensity. The moving regions, also called *blobs*, are then defined by associating a set of 2D features like density or position. The classification algorithm processes the blobs and provides the list of labeled physical objects of interest using 2D geometrical features and 3D dimensions. A merge and split algorithm is applied on the blobs to obtain a more reliable list of physical objects corresponding to the model of expected objects (e.g. vehicle, person). A set of 3D features such as 3D position, width and height are then computed for each of blob. The association of a blob and the corresponding set of 3D features are called *mobile object*. The mobile objects are classified by using probabilistic distribution of the 3D features into predefined classes (e.g. vehicle, person). The reference image is updated by discriminating the real mobile objects from the regions of change in the current image compared with the reference image [Tornieri et al., 2002].

- Spatio temporal analysis.
  The list of physical objects of interest is then processed by spatio-temporal analysis. A graph containing the detected mobile objects and a set of links between object detected at time t and $t-1$ is obtained using frame to frame comparison. In cases where several cameras are used, with overlapping fields of view, the mobile objects are fused to obtain a unique representation of the mobile objects observed by the different cameras. The 3D features of the fused mobile objects are calculated from the previously computed features. Long term trackers can be used to add robustness to the tracking results. A long term track mainly consists in: (a) computing a set of paths representing the possible trajectories of the physical objects of interest (e.g. vehicle, person), (b) tracking the objects with a predefined delay T to compare the evolution of the different paths, (c) choosing at each frame the best path to update the object characteristics [Avanzi et al., 2001].

- Behaviour analysis.
  Finally, the tracked physical objects of interest are given to the event detection module. Depending on the type of scenarios to recognise, different event recognition algorithms can be used:

– Bayesian network can be used to deal with event uncertainty [Moenne-Locoz et al., 2003],

– AND/OR trees can be used to deal with scenarios with a large variety of visual invariants (fighting) [Cupillard et al., 2002],

– for events with multiple physical objects involved in complex relationships, the recognition algorithm is based on a constraint network [Vu et al., 2003].

We call *action*, a simple and short event in time whereas *event* is a set of *action*. Information about the context is necessary to recognise *event*.

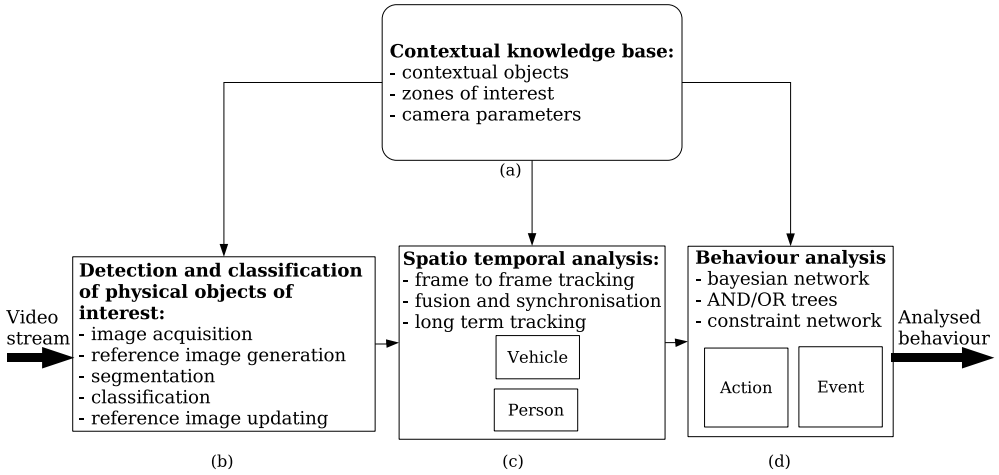

Figure A.1: The VSIP framework: (a) the contextual knowledge base provides information about the context to the different tasks of VSIP, (b) the physical objects of interest are detected and classified into predefined classes, (c) the objects are then tracked using spatio temporal analysis. (d) Finally, depending on the behaviour to be analysed, different methods are used to identify them.

Each of these tasks uses information provided by the contextual knowledge base. The contextual knowledge base contains information about the context of the scene:

- The position of the contextual objects (furniture such as chair or desk).
- The localisation of the zones of interest (forbidden zone, safe zone, etc...).
- The characteristics of the camera (the calibration matrix and the position of the camera).
- The semantic associated to each contextual object to be used in particular by behaviour analysis to infer high level scenario.

The goal of this work has been to design a component which can be integrated in any video understanding system such as VSIP. This component aims at helping the behaviour analysis task in order to refine the analysed behaviour. This component follows the spatio temporal analysis task in the treatment chain (figure A.2). Indeed, the filtering posture task needs information about the previous postures of the recognised person. This information is given by the tracking task. The filtered postures are then provided to the behaviour analysis task.

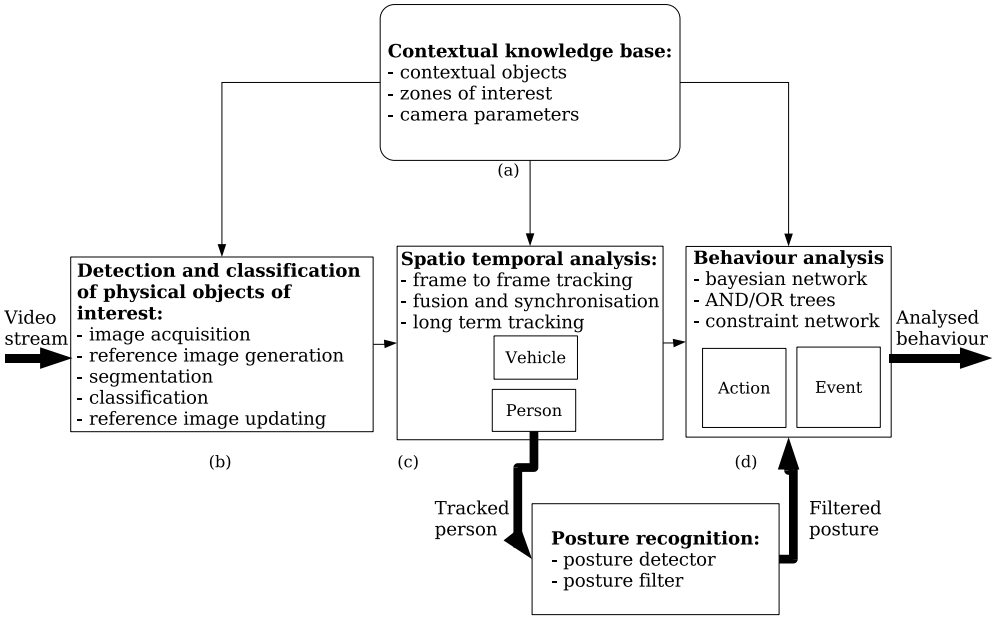

Figure A.2: The posture recognition task uses information provided by the spatio temporal analysis of detected person (c). The filtered postures are then provided to the behaviour analysis task (d).

A.2 3D Posture Avatar Implementation

As seen in chapter 4, the 3D posture avatar implementation has been made with the Mesa Library [Mesa, 2006]. Mesa is a 3D graphics library with an API (Application Programming Interface) which is very similar to OpenGL library [OpenGL, 2006]. We used Mesa because it is based on C language and well adapted to real time tasks.
Mesa handles different matrix stack operations to treat the 3D object modeling and displaying. The different matrix stacks are associated to each matrix modes which are:

- GL_MODELVIEW: the matrix associated with the scene modeling
- GL_PROJECTION: the matrix which characterises how the scene is visualised
- GL_TEXTURE: the matrix associated with the texture.

The GL_MODELVIEW is the mode defining how the different body parts are positioned in the 3D space.
Animation of an articulated object requires the handling of matrix to represent rotation and translation. Mesa library provides useful functionalities to achieve this task. The two main Mesa functions that move an object are *glTranslatef* and *glRotatef*. When *glTranslatef* (respectively *glRotatef*) is called, the first matrix of the current matrix stack is modified to take into account the translation (resp. rotation). So, when an object is drawn after a call of *glTranslatef* (resp. *glRotatef*), the drawing take into account this translation (resp. rotation).
By keeping the notation of chapter 4, the different transformation matrices are computed with:

$$M_X(\alpha) = glRotatef(\alpha, 1.0, 0.0, 0.0); \tag{A.1}$$
$$M_Y(\beta) = glRotatef(\beta, 0.0, 1.0, 0.0); \tag{A.2}$$
$$M_Z(\gamma) = glRotatef(\gamma, 0.0, 0.0, 1.0); \tag{A.3}$$
$$M_T([x, y, z]^T) = glTranslatef(x, y, z); \tag{A.4}$$
$$M_S([S_x, S_y, S_z]^T) = glScalef(S_x, S_y, S_z); \tag{A.5}$$

There are two very useful functions to manage the matrix stack *glPushMatrix* and *glPopMatrix*. The *glPushMatrix* function pushes the current matrix by one, duplicating the current matrix. After a *glPushMatrix* call, the matrix on the top of the stack is identical to the one below it. The *glPopMatrix* function pops the current matrix stack, replacing the current matrix with the one below it on the stack.
We propose an algorithm to animate 3D human body model using the Mesa library. The algorithm must be able to deal with the human body constraints. For example, when the left thigh moves, the left shin and the left foot must follow

this movement. This is done by using the *glPushMatrix* before the drawing of each leg parts.
Another important point is the rotation of the different body parts. The rotation must be done in the body part referential. To rotate the body primitive associated to the joint, first the body primitive is translated to the origin by using the default position information. Then the rotation is done. Finally the inverse translation is made to replace correctly the body primitive. The code that animates our 3D human model is described in algorithm 8. In OpenGL, the transformation matrices should be given in the opposite order than they should be applied.

Algorithm 8 *drawWholeBody*()

```
glTranslatef(this->hips.pos[0],this->hips.pos[1],this->hips.pos[2]);
 glPushMatrix();
    glScalef(2.5*scale,2.5*scale,2.5*scale);
          glTranslatef(this->hips.default_pos[0],this->hips.default_pos[1],this-
>hips.default_pos[2]);
    glRotatef(this->hips.rot[0],1.0f,0.0f,0.0f);
    glRotatef(this->hips.rot[1],0.0f,1.0f,0.0f);
    glRotatef(this->hips.rot[2],0.0f,0.0f,1.0f);
    drawHip();
       glTranslatef(-this->hips.default_pos[0],-this->hips.default_pos[1],-this-
>hips.default_pos[2]);
    glPushMatrix();
                                  glTranslatef(this->abdomen.default_pos[0],this-
>abdomen.default_pos[1],this->abdomen.default_pos[2]);
      glRotatef(this->abdomen.rot[0],1.0f,0.0f,0.0f);
      glRotatef(this->abdomen.rot[1],0.0f,1.0f,0.0f);
      glRotatef(this->abdomen.rot[2],0.0f,0.0f,1.0f);
      drawAbdomen();
                                glTranslatef(-this->abdomen.default_pos[0],-this-
>abdomen.default_pos[1],-this->abdomen.default_pos[2]);
      drawhair();
      drawHead();
      drawNeck();
      drawChest();
      drawRight_Collar();
      drawLeft_Collar();
      glPushMatrix();
        drawRightArm();
      glPopMatrix();
      glPushMatrix();
        drawLeftArm(); {described in algorithm 10}
      glPopMatrix();
    glPopMatrix();
    glPushMatrix();
      drawRightLeg();
    glPopMatrix();
    glPushMatrix();
      drawLeftLeg();{described in algorithm 9}
    glPopMatrix();
 glPopMatrix();
```

Algorithm 9 *drawLeftLeg*(M)

```
glTranslatef(this->left_hip.default_pos[0],this->left_hip.default_pos[1],this->left_hip.default_pos[2]);
glRotatef(this->left_hip.rot[0],1.0f,0.0f,0.0f);
glRotatef(this->left_hip.rot[2],0.0f,0.0f,1.0f);
drawLeft_Thigh();
  glTranslatef(-this->left_hip.default_pos[0],-this->left_hip.default_pos[1],-this->left_hip.default_pos[2]);
glPushMatrix();
                                glTranslatef(this->left_knee.default_pos[0],this->left_knee.default_pos[1],this->left_knee.default_pos[2]);
   glRotatef(this->left_knee.rot[0],1.0f,0.0f,0.0f);
   drawLeft_Shin();
                               glTranslatef(-this->left_knee.default_pos[0],-this->left_knee.default_pos[1],-this->left_knee.default_pos[2]);
   drawLeft_Foot();
glPopMatrix();
```

Algorithm 10 *drawLeftArm*(M)

```
                            glTranslatef(this->left_shoulder.default_pos[0],this->left_shoulder.default_pos[1],this->left_shoulder.default_pos[2]);
glRotatef(this->left_shoulder.rot[0],1.0f,0.0f,0.0f);
glRotatef(this->left_shoulder.rot[1],0.0f,1.0f,0.0f);
glRotatef(this->left_shoulder.rot[2],0.0f,0.0f,1.0f);
drawLeft_Shoulder();
                           glTranslatef(-this->left_shoulder.default_pos[0],-this->left_shoulder.default_pos[1],-this->left_shoulder.default_pos[2]);
glPushMatrix();
                              glTranslatef(this->left_elbow.default_pos[0],this->left_elbow.default_pos[1],this->left_elbow.default_pos[2]);
   glRotatef(this->left_elbow.rot[0],1.0f,0.0f,0.0f);
   glRotatef(this->left_elbow.rot[1],0.0f,1.0f,0.0f);
   glRotatef(this->left_elbow.rot[2],0.0f,0.0f,1.0f);
   drawLeft_Forearm();
                             glTranslatef(-this->left_elbow.default_pos[0],-this->left_elbow.default_pos[1],-this->left_elbow.default_pos[2]);
   drawLeft_Hand();
glPopMatrix();
```

A.3 Virtual Camera Implementation

As described in chapter 5, a virtual camera is designed by computing two transformation matrices:

- the camera transformation: M_{CT}
- the perspective transformation: M_{PT}

The Glu library (OpenGL Utility library) provides two useful function to compute these two matrices: *gluPerspective* and *gluLookAt*.
By keeping the notations of chapter 5, the both transformation matrices are computed as:

$$M_{CT} = gluLookAt(eye[0], eye[1], eye[2], center[0], center[1], center[2], up[0], up[1], up[2]);$$
$$M_{PT} = gluPerspective(fovy, aspect, znear, zfar);$$

As seen previously in OpenGL, transformations are computed by considering the last given operation. Moreover the matrices should be associated to the correct matrix stack of the Mesa library. The perspective transformation is associated to the $GL\_PROJECTION$ matrix stack whereas the camera transformation is associated to the $GL\_MODELVIEW$ stack. The virtual camera is then initialised with the algorithm 11.

Algorithm 11 *initialiseVirtualCamera*()

```
glMatrixMode(GL_PROJECTION);
glLoadIdentity();
gluPerspective(fovy,aspect,znear,zfar);
glMatrixMode(GL_MODELVIEW);
glLoadIdentity();
gluLookAt(eye[0],eye[1],eye[2],center[0],center[1],center[2],up[0],up[1],up[2]);
```

A.4 Prototype Implementation

This thesis work is leaded in cooperation with STMicroelectronics Rousset under PS26/27 project. This project aims to propose a smart environment based on intelligent cameras. Three partners are involved in the project for different tasks:

- STMicroelectronics: a specialist in image sensor (CMOS sensor)
- CMM: segmentation task
- INRIA: human posture recognition task

During this work, we have implemented a prototype to demonstrate the results and to integrate the algorithms of the different partners. The prototype has been defined as a graphic user interface (GUI) composed of three parts as shown in figure A.3:

- the segmentation parts is composed of the current image and the binary image, and of different displaying: bounding box, centroid, recognised action.
- the detailed posture recognition part: an image representing the detected detailed posture is highlighted in green,
- the general posture recognition part: a green curve representing the recognised general postures in time and a red curve representing the filtered general postures in time.

Moreover a menu file allows to load the image sequence and the associated parameters described below.
A tool bar is also implemented to easily navigate in the sequence with the classical button:

- play button: launch the posture recognition
- pause button: pause the posture recognition
- stop button: re-initialise the posture recognition.

The prototype has been implemented in QT because of its portability under Linux and Windows and its C++ like.

Different parameters are needed to make operational the prototype:

- the repository of the image sequence to treat,
- the reference image,
- the perspective matrix,
- the virtual camera,

Figure A.3: The prototype shows the results obtained with the proposed human posture recognition algorithm.

- the different posture parameters, in particular the rotation step.

When the posture recognition is launching, first the segmentation task is made. One of the segmentation algorithm is applied to obtain a binary image. The object of interest are then determined by a connexity analysis. Then these objects are classified to determine the different people evolving in the scene. For the studied video sequence, the classification is just based on the quantity of pixel of the detected objects, since in the scene only people move.

The human posture recognition approach is then applied to each detected person. According to the recognition, the detailed and generated posture recognition parts are updated.

Finally, the actions are determined by using the filtered postures and the results is displaying on the image.

Appendix B

Complete set of confusion matrices

In this appendix, all the confusion matrices obtained with synthetic data are given for the horizontal and vertical projections representation. The recognition is made for the optimal rotation step of 36 degrees. In the following tables the number represents the type of the detailed postures of interest:

- 0: standing with left arm up
- 1: standing with right arm up
- 2: standing with arms near the body
- 3: T-shape posture
- 4: sitting on a chair
- 5: sitting on the floor
- 6: bending
- 7: lying with spread legs
- 8: lying with curled up legs on right side
- 9: lying with curled up legs on left side

The table B.1 gives the confusion matrix for all the viewpoints and the 19 tables (B.2- B.20) give the confusion matrices for each viewpoint. The ith point of view corresponds to a camera localised at $\beta_c = i * 5$ degrees. The rows of a matrix correspond to the recognised postures and the columns correspond to the ground-truth ones. These tables may be used to define ambiguous postures and a recognition likelihood according to the viewpoint.

	0	1	2	3	4	5	6	7	8	9
0	3862	1265	23	523	6	0	37	103	245	100
1	1367	3961	771	711	94	0	56	138	45	256
2	746	724	5917	703	236	215	21	6	0	15
3	421	470	0	4068	228	21	0	55	0	0
4	146	133	19	0	4969	1266	157	206	127	155
5	45	61	4	0	1238	5206	801	197	199	251
6	78	69	106	0	14	0	5639	360	94	89
7	40	54	0	835	36	42	77	4710	474	225
8	88	60	0	0	7	54	31	488	4042	1996
9	47	43	0	0	12	36	21	577	1614	3754

Table B.1: Confusion matrix for detailed postures recognition for H. & V. projections for synthetic data.

	0	1	2	3	4	5	6	7	8	9
0	185	49	18	42	0	0	0	0	0	0
1	113	249	48	60	0	0	0	0	0	0
2	27	33	294	53	0	0	0	0	0	0
3	35	29	0	205	0	0	0	0	0	0
4	0	0	0	0	338	2	0	0	0	0
5	0	0	0	0	22	357	37	0	0	0
6	0	0	0	0	0	0	323	9	0	0
7	0	0	0	0	0	0	0	283	23	0
8	0	0	0	0	0	1	0	18	284	121
9	0	0	0	0	0	0	0	50	53	239

Table B.2: Confusion matrix for detailed postures recognition for H. & V. projections for synthetic data according to the 0th point of view.

	0	1	2	3	4	5	6	7	8	9
0	169	94	5	40	0	0	0	0	0	0
1	125	224	55	69	0	0	0	0	0	0
2	32	14	300	50	0	0	0	0	0	0
3	34	28	0	201	8	0	0	0	0	0
4	0	0	0	0	332	0	0	0	0	0
5	0	0	0	0	20	360	44	0	0	0
6	0	0	0	0	0	0	316	11	0	0
7	0	0	0	0	0	0	0	291	33	2
8	0	0	0	0	0	0	0	40	275	112
9	0	0	0	0	0	0	0	18	52	246

Table B.3: Confusion matrix for detailed postures recognition for H. & V. projections for synthetic data according to the 1th point of view.

	0	1	2	3	4	5	6	7	8	9
0	170	115	0	49	0	0	0	0	0	0
1	112	187	72	67	0	0	0	0	0	0
2	41	36	288	53	0	0	0	0	0	0
3	37	22	0	191	18	0	0	0	0	0
4	0	0	0	0	319	0	0	0	0	0
5	0	0	0	0	23	360	48	0	0	0
6	0	0	0	0	0	0	312	10	0	0
7	0	0	0	0	0	0	0	278	28	9
8	0	0	0	0	0	0	0	39	271	95
9	0	0	0	0	0	0	0	33	61	256

Table B.4: Confusion matrix for detailed postures recognition for H. & V. projections for synthetic data according to the 2th point of view.

	0	1	2	3	4	5	6	7	8	9
0	136	65	0	52	0	0	0	0	0	0
1	139	185	60	57	0	0	0	0	0	0
2	42	75	300	47	3	0	0	0	0	0
3	43	35	0	204	23	0	0	0	0	0
4	0	0	0	0	304	3	0	0	0	0
5	0	0	0	0	30	325	51	0	0	0
6	0	0	0	0	0	0	309	12	0	0
7	0	0	0	0	0	32	0	272	41	20
8	0	0	0	0	0	0	0	41	231	89
9	0	0	0	0	0	0	0	35	88	251

Table B.5: Confusion matrix for detailed postures recognition for H. & V. projections for synthetic data according to the 3th point of view.

	0	1	2	3	4	5	6	7	8	9
0	220	63	0	40	0	0	0	0	0	0
1	68	215	55	54	6	0	0	0	0	0
2	21	42	305	45	5	0	0	0	0	0
3	51	40	0	221	20	0	0	0	0	0
4	0	0	0	0	257	15	0	0	0	0
5	0	0	0	0	36	335	53	31	0	0
6	0	0	0	0	0	0	304	14	0	0
7	0	0	0	0	36	10	0	271	38	25
8	0	0	0	0	0	0	0	16	238	86
9	0	0	0	0	0	0	3	28	84	249

Table B.6: Confusion matrix for detailed postures recognition for H. & V. projections for synthetic data according to the 4th point of view.

	0	1	2	3	4	5	6	7	8	9
0	210	58	0	44	0	0	0	0	0	0
1	65	208	77	59	9	0	0	0	0	0
2	41	46	283	52	8	0	0	0	0	0
3	44	47	0	205	15	0	0	0	0	0
4	0	1	0	0	295	67	0	29	0	0
5	0	0	0	0	33	284	46	23	2	1
6	0	0	0	0	0	0	309	16	0	0
7	0	0	0	0	0	0	0	267	35	26
8	0	0	0	0	0	0	0	5	247	94
9	0	0	0	0	0	9	5	20	76	239

Table B.7: Confusion matrix for detailed postures recognition for H. & V. projections for synthetic data according to the 5th point of view.

	0	1	2	3	4	5	6	7	8	9
0	229	40	0	46	2	0	0	0	0	0
1	63	230	67	59	11	0	0	0	0	0
2	27	37	293	48	9	0	0	0	0	0
3	39	52	0	207	13	0	0	0	0	0
4	2	1	0	0	288	79	5	25	0	0
5	0	0	0	0	29	276	34	40	19	26
6	0	0	0	0	0	0	309	12	0	0
7	0	0	0	0	0	0	0	248	27	15
8	0	0	0	0	4	0	5	16	230	105
9	0	0	0	0	4	5	7	19	84	214

Table B.8: Confusion matrix for detailed postures recognition for H. & V. projections for synthetic data according to the 6th point of view.

	0	1	2	3	4	5	6	7	8	9
0	216	48	0	44	2	0	0	0	0	0
1	55	214	72	59	12	0	0	0	0	0
2	56	41	288	40	10	0	0	0	0	0
3	32	55	0	217	14	0	0	0	0	0
4	1	2	0	0	279	76	9	7	1	9
5	0	0	0	0	35	275	40	51	25	33
6	0	0	0	0	0	0	297	15	0	3
7	0	0	0	0	0	0	0	245	20	12
8	0	0	0	0	0	4	8	21	225	100
9	0	0	0	0	8	5	6	21	89	203

Table B.9: Confusion matrix for detailed postures recognition for H. & V. projections for synthetic data according to the 7th point of view.

	0	1	2	3	4	5	6	7	8	9
0	219	31	0	31	2	0	0	0	0	0
1	49	218	74	61	3	0	0	0	0	0
2	58	58	286	47	26	0	0	0	0	0
3	31	51	0	221	8	0	0	0	0	0
4	3	2	0	0	275	65	11	27	7	7
5	0	0	0	0	46	274	44	28	35	42
6	0	0	0	0	0	0	305	12	1	8
7	0	0	0	0	0	0	0	239	28	2
8	0	0	0	0	0	15	0	26	194	101
9	0	0	0	0	0	6	0	28	95	200

Table B.10: Confusion matrix for detailed postures recognition for H. & V. projections for synthetic data according to the 8th point of view.

	0	1	2	3	4	5	6	7	8	9
0	208	45	0	33	0	0	0	0	0	0
1	44	221	13	49	6	0	0	0	0	0
2	79	52	338	37	29	0	0	0	0	0
3	26	39	0	240	7	0	0	1	0	0
4	1	3	0	0	256	45	16	33	6	10
5	0	0	0	0	62	277	35	20	40	49
6	2	0	9	0	0	0	307	16	8	8
7	0	0	0	1	0	0	2	231	35	9
8	0	0	0	0	0	27	0	33	180	92
9	0	0	0	0	0	11	0	26	91	192

Table B.11: Confusion matrix for detailed postures recognition for H. & V. projections for synthetic data according to the 9th point of view.

	0	1	2	3	4	5	6	7	8	9
0	219	51	0	36	0	0	0	10	0	0
1	33	234	11	42	6	0	1	1	0	0
2	77	36	309	41	29	0	3	1	0	0
3	20	34	0	241	8	0	0	0	0	0
4	3	4	0	0	280	75	20	33	29	29
5	0	0	0	0	35	278	39	4	23	38
6	8	1	40	0	0	0	292	20	8	5
7	0	0	0	0	0	0	5	232	32	6
8	0	0	0	0	2	7	0	35	178	101
9	0	0	0	0	0	0	0	24	90	181

Table B.12: Confusion matrix for detailed postures recognition for H. & V. projections for synthetic data according to the 10th point of view.

	0	1	2	3	4	5	6	7	8	9
0	229	64	0	36	0	0	0	14	0	2
1	58	242	22	21	3	0	15	15	0	0
2	51	23	281	33	32	0	9	3	0	0
3	6	14	0	243	7	0	0	0	0	0
4	2	5	0	0	279	77	25	25	29	40
5	0	0	0	0	38	283	42	0	14	18
6	13	11	57	0	0	0	246	18	10	9
7	0	1	0	27	0	0	23	229	10	5
8	0	0	0	0	1	0	0	24	197	96
9	1	0	0	0	0	0	0	32	100	190

Table B.13: Confusion matrix for detailed postures recognition for H. & V. projections for synthetic data according to the 11th point of view.

	0	1	2	3	4	5	6	7	8	9
0	227	62	0	27	0	0	0	18	0	6
1	60	203	28	25	0	0	16	22	1	0
2	42	66	332	35	34	0	5	2	0	5
3	0	2	0	236	8	0	0	0	0	0
4	10	17	0	0	295	81	21	18	29	31
5	0	0	0	0	23	279	49	0	15	16
6	10	3	0	0	0	0	249	21	7	12
7	7	0	0	37	0	0	20	220	13	10
8	0	0	0	0	0	0	0	24	198	102
9	4	7	0	0	0	0	0	35	97	178

Table B.14: Confusion matrix for detailed postures recognition for H. & V. projections for synthetic data according to the 12th point of view.

	0	1	2	3	4	5	6	7	8	9
0	209	68	0	3	0	0	0	18	4	5
1	46	180	21	14	0	0	22	23	3	15
2	48	58	339	37	34	0	4	0	0	0
3	3	4	0	222	11	0	0	0	0	0
4	23	23	0	0	258	126	7	9	17	16
5	0	0	0	0	57	234	48	0	6	12
6	7	8	0	0	0	0	257	25	8	8
7	11	2	0	84	0	0	22	224	22	11
8	9	5	0	0	0	0	0	25	203	124
9	4	12	0	0	0	0	0	36	97	169

Table B.15: Confusion matrix for detailed postures recognition for H. & V. projections for synthetic data according to the 13th point of view.

	0	1	2	3	4	5	6	7	8	9
0	203	69	0	0	0	0	0	14	26	4
1	51	197	25	8	21	0	2	23	3	35
2	41	59	335	25	16	28	0	0	0	6
3	1	0	0	201	8	0	0	1	0	0
4	25	9	0	0	248	72	25	0	5	1
5	0	0	0	0	67	260	30	0	9	8
6	12	10	0	0	0	0	298	28	8	5
7	10	13	0	126	0	0	5	234	15	14
8	12	0	0	0	0	0	0	25	203	127
9	5	3	0	0	0	0	0	35	91	160

Table B.16: Confusion matrix for detailed postures recognition for H. & V. projections for synthetic data according to the 14th point of view.

	0	1	2	3	4	5	6	7	8	9
0	204	82	0	0	0	0	0	11	45	9
1	60	189	18	5	13	0	0	25	3	35
2	24	24	340	15	0	62	0	0	0	4
3	4	5	0	201	9	0	0	5	0	0
4	22	18	0	0	179	75	17	0	2	3
5	4	8	2	0	154	223	32	0	8	6
6	15	15	0	0	5	0	311	31	7	6
7	10	15	0	139	0	0	0	234	18	15
8	10	4	0	0	0	0	0	26	197	119
9	7	0	0	0	0	0	0	28	80	163

Table B.17: Confusion matrix for detailed postures recognition for H. & V. projections for synthetic data according to the 15th point of view.

	0	1	2	3	4	5	6	7	8	9
0	203	86	0	0	0	0	14	7	57	19
1	79	193	0	2	4	0	0	20	7	50
2	15	13	339	16	1	58	0	0	0	0
3	5	7	0	199	16	0	0	14	0	0
4	15	11	19	0	183	85	1	0	2	5
5	9	9	2	0	151	217	39	0	2	2
6	10	18	0	0	5	0	296	32	12	7
7	2	9	0	143	0	0	0	231	16	11
8	7	13	0	0	0	0	10	21	177	111
9	15	1	0	0	0	0	0	35	87	155

Table B.18: Confusion matrix for detailed postures recognition for H. & V. projections for synthetic data according to the 16th point of view.

	0	1	2	3	4	5	6	7	8	9
0	200	80	0	0	0	0	11	6	51	29
1	79	182	37	0	0	0	0	9	20	54
2	9	4	323	17	0	34	0	0	0	0
3	4	6	0	206	18	1	0	15	0	0
4	21	27	0	0	159	164	0	0	0	4
5	17	18	0	0	179	161	44	0	1	0
6	1	3	0	0	4	0	300	28	14	8
7	0	8	0	137	0	0	0	242	19	16
8	22	22	0	0	0	0	5	24	166	105
9	7	10	0	0	0	0	0	36	89	144

Table B.19: Confusion matrix for detailed postures recognition for H. & V. projections for synthetic data according to the 17th point of view.

	0	1	2	3	4	5	6	7	8	9
0	206	95	0	0	0	0	12	5	62	26
1	68	190	16	0	0	0	0	0	8	67
2	15	7	344	12	0	33	0	0	0	0
3	6	0	0	207	17	20	0	19	0	0
4	18	10	0	0	145	159	0	0	0	0
5	15	26	0	0	198	148	46	0	0	0
6	0	0	0	0	0	0	299	30	11	10
7	0	6	0	141	0	0	0	239	21	17
8	28	16	0	0	0	0	3	29	148	116
9	4	10	0	0	0	0	0	38	110	125

Table B.20: Confusion matrix for detailed postures recognition for H. & V. projections for synthetic data according to the 18th point of view.

Appendix C

Quaternion

In this section a discussion is given on the different rotation representations and in particular about the quaternion representation.
There exist three usual rotation representations:

- Euler representation,
- axis representation,
- and quaternion representation.

As seen in chapter 4, the joints parameters are the three angles associated to each articulation of the 3D human body. This representation is sufficient in the field of this work to represent the different postures of interest but shows its limitation for animation purpose. The next step in the generation of synthetic data is the animation of the 3D human body model to acquire realistic gesture. The quaternion is well adapted for this purpose.
In the next section, the different rotation representations are described and in particular the way to transform each representation in quaternion one.

Euler representation

A rotation is represented by three angles according to each axis. These angles are classically named pitch, roll and yaw. The rotation is obtained by multiplying the three rotation matrix associated to each angle in an arbitrary order. As seen previously in (section 4.1), this representation has different drawbacks:

- non-unicity of the representation,
- gimbal lock problem,
- non realistic rotation.

Axis representation

This representation avoid the gimbal lock problem. It is composed of a unit vector which represents the rotation axis and an angle. The main drawback is the interpolation between two rotations.

Quaternion representation

Quaternion is first introduced as an extension to complex numbers. It has latter be used in computer graphics to represent the rotations.
A quaternion q is defined as

$$q = w + xi + yj + zk = (w, (x, y, z)) = (w, v) \tag{C.1}$$

$$\tag{C.2}$$

where i, j, k are all square roots of -1 and w, x, y, z are real numbers.
A unit quaternion is necessary to represent a 3D rotation. A quaternion can be normalised according to its magnitude similarly to Euclidean magnitude for vector:

$$\frac{q}{\sqrt{w^2 + x^2 + y^2 + z^2}} \tag{C.3}$$

$$\tag{C.4}$$

A unit quaternion can be represented as a rotation in a 4-dimensional world, where the (x, y, z) is the rotation axis and w is the angle.
The rotation matrix associated to a given quaternion q is given by:

$$P = \begin{bmatrix} 1 - 2y^2 - 2z^2 & 2xy - 2wz & 2xz + 2wy \\ 2xy + 2wz & 1 - 2x^2 - 2z^2 & 2yz - 2wx \\ 2xz - 2wy & 2yz + 2wx & 1 - 2x^2 - 2y^2 \end{bmatrix} \tag{C.5}$$

It is not easy to associate a quaternion to a given rotation. Usually, the rotation is represented with Euler angles or axis, the the obtained representation is converted in quaternion. The conversion is described in the next sections.

Euler angle to quaternion

Converting Euler angles to a quaternion is depending on the order of the angle multiplication. We suppose here that the rotation is first done according to X-axis α_X, following by Y-axis α_Y and Z-axis α_Z. First the three following quaternion are determined:

$$Q_X = \cos\frac{a}{2}, \sin\frac{alpha_X}{2}, 0, 0 \tag{C.6}$$

$$Q_Y = \cos\frac{b}{2}, 0, \sin\frac{alpha_Y}{2}, 0 \tag{C.7}$$

$$Q_Z = \cos\frac{c}{2}, 0, 0, \sin\frac{alpha_Z}{2} \tag{C.8}$$

$$\tag{C.9}$$

The final quaternion is obtained by:

$$Q = Q_X \triangleright Q_Y \triangleright Q_Z \tag{C.10}$$
$$\tag{C.11}$$

where $\triangleright$ is the quaternion multiplication defined in equation C.13

$$Q_1 \triangleright Q_2 = [w_1 * w_2 - v_1.v_2, (w_1 * v_2 + w_2 * v_1 + v_1 \wedge v_2)] \tag{C.12}$$
$$\tag{C.13}$$

with $*$ is a scalar multiplication, $\wedge$ is a vector cross product, and . is the vector dot product.

Axis angle to quaternion

The conversion of an axis representation (θ, a_x, a_y, a_z) is given by the equations below:

$$w = \cos\frac{\theta}{2} \tag{C.14}$$
$$x = a_x \sin\frac{\theta}{2} \tag{C.15}$$
$$y = a_y \sin\frac{\theta}{2} \tag{C.16}$$
$$z = a_z \sin\frac{\theta}{2} \tag{C.17}$$
$$\tag{C.18}$$

Conclusion

Quaternion are well adapted to model interpolation between two orientations and avoid the gimbal lock problem. Quaternions support spherical linear interpolation (SLERP), which means that points travel along the surface of a sphere as they are moved from one orientation to the next.

Appendix D

Publications of the Author

- **International Journal with Peer-review:**

  [1] Applying 3D Human Model in a Posture Recognition System. Boulay, B. and Bremond, F. and Thonnat, M. *Pattern Recognition Letter, Special Issue on vision for Crime Detection and Prevention.* November 2006, 15(27), pp 1788-1796.

- **International conferences with Peer-review:**

  [1] Posture Recognition with a 3D Human Model. Boulay, Bernard and Bremond, Francois and Thonnat, Monique. *Proceedings of IEE International Symposium on Imaging for Crime Detection and Prevention.* 2005.

  [2] Human Posture Recognition in Video Sequence. B. Boulay and F. Bremond and M. Thonnat. *Proceedings Joint IEEE International Workshop on VS-PETS, Visual Surveillance and Performance Evaluation of Tracking and Surveillance.* October 2003, pp. 23-29.

Appendix E

French Introduction

La reconnaissance de posture de personne est un problème difficile est ambitieux dû au grand nombre de cas possibles. La quantité de posture est directement reliée au degré de liberté du corps humain (i.e. les articulations telles que les épaules ou les genoux). De plus, la morphologie des personnes (la taille ou la corpulence) joue un rôle important dans la perception des postures. Enfin, les vêtements peuvent aussi donner une apparence différente pour la même posture considérée.
Les sections suivantes décrivent les motivations, le contexte et les objectifs de cette thèse en reconnaissance de posture de personne. Ce chapitre est conclu par la structure du manuscrit.

E.1 Motivations

La reconnaissance de posture de personne est une partie importante de la compréhension du comportement car elle permet d'obtenir des informations précises pour la personne étudiée. Le problème de la reconnaissance de posture intervient dans trois principaux types d'application:

- Les applications de **surveillance** peuvent être définies comme le suivi de une ou plusieurs personnes dans le temps pour analyser leurs comportements. La vidéo surveillance ou la domotique sont des exemples typiques où les personnes sont suivies pour analyser leurs activités.

- Les applications de **contrôle** utilise l'information de la posture d'une personne comme une fonction de contrôle. Par exemple, une personne peut interagir avec un ordinateur grâce à une interface intelligente (IHM) basée sur les postures.

- Les applications d'**analyse** nécessitent une information très précise sur la posture. Elles sont typiquement utilisées pour des applications médicales (par exemple en orthopédie), pour la surveillance ou l'entraînement de sportif, ou pour l'animation virtuelle.

Dans ce travail l'approche proposée a pour but de reconnaître la posture de personne pour des applications de surveillance et de contrôle. Nous pensons que les applications d'analyse nécessitent un traitement spécifique pour obtenir la précision souhaitée dans les mesures des différentes parties du corps (taille, localisation dans l'espace 3D, orientation).
Chacune de ces trois types d'applications doivent respecter certaines propriétés classées en trois catégories :

- Le **nombre de contrainte** dont a besoin une application. Par exemple, une contrainte peut être d'avoir une caméra statique, pas d'occlusion, les personnes doivent être face à la caméra, l'éclairage doit être constant, etc. Les applications de surveillance nécessitent d'avoir moins de contrainte que les autres applications puisqu'elles nécessitent gńéralement un fonctionnement automatique dans des environements variés pour une longue période de temps. Les applications de contrôle et d'analyse ont plus de contraintes car elles fonctionnent généralement pour une courte période de temps dans un espace contraint. Par exemple, la personne doit être devant la caméra dans le cas d'une interface homme machine intelligente.

- La **précision** peut être mesurée grâce à la similarité entre la posture reconnue et celle de la personne évoluant dans la scène. Une grande précision n'est pas nécessaire pour une application de surveillance alors qu'elle est importante pour des applications d'analyse et de contrôle. En effet, les applications d'analyse ont besoin de mesures précises pour les différentes parties du corps.

- La **vitesse** d'exécution peut être classée en temps réel et hors ligne. Le temps réel est communement définit comme le calcul qui donne le résultat dans un temps fixé. Ce temps est différent en fonction du but recherché par une application donnée. Les applications de surveillance et de contrôle nécessitent une vitesse d'exécution élevée pour pouvoir détecter certain comportement à temps. Par exemple, lorsqu'une personne interagit avec un ordinateur, les résultats doivent être immédiats. À l'inverse, les applications d'analyse peuvent être traitées hors-ligne.

E.2 Contexte de l'Étude

Il est nécessaire de placer le problème de reconnaissance de posture de personne dans la chaîne du traitement complet de l'interprétation vidéo. Différentes études sur l'interprétation vidéo (aussi appelée analyse de mouvement de personne dans notre cas) ont été proposées ces 20 dernières années :

- Dans [Cedras and Shas, 1995], les auteurs présentent une vue d'ensemble des méthodes pour l'extraction du mouvement avant 1995. Le mouvement de personne est décrit comme la suite de reconnaissance d'action, de reconnaissance des parties du corps et l'estimation de la configuration du corps.

- Dans [Aggarwal and Cai, 1999], le mouvement de personne est interprété comme la succession de trois tâches qui sont les même que citées précédemment dans [Cedras and Shas, 1995] mais nommées différemment : l'analyse de mouvement faisant intervenir les parties du corps humain, le suivi de personne avec une ou plusieurs caméras et la reconnaissance d'activité humaine.

- Dans [Gavrila, 1999], les auteurs décrivent les principaux travaux en analyse du mouvement de personne avant 1998. Ils décrivent différentes méthodes classées en approche 2D avec ou sans modèle de forme et les approches 3D.

- Dans [Moeslund and Granum, 2001], les auteurs donnent une vue d'ensemble sur l'analyse de mouvement de personne avant 2000 et complétée dans [Moeslund, 2003] avant 2002. Un système d'analyse de mouvement de personne est constitué de quatre tâches : l'initialisation, le suivi, l'estimation des postures et la reconnaissance d'action. Une initialisation des données est nécessaire, par exemple un modèle adapté à la personne étudiée peut être initialisé. La tâche de suivi calcule les relations dans le temps de l'objet détecté en trouvant les correspondances dans les images consécutives. Ensuite, l'estimation de posture des personnes détectées est faite. La tâche finale analyse les postures et d'autres paramètres pour reconnaître les actions effectuées par les personnes évoluant dans la scène.

- Dans [Wang et al., 2003], les travaux sur l'analyse de mouvement de personne sont décrits jusqu'à 2001. La taxonomie proposée est composée de cinq tâches : la segmentation des objets en mouvement, la classification des objets détectés, le suivi des personnes, la reconnaissance d'action et la description sémantique. Le but de la description sémantique du comportement des personnes est de "choisir un ensemble raisonnable de mot ou d'expression courte pour décrire les comportements des objets en mouvement dans des scènes naturelles".

Le travail de cette thèse a été effectué dans l'équipe ORION localisée à l'INRIA Sophia Antipolis. ORION et une équipe pluridisciplinaires à la frontière de la vison par ordinateur, de l'intelligence artificielle et du génie logiciel. L'équipe a acquis une forte expérience dans ces domaines au cours de ces années. Un des sujets d'intérêt est la compréhension automatique d'image et de vidéo basée sur une connaissance a priori. Le travail proposé prend place dans ce contexte. Comme dans [Wang et al., 2003], un framework général de la tâche d'interprétation vidéo peut être décrit de la vision bas niveau à la vision haut niveau (figure E.1) :

- la segmentation des objets,

- la classification des objets,

- le suivi des personnes,

- la reconnaissance de posture des personnes,

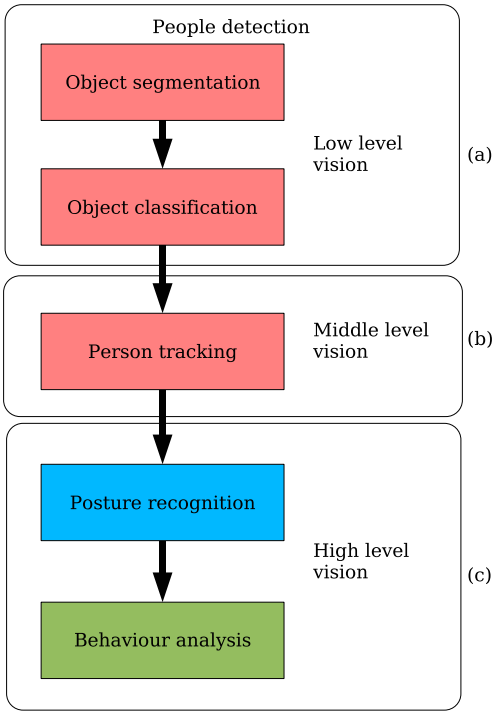

Figure E.1: Un framework général pour la tâche d'interprétation vidéo. La tâche est composée de : (a) une tâche de vision bas niveau qui détecte les personnes évoluant dans la scène, (b) une tâche de vision de niveau intermédiaire qui suit les personnes détectées et (c) une tâche de vision haut niveau qui reconnaît la posture et analyse le comportement en fonction des informations calculées précédemment.

- l'analyse du comportement des personnes.

La première étape d'une telle approche est de détecter les personnes évoluant dans une séquence vidéo. La détection de personne est importante pour les tâches suivantes telles que le suivi des personnes ou l'analyse de comportement. La détection des personnes est généralement réalisée par une tâche de segmentation et de classification. Les personnes sont ensuite suivies tout au long de la séquence vidéo. Finalement, l'analyse du comportement des personnes est faite en utilisant les informations calculées pendant les tâches précédentes. Le placement de la tâche de reconnaissance de posture dans la chaîne de traitement est discuté dans le chapitre 3. En particulier, nous présenterons pourquoi cette tâche a besoin de l'information temporelle fournit par la tâche de suivi des personnes. De plus la solution au problème de l'interprétation vidéo proposée par l'équipe Orion est décrite dans l'annexe A.

E.3 Objectifs

Le but de ce travail est de proposer une approche générique pour reconnaître la posture d'une personne entière à partir de séquence vidéo. L'approche doit être générique pour s'adapter au plus de situation possible.

L'approche prend place dans une tâche plus générale qui est l'interprétation vidéo. Elle utilise l'information calculée par la tâche de détection et fournit un résultat à la tâche d'interprétation vidéo. La tâche de détection des personnes donne des informations sur les personnes évoluant dans la scène telle que leurs positions et leurs dimensions. Les personnes sont généralement représentées par leurs silhouettes binaires. Puisque l'approche proposée utilise ces silhouettes pour déterminer la posture, l'approche doit être efficace pour différent types de silhouette (parfaite ou bruitée).

L'approche proposée doit respecter les propriétés listées précédemment (i.e. nombre de contrainte, précision et vitesse) car les applications visées sont les applications de surveillance et de contrôle.
Comme vu précédemment dans la section E.1, le nombre de contrainte nécessaire à une application est importante pour proposer une approche générique pour la reconnaissacne de posture. Premièrement, le type de caméra nécessaire est important. En utilisant une seule caméra statique, l'approche peut directement être appliquées à des systèmes existants ou facilement appliquées à des nouveaux systèmes d'interprétation vidéo. Deuxièmement, une certaine indépendance au point de vue de la caméra est une clé importante pour proposer une approche opérationnelle. En effet, si, par exemple, une personne peut être face à la caméra pour une application de contrôle, ce n'est généralement pas possible de demander aux personnes évoluant dans une scène de regarder la caméra pour une application de surveillance.
La précision nécessaire à une application de surveillance n'est pas la même que celle intervenant dans une application de contrôle. Une application de surveillance nécessite une information sur la posture plus générale que celle pour une application de contrôle.
La vitesse d'exécution des applications de surveillance et de contrôle est une propriété très importante. Par exemple, l'application doit être capable de générer une alarme quand une personne tombe (ou même avant) et non 10 minutes plus tard.
Notre travail a pour but de résoudre ces problèmes grâce aux principales contributions suivantes :

- Les avancées faites dans l'infographie et l'animation virtuelle sont utilisées pour proposer un modèle 3D humain adapté à la reconnaissance de posture. Une certaine indépendance du point de vue de la caméra est ainsi acquise en utilisant un modèle 3D humain.

- L'approche hybride proposée pour reconnaître la posture de personne com-

bine des représentations 2D des silhouettes et l'utilisation d'un modèle 3D humain. Les représentations 2D maintiennent un temps réel d'exécution et sont adaptées aux différents types de silhouette.

Ces contributions sont présentées dans les chapitres suivants du manuscrit comme décrit dans la section suivante.

E.4 Structure du Manuscript

Ce manuscrit est structuré en six chapitres.

Le **chapitre 2** présente au lecteur les précédents travaux effectués en reconnaissance de posture de personne. Différentes techniques sont présentées pour les capteurs physiologique et mécanique ainsi que pour les capteurs vidéo. Les capteurs physiologiques, telles que les MEMS (Micro Electro Mechanical System), sont utilisés pour des personnes coopératives alors que les capteurs vidéos sont utilisés pour des personnes non coopératives. Un zoom est fait pour la reconnaissance de posture en décrivant en particulier les méthodes utilisant des marqueurs sur le corps ainsi que celles utilisant des capteurs vidéo. Les techniques utilisant les capteurs vidéo sont classées en techniques 2D et 3D. Chacune de ces techniques a ses forces et ses faiblesses. Le but de cette thèse est de proposer une approche qui combine leurs forces tout en minimisant leurs faiblesses.

Le **chapitre 3** présente nos objectifs et donne un aperçu de l'approche proposée pour reconnaître les postures. Comme expliqué dans la section 1.3, les applications visées sont les applications de surveillance et de contrôle. Ainsi, plusieurs contraintes que doit respecter l'approche ont été identifiées : temps réel, indépendance du point de vue de la caméra, une approche automatique et l'utilisation d'une caméra statique. Une approche hybride est ainsi proposée en combinant des techniques 2D et l'utilisation de modèle 3D humain, pour respecter ces contraintes. De plus une base de connaissance contextuel est utilisée pour piloter la tâche de reconnaissance de posture en donnant des informations sur la scène.

Le **chapitre 4** décrit l'avatar 3D de posture proposé qui est une combinaison d'un modèle 3D humain et d'un ensemble de paramètre correspondant à une posture particulière. Le chapitre montre comment les différentes parties de l'avatar 3D sont animées en fonction des paramètres. Un ensemble de posture d'intérêt est alors identifié et modélisé. Ces postures sont classées de manière hiérarchique de postures générales à postures détaillées.

Le **chapitre 5** présente l'approche hybride proposée qui est composée de deux tâches principales :

- la première tâche calcule la posture de la personne détectée avec l'information provenant seulement de l'image courante et des modèles 3D. Les avatars 3D de posture candidats sont générés en projetant les avatars 3D sur le plan image en définissant une caméra virtuelle qui a les mêmes caractéristiques que la véritable caméra. Chaque avatar 3D de posture est placé dans la scène 3D en fonction du résultat de la tâche de détection des personnes (la position), puis tous les avatars possibles sont orientés pour différentes orientations pour générer toutes les silhouettes possibles. Les silhouettes détectées et générées sont modélisées et comparées à l'aide des représentations 2D pour obtenir la posture.

- la seconde tâche utilise l'information sur la posture provenant des images précédentes. Les postures reconnues sur les images précédentes sont utilisées pour vérifier la cohérence temporelle des postures pour donner la posture la plus probable.

Les différentes représentations 2D des silhouettes utilisées dans l'approche sont aussi décrites dans ce chapitre.

Dans le **chapitre 6**, l'approche proposée est évaluée et optimisée. Un modèle de vérité terrain est proposé pour évaluer l'algorithme de reconnaissance de posture proposé. Des données de synthèse sont générées pour plusieurs points de vue différents afin de comparer les différentes représentations 2D et l'influence des paramètres sur l'approche de reconnaissance de posture proposée. L'approche est testée sur plusieurs séquences vidéo réelles et pour différents types de silhouette (différents algorithmes de segmentation).

Le **chapitre 7** montre comment les postures peuvent être utilisée pour reconnaître des actions ne faisant intervenir qu'une seule personne. Une action est représentée par une machine à états finis. Chaque état est représenté par une ou plusieurs postures. Cette méthode a été testée avec succès pour différents types d'actions telles que la chute (une action importante médicalement parlant) ou la marche.

Finalement, le **chapitre 8** conclut ce travail, en résumant les contributions de cette thèse et en présentant les perspectives à court terme et long terme.

Appendix F

French Conclusion

Durant cette thèse nous avons proposé une nouvelle approche pour la reconnaissance de posture de personne. Cette approche combine des techniques 2D et l'utilisation d'avatar 3D de posture. Les techniques 2D permettent de garder un temps réel d'exécution alors que les avatars 3D de posture permettent une certaine indépendance du point de vue de la caméra.
L'approche proposée prend en entrée la silhouette 2D des régions en mouvement correspondant à la personne détectée ainsi que sa position 3D estimée. L'approche est composée de quatre tâches :

- Les silhouettes des avatars 3D de posture sont générées en projetant les avatars sur le plan image en utilisant une caméra virtuel. Les avatars 3D sont placés dans la scène virtuelle où la personne observée est détectée, ensuite les avatars sont orientés selon différents angles pour générer toutes les silhouettes possibles en accord avec les postures prédéfinies.

- Les silhouettes sont représentées et comparées en fonction de quatre techniques 2D : une représentation géométrique, les moments de Hu, la skeletonisation, et les projections horizontales et verticales.

- La posture de la personne détectée est estimée en gardant la silhouette générée la plus similaire en fonction de la tâche précédente.

- Le tâche de filtrage des postures identifie les postures erronées détectées lors de la tâche précédente en exploitant leur cohérence temporelle. Le principe de stabilité des postures dit que pour un framerate suffisamment élevé la posture reste similaire sur plusieurs images consécutives. L'information fournit par le suivi des personnes détectées est utilisée pour retrouver les postures précédemment reconnues. Ces postures sont ensuite utilisées pour calculer la posture filtrée (i.e. la posture principale) en cherchant la posture qui apparaît le plus souvent sur une courte période.

Un aperçu des contributions effectuées pendant ce travail est donné dans la section suivante. Ensuite une discussion est faite pour montrer en particulier les

limitations de l'approche proposée. Finalement, des travaux futurs sont proposés pour améliorer l'approche dans la dernière section.

F.1 Aperçu des Contributions

- Les **avatars 3D de posture** ont été introduits pour modéliser les postures de personne à reconnaître comme décrit dans le chapitre 4. Les avatars sont inspirés par les précédents travaux en imagerie virtuelle qui ont été adaptés en proposant un modèle simplifié pour la reconnaissance de posture. Un avatar 3D de posture est composé d'un modèle 3D humain, qui définit les relations entre les 20 parties du corps, d'un ensemble de 23 paramètres qui définit les positions des différentes parties du corps, et d'un ensemble de primitives géométriques qui définit l'aspect visuel des différentes parties du corps. Le modèle 3D humain proposé contient aussi 10 joints qui sont les principales articulations du corps humain et les 20 parties du corps. Les articulations sont représentées avec les angles d'Euler pour modéliser les 9 postures d'intérêt : debout avec un bras levé, debout avec les bras le long du corps, debout les bras écartés, assis sur une chaise, assis parterre, penché, couché sur le dos les jambes tendus, et couché sur le côté recroquevillé. Les primitives du corps sont représentées avec un maillage de polygone pour obtenir un modèle 3D humain réaliste pour pouvoir entre autre générer des silhouettes de synthèse proche de celles du monde réel. De telles primitives améliorent la qualité de la reconnaissance. Les avatars 3D de posture sont indépendants des primitives du corps utilisées pour représenter les différentes parties du corps. En effet, différentes primitives peuvent être utilisée pour visualiser différents types d'avatar adaptés aux personnes observées. Les paramètres des articulations peuvent être modifiés pour modéliser les postures intermédiaires. De plus, les avatars 3D de postures sont classés de manière hiérarchique des postures générales aux postures détaillées.

- Une nouvelle **approche hybride** est proposée dans le chapitre 5 pour reconnaître les postures dans des séquences vidéo. L'approche est basée sur la caractérisation des silhouettes associées aux personnes détectées. L'approche combine des techniques 2D et l'utilisation des avatars 3D de posture décrits précédemment. Les techniques 2D sont utilisées pour garder un temps réel d'exécution alors que les avatars 3D sont utilisés pour obtenir une certaine indépendance au point de vue de la caméra. Plusieurs techniques 2D ont été testées pour représenter les silhouettes :

  - la première est basée sur la combinaison de différentes valeurs géométriques telles que l'air, le centre de gravité, l'orientation, l'eccentricité, et la compacité,
  - la seconde utilise les sept moments de Hu,
  - la troisième appelée skeletonisation, utilise des points caractéristiques du contour,

  - et la dernière est basée sur les projections horizontales et verticales de la silhouette.

  Les techniques 2D sont choisies en fonction de la qualité de la silhouette (qui dépend de la segmentation) et du temps d'exécution nécessaire pour représenter une silhouette donnée.

- Une **évaluation des représentations des silhouettes** a été effectuée dans le chapitre 6. Les avatars 3D de posture sont assez réalistes pour générer des données de synthèse pour différents points de vue. L'avatar 3D utilisé dans la génération des données de synthèse est différent de l'avatar 3D intervenant dans le processus de reconnaissance. Il y a trois principaux avantages à utiliser des données de synthèse :

  - Premièrement, beaucoup de donnée peut être généré pour n'importe quelle point de vue et l'avatar virtuel peut être placé n'importe où.
  - Deuxièmement, l'approche peut être étudiée suivant différent problèmes : la qualité de la segmentation, les postures intermédiaires, les postures ambiguës et la variabilité entre la personne observée et l'avatar 3D.
  - Finalement, un fichier de vérité terrain peut être automatiquement associé à chaque étape de la génération des données de synthèse.

  Une base de donnée de posture de synthèse a été générée pour 19 points de vue, 10 postures et pour 360 orientations (tous les degrés). Les points de vue sont localisés sur un quart de cercle autour de la personne tous les 5 degrés de 0 à 90 degrés donnant 68400 silhouettes. L'approche proposée a été validée sur des données de synthèse et réelles. Les projections horizontales et verticales donnent de meilleurs taux de reconnaissance que les représentations géométriques, la skeletonisation et les moments de Hu car cette représentation est plus robuste aux silhouettes bruitées et aux postures intermédiaires.

- Une **caractérisation** exhaustive **des postures ambiguës** a été effectuée à l'aide de la base de donnée des silhouettes de synthèse dans le chapitre 6. Les cas ambigus apparaissent quand les silhouettes représentant des postures différentes ont la même projection sur le plan image pour un point de vue donné. L'ambiguïté est alors caractérisée par une posture et une orientation pour un point de vue donné. Ces cas dépendent de la technique 2D utilisé pour représenter les silhouettes. Cette connaissance a priori peut être utilisée dans le processus de reconnaissance pour associer une valeur de confiance aux postures reconnues.

- Le résultat de l'approche proposée, les postures reconnues, a été utilisé pour la **reconnaissance d'action** dans le chapitre 7. Les actions visées sont des actions où seulement la personne considérée intervient. Les actions

sont modélisées par une machine à états finis dont chaque état est composé d'une ou plusieurs postures et d'un nombre minimal/maximal d'occurrence consécutive de ces postures. L'approche a été testée avec succès pour la détection de la chute et de la marche. L'action "chuter" est basée sur les postures générales alors que l'action "marcher" utilise des postures détaillées. Une nouvelle posture, la posture marche, a été facilement ajoutée à l'ensemble des postures d'intérêt et montre ainsi l'adaptabilité de notre approche.

- De plus, durant ce travail plusieurs outils ont été développés :

  - Le premier outil consiste en un **moteur 3D** capable de visualiser et de manipuler les avatars 3D de posture en faisant bouger les différentes parties du corps. De plus il permet d'extraire les silhouettes en fonction d'une caméra virtuelle. Le moteur est basé sur la librairie Mesa, en combinant plusieurs transformations telles que des rotations ou des translations pour animer les avatars 3D de posture. Ce moteur est un composant pour les outils décrits dans la suite.

  - Le second outil permet l'**animation de l'avatar 3D de posture** et de définir les paramètres associés à l'avatar 3D considéré. Chacune des parties du corps de l'avatar 3D peut être sélectionnée, et les paramètres correspondants aux articulations de la partie sélectionnée peuvent être modifiés pour obtenir l'avatar de posture désiré. Les paramètres sont sauvegardés et utilisés avec le moteur 3D précédemment décrit pour pouvoir afficher l'avatar 3D de posture ainsi définit.

  - Un troisième outil **génère de manière exhaustive des données de synthèse** en définissant différents points de vue et en donnant différentes orientations à l'avatar 3D de posture.

  - Le quatrième outil **génère des silhouettes de synthèse en utilisant des trajectoires**. Une scène virtuelle est observée depuis le haut (dans la direction verticale), l'utilisateur dessine une trajectoire et choisi pour les points importants de celle-ci la posture désirée. L'outil génère alors automatiquement la vidéo d'un avatar se déplaçant sur la trajectoire en prenant les postures désirées. Cet outil est utile dans un but démonstratif.

  - Le dernier outil est un **prototype pour reconnaître les postures** de personnes évoluant dans une séquence vidéo qui intègre la chaîne complète de traitement de l'acquisition à la reconnaissance de posture. Le prototype est une interface graphique permettant de visualiser les résultats obtenus avec l'approche proposée. Une description de ce prototype est donnée dans l'annexe A.

F.2 Discussion

Dans la section 3.1.2, plusieurs contraintes ont été identifiées pour proposer une approche générique : le temps réel, l'indépendance du point de vue de la caméra, une approche complètement automatique, et l'utilisation d'une seule caméra statique. Nous détaillons dans la suite comment ces contraintes ont été respectées.

- Le temps réel. L'algorithme proposé est capable de traiter entre 5 et 6 images par seconde en utilisant un flux de vidéo. Il a été montré que l'algorithme est efficace pour reconnaître certaines actions telles que la chute ou la marche. Ce temps de traitement est possible grâce à l'utilisation de représentation 2D des silhouettes.

- L'indépendance du point de vue de la caméra. Dans la section 6.3.2, l'approche a montré son indépendance au point de vue de la caméra. La caméra virtuelle permet la génération des silhouettes des avatars 3D de posture en utilisant le même point de vue que la véritable caméra. Ainsi, une caméra virtuelle peut être associée à une caméra réelle pour toute position et orientation de celle-ci.

- Une approche automatique. L'approche proposée est complètement automatique et peut être facilement adaptée à n'importe quelle séquence vidéo. De plus, cette approche peut être adaptée à différents types d'application en modifiant l'ensemble des postures d'intérêt. Une nouvelle posture d'intérêt peut être définit en déterminant un ensemble de paramètre spécifique (les angles d'Euler des articulations) pour représenter la posture désirée telle que la posture "marche".

- Une seule caméra statique. L'approche fonctionne avec une seule caméra statique en utilisant une base de connaissance a priori associée à la scène considérée. En particulier, la matrice de calibration de la caméra permet de calculer une approximation de la position 3D des personnes évoluant dans la scène et d'initialiser la caméra virtuelle.

L'approche est robuste à différent type de segmentation. L'approche a été testé avec l'algorithme "*watershed segmentation*" (qui a tendance à donner des silhouettes bruitées sur segmentées), avec l'algorithme "*VSIP segmentation*" (qui donne des silhouettes sous segmentées avec quelques trous) et avec l'algorithme de segmentation associé aux séquences d'analyse de la démarche (qui donne des silhouettes très bruitées).
Cependant l'approche proposée montre certaines limitations.
Le principal inconvénient de l'approche est sa limitation en terme de posture d'intérêt. La première raison de cette limitation est le temps de calcul. Le temps de calcul augmente lorsque le nombre de posture d'intérêt considérée augmente, limitant ainsi le nombre de posture considérée pour garder un temps de traitement rapide. La seconde raison est le pouvoir de discrimination entre les postures. Si plus de postures sont considérées le nombre de cas ambigu va augmenté rendant

les résultats de la reconnaissance non fiables.
Le second problème est le temps de calcul. La génération de silhouette des avatars 3D de posture est l'étape la plus coûteuse en terme de temps de calcul. Le temps nécessaire à la génération de 100 silhouettes correspondant à 10 avatars de postures et à un pas de rotation de 36 degrés est d'environ 1.28 seconde. En ne générant les silhouettes seulement lorsque la personne détectée a effectué un déplacement suffisant dans la scène, le temps de traitement est de 5 à 6 images par seconde. Pour réduire ce temps de calcul, des améliorations sont nécessaires.
De plus, nous avons fait l'hypothèse que la personne observée était isolée. Mais, cette personne peut être en partie cachée par des objets du contexte, ou elle peut interagir avec d'autre personnes.
Enfin, dans l'approche proposée l'avatar 3D de posture n'est adaptée à la personne étudiée en ne prenant en compte seulement la hauteur de celle-ci.

F.3 Travaux Futurs

Ce travail peut être améliorer de différentes manières classées en perspectives à court et long termes.

F.3.1 Perspectives à Court Terme

Occlusion

La scène virtuelle peut être prise en compte pour résoudre le problème des occlusions. Un modèle 3D de la scène peut être affichée en même temps que l'avatar 3D. En positionnant correctement l'avatar 3D dans la scène, une silhouette occludée peut être extraite et comparée avec celle détectée. Ici, la technique de Z-buffer, décrite dans la section 5.2.2.3, ne peut plus être utilisée pour extraire la silhouette puisque dans la scène il n'y a plus seulement l'avatar 3D mais aussi les objets contextuels. Une simple segmentation couleur peut être envisagée pour résoudre ce problème en coloriant les objets du contexte avec la même couleur que celle du fond. Un exemple d'une silhouette occludée est donnée dans la figure F.1.

Gestion des déformations avec la caméra virtuelle

Durant ce travail, différents tests ont été réalisés à l'aide d'un capteur CMOS équipé d'un objectif grand angle (figure F.2). Utilisé de tel capteur implique des déformations géométriques au niveau de l'image. Le modèle utilisé pour la caméra virtuelle peut être amélioré afin de prendre en compte les déformations de la caméra réelle dans le but d'obtenir des silhouettes déformées. Les silhouettes déformées pourront donc être comparées directement avec la silhouette de la personne détectée. Le modèle linéaire utilisé lors de l'étape de calibration de la caméra n'est plus valide pour ce type d'image et une autre méthode de calibration

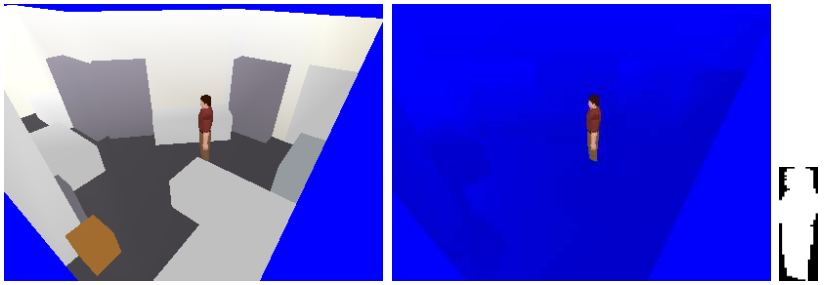

Figure F.1: Les objets du contexte et l'avatar 3D sont affichés dans la scène virtuelle. Les objets sont coloriés en bleu pour pouvoir faire une simple segmentation couleur afin d'obtenir une silhouette occludée.

doit être envisagée pour gérer ces déformations.

Figure F.2: Déformations géométriques observables sur une image provenant d'un capteur CMOS muni d'un objectif grand angle.

F.3.2 Perspectives à Long Terme

Adaptabilité dynamique des primitives du modèle humain

Durant ce travail, le modèle 3D est automatiquement adapté à la personne étudiée en considérant seulement sa hauteur 3D. Plus d'information sur la personne doit être calculée pour initialiser un avatar 3D plus proche de celle-ci. Ces informations pourraient être sa corpulence, ou les habits portés par la personne considérée par exemple. Cet avatar 3D permettrait la génération de silhouette plus précise et donc améliorer les taux de reconnaissance de notre approche. L'information de la corpulence peut déjà être gérée par le modèle d'avatar 3D proposé en jouant sur la taille des différentes primitives du corps. Une solution pour intégrer l'information des vêtements est d'avoir plusieurs primitives 3D

associées à différents types de vêtements. Ceci peut être simplement atteint en définissant des primitives qui représentent les différentes parties du corps pour un vêtement donné. Par exemple, une primitive peut être défini pour représenter une tête couverte d'un chapeau. Le moteur 3D permettant d'afficher les avatars 3D de postures, montrera une tête plus complexe en terme de géométrie.

Variabilité de l'avatar 3D de posture ou reconnaissance de geste

L'approche proposée est basée sur l'utilisation d'avatar 3D de posture statique et peut ainsi induire des erreurs de reconnaissance pour les postures intermédiaires comme montré à la section 6.3.3. Lorsqu'un avatar 3D de posture est reconnu, les paramètres de celui-ci pourraient être changés pour obtenir une silhouette plus proche de celle détectée. Cette amélioration pourrait autoriser la reconnaissance de geste. Une thèse sur le sujet de la reconnaissance de geste a démarrée dans l'équipe ORION.
Un autre point concernant la reconnaissance de geste est la génération de données de synthèse. Comme décrit dans la section 6.3.1, les données de synthèse peuvent être utilisées pour évaluer facilement un algorithme de reconnaissance de posture. La même analogie peut être faite pour les algorithmes de reconnaissance de geste. Une amélioration doit alors être faite au niveau de la représentation des rotations des articulations de l'avatar 3D. En effet, la représentation actuelle est basée sur les angles de Euler qui n'est pas adaptée pour effectuer des animations. Les quaternions peuvent être utilisés pour représenter les rotations comme décrit dans l'annexe C pour pouvoir animer l'avatar 3D.

Choix de la représentation 2D des silhouettes

Nous avons vu que le choix de la représentation 2D d'une silhouette dépendait de la qualité de la dite silhouette. Nous avons montré que les projections horizontale et verticale donnaient les meilleurs résultats pour différents types de segmentation dans la section 6.4. Mais cette représentation ne permet pas d'extraire facilement des informations plus précises concernant la personne étudiée (localisation de ses mains, ou de sa tête par exemple) comme pourrait le faire la skeletonisation. Une tâche intéressante serait de pouvoir choisir automatiquement la représentation 2D en évaluant la qualité de la segmentation et en tenant compte de l'information nécessaire par l'application.

Amélioration du temps de calcul

La principale limitation de lapproche proposée est le temps de calcul nécessaire à la génération des silhouettes des avatars 3D de posture. Une façon évidente de réduire celui-ci est de générer moins de silhouettes. Un automate peut être utilisé

pour représenter les transitions possibles entre les postures. La reconnaissance de la posture d'une personne dans une image donnée peut être utilisée pour guider la reconnaissance de la posture de la même personne dans l'image suivante. En particulier, en permettant de prédire quels avatars 3D de posture utiliser. L'ensemble des postures d'intérêt serait donc adapté automatiquement en ne considérant seulement les postures autorisées. Ce traitement devrait réduire le temps de calcul. De plus, l'information sur l'orientation de la personne peut aussi être utilisée pour générer seulement les silhouettes pour un nombre d'orientation de l'avatar très restreint. Un algorithme décrit dans [Zuniga et al., 2006] propose de classer des objets détectés dans une séquence vidéo (sous forme de silhouette) en déterminant le parallélépipède 3D contenant cet objet. L'orientation de ce parallélépipède peut être utilisée comme approximation de l'orientation de la personne évoluant dans la scène.

Segmentation hiérarchique

L'approche de reconnaissance de posture de personne est basée sur l'étude d'une silhouette binaire. Une amélioration peut être faite en considérant plus qu'une seule région, mais un ensemble de régions constituant la silhouette. En utilisant, une telle segmentation hiérarchique, les différentes régions peuvent être utilisées pour localiser les différentes parties du corps humain afin d'aider à l'initialisation de l'avatar 3D de posture.

Amélioration de la segmentation

La posture reconnue (et donc la silhouette de l'avatar 3D) peut être utilisée pour améliorer l'étape de segmentation en aidant à la paramétrisation de l'algorithme de segmentation. La silhouette reconnue permettrait de détecter quelles parties de la personne ne sont pas sur la silhouette ou quels pixels de la silhouette n'appartiennent pas à la personne et donc donner une indication sur comment faire évoluer les paramètres de la segmentation pour obtenir de meilleures silhouettes.

Bibliography

[Agarwal and Triggs, 2006] Agarwal, A. and Triggs, B. (2006). Recovering 3d human pose from monocular images. *IEEE Transactions on Patten Analysis and Machine Intelligence*, 28(1).

[Aggarwal and Cai, 1999] Aggarwal, J. and Cai, Q. (1999). Human motion analysis : a review. *Computer vision Image understanding.*

[Al-Zahawi, 2006] Al-Zahawi, H. (2006). The computing research association. http://www.cra.org/Activities/craw/dmp/awards/2002/alzahawi/Project.html.

[Ardizzone et al., 2000] Ardizzone, E., Chella, A., and Pirrone, R. (2000). Pose classification using support vector machine. In *Proceedings of IEEE International Joint Conference on Neural Networks IJCNN 2000*, volume 1, pages 317–322.

[Ascension, 2006] Ascension (2006). Ascension technology corporation. http://www.ascension-tech.com/products/motionstarwireless.php.

[Aslan and Tari, 2005] Aslan, C. and Tari, S. (2005). An axis-based representation for recognition. In *Proceedings of the tenth IEEE International Conference on Computer Vision (ICCV).*

[Athitsos and Sclaroff, 2001] Athitsos, V. and Sclaroff, S. (2001). 3d hand pose estimation by finding appearance-based matches in a large database of training views. In *Proceedings of IEEE Workshop on Cues in Communication.*

[Aubel et al., 2000] Aubel, A., Boulic, R., and Thalmann, D. (2000). Real-time display of virtual humans: levels of details and impostors. *Circuits and Systems for Video Technology*, 10:207–217.

[Avanzi et al., 2001] Avanzi, A., Bremond, F., and Thonnat, M. (2001). Tracking multiple individuals for video communication. In *Proceedings of the International Conference on Image Processing.*

[Avanzi et al., 2005] Avanzi, A., Bremond, F., Tornieri, C., and Thonnat, M. (2005). Design and assessment of an intelligent activity monitoring platform. *EURASIP journal on applied signal processing, special issue in Advances in intelligent vision systems: method and applications*, pages 2359–2374.

[Barron and Kakadiaris, 2000] Barron, C. and Kakadiaris, I. (2000). Estimating anthropometry and pose from a single image. *Computer Vision and Pattern Recognition*, 1:669–676.

[Barron and Kakadiaris, 2003] Barron, C. and Kakadiaris, L. A. (2003). On the improvement of anthropometry and pose estimation from a single uncalibrated image. *Machine Vision and Applications*, 14:229–236.

[Barrow et al., 1977] Barrow, H., Tenenbaum, J., Bolles, R., and Wolf, H. (1977). Parametric correspondence and chamfer matching: two new techniques for image matching. In *Proceeding 5th International Joint Conference Artificial Intelligence*, pages pp 659–663.

[Baumberg and Hogg, 1995] Baumberg, A. and Hogg, D. (1995). An adaptive eigenshape model. In *Proceedings of British Machine Vision Conference, BMVC 95*.

[Belongie et al., 2002] Belongie, S., Malik, J., and Puzicha, J. (2002). Shape matching and object recognition using shape contexts. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(24):509–522.

[Bioengineering, 2006] Bioengineering (2006). Bts bioengineering. http://www.bts.it/eng/default.htm.

[Bobick and Davis, 2001] Bobick, A. F. and Davis, J. W. (2001). The recognition of human movement using temporal templates. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(3).

[Borg et al., 2006] Borg, M., Thirde, D., Ferryman, J., Fusier, F., Valentin, V., Bremond, F., and Thonnat, M. (2006). A real-time scene understanding system for airport apron monitoring. In *Proceedings of 2006 IEEE International Conference on Computer Vision Systems*.

[Borgefors, 1986] Borgefors, G. (1986). Distance transformations in digital images. In *Computer Vision Graphics, and Image Processing*, volume 34, pages pp. 344–371.

[Boulay et al., 2005] Boulay, B., Bremond, F., and Thonnat, M. (2005). Posture recognition with a 3d human model. In *Proceedings of Imaging for Crime Detection and Prevention*.

[Bradski and Davis, 2002] Bradski, G. R. and Davis, J. W. (2002). Motion segmentation and pose recognition with motion history gradients. *Machine Vision and Applications*, pages 174–184.

[Bremond, 1997] Bremond, F. (1997). *Environnement de résolution de problèmes pour l'interprétation de séquences d'images*. PhD thesis, Université de Nice-Sophia Antipolis.

[Cavallaro et al., 2004] Cavallaro, A., Salvador, E., and Ebrahimi, T. (2004). Detecting shadows in images sequences. *Visual Media Production.*

[Cedras and Shas, 1995] Cedras, C. and Shas, M. (1995). Motion-based recognition: a survey. *Image Vision Comput*, 13(2):129–155.

[Center, 2006] Center, D. H. R. (2006). Non-invasive and unrestrained monitoring of human respiratory system by sensorised environment. http://www.dh.aist.go.jp/research/enabling/SELF/index.php.en.

[Chen et al., 2006] Chen, H.-s., Chen, H.-T., Chen, Y.-W., and Lee, S.-Y. (2006). Human action recognition using star skeleton. In *Proceedings of the 4th ACM international workshop on Video surveillance and sensor networks*, pages 171–178.

[Codamotion, 2006] Codamotion (2006). Codamotion: the science of real-time motion capture and analysis. http://www.charndyn.com/.

[Cohen and Li, 2003] Cohen, I. and Li, H. (2003). Inference of human postures by classification of 3d human body shape. In *Proceedings of IEEE International Workshop on Analysis and Modeling of Faces And Gestures, ICCV 2003.*

[Cohen et al., 2001] Cohen, I., Medioni, G., and Gu, H. (2001). Inference of 3d human body posture from multiple cameras for vision based user interface. In *5th World Multi-Conference on Systemics, Cybernetics and Informatics.*

[Cucchiara et al., 2003] Cucchiara, R., Prati, A., and Vezzani, R. (2003). Domotics for disability: smart surveillance and smart video server. In *Proceedings of 8th Conference of the Italian Association of Artificial Intelligence - Workshop on "Ambient Intelligence"*, pages 46–57.

[Cupillard et al., 2002] Cupillard, F., Bremond, F., and Thonnat, M. (2002). Behaviour recognition for individuals, groups of people and crowd. In *Proceedings of the International Conference on Intelligent Distributed Surveillance Systems.*

[D'Apuzzo et al., 1999] D'Apuzzo, N., Plankers, R., Fua, P., Gruen, A., and Thalmann, D. (1999). Modeling human bodies from video sequences. *Videometrics VI, Part of IS&T/SPIE's Symposium on Electronic Imaging '99.*

[Dedeoglu et al., 2006] Dedeoglu, Y., Toreyin, B. U., Gudukbay, U., and Cetin, A. E. (2006). Silhouette-based method for object classification and human action recognition in video. In *Proceedings of European Conference on Computer Vision in Human Computer Interaction*, pages 64–77.

[Delamarre and Faugeras, 2001] Delamarre, Q. and Faugeras, O. (2001). 3d articulated models and multi-view tracking with silhouettes. *Special Issue on Modelling People, Computer Vision and Image Understanding*, pages 328–357.

[Doermann and Mihalcik, 2000] Doermann, D. and Mihalcik, D. (2000). Tools and techniques for video performance evaluation. In *Proceedings of the International Conference on Pattern Recognition (ICPR'00)*, pages pages 167–170.

[Erdem et al., 2006] Erdem, A., Erdem, E., and Tari, S. (2006). Articulation prior in an axial representation. *International Workshop on the Representation and use of Prior Knowledge in Vision*.

[Fujiyoshi and Lipton, 1998] Fujiyoshi, H. and Lipton, A. J. (1998). Real-time human motion analysis by image skeletonization. In *Proceedings of the Workshop on Application of Computer Vision*.

[Fujiyoshi et al., 2004] Fujiyoshi, H., Lipton, A. J., and Kanade, T. (2004). Real-time human motion analysis by image skeletonization. *IEICE Transactions on Information and Systems*, E87-D(1):113–120.

[Gavrila, 1999] Gavrila, D. (1999). The visual analysis of human movement: A survey. *Computer Vision and Image Understanding*, 73(1):82–98.

[Gavrila and Davis, 1996] Gavrila, D. and Davis, L. (1996). 3d model-based tracking of humans in action: a multi-view approach. *Computer Vision and Pattern Recognition*, pages 73–80.

[Georis et al., 2006] Georis, B., Maziere, M., Bremond, F., and Thonnat, M. (2006). Evaluation and knowledge representation formalisms to improve video understanding. In *Proceedings of the International Conference on Computer Vision Systems (ICVS'06)*.

[H-Anim, 2006] H-Anim (2006). Humanoid animation working group. http://h-anim.org/. The H-anim specifications.

[Haritaoglu et al., 1998a] Haritaoglu, I., Harwood, D., and Davis, L. S. (1998a). Ghost: A human body part labeling system using silhouettes. In *Proceedings of 14th International Conference on Pattern Recognition, Brisbane, Australia*.

[Haritaoglu et al., 1998b] Haritaoglu, I., Harwood, D., and Davis, L. S. (1998b). W4: Who? when? where? what? a real time system for detecting and tracking people. In *Proceedings of 3rd International Conference on Face And Gesture Recognition, Nara, Japan*.

[Intersense, 2006] Intersense (2006). Intersense: Sensing every move. http://www.isense.com/products.aspx?id=45&.

[Iwasawa et al., 1999] Iwasawa, S., Ohya, J., Takahashi, K., Sakaguchi, T., Kawato, S., Ebihara, K., and Morishima, S. (1999). Real-time, 3d estimation of human body postures from trinocular images. In *Proceedings of IEEE International Workshop on Modelling People*.

[Ju et al., 1996] Ju, S. X., Black, M. J., and Yacoob, Y. (1996). Cardboard people: A parameterized model of articulated image motion. In *Proceedings of 2nd International Conference on Automatic Face and Gesture-Recognition*, pages 38–44.

[Kameda et al., 1993] Kameda, Y., Minoh, M., and Ikeda, K. (1993). Three dimensional pose estimation of an articulated object from its silhouette image. *ACCV93*.

[Kelly et al., 2002] Kelly, K. E., Phillips, C. L., Cain, K. C., Polissar, N. L., and Kelly, P. B. (2002). Evaluation of a nonintrusive monitor to reduce falls in nursing home patients. *American Medical Directors Association*.

[Kemp et al., 1998] Kemp, A., Janssen, A., and Van Der Kamp, B. (1998). Body position can be monitored using miniature accelerometers and earth magnetic field sensors. *Electroencepha Clinical Neurophysiology*, 109:484–488.

[Lerallut, 2006] Lerallut, R. (2006). *Modelisation et Interpretation d'Images a l'aide de Graphes*. PhD thesis, Ecole des Mines de Paris.

[Loncaric, 1998] Loncaric, S. (1998). A survey of shape analysis techniques. *Pattern Recognition*, 31(8):983–1001.

[Mariano et al., 2002] Mariano, V., Min, J., Park, J.-H., Kasturi, R., Mihalcik, D., Doermann, D., and Drayer, T. (2002). Performance evaluation of object detection algorithms. In *Proceedings of the International Conference on Pattern Recognition (ICPR'02)*.

[Mesa, 2006] Mesa (2006). The mesa 3d graphics library. http://www.mesa3d.org/.

[Mittal et al., 2003] Mittal, A., Zhao, L., and Davis, L. S. (2003). Human body pose estimation using silhouette shape analysis. In *Proceedings of IEEE Conference on Advanced Video and Signal Based Surveillance (AVSS'03)*.

[Moenne-Locoz et al., 2003] Moenne-Locoz, N., Bremond, F., and Thonnat, M. (2003). Recurrent bayesian network for the recognition of human behaviors from videos. In *Proceedings of the 3rd International Conference on Computer Vision Systems*, pages 68–77.

[Moeslund, 2003] Moeslund, T. B. (2003). *Computer vision-based motion capture of body language*. PhD thesis, Aalborg University.

[Moeslund and Granum, 2000] Moeslund, T. B. and Granum, E. (2000). 3d human pose estimation using 2d-data and an alternative phase space representation. In *Procedure Humans 2000*.

[Moeslund and Granum, 2001] Moeslund, t. B. and Granum, E. (2001). A survey of computer vison-based human motion capture. *Computer Vision and Image Understanding*, 81:231–268.

[Mori and Malik, 2002] Mori, G. and Malik, J. (2002). Estimating human body configurations using shape context matching. In *Proceedings of European Conference on Computer Vision*.

[Noury et al., 2004] Noury, N., Barralon, P., Virone, G., Rumeau, P., and Boissy, P. (2004). Un capteur intelligent pour detecter la chute: fudion multi-capteurs et decision a base de regles. *Communication C2I-2004-78*, pages 1–8.

[OpenGL, 2006] OpenGL (2006). The opengl home. http://www.opengl.org/.

[Panini and Cucchiara, 2003] Panini, L. and Cucchiara, R. (2003). A machine learning approach for human posture detection in domotics applications. In *Proceedings of the 12th Internatiional Conference on Image Analysis and Processing (ICIAP'03)*.

[Park et al., 2000] Park, J.-S., Oh, H.-S., Chang, D.-H., and Lee, E.-T. (2000). Human posture recognition using curve segments for image retrieval. In *Proceedings of SPIE Storage and Retrieval for Media Databases 2000*, volume 3972, pages 2–11.

[Rosales, 1998] Rosales, R. (1998). Recognition of human action using moment-based features. Report BU 98-020, Boston University Computer Science, Boston, MA 02215.

[Rosales and Sclaroff, 2000a] Rosales, R. and Sclaroff, S. (2000a). Inferring body pose without tracking body parts. In *Proceedings of IEEE Computer Vision and Pattern Recognition*.

[Rosales and Sclaroff, 2000b] Rosales, R. and Sclaroff, S. (2000b). Specialized mappings and the estimation of human body pose from a single image. In *Proceedings of IEEE Workshop on Human Motion*.

[Rosales et al., 2001] Rosales, R., Siddiqui, M., Alon, J., and Sclaroff, S. (2001). Estimating 3d body pose using uncalibrated cameras. Technical report, Boston University.

[Rosenfeld and Kak, 1976] Rosenfeld, A. and Kak, A. C. (1976). *Digital Picture Processing*. Compter science and applied mathematics.

[Sarkar et al., 2005] Sarkar, S., Phillips, P. J., Liu, Z., Vega, I. R., Grother, P., and Bowyer, K. W. (2005). The humanid gait challenge problem: Data sets, performance, and analysis. *IEEE transactions on Pattern Analysis and Machine Intelligence*, 27(02):pp.162–177.

[Shakhnarovich et al., 2003] Shakhnarovich, G., Viola, P., and Darrell, T. (2003). Fast pose estimation with parameter sensitive hashing. Technical report, MIT - Artificial Intelligence Laboratory.

[Shimada et al., 2001] Shimada, N., Kimura, K., and Shirai, Y. (2001). Real-time 3d hand posture estimation based on 2d appearance retrieval using monocular camera. In *Proceedings of International Workshop on Recognition, Analysis, and Tracking of Faces and Gestures in Real-Time Systems (RATFG-RTS)*, pages 23–30.

[Sidenbladh et al., 2002] Sidenbladh, H., Black, M. J., and Sigal, L. (2002). Implicit probabilistic models of human motion for synthesis and tracking. In *European Conference on Computer Vision*, volume 1, pages pp. 784–800.

[Sminchisescu and Telea, 2002] Sminchisescu, C. and Telea, A. (2002). Human pose estimation from silhouettes : A consistent approach using distance level sets. In *Proceedings of International Conference on Computer Graphics, Visualization and Computer Vision (WSCG 2002)*.

[Tanikawa and Takahashi, 2003] Tanikawa, T. and Takahashi, K. (2003). Remarks on neural-network-based human body posture estimation from human silhouette image. In *Proceedings of International Conference on Management of Innovation and Technology 2003*.

[Thayananthan et al., 2003] Thayananthan, A., Stenger, B., Torr, P., and Cipolla, R. (2003). Shape context and chamfer matching in cluttered scenes. In *Proceedings of Conference on Computer Vision and Pattern Recognition*, volume 1, pages 127–133.

[Tornieri et al., 2002] Tornieri, C., Bremond, F., and Thonnat, M. (2002). Updating of the reference image for visual surveillance systems. In *Proceedings of the International Conference on Intelligent Distributed Surveillance Systems*.

[Tsai, 1986] Tsai, R. (1986). An efficient and accurate camera calibration technique for 3d machine vision. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pages pp. 364–374.

[Veltkamp and Hagedoorn, 2001] Veltkamp, R. C. and Hagedoorn, M. (2001). State of the art in shape matching. *Principles of visual information retrieval*, pages 87–119.

[Vicon, 2006] Vicon (2006). Vicon. http://www.vicon.com/.

[Viper, 2006] Viper (2006). Viper: The video performance evaluation ressource. http://viper-toolkit.sourceforge.net/.

[Vosinakis and Panayiotopoulos, 2001] Vosinakis, S. and Panayiotopoulos, T. (2001). Simhuman: A platform for real-time virtual agents with planning capabilities. In *IVA'01 workshop*.

[VRML, 2006] VRML (2006). The virtual reality modeling language. http://www.web3d.org/x3d/specifications/vrml/ISO-IEC-14772-VRML97/. The VRML specifications.

[Vu et al., 2002] Vu, T., Bremond, F., and Thonnat, M. (2002). Temporal constraints for video interpretation. In *Proceedings of the 16th European Conference on Artificial Intelligence*.

[Vu et al., 2006] Vu, V. T., Bremond, F., Davini, G., Thonnat, M., Pham, Q.-C., Allezard, N., Sayd, P., Rouas, J.-L., Ambellouis, S., and Flancquart, A. (2006). Audio-video event recognition system for public transport security. In *Proceedings of ICDP*.

[Vu et al., 2003] Vu, V. T., Bremond, F., and Thonnat, M. (2003). Automatic video interpretation: a novel algorithm for temporal scenario recognition. In *Proceedings of the 18th International Joint Conference on Artificial Intelligence*.

[Wang et al., 2003] Wang, L., Hu, W., and Tan, T. (2003). Recent developments in human motion analysis. *Pattern Recognition*, 36:585–601.

[Williams et al., 1998] Williams, G., Doughty, K., Cameron, K., and Bradley, D. (1998). A samrt fall and activity monitor for telecare applications. In *Proceedings of 20th International conference IEEE-EMBS*, pages 1151–1154.

[Wren et al., 1997] Wren, C., Azarbayejani, A., Darrell, T., and Pentland, A. (1997). Pfinder: Real-time tracking of the human body. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(7):780–785.

[Wu, 2000] Wu, G. (2000). Distinguishing fall activities from normal activities by velocity characteristics. *Journal of Biomechanics*, 33:1497–1500.

[Yamamoto et al., 1998] Yamamoto, M., Sato, A., Kawada, S., Kondo, T., and Osaki, Y. (1998). Incremental tracking of human actions from multiple views. *Computer vison and pattern recognition*, pages 2–7.

[Zhao et al., 2004] Zhao, J., Li, L., and Keong, K. C. (2004). A model-based approach for human motion reconstruction from monocular images. In *Proceedings of 2nd International Conference on Information Technology for Application (ICITA)*.

[Zhao and Nevatia, 2004] Zhao, T. and Nevatia, R. (2004). Tracking multiple humans in complex situations. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(9).

[Zuniga et al., 2006] Zuniga, M., Bremond, F., and Thonnat, M. (2006). Fast and reliable object classification in video based on a 3d generic model. In *VIE*.