UCL
Université
catholique
de Louvain

Université catholique de Louvain
Faculté des Sciences Appliquées
LABORATOIRE DE TÉLÉCOMMUNICATIONS
ET
TÉLÉDÉTECTION

B - 1348 Louvain-la-Neuve
Belgique

Dual Bayesian and Morphology-based Approach for Markerless Human Motion Capture in Natural Interaction Environments

Pedro Correa Hernández

Thèse présentée en vue de l'obtention du grade de
Docteur en Sciences Appliquées

Examination Committee:

Benoît MACQ (UCL) - *Supervisor*
Ferran MARQUÉS (UPC, Spain) - *Supervisor*
Xavier MARICHAL (Alterface)
Christophe DE VLEESCHOUWER (UCL)
Justus PIATER (ULg)
Luc VANDENDORPE (UCL) - *President*

June 2006

Acknowledgements

It is often said that the achievement of a PhD thesis is a lonesome desert crossing. And even if that though is far from being overstated, I have had the chance to count on a solid team that has been my personal and permanent oasis during this journey. I thus wish to express my most sincere gratitude to the following people without whom - and I am afraid this will sound much more *cliché* than it really is - this work would have not been possible...or in any case much less fun.

My very first thanks go to the wife of my dreams, Aurélie (to whom I have dedicated the first letter of each chapter), and also my parents and brother for giving half their lives and all their love to support me and my work. Perhaps this is not the most usual place to say this but I am really proud of having you all, really.

I also want to express my gratitude to my first co-supervisor, Prof. Benoît Macq, for thinking of me in the first place, four years back, to start this enthralling research topic. Ever since he has put all the necessary means and contacts in place (along with his optimism and kindness) in order to create the basis for a fruitful itinerary, in both research and human terms. Dr. Xavier Marichal and all the Alterface S.A team, for welcoming me so warmly in their work environment as one of theirs, and putting their know-how and means to help my work. Special thanks to Willy and Xavier Sohy for their time, and the design team (Laurence, Kily, Luc et Nico) for the good vibes.

Prof. Ferran Marqués, also co-supervisor of this PhD, has been the closest research partner from the very beginning. His unconditional scientific support has been essential and his human touch and kindness during the last four years have made him what any other supervisor would have feared: a close friend. The stay at the UPC was one of the most enriching periods of my life, and his welcome is responsible for much of it. So as my friends over there: Hugo, Joel, Camilo, Carles and Maribel.

Jacek Czyz's help in the last stages of the Thesis has also been crucial. The collaboration with this talented scientific (and musician) has been extremely fruitful and pleasant. Toshiyuki Umeda's unbelievable programming skills have also been of great help. And actually, the whole TELE

team at the UCL have helped creating a very social and inspirational work environment. Special mentions to Jérôme, Damien, Atonin, Fod, Mathieu and Cédric, among all the others.

I would also like to thank Christophe De Vleeschouwer and Prof. Justus Piater, that as jury members have helped to improve the final document, with such constructive remarks.

And finally, very special thanks to my good friends in Belgium (and around the world) that make my life away from my homeland so incredibly easy. They know who they are.

Contents

List of Figures

1 Introduction

True interactivity is not about clicking on icons or downloading files, it is about encouraging communication.

Ed Scholssberg

It would appear that we have reached the limits of what it is possible to achieve with computer technology, although one should be careful with such statements, as they tend to sound pretty silly in 5 years.

John Von Neumann (ca. 1949)

1.1 Context of the Work

OVER the past few years the development of human-computer interfaces that enable a more natural man-machine interaction has witnessed a world-wide expansion within the computer vision domain. The evolution towards the ultimate intelligent machine (capable of autonomous thinking, understanding and thus having the ability to perceive its surrounding environment) took off forcing the users to adapt very strongly to the dictations of technological limitations (dichromatic text screens, keyboards, etc). Lately, research and technological developments (higher computation speeds and higher available

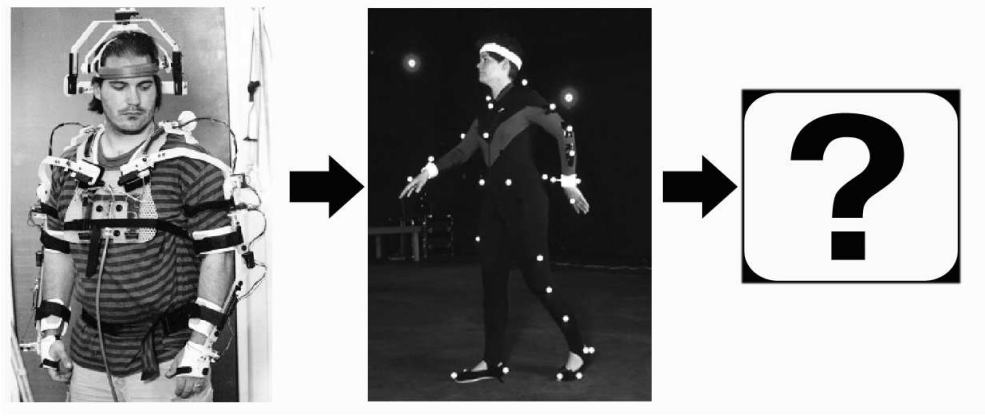


Figure 1.1: *Evolution of the motion capture devices*

memory being for instance critical advances in the field of computer vision), have contributed in making the situation begin to evolve towards a reverse framework where the machine endorses the adaptation charge in order to understand the human communication.

Currently, the development of human-computer interfaces is a very active area of research. Indeed, familiar HCI tools (such as keyboards and mice) are proving to be limiting the creativity, speed and naturalness when interacting with the machine [Pavlovic et al. 1997]. Furthermore, it is even widely believed that the means we use to communicate with the computer may create in the long run an efficiency bottleneck as computational capabilities and the information flow continue to skyrocket [Pavlovic et al. 1997]. Thus, worldwide research towards novel forms of man-machine communication has been greatly driven forward.

Since gestures are the most natural and spontaneous form of non-verbal communication in human societies, gestural interfaces have quite naturally become one of the preferred options to tackle this problem. First attempts to address this issue lead to the so-called data-gloves [Fels and Hinton 1993], [Sturman and Zeltzer 1994] [Quam 1990] and other mechanical encumbering devices tethered with cables (Fig. 1.1).

Although these instruments are still used nowadays, they have been grad-

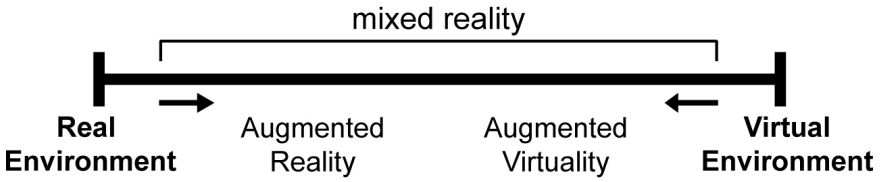


Figure 1.2: *Milgram's Reality-Virtuality Continuum.*

ually confined to very specialized areas, in which high precision rates are compulsory and have priority over natural interaction. Nowadays there is still indeed an inevitable trade-off between precision and natural interaction to be made.

This gradual evolution towards a more natural interaction emerged with Myron's Krueger visionary work in the mid-1970s in which the computer responded to vague gestures of the audience by interpreting, and even anticipating, their actions [Krueger 1991; 1985]. Since then, gestural interfaces have experienced a dramatic increase of attention in the recent years with the emergence of virtual reality and even more with *augmented reality*. This term has been defined as "a computer generated, interactive, three-dimensional environment in which a person is immersed" [Aukstakalnis and Blatner 1992], the degree of immersion being the key factor to differentiate virtual reality from augmented reality. To this matter, Milgram [Milgram and Kishino 1994] describes a taxonomy that identifies how augmented reality and virtual reality works are related. This well-known taxonomy is illustrated in Figure 1.2.

Last but not least, the recent progress in speech recognition has again made gestural interaction very topical in order to follow Bolt's seminal work "put that there" [Bolt 1980] towards robust multimodal speech-gesture interaction. Gestures can indeed be combined with speech and thus used as complementary or even redundant information, increasing robustness and naturalness (by allowing the free use of pronouns for example) [Kettebekov et al. 2005].

In order to extract the user's gestures and postures in a non-invasive manner, some human features (the choice of these features will be discussed later) need to be detected, located and followed (or *tracked*). In the mean

time this stage has to remain as invisible to the user as possible. To achieve this, new detection and tracking methods are needed. Indeed, non invasive gestural interaction conceals considerable challenges, and the feature extraction issue for instance is still far from being solved, as classic feature extraction techniques like skin-color detection begin to meet their limitations.

The main challenge of natural interaction algorithms is to make machines able to recognize purposeful motor activities, making processing algorithms able to deal with the great variety of shapes and styles a human gesture can assume.

Tracking human features (those necessary to extract and interpret a human motion) is indeed an inherently difficult problem since they have irregular and often unpredictable trajectories, and are furthermore obviously dependent, leading to frequent occlusions and/or fusions. Furthermost, clothing is not known a priori, can be textured and changing over time, and is frequently misdetected as skin.

1.2 Goal and Motivations

In the line of man-machine interfaces, our goal with this thesis has been to tackle the problem of novel forms of human-computer interfaces (HCI) that would allow an enhancement of creativity (or at least to avoid the creativity bottleneck that begins to rise with standard HCI tools).

Basically, two approaches have been present up to this point in this area (see next section for details). Firstly, approaches using complex models and/or machine learning as well as extensive hardware (at least three monoscopic or even stereoscopic cameras), very often extracting action recognition and/or whole body pose and focusing on the accuracy of the motion extraction. These approaches are very seldom able to work at real-time paces. On the other hand, we have wanted to stand among the approaches that focused on cheap, real-time interaction. We wanted indeed to explore algorithms that could actually be implemented and applied to nowadays applications (or enhanced versions of todays applications).

It must be pointed out that we do not intend to determine the body posture in every situation. Rather, our purpose is to provide a system that allows **intentional** gestural interaction with a computer. For this reason, we believe that the user can adapt his/her gestures so that auto-occlusions are not very frequent, auto-occlusions being indeed the major downfall of monoscopic approaches. Machine learning algorithms focus indeed in capturing human motion by retrieving whole body pose, whereas we have chosen to only extract a restricted group of human features that will still enable a full body interaction.

Furthermore, we have chosen to deal with an additional constraint: not to use any skin color information in order to extract body parts exposing a certain amount of flesh (typically face and hands). We wanted indeed to explore novel methods of human feature extraction, and skin color is highly environment dependent and may lead in some applications to unstable results [Soriano et al. 2000]. Moreover color can also be unavailable as with infra-red lighting and for instance unusable to detect the feet. It is finally very dependent on the user's clothing (since the amount of visible flesh can very much vary from user to user), and trying to impose a certain recognition pattern (not to wear shorts or to be bare foot, for example) did not feel as going in the right *natural interaction* direction.

These a priori constraints might seem quite strong, and yet as it will be shown - mostly in Chapter 4 and Section 3.5 - they allow applications in a wide range of fields, such as gestural navigation, interactive edutainment, virtual/mixed/augmented reality video-gaming, home training, and even Computer Graphics character animation.

We thus propose a trade-off between naturalness and simplicity -which translates in real-time paces and affordable hardware- and the amount of extracted information.

Note that, even though we consider the real-time human feature extraction, classification and tracking (without global posture or action recognition) as being the major contribution of this work, efficient tools have been developed in order to tackle the whole body pose extraction problem (see Sections 2.4.2 and 5.3).

1.3 Targeted Applications

We will discuss in this section the general application scope or research fields that surround this work, this is, the different applications of man-machine natural interactions. These applications are in their turn part of what is frequently referred to as the "Looking at People" [Pentland 2000] systems which ultimate goal is the achievement of an independent and autonomous machine capable of evolving unnoticed in human environments.

These applications can be organized mainly in five different areas, that are summarized in table 1.1, extracted from Gavrila's survey on human movement analysis [Gavrila 1999]. Yet, we have rearranged the different domains in decreasing order of **natural** interaction with the user, which is the main concern of this thesis.

1. Smart surveillance systems refer to those applications that are able to supplant human perception to a certain extent in monitoring applications. This task has traditionally been accomplished by low-level frame to frame comparison algorithms, incline to frequent false detections. Smart surveillance systems possess an additional semantic layer able to answer in most cases not only what part of the image is changing, but above all if a moving part of the image is indeed a human being. If so, what action he/she is performing, and finally if that action is a "normal" one (with respect to the context) or a "suspicious" one. In these kinds of applications the semantic part needs a human model in order to detect the users, and thus, some sort of human motion detection/capture/analysis. These are, obviously, the applications in which the naturalness of the interaction between (involuntary) user and system is the most important.

2. In virtual reality applications, interaction with the system recreating a perceptually credible and coherent reality will have to be as natural as possible, in order to increase the sense of immersion (Fig. 1.3). In advanced user interfaces, this naturalness will not have such a direct influence on end-user satisfaction, especially because of the long-term training that people already have with traditional interfaces (hybrid natural-haptic interaction could still provide good re-

sults). Still, the more intuitive and natural the interaction, the more creative and easy the use of machines (especially computers) will be. Finally in more specialized applications such as clinical or blockbuster character-animation, accuracy will be the top priority, not natural interaction.

3. In advanced user interfaces, natural interaction is still a key factor with respect to user satisfaction, still not as fundamental as it is in the virtual reality applications. In social interfaces or gesture driven control, the sense of immersion is not compulsory (and is actually the main difference between virtual reality and augmented reality, Fig. 1.2). Nevertheless, natural interaction is still a priority since the aim is to provide novel forms of interaction that decrease the technical dictations to which the user has to adapt. Social interfaces refer to those interfaces that display virtual human-like interlocutors with whom the user can interact in a more personable manner [Turk 1996].

4. Human motion analysis can be used in the fields of choreography and orthopaedy for example, where natural interaction is less a priority generally due to the needed precision. In other cases, like personalized systems for sports training (like golf or tennis), natural interaction systems could bring a substantial plus.

More precisely, we will focus on the second and third areas, i.e. applications dealing with Virtual Reality and advanced user interfaces, which are actually the most (natural) interaction-oriented areas (Fig. 1.3).

1.4 Related Work

Let us now introduce the state-of-the-art context of our research, by presenting approaches that are similar to ours in their purpose. A recent survey by Gavrila [Gavrila 1999] classifies the research in this field in 3 categories: 2D approaches without explicit shape models, 2D approaches with explicit shape models and 3D approaches with articulated models.

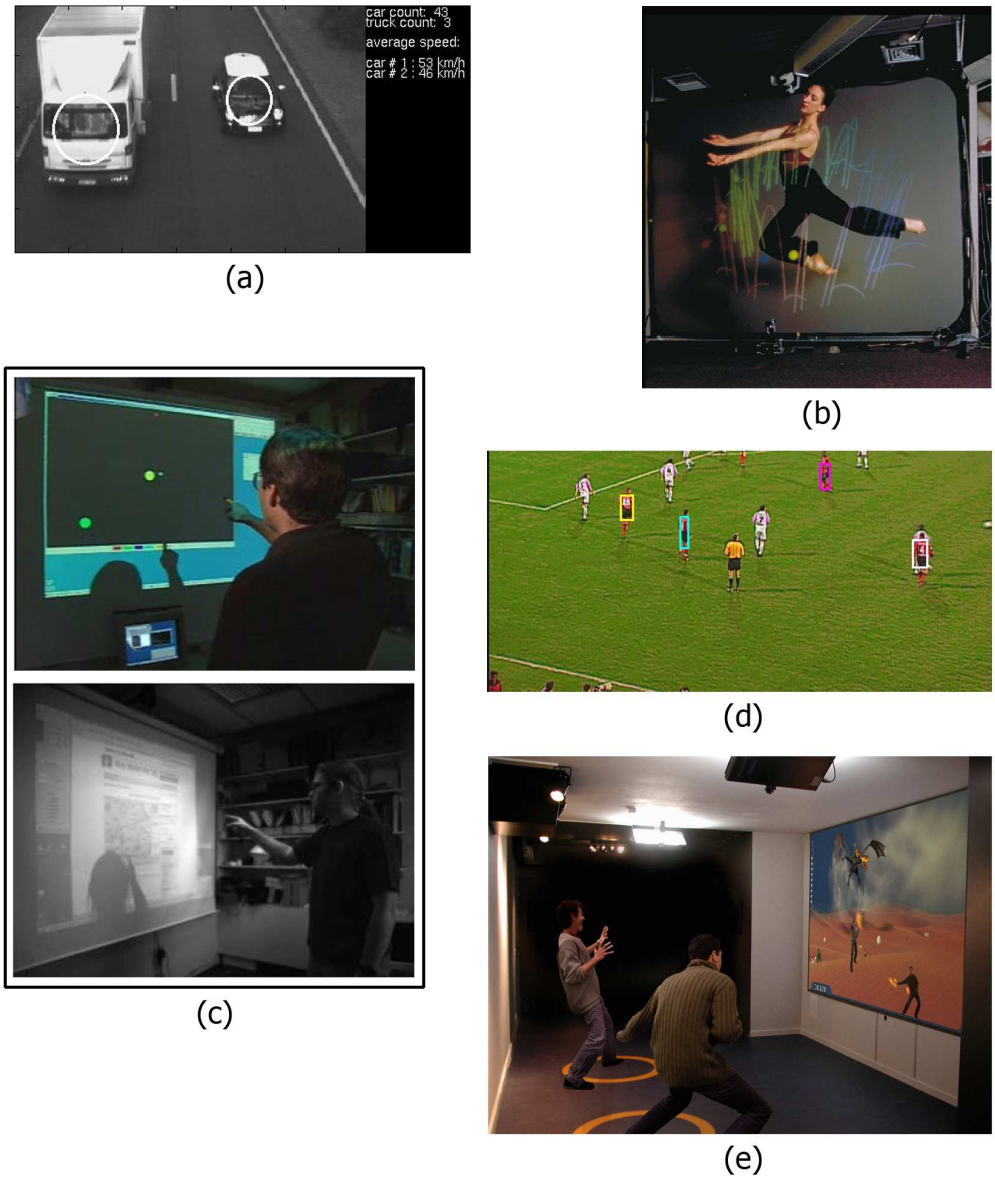


Figure 1.3: *Illustrations of Table 1.1 applications. (a) A smart surveillance application: Counting and monitoring of cars in a highway. (b) Motion analysis example: choreography application. Performer Jennifer DePalo in DanceSpace application [Sukel et al. 1998]. (c) Gesture driven control example: Users interacting with different applications using the integrated system WebGalaxy/pointing detector [Demirdjian and Darrell 2002]. (d) Content-based indexing of sports example [Czyz et al. 2005]. (e): A natural interaction application: "Duel of fire". Courtesy of Alterface.*

General Domain	Specific Area
(1) "Smart" Surveillance Systems	-Access control -Parking lots -Traffic
(2) Virtual Reality	-Interactive virtual worlds -Games -Virtual studios -Character animation -Teleconferencing
(3) Advanced User Interfaces	-Social interfaces -Sign-language translation -Gesture driven control
(4) Motion Analysis	-Content-based indexing of sports video footage -Personalized training in sports -Choreography (dance, ballet) -Clinical studies of orthopedic patients

Table 1.1: *Natural Gesture Applications*

Another survey on *looking at people* algorithms, proposed in [Moeslund and Granum 2001], focuses on general aspects such as the overall structure of the motion capture system and the various types of information being processed. It focuses on 4 particular stages: initialization of the system/model/user, tracking of the motion, pose estimation and action recognition.

2D systems have the advantage of not requiring special equipment (such as stereo cameras [Demirdjian and Darrell 2002]) or heavy 3D articulated human models [Fua et al. 2002]. Therefore, they can perform faster and less expensively than 3D systems while proving to be very useful and with a sufficient precision in a vast scope of different upraising areas, like virtual reality (interactive virtual worlds, teleconferencing) [Douxchamps et al. 2003], advanced user interfaces (gesture driven control) [Sparacino et al. 2000] and motion analysis (gait-based biometrics, avatar animation) [Haritaoglu et al. 2000, R. Urtasun et al. 2005]. Given the real-time nature of our application, we concentrate on the analysis of 2D approaches.

Following Gavrila's criteria, 2D human body tracking systems with articulated human models use shape models in order to extract a hierarchical structure out of the image. A good example of this approach is the work accomplished by Ju et al. [Ju et al. 1996]. Their work describes the creation of an articulated 2D human model as a series of deformable patches, and focus on leg tracking, in order to extract gait information. This work tackles the tracking, pose estimation and recognition steps, and does not require the user to perform a specific initialization pose. This method, very popular as a theoretical basis for articulated human models [Morris and Rehg 1998, Pavlovic et al. 1997, Wu et al. 2003], is seldom followed by itself in 2D, since bidimensional feature extraction in that context can be achieved faster with sufficient precision without such a complex articulated model.

As for 2D approaches without an explicit human model, Fushiyoshi et al.'s work in pedestrian detection [Fujiyoshi and Lipton 1998] provides a real-time silhouette extraction, and a subsequent star skeleton of pedestrians, without a prior initialization of the user, and without an explicit shape model. This silhouette's skeletonisation helps extracting cyclic branch (i.e. extremities) movement and body posture. These two cues afterwards help in distinguishing some basic pedestrian activities such as walking or running.

In the same class of 2D approaches without explicit human model, W4 [Haritaoglu et al. 2000] uses inter-image motion estimation in order to (i) distinguish a generic human silhouette without prior user initialization, (ii) segment it and (iii) track it. The system keeps aspect templates of different body parts in order to track the silhouette in split and merge situa-

tions (i.e. when two silhouettes meet, establish a contact and then separate again). It is also able to recognize some simple actions, as to distinguish between a walking from a standing pedestrian.

Baumberg and Hogg's [Baumberg and Hogg 1995] describe in their work a flexible 2D human boundary tracking that can be created automatically, over many training images. This approach allows creating a generic pedestrian model to which any other walking human data can be compared and rated.

Finally, the Pfinder algorithm [Wren et al. 1997], one of the most popular in the field, uses a statistical representation of different body parts, modelled as blobs. These regions are created by assigning to each pixel feature vectors that are subsequently merged by combining color and spatial information, without using an *a priori* model.

Even if these above-mentioned works have remarkable results in their goals, which is to track and/or monitor whole body motion, they do not contemplate interaction with separate body parts -and thus feature extraction- as a priority. W4 or Baumberg's work have impressive human surveillance capabilities but a fairly coarse body part detection and tracking, and Fujiyoshi's work only needs to approximately detect head and feet in order to achieve its pedestrian detection goal.

Pfinder puts more emphasis on feature extraction and pose estimation. Yet, this work focuses mainly on head and hand extraction, needs a prior user initialization and uses strong flesh-colored color priors.

Table 1.2 summarizes the overview up to this point. Even though it presents the presented works following Gavrila's taxonomy, both the table and the text focus on works similar (in purpose or content or both) to ours. For a more general survey on the existing works related to the "Looking at people" domain, the interested reader is referred to Table 2 of the following work [Gavrila 1999], and Appendix A of this one [Moeslund and Granum 2001].

Let us now introduce other works that fall in a more general state-of-the-art scope.

	Using Explicit Human Model	**Without Explicit Human Model**
2D	[Ju et al. 1996]	[Fujiyoshi and Lipton 1998] [Haritaoglu et al. 2000] [Baumberg and Hogg 1995] [Wren et al. 1997]
3D	[Urtasun and Fua 2004] [Fua et al. 2002]	[Demirdjian and Darrell 2002]

Table 1.2: *Related Work Recapitulation*

Using silhouettes for determining body posture has been used by many researchers, see for example the following works: [Deutscher et al. 2000],[Sminchisescu and Telea 2002],[Sminchisescu et al. 2005],[Agarwal and Triggs 2004],[Brand 1999],[Rosales and Sclaroff 2000],[Delamarre and Faugeras 1999]. Davis and Bobick [Davis and Bobick 2001], as well as Yilmaz and Shah [Yilmaz and Shah 2005] perform directly action recognition from a moving human silhouette, bypassing the pose estimation step. Delamarre and Faugeras [Delamarre and Faugeras 1999], Sminchisescu and Telea [Sminchisescu and Telea 2002], Deutscher *et al.* [Deutscher et al. 2000] suppose a parametric 3D model of the body is known. The full body pose is recovered by optimizing a likelihood function defined over silhouette measurements. Note that Sminchisescu *et al.* [Sminchisescu and Telea 2002] also use silhouette skeletons, which are closely related to the geodesic distance used in this paper, however in their work the skeleton serves to smooth out the silhouette in order to stabilize the optimization process.

Given the ambiguities in the projection human pose on the image plane, model-based approaches may converge to local minima of the likelihood function. Alternative approaches to pose estimation/motion capture rely on the direct learning of the mapping between the measured image silhouette and the 3D pose, avoiding the need of accurate 3D modelling. The ad-

vantage over model-based approaches is that the mapping concentrates on the most usual human poses (through the training set). In the work by Rosales *et al.* [Rosales and Sclaroff 2000], this mapping is learned by unsupervised clustering. Brand [Brand 1999] uses an HMM to model the manifold of human body configurations. Motion capture amounts to find the most probable path in the manifold given a sequence of human silhouettes. Agarwal and Triggs [Agarwal and Triggs 2004] propose to use a Relevance Vector Regression, a learning machine approach with good generalization capabilities, to learn the mapping and recover the full 3D pose. Instead of learning the mapping from image measurement to 3D pose, Mori and Malik [Mori and Malik 2002] look for the closest match between the 2D edge image and a set of hand-labelled 2D exemplars containing a variety of different configuration. The 3D pose can then be reconstructed from the position of the body joints.

The drawback with the learning approach is that the algorithm might fail to recover the pose in case of postures and/or motion that were not represented in the training set. Also the flexibility is limited as the algorithm must be re-trained when the type of configuration changes, for example from a biped stand-up configuration to a wheelchair configuration. Moreover there are a certain number of applications that do not require the full body posture but rather a real time, accurate localization of some key features, without the specific need of a semantic layer (i.e. interactive edutainment, incrustation of virtual objects in mixed/augmented reality, virtual interaction, avatar animation, gestural navigation, video gaming). These are the applications we are interested in.

In that sense the work on human detection on an assembly of body parts [Forsyth and Fleck 1997],[Ioffe and Forsyth 2001],[Felzenszwalb and Huttenlocher 2000],[Mikolajczyk et al. 2004],[Micilotta and Bowden 2006],[Micilotta and Bowden 2004],[Ren et al. 2005] is relevant to ours. In the work by Ioffe *et al.* [Ioffe and Forsyth 2001], body parts are detected by a simple segment detector and then assembled using constraints on the appearance of a person. Pictorial structure matching is used by Felzenszwalb *et al.* [Felzenszwalb and Huttenlocher 2000] to determine the best matching configuration of a set of rectangular color-based parts detected in the image. The set of model color parts must be given in advance. In this work a strategy to use pairwise constraints between human body parts to recover body configurations from static images is developed. Ren *et al.* [Ren et al.

2005] detect candidate body parts in static images from bottom-up using parallelism cues. Finding the configuration of a human body becomes an assignment problem which is solved using Integer Quadratic Programming. Mikolajczyk *et al.* [Mikolajczyk et al. 2004] use robust human part detectors (face, head, torso and legs) and combine parts with a joint probabilistic body model. The procedure allows to detect humans in static gray-level cluttered images. However the localization of body parts is fairly coarse and the procedure is computationally expensive which makes this approach inappropriate for the gestural interaction applications. Micilotta *et al.* [Micilotta and Bowden 2006], [Micilotta and Bowden 2006] concentrate on the upper body 3D pose estimation by detecting face, torso and hands. Detected parts are then assembled using RANSAC and an *a priori* body configuration likelihood function. 3D pose recovery is performed by matching the silhouette and the edge map of the subject with a collection of examples obtained from a 3D synthetic model. Again, the face and hand detector relies on skin color models, therefore it would not avoid for instance the feet detection or infrared lighting issues described above.

Recently, the ever increasing computational speeds have brought back particle filtering techniques [Doucet et al. 2001] under the spotlight. These techniques, that were set aside in the past for being too greedy are now becoming very popular in 2D [Micilotta and Bowden 2004] but also in 3D [Cohen and Lee 2002, Deutscher et al. 2000], these latter still being unusable for real-time. Micilotta's work [Micilotta and Bowden 2004], close to Pfinder's, also uses skin color recognition, but with less strong priors, as the skin models are built online. Still, the model context seems to be quite rigid - for biped stand-up only - and would not avoid for instance the feet detection or infrared lighting issues that we present hereafter.

The system presented in this work tracks human key features that overall define a specific human gesture using pure silhouette analysis. It falls, following Gavrila's taxonomy [Gavrila 1999], into the *2D without explicit shape models* category, even if we will also discuss the 3D issue and show some promising results in Chapter 4.

Hereafter we present an overview of the algorithm.

1.5 Algorithm Overview

In this work we focus on the extraction of the five human extremities: head, hands and feet, in different configurations and applications. In any case, we will henceforth refer to the detected key features that enable a gestural interaction as *crucial points*, or *CP*. The choice of this particular set of human features will be discussed in Section 2.1.

Following the taxonomy presented in [Moeslund and Granum 2001], our system performs human feature extraction, tracking and pose estimation, without needing a user initialization pose, by applying the steps briefly presented hereafter. How these different stages are organized throughout this document is detailed in Section 1.6.

Firstly the human silhouette is segmented from the background. Then, key features are detected and extracted in intra-image mode using mathematical morphology tools. A geodesic distance map is computed on the body silhouette with respect to its center of gravity (or *CoG*). Analysis of this map on the silhouette contour allows the extraction of all the local geodesic distance maxima, by creating a distance-location function. After this point two different approaches have been followed (Fig. 1.4). A first one dealing exclusively with intra-image information labels the extracted features (Chapter 2). In this intra-image approach, that uses standard mathematical morphology tools, performance is favored vs. real-time paces computation. It is thus aimed at being more detailed and precise. Research in that direction is still in progress at the Image Processing Department of the UPC (Barcelona), under the direction of Prof. Ferran Marqués, in an extensive collaboration with the work presented in this thesis.

Another approach, that uses additional inter-image temporal information, sets the correspondences between detected features in consecutive frames and attributes labels to features, following a Bayesian approach (Chapter 3). In order to disambiguate the detected features into head, left and right foot, left and right hand classes, we follow a Bayesian method that combines the information of the local maxima intensities, a prior statistical distribution of the selected crucial features and the dynamics of the tracked crucial points.

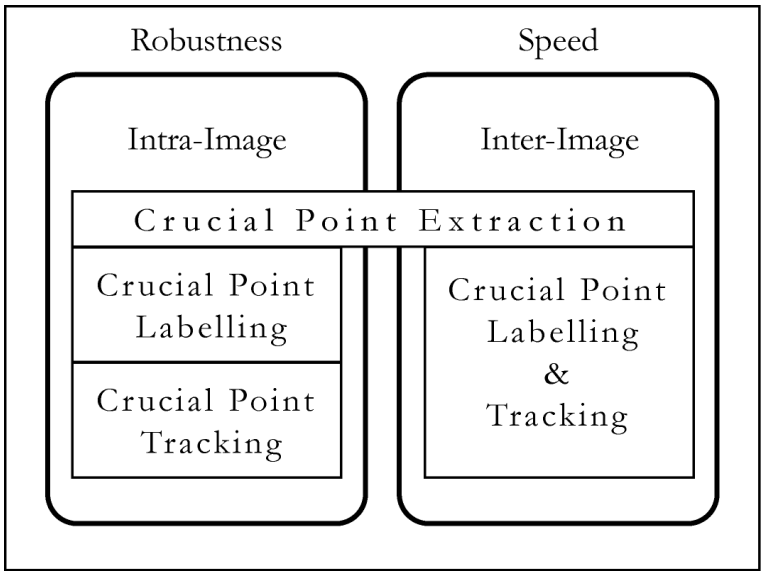


Figure 1.4: *Thesis Global Organization*

1.5.1 About Silhouette Segmentation

Robust silhouette extraction is thus an important step of the pipeline, yet not being strictly part of our developed algorithm. In this work we employ the setup and real-time segmentation described in [Marichal et al. 2003]. Nevertheless, hereafter we present other silhouette extraction systems that are worth the mention.

In practice, there are two classes of silhouette extraction algorithms: those that deal with outdoor, pedestrian-related problems, and those that deal with indoor, gestural interaction ones. The former class typically leads to global behavioral analysis and other performance often needed in smart surveillance applications, i.e: individual segmentation out of a group of people, simple pedestrian behavior detection (walking, running, sitting, jumping) and splitting/merging detection, in order for instance to spot a person leaving a bag behind. They need to be very robust to lighting condition changes while in the other hand only need an approximate silhouette delimitation. Some of the works successfully tackling this problem have already been presented above, such as [Haritaoglu et al. 2000]

using an adaptative background model, or [Fujiyoshi and Lipton 1998] using a similar technique. Also recently, Zhao *et al.* [Zhao and Nevatia 2003], using a model-based segmentation problem and a Markov chain Monte Carlo (MCMC) method in a Bayesian framework (a 3D human shape model projected into 14 different 2D pedestrian configurations) produced very promising results, yet not reaching real-time paces.

Indoor-oriented segmentation algorithms traditionally produce silhouettes precise enough in order to differentiate limbs in real-time. On the other hand, they respond less efficiently to illumination or background changes (even if gradual changes are often permitted), thus being more application and/or context oriented. Besides the system used in our work, another example of indoor segmentation can be found in the already presented [Wren et al. 1997]. Another analogous approach is presented in [Davis and Bobick 1998], yet using an infra-red environment and a black and white camera with an infrared filter.

1.6 Thesis Organization

This thesis presents my personal and collaborative results, following almost the chronological order of this research.

After this introduction, Chapter 2 presents the intra-image section, that focuses on the extraction of the human features that allow a general movement reconstruction. It also introduces the intra-image labelling. Thus, this chapter sets up an intra-image crucial point detection and classification algorithm as a whole. This chapter also presents the image processing tools that will be used in the process. One of these tools, namely the geodesic skeletons (section 2.4.1), will represent the main difference between between this chapter and Chapter 3. It is indeed a powerful tool that allows a detailed analysis of the user's posture information via a condensed structure, being however quite a time-consuming tool.

Chapter 3 presents another motion capture algorithm, faster, more flexible and sharing with the intra-image approach the whole crucial point extraction phase (see Figure 1.4). Using less precise yet faster and more flexible tools, the labelling and tracking algorithm presented in this Chapter could

be seen as a substantial improvement with respect to Chapter 2, within the real-time applications context.

Chapter 4 presents the results of two experiments concerning possible extensions and perspectives of the previous results. Section 4.1 extends the previous approach to two orthogonal cameras and some promising results are presented. Section 4.2 introduces the utility of inverse kinematics in order to reconstruct whole body pose taking only our Crucial Points as input.

Chapter 5 closes this thesis with overall conclusions, its contributions and publications along with perspectives for future research.

Intra-Image Feature Extraction

2

In this Chapter we present a non invasive crucial human feature extraction algorithm, aimed at capturing human movements for gestural interactive applications. The needed mathematical morphology tools are presented and applied to the different steps of the algorithm: pre-processing, human feature extraction, feature labelling. These steps have in common the fact that they work on each image individually, without taking into account any temporal information. Another approach will be proposed in Chapter 3, that focuses on motion information. The chapter will wrap up presenting the results accomplished with this method.

2.1 The Crucial Point Set

RETRACING in an almost chronological manner our research progression, let us first define the set of human features that are the most relevant ones concerning gesture recovery. Traditionally, specialized state-of-the-art optical motion capture devices for computer animation and kinesiology rehabilitation use a bare minimum of 18 nodes: one for each articulation (knees, elbows, hips, shoulders, ankles, wrists, neck) and 5 extremities. This number is frequently increased, and can easily reach two to three times that amount of nodes for blockbuster movie special effects character animation.

Since we focus on real-time interactive applications, the way of capturing human movements has to be as light as possible. Hence, instead of extracting the whole body pose, we have favored a motion-based ap-

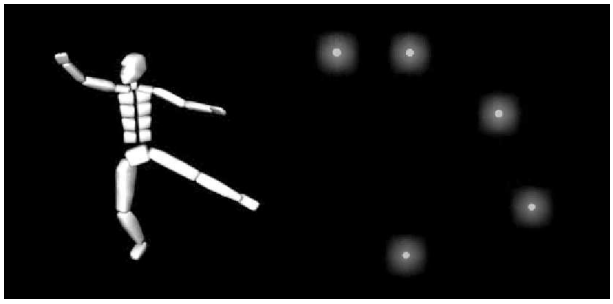


Figure 2.1: *Johansson's pose recovery example.*

proach, by extracting and tracking some key human features that would still make possible a natural gesture interaction. Furthermost, this number of key features has to be the bare minimum that could nevertheless allow a complete body posture recognition. Thereafter, position and movement of these features could be used for pose and gesture recognition purposes.

The choice of extracting only the five extremities was triggered by Johansson's late 70's human perception studies [Johansson 1973]. His seminal work, in 1950 ([Johannson 1950]), described how applying phased movement to several dots among others could not only help to distinguish them and group them (in extension of Gestalt's laws [Ellis 1938] and Gestalt's *law of fate* in particular) but furthermore, that moving dots could also infer to the viewer structural relations between them (much like if they were linked by a rigid rod). His later psychovisual studies indicated that humans can easily recover the pose of familiar biological forms in motion from extremely sparse data (moving dots attached to specific body features).

Among these different moving human features, the bare minimum that still convey a global gesture information are the five extremities. They are indeed those that infer the simplest body structure when moving: since they seem to be linked by rigid segments, they are in fact recreating some kind of childish human stick figure in the viewer's mind. See Fig. 2.1 for a single frame illustrating this concept. The reader is referred to:
`http://www.tele.ucl.ac.be/~pedro/gestural/`
for the complete video sequence.

As our goal is to reconstruct the simplest and nonetheless useful human body model with the less possible information, we use all of the above to define our set of crucial human features to extract. These will thus be the five extremities: the head, the hands and the feet.

Section 2.3.3 will be devoted to the translation of these crucial features set into a mathematically extractable form.

2.2 Image Pre-Processing

In order to extract the 2D posture of the human subject we use one single non calibrated camera.

The user silhouette region is extracted using a real-time segmentation technique [Marichal et al. 2003], which has very similar operating specifications as Pfinder's [Wren et al. 1997] or other analogous advanced silhouette-based applications. As typically assumed for this kind of applications, we expect the scene to be significantly less dynamic than the user and having overall changes that are small or gradual. The segmentation is achieved using a Walsh-Hadamard transform on blocks of 4×4 pixels (Fig. 2.2). First, the module calculates the Walsh-Hadamard transform of the background image. Afterwards, the module compares the values of the Walsh-Hadamard transform of both the current and the background images. When the change rate is higher than a threshold that has been initially set, this module sets the area as foreground (Fig. 2.3). For a complete description of the segmentation approach, the reader is referred to [Marichal et al. 2003, Wren et al. 1997].

2.3 Crucial Point Extraction

To locate the Crucial Point (or *CP* in the sequel) on the silhouette there are several steps to follow, each of them using a well-known image processing tool. They are each detailed in this chapter. Here is a quick overview of the whole process:

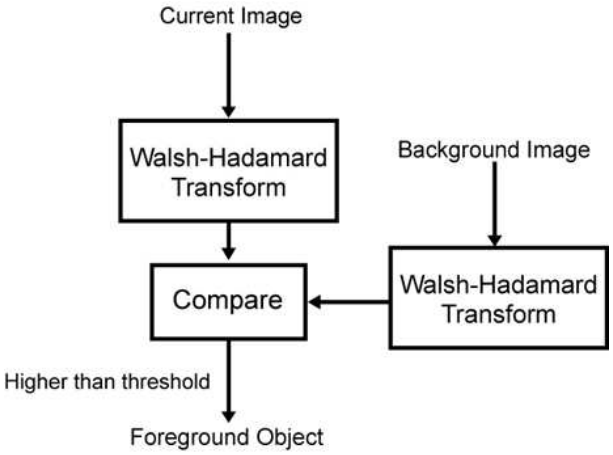


Figure 2.2: *Walsh-Hadamard segmentation scheme.*

Figure 2.3: *Segmentation example: Original image and extracted silhouette*

1. Center of Gravity Computation (Image moments).
2. Geodesic Map Computation, with respect to the CoG.
3. Geodesic Map border tracking.
4. Creation of the Geodesic Distance function ($f(p)$).
5. Simplification of $f(p)$ and creation of the $g(p)$ function via an opening by reconstruction on $f(p)$.
6. Location of the $f(p)$ local maxima on the silhouette border.

2.3.1 Geodesic Distance Map Computation

We have previously defined (section 2.1) the set of crucial points to extract, considering the targeted applications. These are the five extremities. As we want to extract them automatically (and robustly), we need to find a mathematical expression for these features. To do so, we use the fact that these points are the five most prominent human features in the user silhouette area (five at most, taking self-occlusions into account). And these *prominence* or *extremity* notions can be translated in terms of geodesic distances from the most central point of the body to the silhouette boundary: they are the points of the silhouette border that represent local geodesic distance maxima with respect to the CoG. In other words, the five extremities are the points of the silhouette located farther away (in the geodesic sense, see below) from the Center of Gravity (CoG), while still being on the border of the user silhouette area.

Geodesic Distances and Geodesic Maps

The geodesic distance between two points x and y in a connected set X is the infimum of the paths length between x and y in X, if such path exists [Lantuejoul and Beucher 1981, Lantuejoul and Maisonneuve 1984]:

$$d_X(x,y) = inf\{l(C_{(x,y)}) \mid C_{(x,y)}\ is\ a\ path\ between\ x\ and\ y\ inside\ X\} \quad (2.1)$$

If there are no such paths, we set $d_X(x,y) = \infty$.

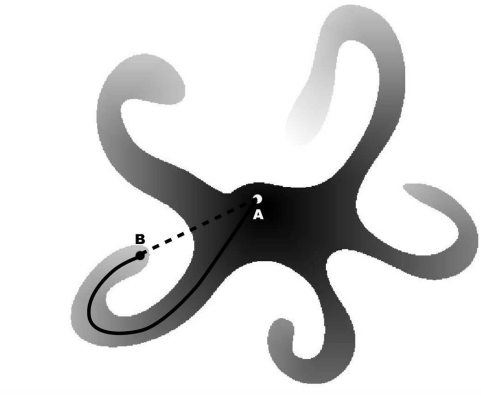


Figure 2.4: *Geodesic distance map example. The geodesic distances with respect to point A of all region pixels are displayed in increasing grayscale value. Euclidean (dashed) and geodesic (solid line) distances from point A to point B are compared*

Another approach, closer to the point of view of its application in our algorithm is the following. Suppose $P = \{p_1, p_2, ..., p_n\}$ is a path in a connected domain between pixels p_1 and p_n, i.e. p_i and p_{i+1} are connected neighbors for $i \in \{1, 2, ..., n-1\}$ and p_i belongs to the domain for all i. The path length $l(P)$ is defined as the sum of the neighbor distances d_N between adjacent points in the path.

$$l(P) = \sum_{i=1}^{n-1} d_N(p_i, p_{i+1}) \tag{2.2}$$

In other words, the geodesic distance between two points is the sum of neighbor distances between the two points taking into account only the possible paths inside the connected domain. In our case the domain is formed by the user silhouette area. Figure 2.4 presents an example of geodesic distance computed from a point A to a point B inside a given region.

Moreover, we call the *geodesic ball* of radius n and of center $c \in X$ the set $B_X(c, n)$ defined by:

$$B_X(c, n) = \{c' \mid c' \in X,\ d_X(c, c') \leq n\} \tag{2.3}$$

Suppose now that X is equipped with its associated geodesic distance d_X. Given $n \geq 0$, we consider the structuring function [Serra 1988] mapping each pixel $c \in X$ to the geodesic ball $B_X(c, n)$ of radius n centered at c. This leads to the definition of the geodesic dilation of size n $\delta_X^n(Y)$ of a subset Y inside of X, as

$$\delta_X^n(Y) = \bigcup_{c \in Y} B_X(c, n) = \{c' \mid c' \in X \; and \; \forall \, c' \in X, \exists c \in Y, d_X(c, c') \leq n \; \} \tag{2.4}$$

Figure 2.5 (c) and (d) illustrate a geodesic dilation. This geodesic dilation will be used, below, in the computation of the geodesic distance map.

For a more in depth presentation, analysis and applications of geodesic distances, interested readers are referred to [Cuisenaire 1999, Serra 1988; 1982].

Geodesic Maps Computation

We have seen before (Eq. 2.1) how the geodesic distance was defined. Moreover, when in a certain connected domain X the distances of all points are computed with respect to a single point of the same domain, a *geodesic distance map* is created. In that case, we refer to the selected point with respect to which all distances are computed as *reference point* (point A in Fig. 2.4).

We construct the geodesic distance map by successive steps so as the geodesic distance from the center of gravity C to any point x in the silhouette A is defined as

$$d_A(C, x) = n \Leftrightarrow x \in \; \delta_A^n(C) \; \text{ and } \; x \notin \; \delta_A^{n-1}(C) \tag{2.5}$$

where $\delta_A^n(C)$ is the geodesic dilation as defined in Eq. (2.4), centered in the CoG C inside of silhouette A, with a Structuring Element SE of size n. See Figure 2.5 for an illustration of this equation.

In terms of computational complexity, geodesic dilation behaves as a standard morphological dilation. The most commonly used algorithms in order to achieve a single morphological dilation δ are the parallel ones (Vincent, 1990, 1991b), having a $\mathcal{O}(N^2)$ complexity in a $N \times N$ pixel image.

With the required number of operations, this geodesic map computation can thus lead to $\mathcal{O}(N^3)$ complexities. Yet, much faster approaches have also been implemented, using for instance contour-processing algorithms [van Vliet and Verwer 1988], which complexity is reduced to $\mathcal{O}(l_{max} \cdot d)$ where l_{max} is the maximal size of the image contour, with a SE of radius d. Vincent [Vincent 1991b] proposes an algorithm using both the contours of A and the contours of the SE. This algorithm has a complexity proportional to the product of the number of pixels in each contour, which means $\mathcal{O}(l \cdot d)$ for a circular element of radius d. Other algorithms such as those based on chains and loops proposed in [Schmitt 1989] provide an efficient propagation function algorithm particularly recommenced for binary geodesic operations such as this geodesic distance map creation [Maisonneuve and Schmitt 1989], since they only need to scan the image twice. The interested reader is referred to these works as well as to [Cuisenaire 1999] for more details on the optimization of mathematical morphology operations.

Center of Gravity

In our case, the geodesic distance maps are always going to be computed taking the Center of Gravity (CoG) as the reference point. The CoG coordinates $(\bar{x}, \bar{y})$ of the silhouette are estimated by computing an average of the region pixels [Soille 2003]:

$$\bar{x} = \frac{\sum_{(x,y)\in S} x}{\sum_{(x,y)\in S} I}, \qquad \bar{y} = \frac{\sum_{(x,y)\in S} y}{\sum_{(x,y)\in S} I} \tag{2.6}$$

Where we have denoted I the whole binary image, and $S = \{(x, y) \mid I(x, y) = 1\}$ the pixels forming the user silhouette area.

It is possible that the center of gravity sometimes falls outside the user silhouette area. For our purposes, the geodesic distance map reference point needs to be located inside the domain boundaries in order to be computed properly (i.e Fig. 2.5). Hence, in those cases the CoG is pulled inside the silhouette area, to the position of the closest (in terms of Euclidean distance) silhouette inner pixel.

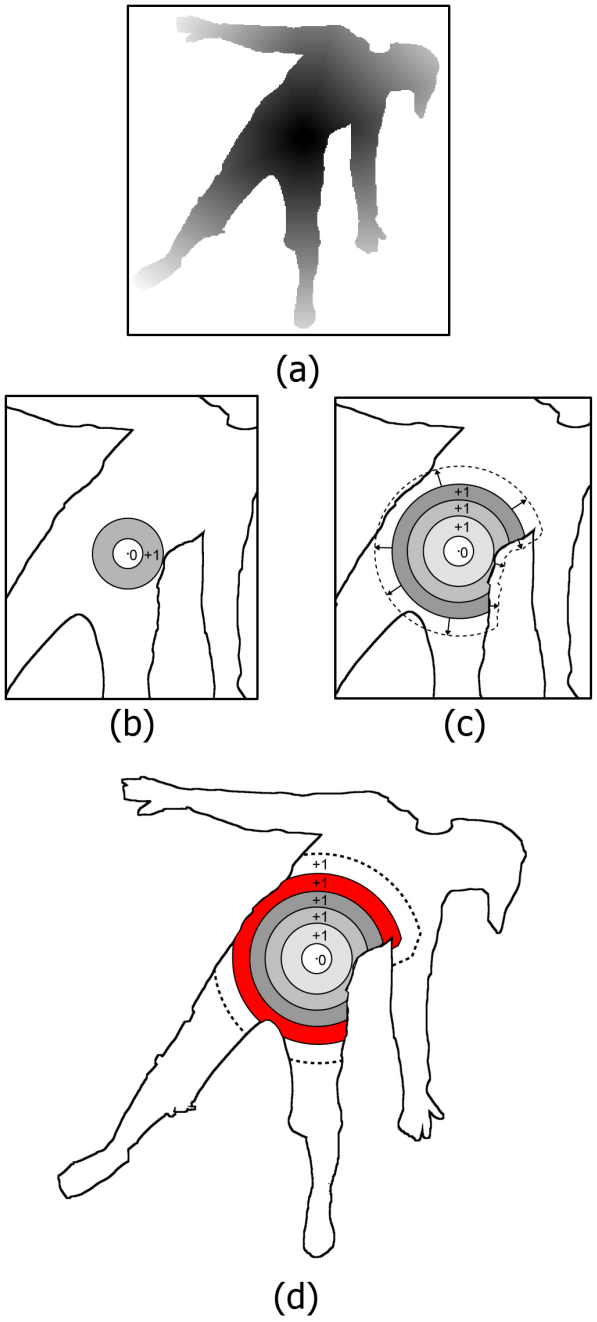


Figure 2.5: *Equation (2.5) illustration. (a): Final Geodesic Map Computation. (b) to (d): Successive iterations of the algorithm . (b): First two geodesic dilations with the silhouette. Gray area represents distance 1 with respect to the CoG. (c): A geodesic dilation step (dashed line) in detail. (d): Previous dilation result in dark gray. Dashed line represents next dilation.*

2.3.2 Geodesic Distance Map Computation Optimization

As we have presented above, the standard geodesic distance map calculation using Equations (2.1) and (2.5) is prohibitive for real-time applications, as it requires to operate a large number of consecutive dilations on the silhouette. Furthermore, in our application only the geodesic distance on the silhouette contour is needed (and not on every point of the segmented region). Due to the previous reasons, a fast geodesic block-distance which can be achieved in real time on standard PC hardware was computed [Umeda et al. 2004]. It is a particularly well suited alternative approach to the possible optimizations cited in Section 2.3.1.

The first step is to decompose the silhouette into connected horizontal slices of one pixel height. For storage efficiency purposes, each slice is then defined by its first and last pixels. A list of slices is then created, based on connectivity: starting from the slice containing the center of gravity, we add to the list the slice that is connected to the current slice. The geodesic distance computation on the contour is done recursively by going through each slice in the list, as detailed below (Fig. 2.6):

1. In the first element of the list l_0 (i.e. the slice that contains the gravity center of the object) search for the center of gravity and assign to this pixel a geodesic distance d_0 equal to zero.
2. Set the pointer to the previously treated slice p to l_0:$p \leftarrow l_0$
3. Set the pointer to the current slice c to l_1:$c \leftarrow l_1$
4. Repeat the following steps until the end of the list
   - Among the pixels in the current slice c, find the nearest pixel P, in terms of Manhattan distance, to the one of shortest geodesic distance d_{i-1} in the previous slice p. Mark the distance between these two pixels as D. This pixel P is the one of shortest geodesic distance for the current slice. Its approximated geodesic distance is $d_i = d_{i-1} + D$.
   - Compute the geodesic distance value for the first and last pixels of slice c as the sum of the block distance to the point with shortest distance P and d_i.

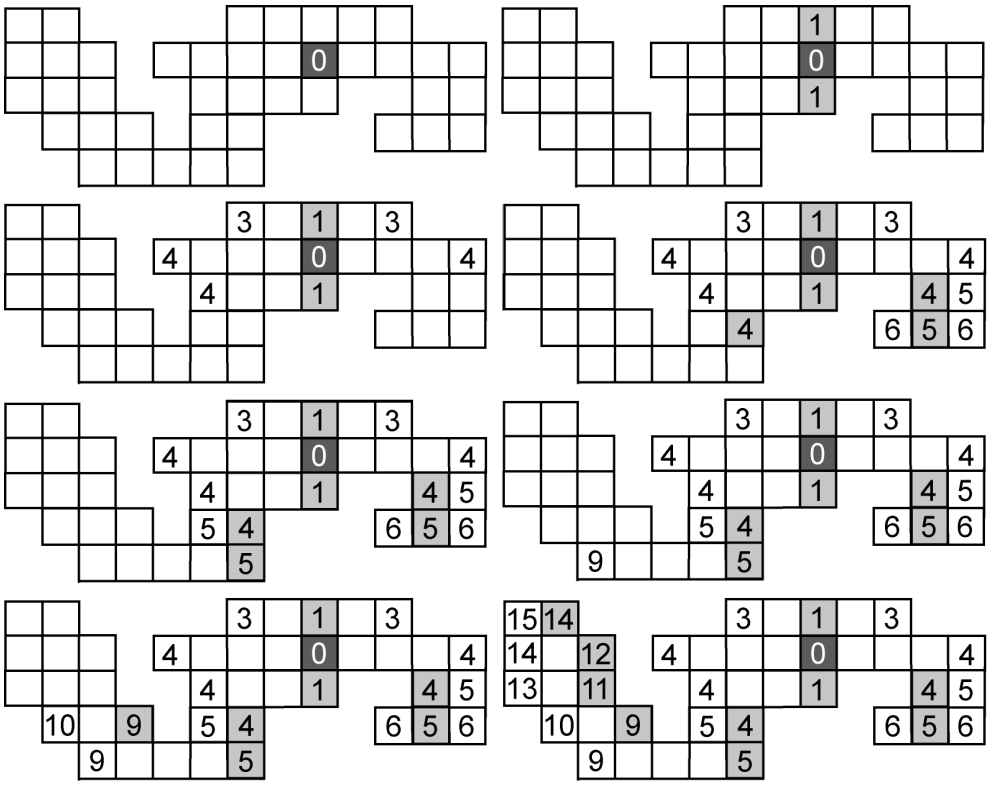


Figure 2.6: *Illustration of the geodesic distance map computation optimization. Distance 0 corresponds to the center of gravity. Squares in light gray correspond to the nearest pixels to gravity center (in the first iteration), or to the previous light gray pixel (in the general case).*

- Set $p \leftarrow c$ and $c \leftarrow l_{i+1}$

The algorithm does not only scan the silhouette upwards and downwards from the starting point (the CoG) but recursively executes itself at the end of each slice exploration. It thus achieves a breadth-first search, and could actually be compared to the Dijkstra's shortest path computation algorithm [Dijkstra 1959], with the restrictions of running in a 4-connected domain and where all edges have the same cost. Since each slice is scanned once in order to find its extremities the computational complexity is linear.

2.3.3 Analysis of the Geodesic Distance Function

The geodesic distance map, either created using Equations (2.1) and (2.5) by applying successive dilations and intersections until idempotence, or computed real-time with the optimization introduced in Section 2.3.2, yield the geodesic distances of all pixels in the silhouette border with respect to the CoG. The next step is thus the analysis of these geodesic distances variations along the silhouette border, in order to exam its dynamics. To do so, we perform a **contour-tracking** of the silhouette border (Fig 2.7 (a), in red). We perform an 8-connected contour-tracking, since it is slightly faster and produces contours with less pixels (thinner lines in the contour angles) than a 4-connected one. Each pixel on the final contour has indeed only 2 neighbors.

We then compute a one-dimensional function $f(p)$ linking each pixel position p in the silhouette border with its geodesic distance with respect to the CoG (Fig 2.7 (b)).

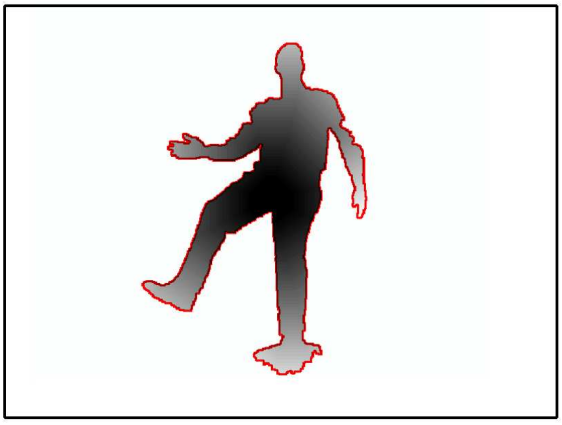
This function ($f(p)$), that we will henceforth refer to as *geodesic distance function*, yields the local maxima associated to the crucial points, as well as other local maxima due to segmentation noise. The aim of all segmentation algorithms being to minimize these noisy peaks, their local maxima will thus usually be minor than actual CPs.

Noise peaks present lower dynamics than peaks associated to crucial points. Therefore, they are removed by applying an H-maxima operator [Soille 2003]. This operator is obtained by applying an opening by reconstruction (γ^{rec}) to the function $f(p-H)$ that provides us with a function $g(p)$.

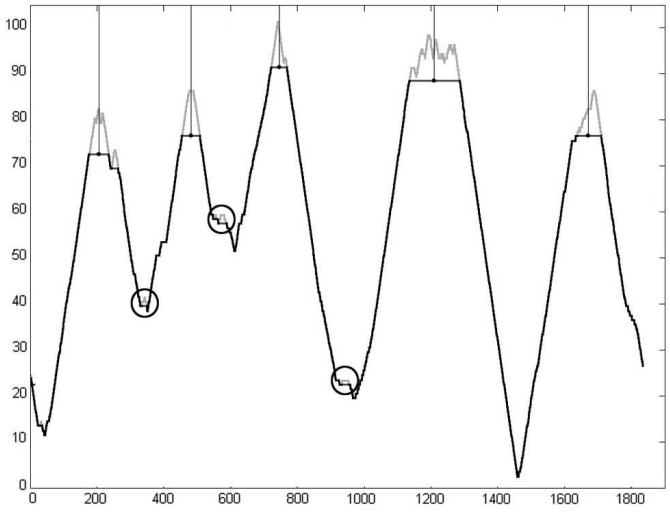
$$g = \gamma_f^{rec}(f-H) = \delta_f^{\infty}(f-H) \tag{2.7}$$

where $\delta_f^{\infty}(f-H)$ is the unidimensional geodesic dilation of the function $(f-H)$ within f, iterated until idempotence. The parameter H is a constant value related to the estimated noise dynamics.

Figure 2.7 (a) presents the geodesic distance map created using the silhouette extracted in Fig. 2.3 as input. The function representing the geodesic distance evaluated on the silhouette is shown before ($f(p)$) and after ($g(p)$)



(a)



(b)

Figure 2.7: *Geodesic distance map for a given silhouette (a). Lighter shades of gray correspond to longer geodesic distances. Contour-tracked silhouette border is displayed in black. (b) displays the geodesic distance function after contour-tracking the geodesic map. The function representing the geodesic distance evaluated on the silhouette is shown before ($f(p)$) and after ($g(p)$) the filtering, in gray and black respectively. The removed noisy local maxima are encircled and the selected local maxima (CP) are marked with vertical lines.*

the filtering in Fig. 2.7 (b).

Parameter H is the factor that will allow to trade off noise removal against local maxima detection sensitivity. It is estimated and fixed once and for all depending on the camera, the working environment and the application context. Since in this chapter the acceptable number of extracted CP ranges from 2 to 5, this parameter will be chosen as to avoid the case where the number of local maxima present in $g(p)$ is higher than the number of Crucial Points that are really present. Since it is always possible to find a value of H high enough to remove all noisy maxima, the trade-off consists in taking the case where H is the lowest possible, yet high enough to have all segmentation noise indeed removed.

Its choice is obviously also strongly dependent on the size of the structuring element (Fig. 2.8), when using the standard procedure described in Section 2.3.1. Concerning the structuring element (SE) shape, only approximations of the circular shape are used, since we do not seek to have any particular direction privileged. Hence, convex and isotropic forms like squares or octagons are recommended, versus less compact shapes like crosses. These latter can indeed produce less regular results, by privileging certain directions and producing more irregular geodesic distance maps, which translates into a higher number of local maxima in the distance function along the silhouette border. See Figure 2.12 for an illustrated example.

As for the choice of the structuring element size, it is again subject to a trade-off: the bigger its size, the fastest the geodesic map computation; the smallest its size, the more detailed the map, and thus the more detailed the distance values on the border of the silhouette.

Figure 2.8 and Table 2.3.3 illustrate the influence of the structuring element size, of parameter H, and their dependance. It displays the number of local maxima that appear for each distance function computation, according to different values of SE size and H parameter. The function displayed in Fig. 2.8 (a2) is computed with the smallest structuring element (square-shaped, with a 3 pixels side) and an opening by reconstruction of the function obtained with a value of H equal to 2. We can observe the important influence of the noise, before the H-maxima operator: 9 local maxima were detected. However, only 5 local maxima (actually **the** 5 local maxima we

are seeking to extract) remain after the H-maxima operator. Note that the function created using a SE of size 9 and opened by reconstruction with H equal 2 also allows the extraction of the same 5 extremities, yet being 3 times faster.

	Figure a2. SE=3, H=2	**Figure b2. SE=9, H=2**	**Figure b2'. SE=9, H=6**
Before H-thresholding	9	**8**	6
After H-thresholding	5	**5**	4

Table 2.1: *Results referring to Figure 2.8. Columns display results (number of extremities found) for different SE and H parameter values configuration, displayed in Fig. 2.8. The goal being to locate 5 different extremities after H-maxima operator, the best trade-off computation time/accuracy is displayed in bold.*

In order to produce results at fast paces, this mathematical morphology approach is not privileged since the overall algorithm will only use the geodesic distances computed along the border. It is still nevertheless a very convenient approach when developing and/or improving this kind of algorithms, since it produces very detailed and visual information. It is thus quite an appealing tool and the actual underlying technique in the optimized approach.

Each local maxima i still present after the filtering step is selected and associated with a point in the given frame t: $z_t^{(i)} = (x, y)$. In cases of self-occlusions, the number of detected crucial point candidates can narrow down to two. Hence, in this first approach the number of extracted crucial points ranges from 2 to 5. The value of parameter H is chosen as to extract a number of CP that stay within that range. In practice, once H is chosen accordingly with respect to camera noise and lighting environment, this value does not need to change substantially. However, if for a punctual reason the number of extracted CP should become higher than 5, the frame is recomputed using a slightly higher value of H. A more flexible approach will be presented in Chapter 3.

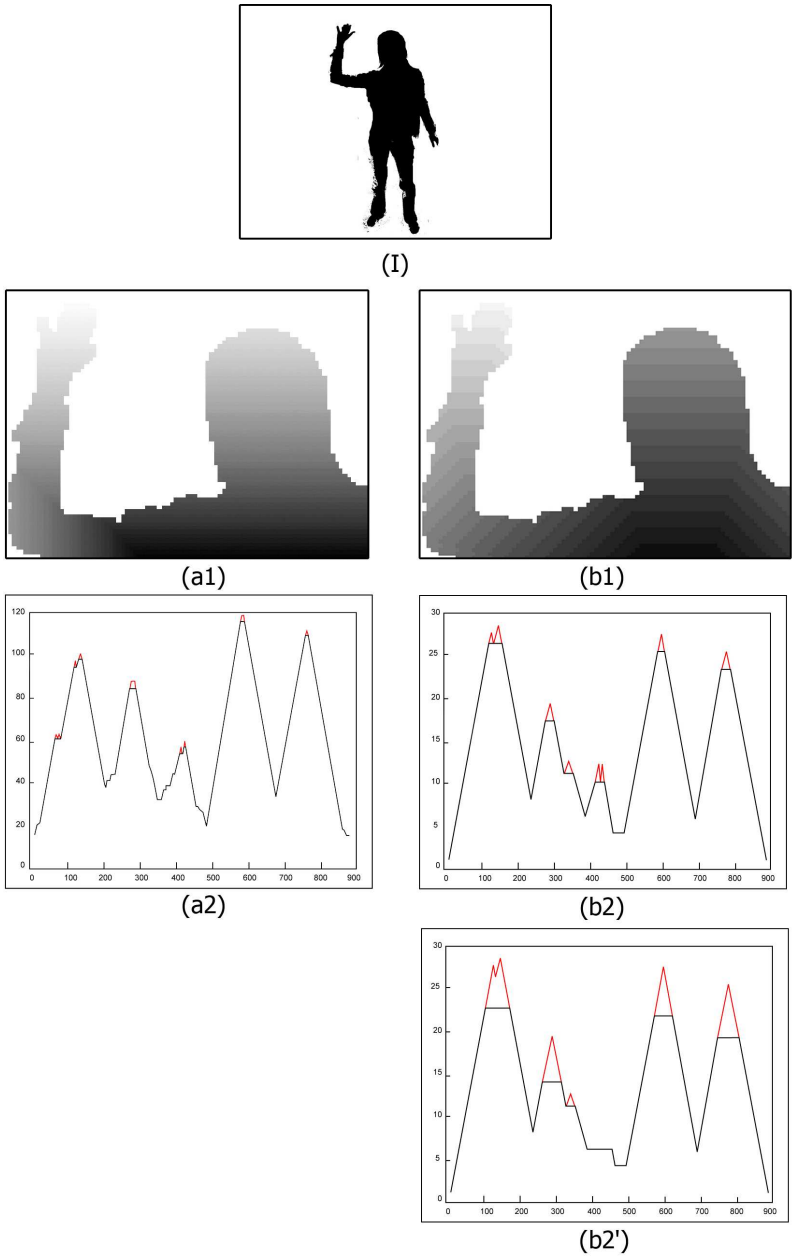


Figure 2.8: *Influence of SE and parameter H in distance function. Column ai displays a geodesic map detail and a distance function obtained with a SE of size 3 and H equal to 2. Column bi was computed with an SE of size 9 and H equal to 2 (b2) and 6 (b2′) respectively. Peaks in red represent the original function before H-maxima operator.*

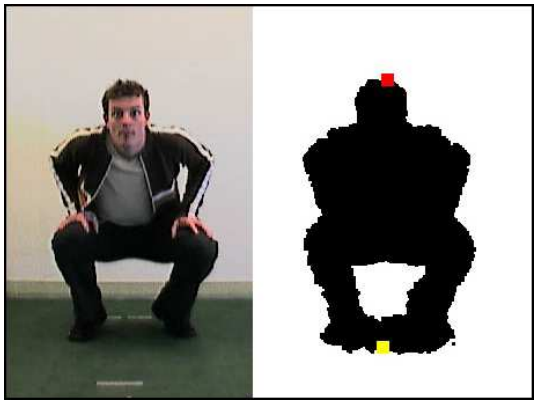


Figure 2.9: *Head and Foot extraction in presence of shadow segmented as noise between the feet.*

It is interesting to note that after applying the H-maxima operator on the distance function, local peaks are changed into local constant values. The extracted CP is then located in the pixel of the silhouette border corresponding to the middle of the flat zone value (Fig. 2.7 (b)). An important outcome of this fact is that when both feet are connected to one another in the silhouette due to a segmentation error, the extracted CP will be placed in the middle of the region connecting the feet. Hence, in those cases even if both feet can not be retrieved, the error does not have a dramatic impact, since the only foot label will be located near both feet (Fig. 2.9)

The function ($g(p)$) provides us as well with the *intensities* $I_t^{(i)}$ of each local maxima, which are their geodesic distance dynamics [Soille 2003]. These intensities are computed as the smaller distance measured from each local maxima peak after filtering, to each one of its two surrounding minima. Intensities help us quantifying the reliability of each local maxima after being selected. This is an important information that will be used during the inter-image phase (Chapter 3).

2.3.4 Results

For performance quantification purposes we have analyzed a representative sample sequence. The test sequence is 800 frames long and displays a vast scope of standing human gesticulations. It was recorded and results were computed at real-time paces, in the Alterface S.A installations. Hence, it also represents a good sample of realistic working conditions. The complete sequence is available following the link:
`http://www.tele.ucl.ac.be/~pedro/gestural/`

The CP extraction algorithm has a 94% average accuracy rate on the sequence. Even though this is a very high proportion of correct extractions, it needs however to be nuanced. First of all, this is the initial step of the overall motion capture chain. Hence, very high accuracy rates are needed in order to keep acceptable end results. Also, the error rate must be distinguished between the *no detection* errors and the *misdetection* errors. This algorithm can indeed only extract features that are visible in the silhouette, this is, the hands can only be located if not occluded by the rest of the body. This being a first limitation, errors are nevertheless counted as *no detections* each time a feature is visible but is not extracted (typically because its intensity is lower than the H value). On the other hand, parameter H is chosen so as to minimize the misdetection errors, this is, the extraction of extremities produced by noise. This second type of error is more damaging for the sequel, nevertheless, even if both errors are counted separately, they are given an equal weight in the overall average accuracy rate.

Both rates thus translate the H value trade-off. For this sequence, as for all the other test sequences, parameter H is fixed once and for all after having a single test on the segmentation, that is, after estimating the noise amplitude.

In order to quantify the error rates, we assume that the region of the hand extends from the tip of the fingers to the wrist, the feet from the ankle to the tip of the toes, and the head region corresponds to the skullcap. A *misdetection* error is thus counted each time a CP is extracted outside these regions. A *no detection* error is counted each time one of these regions are distinguishable by visual inspection but is not extracted.

The average error rates are:

- Misdetections in 15 frames = 2%.
- No detections in 32 frames = 4%.
- **Global Average Error Rate = 6%**.

Some other interesting results are depicted in Figure 2.12. It presents 2 geodesic distance maps and their respective $f(p)$ (before the H-maxima operator) and $g(p)$ (after the H-maxima operator) functions. It is also an illustration of the SE choice influence. As discussed in section 2.3.3, we can see that functions in column bi, computed with a cross-shaped SE, behave in a much more irregular manner than functions $f(p)$ and $g(p)$ of column ai (computed with a square SE). This translates in a higher number of local maxima in function $f(p)$. Moreover, the vertical direction appears privileged, the local maxima corresponding to the head appearing thus much less strong.

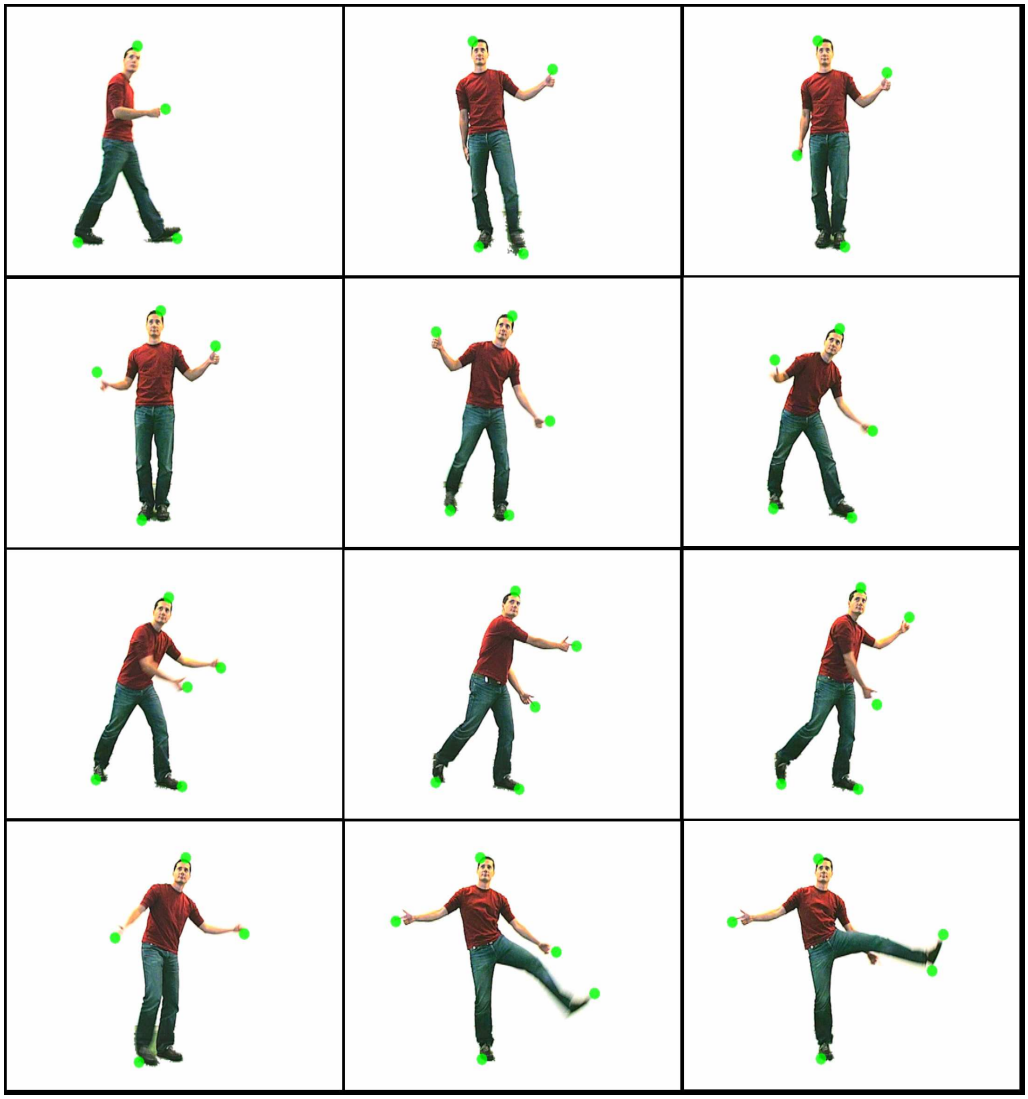


Figure 2.10: *Key frames of a real segmentation live sample sequence. Sequence about 800 frames long, shot using the Alterface infrastructures. The whole acquisition and processing was performed real-time. Part I.*

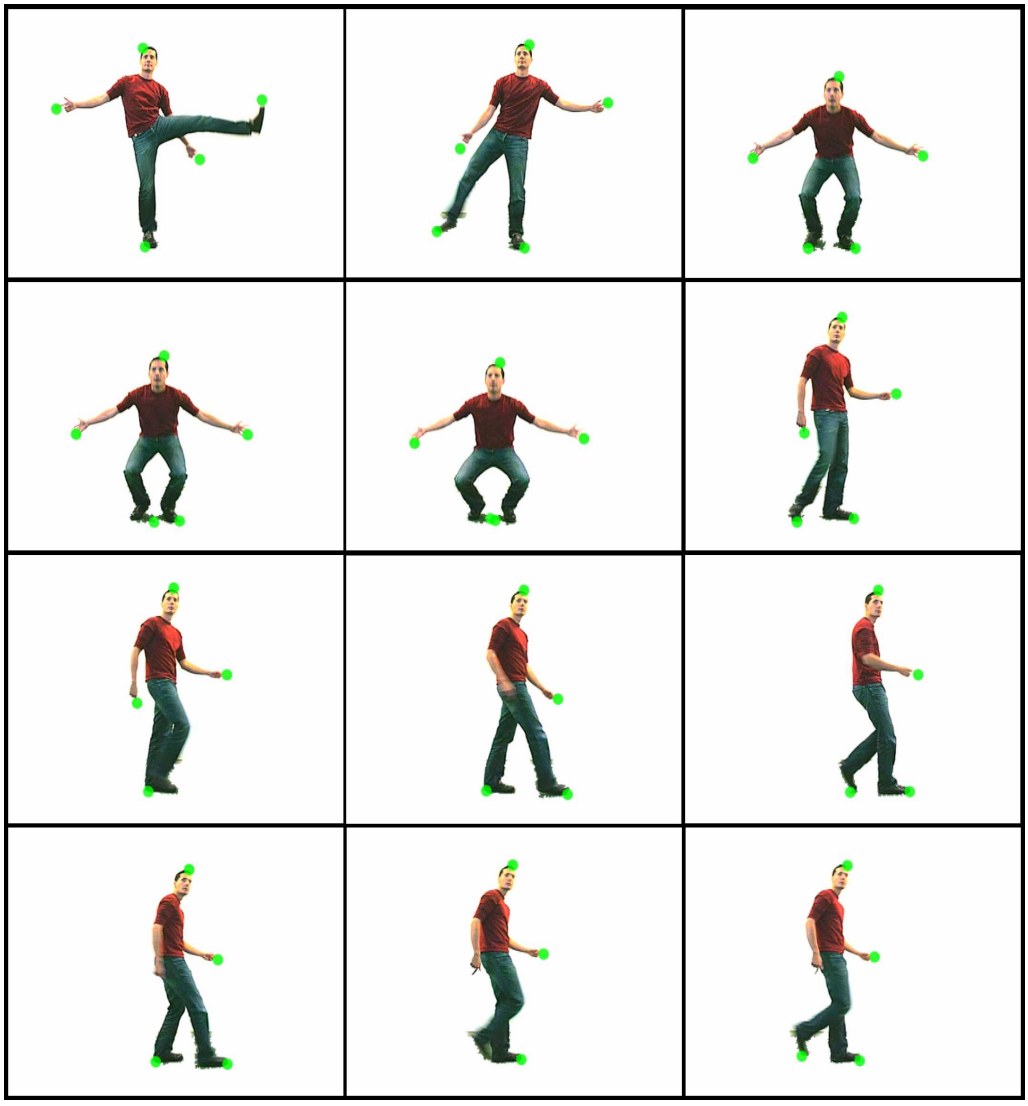


Figure 2.11: *Key frames of a real segmentation live sample sequence. Sequence about 800 frames long, shot using the Alterface infrastructures. The whole acquisition and processing was performed real-time. Part II.*

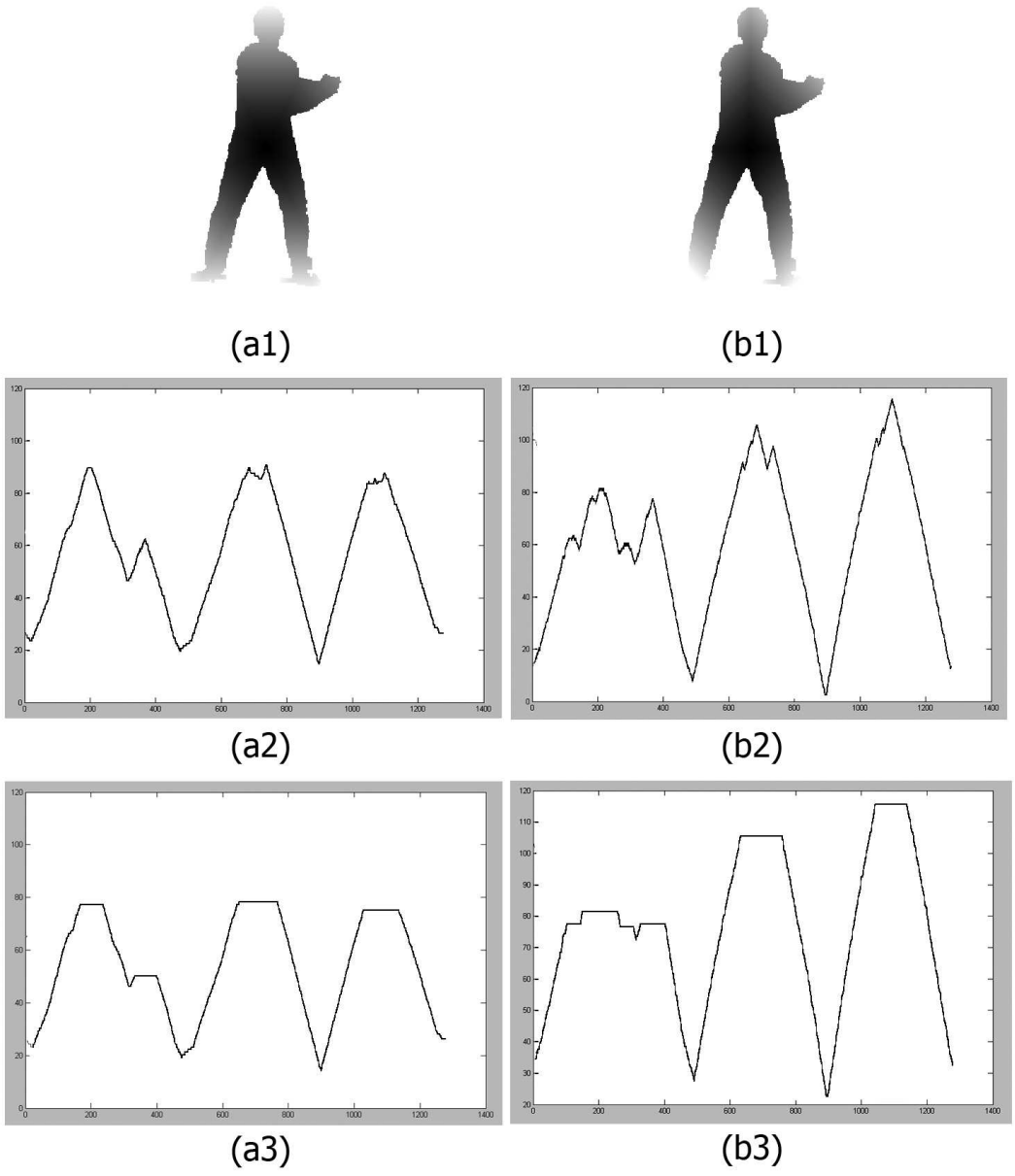


Figure 2.12: *Geodesic Distance Map and Distance Functions f(p) and g(p) illustration, with two different Structuring Elements. Row ai displays the results obtained with a pseudo-circular SE (an octagon inscribed in a 5×5 square). Row bi display results obtained with a cross (inscribed in a 5×5 square). Lines $a2/b2$ and $a3/b3$ display functions f(p) and g(p), respectively.*

2.4 Intra-Image Classification

Once the crucial points are extracted the next step is to classify them; that is, to associate each extracted crucial point with its corresponding human feature. We present hereafter an image processing tool that will help us achieve this goal.

2.4.1 Morphological Skeletons

Skeletonisation is a standard image processing tool that computes all the pixels that are equidistant to the shape's border in order to condense a shape into a group of lines that still defines the original nature of the figure (it preserves the extent and connectivity of the original region).

It was first introduced by Blum in 1961 as the *medial axis transformation* [Blum 1961], based on the *grassfire* analogy: assuming a grassfire was started from the boundary of a set $X \subset \mathbb{Z}^2$ and propagating within it at constant speed, the skeleton $S(X)$ of X is the set of pixels where different grassfire fronts meet. A more formal definition was then proposed by Calabi [Calabi and Harnett 1966], based on the notion of maximal ball: the skeleton of X is defined as the locus of its maximal balls for the used metrics, which actually lead to the local maxima of the distance function of X [Vincent 1991a]. In this latter approach, the skeleton $S_k(X)$ represents the local maxima of a $D_C(X)$ distance function, where C denotes the underlying grid $C \subset \mathbb{Z}^2 \times \mathbb{Z}^2$ defining the neighborhood relation between pixels. C is usually a square grid (of 4- or 8-connectivity). In our case, the square grid with an 8-connectivity will be privileged.

$$S_k(X) = \{p \in X \mid \forall p' \in N_C(p), [D_C(X)](p') \leq [D_C(X)](p)\} \tag{2.8}$$

where $N_C(p)$ denotes the set of the neighbors of a pixel $p \in \mathbb{Z}^2$ according to a grid C. The result of this latter approach is unfortunately a skeleton that is not necessarily connected, even when the original set is.

Another alternative approach has been to construct morphological skeletons by successive thinnings of the original shape, and thus rejoining the

original *firegrass* analogy, at least in its concept. These approaches, that greatly vary in their implementation, have the advantage of extracting an object of unit-width which is close to the skeleton by maximal balls, while preserving the homotopy of the original set. They consist in removing successive peels of the set by means of homotopic thinnings until idempotence [Serra 1982].

In our algorithm an approach by successive peelings using morphological operators is used, since it preserves the following properties, essential for the purposes of the algorithm:

- Homotopy. It must preserve the original image topology. In other words the original silhouette and the corresponding skeleton need to have the same Euler number, given by

$$E_N = F - E + V \tag{2.9}$$

  where F denotes the number of faces, E the number of edges and V the number of vertexes.

- Homogeneous thickness, which must be as small as possible. One-pixel thickness is the optimum.

- Mediality: the skeleton should lie in the middle of all borders.

However, the used skeletonisation algorithm is non-optimized. The interested reader is referred to the following faster algorithms. The approach presented in [Meyer 1990] uses an extension of the *chains and loops* approach already cited in the context of morphological dilation optimization in Section 2.3.1, where it is applied in the creation of Euclidean skeletons. This approach requires nevertheless a cumbersome neighborhood analysis and has a rather poor flexibility. Also, [Vincent 1991a] presents an efficient and more flexible skeletonisation approach, based on queues of pixels. [Maragos and Schafer 1986] claim for instance that optimized skeletonisation algorithms can reduce the quadratic computational complexity of standard skeletonisation algorithms to a linear one.

Figure 2.13 shows a skeletonisation example, and also how sensitive to noise this procedure is.

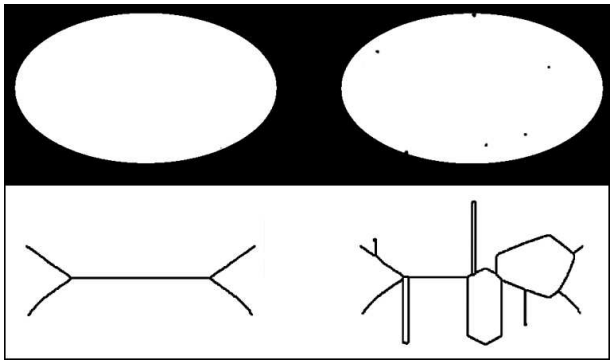


Figure 2.13: *Skeletonisation example. The upper row displays the original binary shapes. Note how small the variation in the original shape has been, and how dramatic the impact on the resulting skeleton*

In our case, the morphological skeleton is used as a paradigm of the actual anatomical skeleton (a step further from the simplest human stick figure model introduced in Section 2.1). From the gestural point of view, this skeleton carries as much information as the whole silhouette shape, yet being only represented by a set of lines. It is thus a compact form to carry anatomical information, in order to ease the characterization of each extracted crucial point into a specific human feature, among other possible applications.

Nevertheless, as we have seen in Fig. 2.13, morphological skeletons are very sensitive to noise, since every noise point is connected to the original skeleton by a line. In our case, noisy bifurcations appear for every brisk silhouette border irregularity (Fig. 2.16 (a)). Indeed, as proven in [Maragos and Schafer 1986], the transformation of a binary image into its morphological skeleton is not continuous, and notches in the boundaries create new *bones* in the skeleton. Fig. 2.14 illustrates this fact: the original shape presents a triangular notch in the boundary of the rectangle of width (w/n) and height (h/n). When $n \rightarrow \infty$, the limit of the sequence of shapes is the rectangle without the notch.

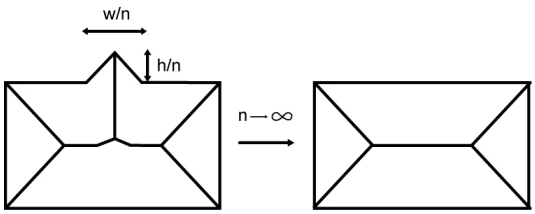


Figure 2.14: *Illustration of the discontinuous branch creation in a morphological skeleton. The triangular notch on the rectangle creates an additional branch. The branch disappears discontinuously as the dimensions of the notch tend to* 0.

2.4.2 Selective pruning and robust skeletons

Since we seek the most compact form for translating extremities morphology, it is essential to work with skeletons that have only relevant branches; this is, skeletons that are exclusively composed of paths connecting the extremities of the user with each other. To do so, we need to operate a *selective pruning*; this is, to remove every pixel belonging to a noisy branch, without altering branches connecting to a CP. Skeleton pruning removes iteratively every branch extremity pixels, either until idempotence is reached,

$$P(S_k) = (X \bigcirc E)^{\infty} \tag{2.10}$$

or until a given number of iterations is reached (parametrical pruning)

$$P(S_k) = (X \bigcirc E)^{(n)} \tag{2.11}$$

where S_k denotes the binary skeleton, E denotes the SE used in order to detect extremity pixels, and $\bigcirc$ is the extremity pruning operator.

We thus perform a non parametrical selective pruning in order to construct a robust skeleton. Noisy branches are removed from the morphological skeleton using the prior information extracted from the silhouette, i.e. the location of the crucial points. This way only the extremity pixels from the noisy branches are removed, one by one until a junction is reached (Fig. 2.15). A morphological skeleton of a silhouette before and after selective pruning is depicted in Figure 2.16.

This robust skeleton represents a useful tool for a subsequent analysis of the silhouette, since it is a condensed expression of the latter, and creates

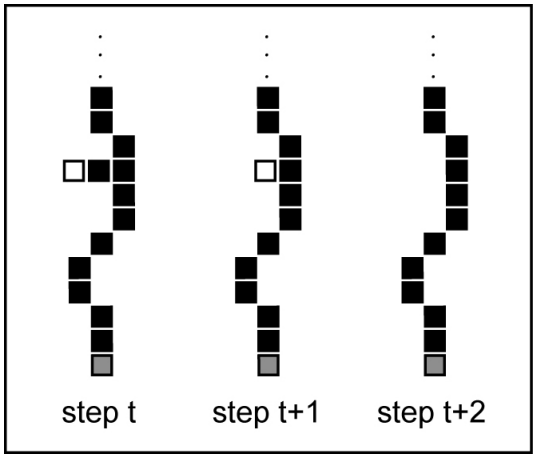


Figure 2.15: *Selective Pruning Illustration. Pixels belonging to a morphological skeleton are depicted in black. A skeleton extremity corresponding to a Crucial Point is depicted in gray. A pixel corresponding to a skeleton extremity is depicted in white. At each step, selective pruning is performed on non crucial point skeleton extremities, until a junction is found.*

a measurement space between crucial features. The next section presents the practical utility of morphological skeletons and, more precisely, selectively pruned skeletons in detail, in order to label crucial features.

2.4.3 Feature Classification

In the intra-image approach, we use the geodesic distances between crucial points in order to classify them. We use the previously computed clean morphological skeleton as a set of paths than enable us to estimate which points are geodesically closest to the rest, and which are the most isolated. Hence, it allows us to pre-classify the different extracted features as isolated/grouped extremities.

Indeed, the human anatomy is constructed in such a way that it allows a grouping of the five extremities in two distinct groups, in terms of geodesic distance. The head and the feet are the most far apart from each other since they are actually separated by the whole body height (chest

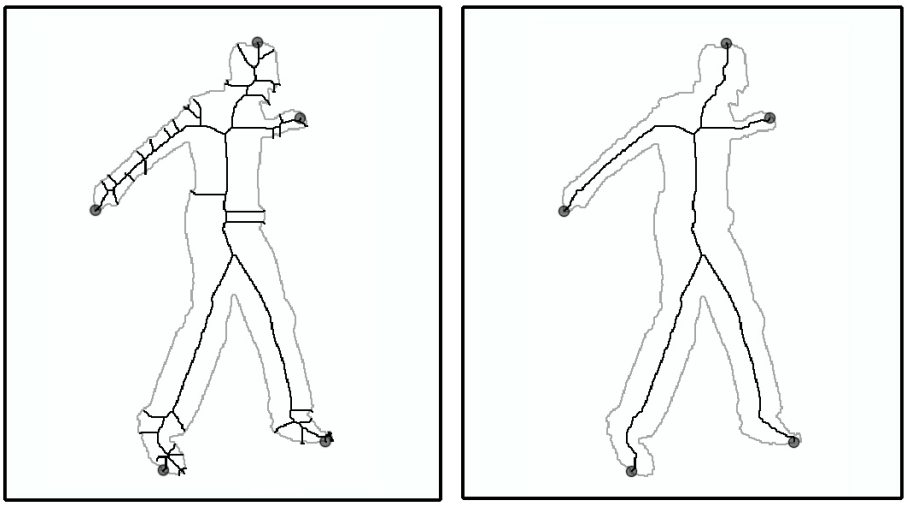


Figure 2.16: *Robust Skeletonisation example. Extracted crucial points are displayed as discs. Morphological skeleton is displayed in bold before (a) and after (b) selective pruning.*

and legs, essentially). As for the hands they are basically separated from the head by the arms. Geodesically speaking, we can thus merge head and hands in one group, and both feet into another one.

Since these properties are the result of geodesic computations and hence a certain reflect of human skeletal distances they are always verified provided we have the means (like stereovision cameras and/or a set of multiple cameras) to avoid perspective aberrations and/or occlusions. The more context-based the application is the simpler the set-up can be, in order to avoid those problems. For instance, in the kind of applications we target (i.e. gestural interaction for navigation, virtual gaming and edutainment), the stand-up frontal position is the most widespread. Hence, in these targeted applications a frontal camera is sufficient in order to achieve good results (see section 2.4.4).

Morphological and geodesic skeletons prove thus to be a powerful tool in order to analyze in detail a given silhouette, yet on the other hand they are very time consuming, as we have seen earlier. A more flexible and fast

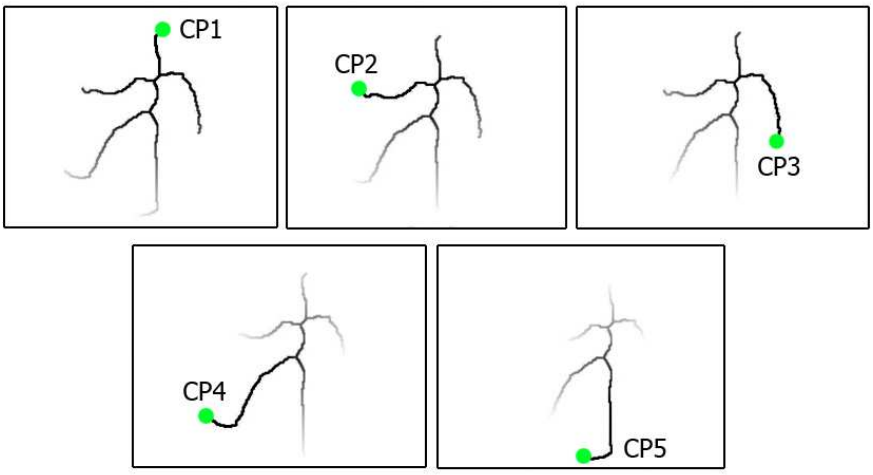


Figure 2.17: *Computation of the five geodesic distance maps related to the five crucial points detected in Figure 2.7. Higher distance values have lighter shades of gray. Crucial points with respect to which each geodesic skeleton is computed are depicted by discs.*

labelling method is presented in Chapter 3.

In order to classify every extracted crucial point into one of the two *head-hands* or *feet* group, the cleaned skeleton is used as a seed to create a set of geodesic skeletons. Geodesic skeletons are geodesic distance maps that use the clean geodesic skeleton as the original domain. For each of the previously extracted crucial points, a geodesic distance map is computed using the crucial point as reference point (Fig. 2.17). For each geodesic map we will henceforth refer to the crucial point with respect to which the map is computed as the *reference point* of the map. Hence, the algorithm computes as many geodesic distance maps as extracted crucial points (2 at least, 5 at most). Figure (Fig. 2.17) illustrates the set of geodesic skeletons corresponding to a given silhouette (Fig. 2.3).

Each crucial point is thus a reference point in one geodesic skeleton map. Using the whole set of geodesic skeletons we actually have for each map, and thus for each crucial point, the computation of all the other CP geodesic distances with respect to the crucial point being the reference in

that map. Hence, summing up all the crucial points distances with respect to the reference point, we actually compute this latter global positioning with respect to all the other extremities.

Using a *classification table* as the one presented in Table 2.2, after summing up the values of each row we obtain what we will refer in the sequel as *scores*. We can see that scores form two distinct groups. The two highest values form the feet group, the other form the head-hands group. Table 2.2 corresponds to the labelling of the silhouette extracted in Fig. 2.3, and is created using the set of geodesic skeletons of Fig. 2.17. Each row i corresponds to values extracted from Map i, corresponding in its turn to the map that has CPi as a reference point. Hence, rows present geodesic distances from reference point i to each one of the other CP in that particular map.

	C.P. 1	**C.P. 2**	**C.P. 3**	**C.P. 4**	**C.P. 5**	**Final Scores**
Map 1	0	21	25	54	60	160
Map 2	21	0	31	82	78	212
Map 3	25	31	0	73	79	208
Map 4	54	82	73	0	70	**279**
Map 5	60	78	79	70	0	**287**

Table 2.2: *Classification table associated to the set of geodesic maps of Figure 2.7. For each Map i, all distances in the same row are computed with respect to Crucial Point i. The two highest scores are marked in bold, and form the feet group.*

After having classified the two crucial points corresponding to the feet, the Crucial Point (*CP*) representing the head has to be pointed out from the head-hands group in order to finalize the crucial point classification. To do so, an a priori anatomic information is used: the head is the CP that should be in the prolongation or the closest to the line passing trough the CoG and having the same slope as the angle of the torso.

In order to approximate the angle of the torso we compute the principal axis of the silhouette torso. First of all, in order to have a cleaner shape leading to a more accurate principal axis angle, the silhouette is eroded so that only the torso area remains, as proposed in [Ohya et al. 2002].

The shape of the corresponding structuring element has little influence in

the result (as far as it remains an isotropic convex shape as the one discussed in Section 2.3.3).Its size has to be chosen wisely, since the shape of the remaining area has a great influence in the angle computation when using the principal axis method [Cortadellas et al. 2004]. Hence, this parameter has been left as an initial one.

The torso angle is computed as the principal axis of the torso area. This axis is the one that has the best fit inside the eroded silhouette region. It is obtained by minimizing with respect to the angle θ the sum of the squared distances of each silhouette point to an axis passing through the coordinates (α, β) and having a slope $\tan\theta$:

$$\min_{\{\theta,\alpha,\beta\}} \sum_{(x,y)\in S} [(x-\alpha)\sin\theta - (y-\beta)\cos\theta]^2 \tag{2.12}$$

where S corresponds to the torso area. This minimization results in [Haralick and Shapiro 1992]:

$$\theta = \frac{1}{2} tan^{-1}\left(\frac{2\mu_{11}}{\mu_{20}-\mu_{02}}\right) \tag{2.13}$$

$$\alpha = \overline{x}, \quad \beta = \overline{y} \tag{2.14}$$

where $\overline{x}, \overline{y}$ are the coordinates of its center of gravity (Section 2.3.1) and for each pair of non-negative integers (j, k), μ_{jk} are the central moments of the silhouette, given by

$$\mu_{jk} = \sum_{(x,y)\in S} (x-\bar{x})^j (y-\bar{y})^k \tag{2.15}$$

See [Haralick and Shapiro 1992] for more details.

Once the angle of the torso is computed, the extracted crucial point having the closest angle to the torso angle is classified as being the head. The other two points left in the head/hands group are thus classified as hands.

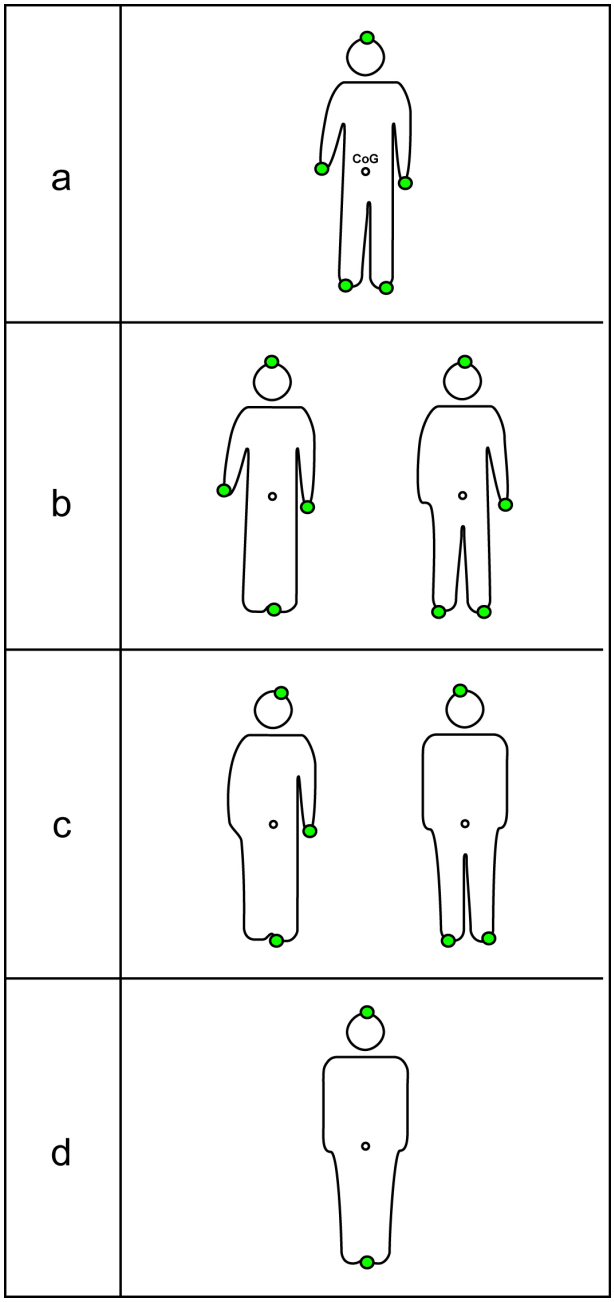


Figure 2.18: *Possible Crucial Points labelling configurations. Row (a) considers a 5 CP extraction and rows (b),(c) and (d) display the extraction of 4,3 and 2 CP, respectively. Effectively extracted crucial points are depicted as discs.*

The number of extracted crucial points ranges from 2 to 5. Note that in the following taxonomy (illustrated in Fig. 2.18) we refer to a *feet* configuration where the feet are joined or connected and thus being extracted as a single point in the middle of the two, while a *foot-foot* configuration clearly extracts both feet.

- When 2 extremities are extracted, it only allows a *Head-Feet* configuration.
- When 3 extremities are extracted, it allows a *Head-Hand-Feet* or a *Head-Foot-Foot* configuration.
- When 4 extremities are extracted, it allows a *Head-Hand-Hand-Feet* or a *Head-Hand-Foot-Foot*.
- The 5 extremities are extracted it only allows a *Head-Hand-Hand-Foot-Foot* standard configuration.

Each configuration follows analog classification procedures, but the differences are detailed hereafter. For the 5 CP configuration, the labelling is achieved as detailed above. For the 2 CP configuration, the labelling is achieved by classifying as head the CP located closest to the angle of the torso, the foot being the CP in contact with the floor.

For the 3 and 4 CP configurations, some heuristics are needed. Both cases have two possible labelling configurations (Fig 2.18 (b) and (c)): a *Foot-Foot* configuration and a sole Foot configuration. Let us discuss the 3 CP case, the 4 CP case being analogous.

Some measurements involving both Euclidean and geodesic distances are performed in order to determine which one of the two, whether the $Head - Foot - Foot$ or the $Head - Hand - Foot$, is indeed correct. In practice, the verification is carried on the CP that is neither the lowest one (which has to be a foot) nor the crucial point already labelled as *Head*. This particular CP will be denoted C in the sequel (Fig. 2.19). Indeed, let us denote l_1 the geodesic (skeletal) distance from the CoG to point $Foot$, and l_2 the geodesic distance from the CoG to point C, and finally $\triangle y$ the vertical difference in Euclidean distance between C and point $Foot$. Note that since the CoG is very unlikely to be part of the skeleton, the geodesic distance is computed from a horizontal projection of the CoG to the skeleton.

Finally, we will denote CoG_{down} the Euclidean vertical distance between the CoG and the lowest point of the silhouette (point Foot). Three tests are thus carried:

1. Parameter l_1 being the leg distance, C will have an increasing (decreasing) probability of being a hand as its distance to the CoG l_2 is higher than (closer to) l_1. We can thus take a security coefficient $\tau = \frac{1}{3}\, l_1$ and set the first test as:

$$l_2 \overset{?}{>} l_1 \, + \, \tau. \tag{2.16}$$

2. Following the same reasoning (and thus that the hand is assumed to be at most at half height of the CoG), point C will have increasing (decreasing) chances of being a hand, as the parameter $\triangle y$ increases (decreases). We can thus set the second test as being:

$$\triangle y \overset{?}{>} \frac{1}{2}\, CoG_{down}. \tag{2.17}$$

3. The previous test links the vertical position of point C with the vertical position of point *Foot*. We thus perform a third test on the vertical position of Point C alone,

$$C_y \overset{?}{>} CoG_y \tag{2.18}$$

C_y and CoG_y being the vertical coordinates of C and the CoG, respectively.

If two out of the three tests are affirmative, then point C is labelled as a hand, otherwise it is labelled as a foot. See Figure 2.19 for a labelling situation where these tests apply. In this example, $l_1 = 223$, $l_2 = 318$, $\triangle y = 135$ and $CoG_{down} = 221$. Point C passes the first two tests but not the last one, since it is indeed located below the CoG. Point C is thus labelled as *Hand*.

As it will be presented in Chapter 3, these above classification heuristics will be extended and formalize as priors in a Bayesian framework.

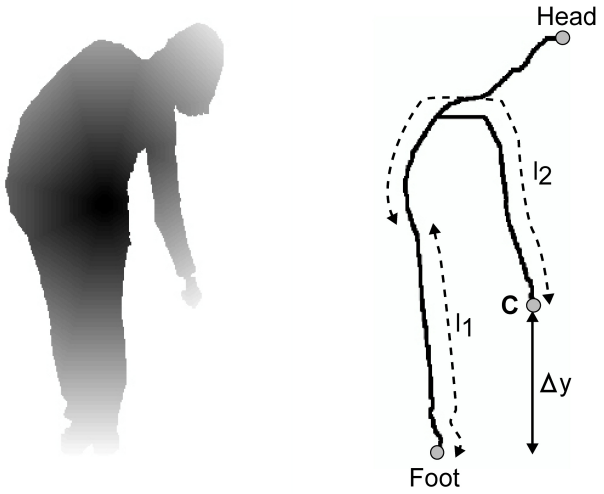


Figure 2.19: *Intra-Image labelling illustration.* l_1 *denotes the geodesic distance from CoG to 'Foot'.* l_2 *denotes the geodesic distance from CoG to CP 'C'.* $\triangle y$ *denotes the vertical Euclidean distance between CoG and 'Foot'.*

Holes and Loops

As we have seen, this algorithm has been adapted to handle the situations where less than five crucial points have been detected (the rest of crucial points being omitted by self occlusions), see section 2.4.4 for results. Indeed, as we can notice in Figure 2.19, even in the cases where less than five crucial points are visible, the geodesic distance map remains coherent and the morphological hypothesis as well.

In those cases however, *holes* inside the silhouette sometimes appear, this is, connected areas that belong to the background and are nevertheless included inside the silhouette area. This translates in loops in the morphological skeleton.

Since the distance function is performed over the contour of the silhouette, as long as the whole silhouette remains a connected domain inner holes do not perturb directly the distance functions $f(p)$ and $g(p)$. They do however introduce an indirect perturbation in the geodesic distance map computation, yet not a dramatic one. The following explanation focuses

on the standard geodesic map computation by successive morphological dilations, but is also applies to the optimized Dijkstra slices approach (Section 2.3.2). Indeed, if the geodesic dilations encounter the background hole at step t, successive dilations will split inside the connected domain, surrounding the hole and colliding again on the other side of the hole at step $t + n$, creating a local maxima (see Figure 2.20). These local maxima are however usually quite small. In the case of the arms, first of all the clothed arms width/length ratio do not allow the creation of important hole surfaces. As for the legs, the corresponding maxima is actually located in the center of the two colliding feet, and thus correspond to an actual CP location. In Figure 2.20, the little influence of the holes is visible. The one created by the arm is too small to have any influence on the geodesic map, while the hole created by the legs is big enough to have some influence on the geodesic map computation, but not any on the distance function $g(p)$. Figure 2.21 presents the skeleton and final classifications results of the same frame.

As for the geodesic skeletons, the perturbation is analogous to the one introduced by the holes over the border $f(p)$ and $g(p)$ functions. Figure 2.20 also presents the overall results (and skeleton) for this configuration.

Note that in this first approach no feature tracking is achieved. The results displayed hereafter (Section 2.4.4) are thus achieved without any temporal tracking. A temporal tracking for this first approach will actually be presented in Chapter 4, where the same technique is used with two orthogonal cameras. Indeed, this approach was first designed in order to be used in that kind of infrastructure, where the information redundance of the two orthogonal cameras would have compensated possible errors introduced by the above heuristics.

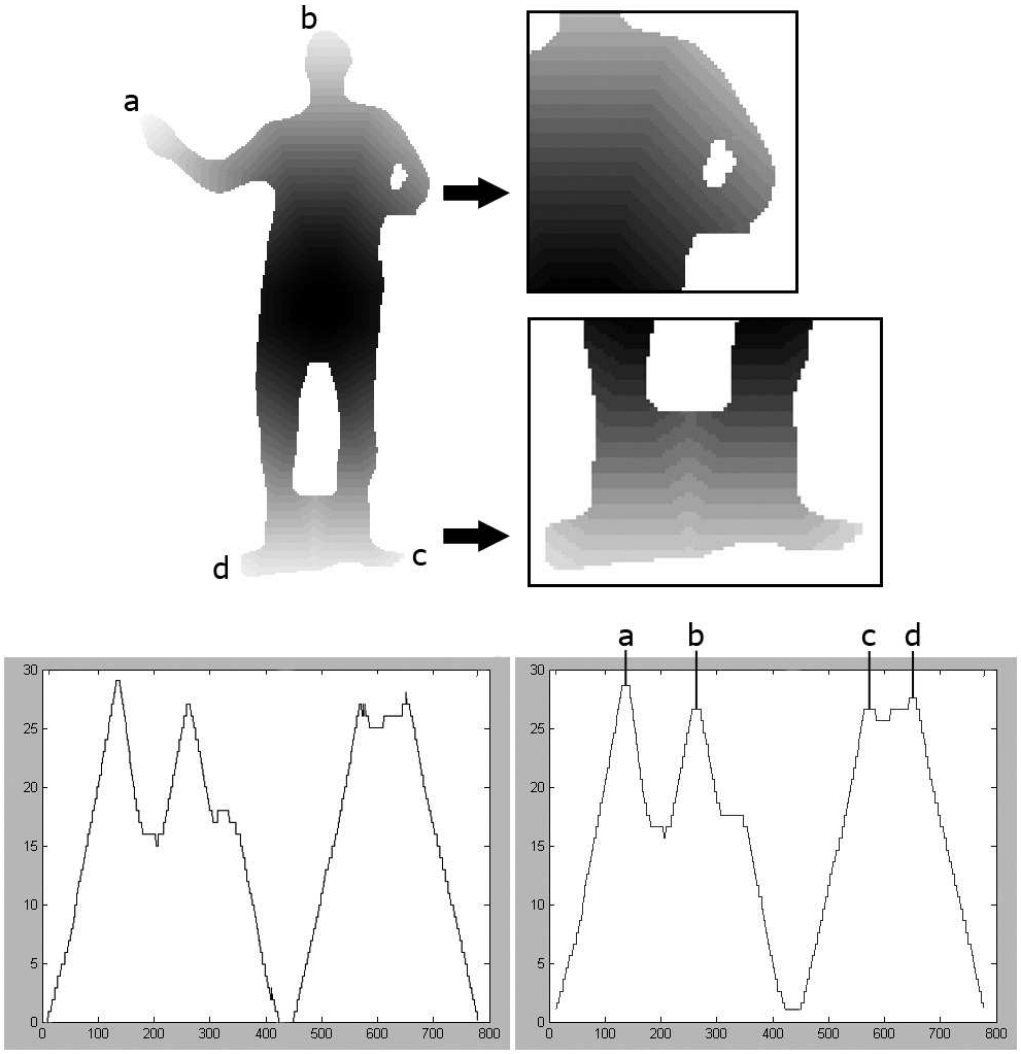


Figure 2.20: *Illustration of the influence of holes inside the silhouette. Points a, b, c and d represent the extracted crucial points. Their location is presented both in the g(p) function and in the frame domain. Functions $f(p)$ and $g(p)$ are depicted in the lower left and right hand sides of the Figure, respectively.*

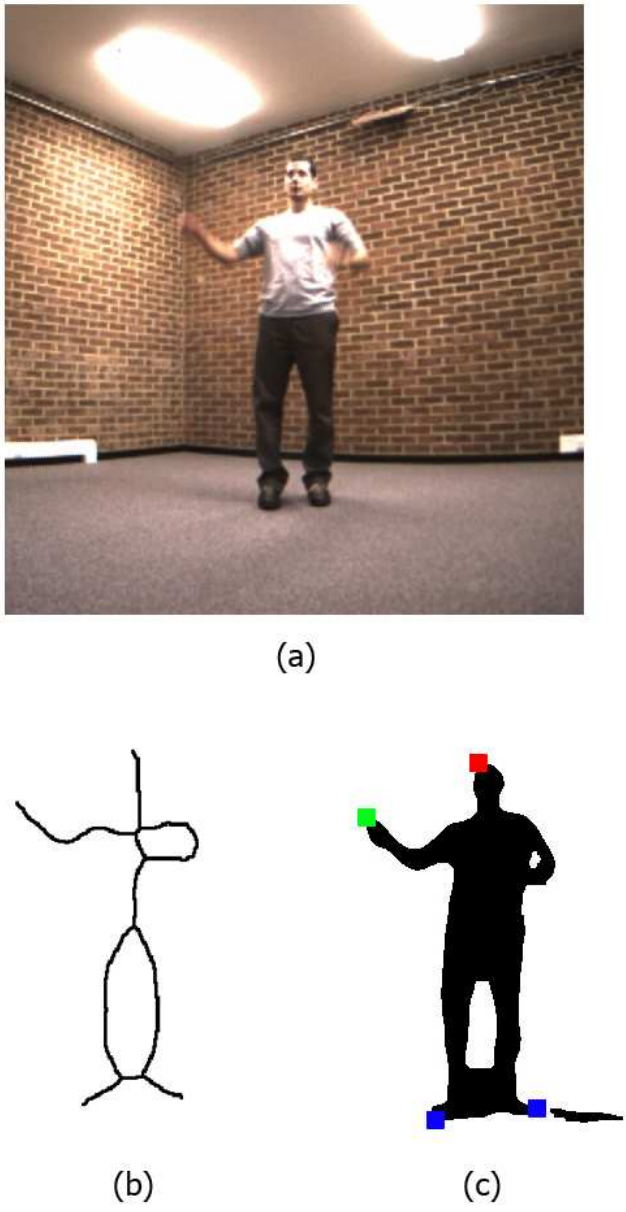


Figure 2.21: *Final results on the previously commented loop silhouette. The original frame is presented (a). The pruned morphological skeleton is depicted in (b). Final results are presented in (c).*

2.4.4 Results

Let us now review the overall algorithm performance, as well as some interesting particular cases.

Morphological Skeletons

As commented above, robust morphological skeletons obtained after a selective pruning convey highly relevant information in order to analyze a given human morphology under a very compact form. This tool is thus an important intermediate contribution. Hereafter (Figures 2.22 and 2.23) we present a series of key frames of a general stand-up biped movement sequence.

Extraction and Labelling Results

Concerning the extraction and labelling under CP self-occlusions, Figure 2.24 presents in detail two examples of CP extraction and labelling when one or more Crucial Points are hidden inside the border of the silhouette. The upper row illustrates the extraction and labelling of a 4 CP configuration, with one hand occluded and the other hand creating a hole/loop situation (as presented in Section 2.4.3). The lower row presents the extraction and labelling of a 3 CP configuration, using the labelling procedure presented in Section 2.4.3. As discussed in that Section, we can thus see that the algorithm supports these configurations. A complete set of other various postures is presented hereafter.

Overall algorithm performance quantification is presented in Table 2.3. It was built using real and synthetic (perfect segmentation) sequences, of around 700 frames in total length. An example of synthetic sequence is presented in Figures 2.25 and 2.25. A real sequence keyframes are displayed in Figure 2.10 and 2.11. Values corresponding to perfectly segmented sequences are displayed in brackets. It displays the average error produced by the algorithm when classifying each different feature. A classification error was counted each time a feature was misclassified, and a

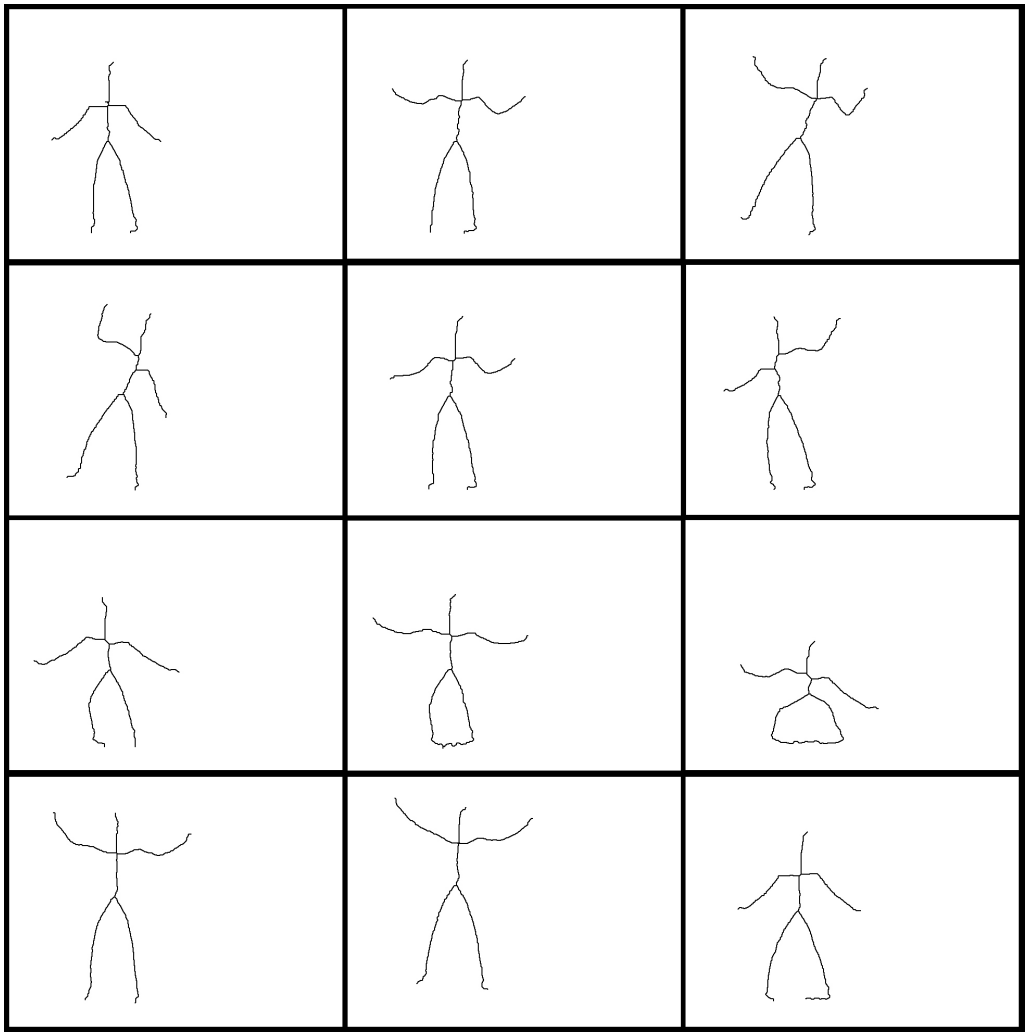


Figure 2.22: *Robust morphological skeletons sequence. 24 key frames on a 500 frame sequence. Part I.*

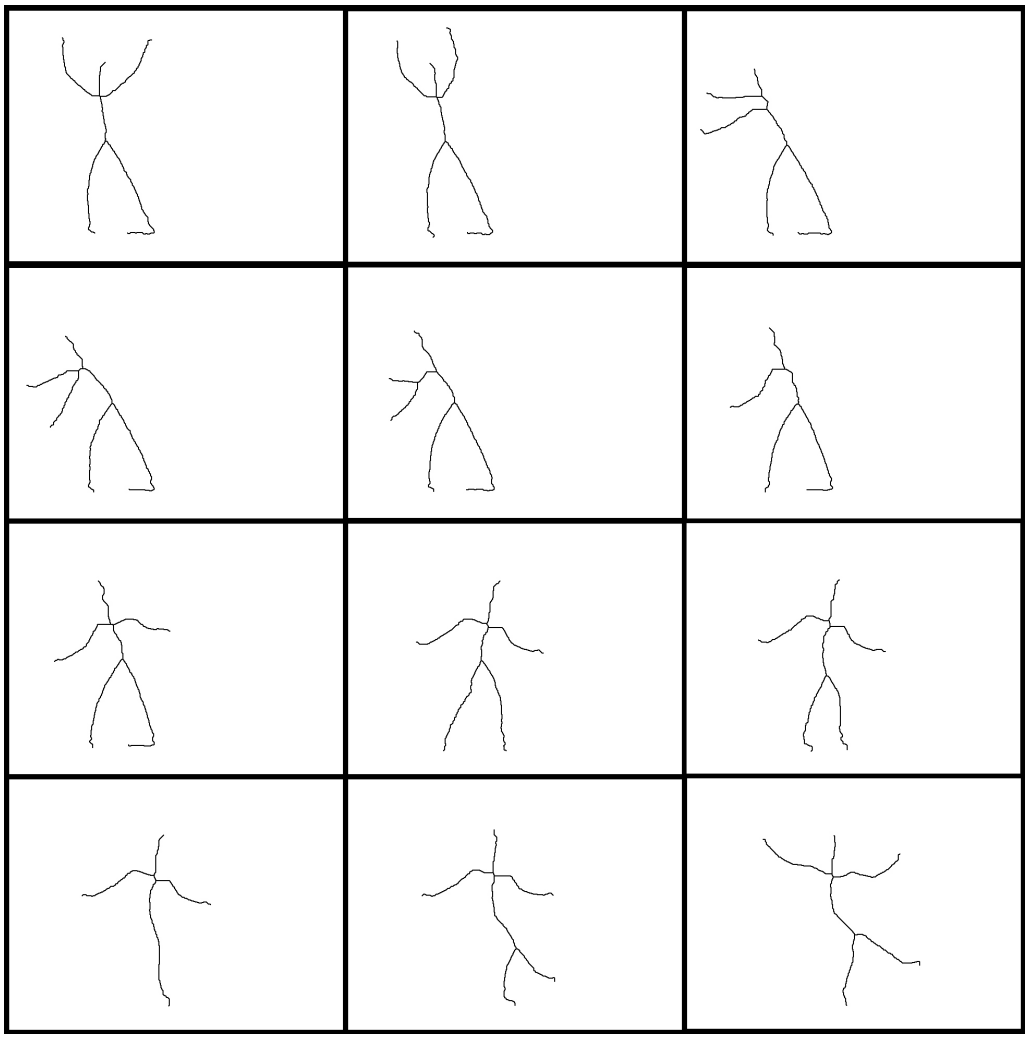


Figure 2.23: *Robust morphological skeletons sequence. 24 key frames on a 500 frame sequence. Part II.*

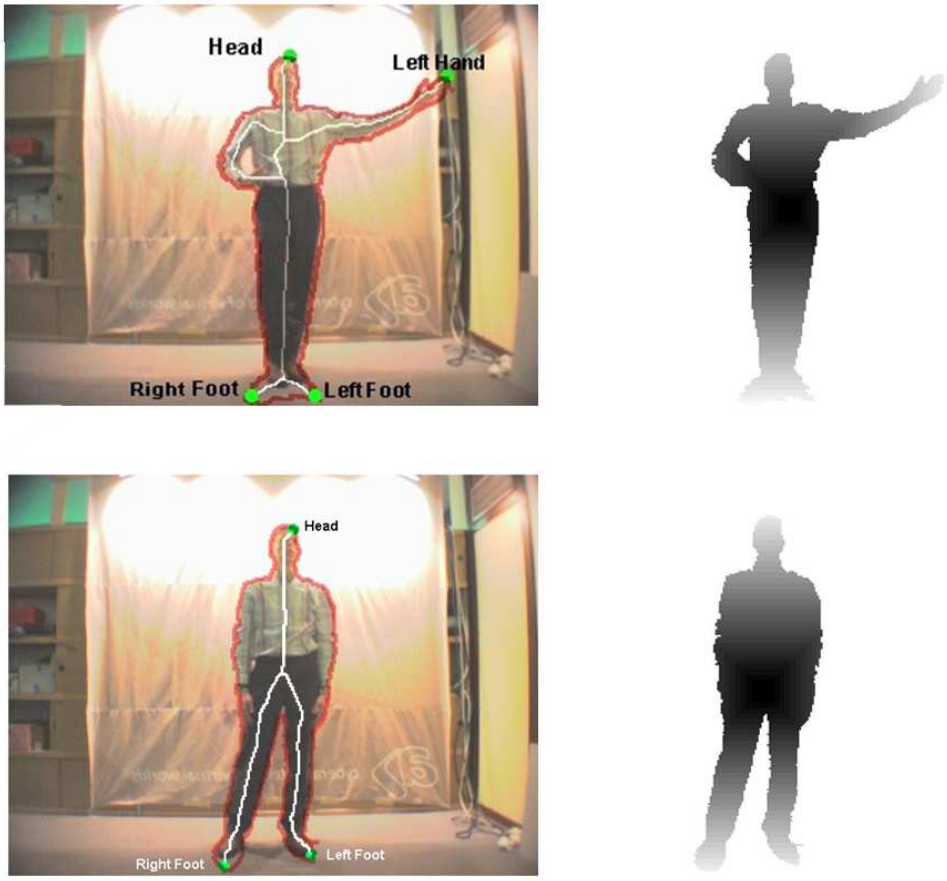


Figure 2.24: *Two examples of CP extraction and labelling in the presence of self-occlusions. The upper row illustrates the extraction and labelling of 4 CP with one hand occluded, the other hand creating a hole/loop situation. The lower row presents the extraction and labelling of a 3 CP configuration*

	Left Hand	Right Hand	Head	Left Foot	Right Foot	Avg. Error Rate (%)
Classification Error (%)	3.15 (1.84)	2.63 (2.36)	1.84 (1.57)	11.05 (3.15)	10.26 (2.89)	- -
No Detection (%)	2.63 (2.10)	3.15 (1.57)	1.31 (3)	2.36 (1.57)	3.42 (1.31)	- -
Error Rate (%)	6 (4)	5.7 (4)	2.45 (3)	14 (5)	14.1 (4.35)	**8.44 (4)**

Table 2.3: *Overall intra-image CP extraction and classification performance.*

no detection error was counted each time a visible feature was missed. The error rate row presents the average errors for each different feature, and the global average error rate of the algorithm is presented in the last column, in bold. This average error rate is about 9% for natural sequences and 4% for synthetic ones.

Figures 2.25 and 2.26 present global results on a synthetic (hence perfectly segmented) sequence with 192 frames. This sequence presents the results on a complex *ballet* movement, including spins and very active limbs. Also note how the silhouette becomes continuously bigger as the camera gets closer, without altering the results. This sequence sample illustrates the typical overall algorithm performance for synthetic results (without taking into account segmentation imperfections), i.e.:

- As long as the Crucial Points are detached from the silhouette, they are **extracted** with an average 98% accuracy.
- Even in quite complicated configurations Crucial Points are **labelled** with an overall 95% average accuracy.
- Since no tracking is present up to this point, CP are labelled in three distinct groups: head, hands and feet (respectively displayed in red, green and blue). CP are thus labelled, however not *followed*, contrary to the inter-image approach described in Chapter 3.

- A typical mislabelling appears between hand and head as both CP interact with each other (by fusing in the silhouette), or when an arm is positioned as prolongation of the torso (i.e last frame in Figure 2.26).

Finally, figures 2.27 and 2.28 illustrate an interesting example. We have seen in this chapter (Section 2.3.3) how parameter H was used in order to minimize the effects of segmentation noise. Moreover, the algorithm can also be tuned, via this parameter, to adapt the extent of the feature extraction. It is indeed possible to adapt the sensitivity of the Crucial Point detection algorithm, in order either to be kept fairly low or on the contrary to go to a very detailed CP extraction. This way we will be able to extract respectively only the most prominent features (Head and Feet in the case of an upright position) or go up to a finger extraction level, provided the resolution of input images (and accuracy of segmentation) is sufficient. Figure 2.27 presents the original snapshot (a), as well as the original geodesic distance map (b), morphological skeleton (c) and $f(p)$ function (d). Note that this latter counts 18 local maxima. Figure 2.28 illustrates the two different approaches commented above. In column ai parameter H was chosen as to remove all noisy maxima, yet being low enough to extract a maximum number of silhouette extremities. Only 11 local maxima remain in the $g(p)$ function (a1): the head, the feet, and 8 fingers out of 10. Hence the 11 branches left in the selectively pruned skeleton ($a2$). On the other hand, column bi depicts the results using a higher value of parameter H. Only 5 local maxima are left, and thus only the five extremities are extracted.

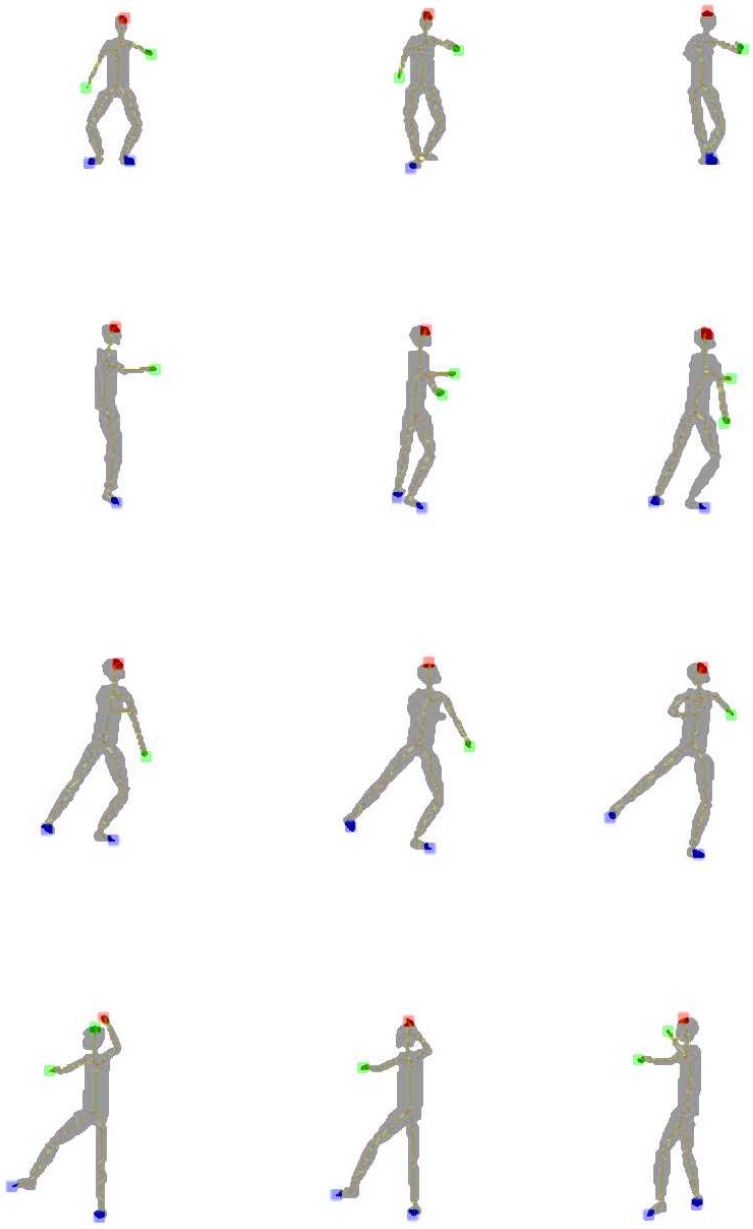


Figure 2.25: *'Ballet' sequence key frames. Total length: 192 frames. Part I.*

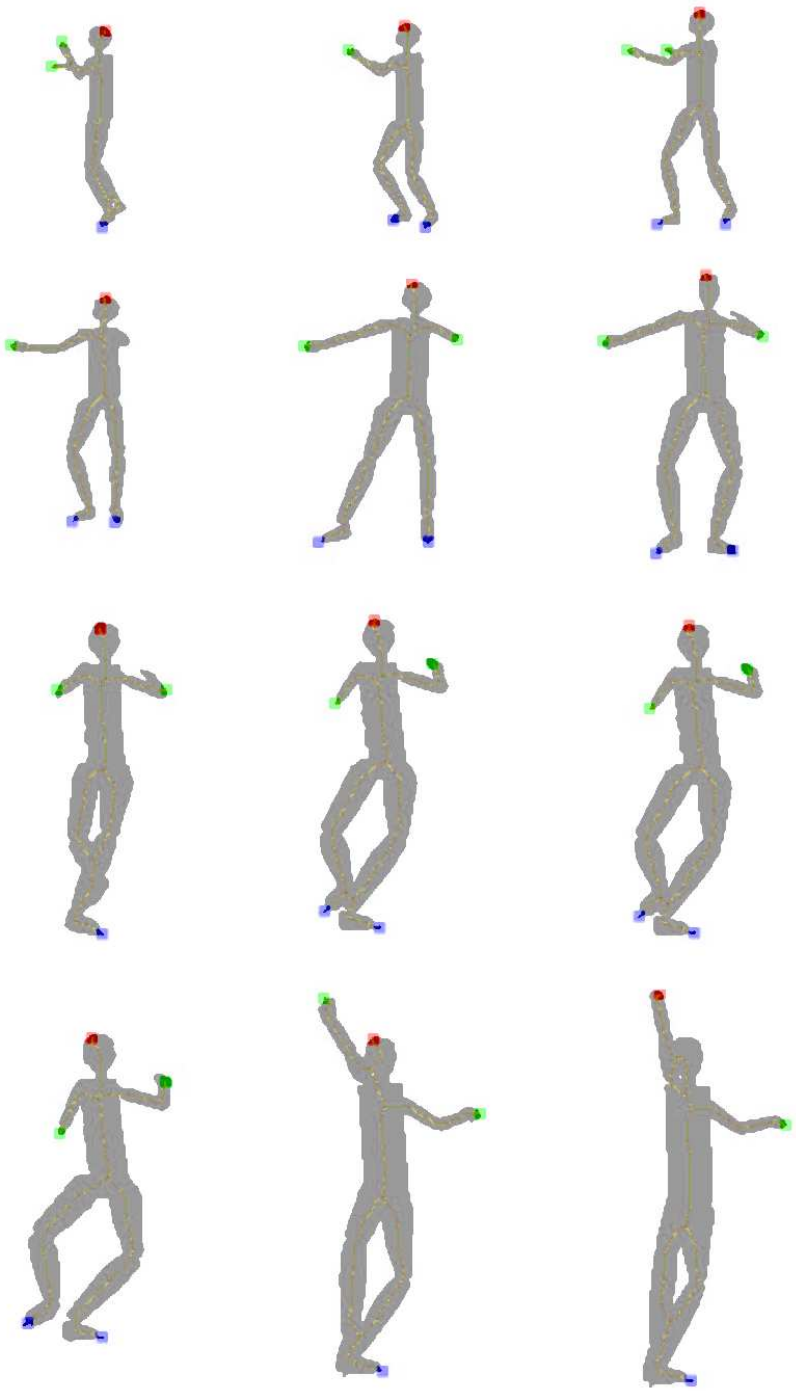


Figure 2.26: *'Ballet' sequence key frames. Total length: 192 frames. Part II.*

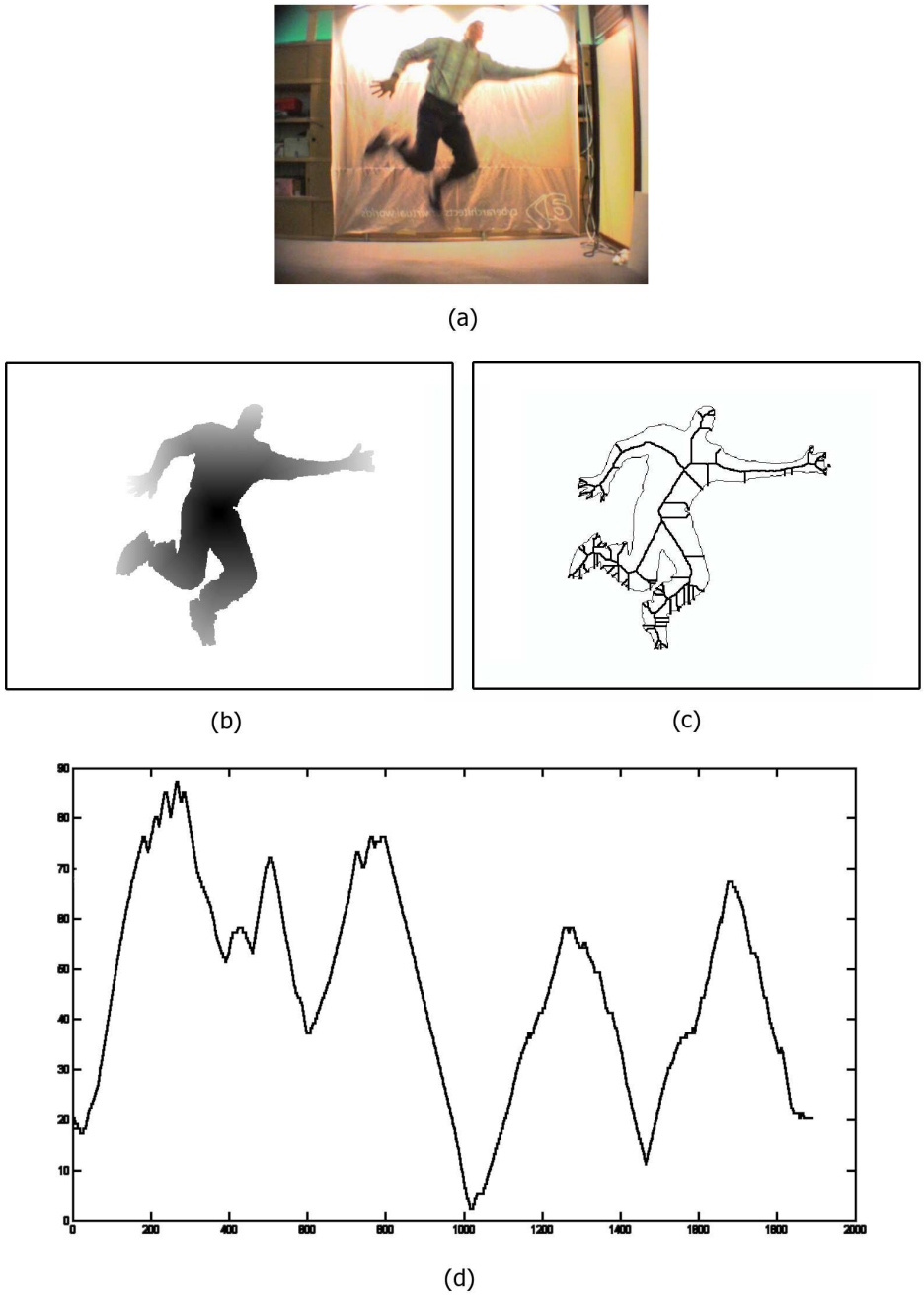


Figure 2.27: *Illustration of the multi-scale use of CP extraction (I). (a) displays the original frame, (b) the geodesic distance map, (c) the original morphological skeleton and (d) the original distance function. See Fig. 2.28 for a presentation of a multi-scale feature extraction.*

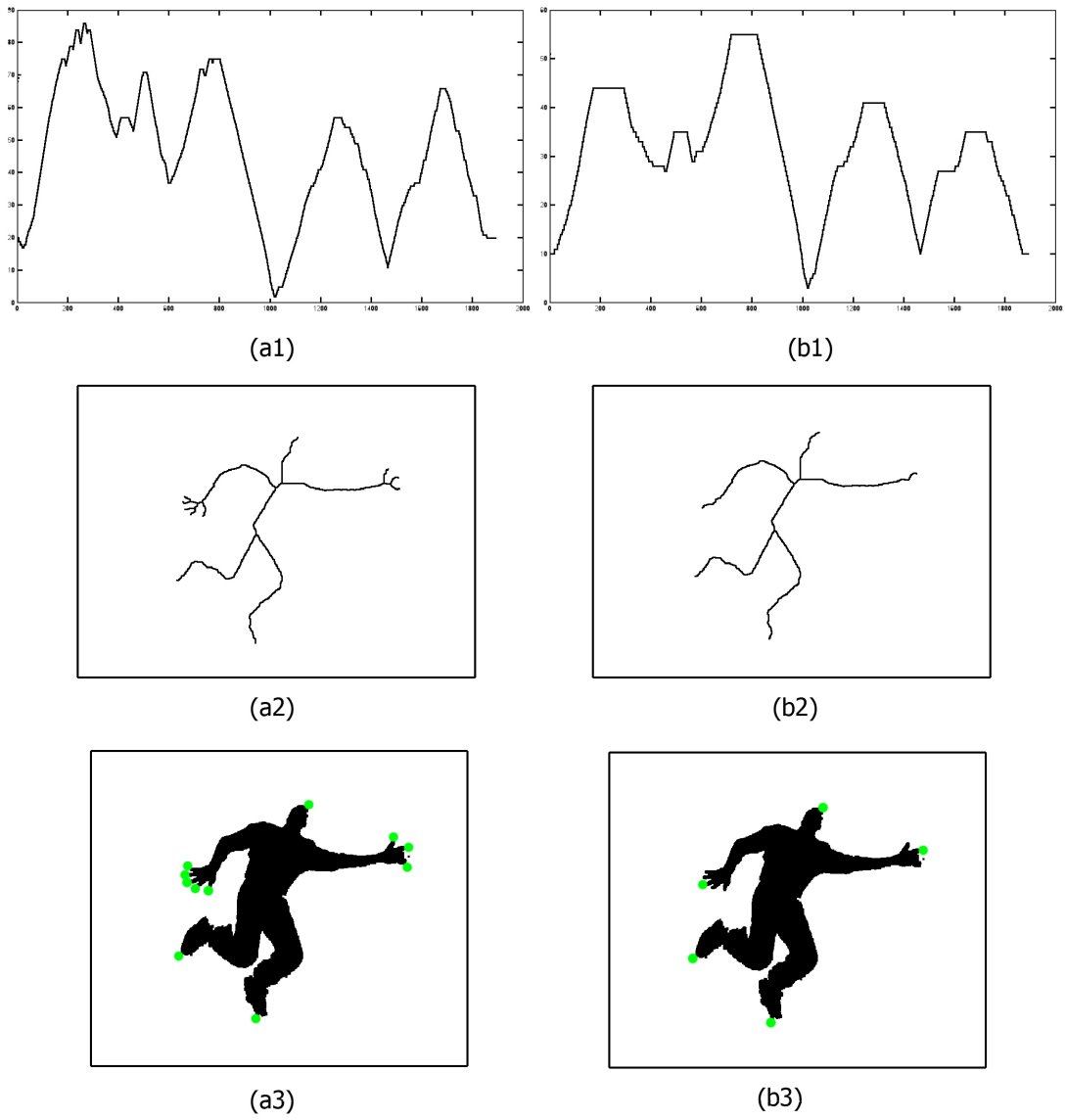


Figure 2.28: *Illustration of the multi-scale use of CP extraction (II). Column ai displays the results of a high-resolution feature extraction (and thus a low value of H). Column bi displays a lower-level CP extraction. Effectively extracted Crucial Points are depicted as discs ((a3) and (b3)).*

2.5 Conclusion

A novel method to analyze the human silhouette is presented. It allows the extraction of human prominent features using low-level standard image processing tools, among which the geodesic distance map is the most important one (Section 2.3.1). These tools allow to extract crucial human features without expensive material nor skin detection.

The standard geodesic distance map, presented as a succession of geodesic dilations, help us visualize and present the challenges, steps and results of the CP extraction algorithm. It is moreover at the core of the final approach. Nevertheless, we have seen that it has prohibitive computational complexity ($\mathcal{O}(N^3)$). A novel alternative, much less computationally complex, is presented in Section 2.3.2. This alternative, which is an adaptation of the Dijkstra's shortest path computation algorithm ([Dijkstra 1959]) in a 4-connected domain, is a particularly well suited alternative approach to the possible optimizations cited in Section 2.3.1. It allows a geodesic distance map creation in a single pass, hence having a computational complexity of $\mathcal{O}(N)$.

Either way, via the geodesic distance map computed with respect to the center of gravity of the user silhouette, we are able to extract Crucial Points that are visible in the silhouette with a 94% average accuracy rate in realistic segmentation situations (Section 2.3.4).

Also, morphological skeletons have been introduced (2.4.1). We have seen that they condense the morphological information of a whole silhouette in a few branches. Unfortunately, standard (and even faster optimized skeletonisation algorithms) always produce *noisy* skeletons. These morphological skeletons contain indeed numerous branches, where a branch is created for each single silhouette border irregularity, disregarding the intensity of this latter. To counter this problem, we have introduced the **morphological skeleton selective pruning** (Section 2.4.2). This algorithm takes the extracted CP and the standard morphological skeleton as input and produces clean skeletons, that only contain branches that connect to an actual CP. Thus these *robust* skeletons introduce a very convenient measurement space between crucial features. As presented in Section 2.4.3, this domain will allow both geodesic and Euclidean measurements, in or-

der to classify the extracted CP into head, hands or feet. This classification is achieved with an average accuracy rate of 95%. This average accuracy rate, combined with the 96% CP extraction accuracy rate, produce a **global extraction and labelling accuracy rate of 91%**.

Inter-Image Feature Labelling and Tracking

3

Chapter 2 has described a novel human feature extraction technique. It has presented a series of standard image processing tools used in new ways for silhouette analysis, in order to achieve real-time feature extraction. The goal of this second approach is strictly the same, yet being more focused in the stochastic domain (and the real-time applications). Sharing portions with the previous approach (the optimized CP extraction module), it introduces the benefits of working with inter-image temporal information, using a very computationally light Bayesian framework. Hence the result of this chapter, that actually represents one of the major contributions of this thesis, is a hybrid approach using both image processing tools and probabilistic tools.

3.1 Crucial Point Extraction

EVEN though the inter-image feature extraction and tracking algorithm chain described in this chapter introduces a completely novel framework, it shares the Crucial Point (CP) extraction module with the intra-image approach, as described in Section 1.6 and illustrated in Fig. 1.4. However, concerning the CP extraction step, it also introduces some subtle yet meaningful changes. Indeed, as it was described in Section 2.3.3, during the local maxima extraction from the $g(p)$ function, each local maxima i still present after the filtering step is selected and associated with a position in the given frame t: $z_t^{(i)} = (x, y)$, where x and y are respectively the horizontal and vertical image coordinates. However, since the range of extracted Crucial Points could only evolve

between 2 and 5, if more than 5 points were extracted, the $g(p)$ function H-thresholding needed to be iterated with a higher H value. The Inter-Image Crucial Point extraction algorithm works similarly, but the range of admissible CPs is not fixed a priori. The H-maxima operation is still used in order to smooth out segmentation imperfections and intrinsically delimit the CP range, yet all $g(p)$ local maxima are taken into account. Since in spite of the filtering process some of these points $z_t^{(i)}$ may be due to noise, in the sequel we call them *crucial point candidates*.

The output of the feature extraction phase is the list of local maxima extracted from function $g(p)$ and corresponding to the most prominent points of the silhouette. For a given frame t, each maximum i corresponds/refers to a pixel on the silhouette boundary (crucial point candidate) with coordinates $z_t^{(i)} = (x, y)$ and intensity $I_t^{(i)}$. In this chapter, this intensity information extracted during the crucial point candidates phase will be essential.

The algorithm classifies each pair $(z_t^{(i)}, I^{(i)})$ into one of the six classes: head, left foot, right foot, left hand, right hand and noise, denoted by $\Omega = \{h, lf, rf, lh, rh, n\}$. It works in two steps:

- Matching previously labelled crucial points with current crucial point candidates: Tracking Step (Section 3.2).
- Labelling new appearing crucial point candidates that were not visible or not detected in the previous frame: Detection Step (Section 3.3).

Therefore, the feature labelling is jointly performed by the tracking and the detection steps.

3.2 Tracking Step

The whole algorithm chain presented in Chapter 2 only involved intra-image information, hence the lack of feature tracking. It was designed in

order to be robust enough in intra-image only and thus able to work without a tracking step (furthermore using the redundant information set-up of two orthogonal cameras described in Section 4.1 for 3D). However, a tracking step for 2D was soon proven unavoidable in order to correct misclassifications and inversions, and globally provide a temporal coherence layer.

In this section, we describe how the temporal continuity of human motion allows to use the previous position of a given crucial point to aid the classification. We also describe how the a priori information on the human extremities natural location and movement can improve the tracking robustness. The use of this a priori information is necessary since a pure predictive tracking is not robust enough against fast and unpredictable movements as those followed by human extremities. The standard frame rates used by the applications we target, traditionally between 10 and 20 fps, can indeed induce frame to frame distances between CPs important enough to create highly disturbing ambiguities. With higher frame rates the a priori information would still be very helpful, yet nevertheless restricted to the detection mode only (Section 3.3) and not during tracking anymore.

We have classified points in six different classes $\Omega = \{h, lf, rf, lh, rh, n\}$. Class n denotes points that have been selected by mistake, that is, the *noise* class. Moreover, the extracted CP candidates in the current frame can also be grouped in points that have a corresponding candidate in the previous frame, and those that have not (this is, points that have appeared in the current frame).

Suppose that part of the crucial points were correctly labelled in the previous frame. We denote these points by $z_{t-1}^{(j)}$ and the associated class labels by $\omega_\alpha^{(j)}$. A candidate $z_t^{(i)}$ in the current frame is then classified using a Maximum a Posteriori (MAP) rule. We compute the probability $P(\omega_\alpha|z_t, z_{t-1})$ for each pair $(z_t^{(i)}, z_{t-1}^{(j)})$ and assign to the point the class that has maximum probability, i.e.

$$\omega^* = argmax_\omega P(\omega_\alpha|z_t, z_{t-1}). \tag{3.1}$$

Since part of the crucial points were already labelled, we classify the candidate points in the remaining possible classes. Using Bayes law, the a

posteriori probability can be written as a product of three factors, i.e.

$$P(\omega_\alpha|z_t, z_{t-1}) = \frac{p(z_t, z_{t-1}|\omega_\alpha)P(\omega_\alpha)}{p(z_t, z_{t-1})} \tag{3.2}$$

$$\propto p(z_t|\omega_\alpha)p(z_t|z_{t-1}, \omega_\alpha)P(\omega_\alpha), \tag{3.3}$$

the normalizing factor $p(z_t, z_{t-1})$ in Equation (3.2) can be discarded since it does not influence the classification.

Let us now detail the influence of the three factors in Equation (3.3).

- The first factor $p(z_t|\omega_\alpha)$ represents all the prior knowledge available on the presence of a crucial point (of class ω_α) at the particular location z_t. This term is computed using a prior probability map, described in Section 3.3.1.
- We model the crucial points kinematics, represented by the second factor, as a Gauss-Markov random sequence or random walk, i.e.

  $$p(z_t|z_{t-1}, \omega_\alpha) = \mathcal{N}(z_t; z_{t-1}, S_\alpha) \tag{3.4}$$

  where the covariance S_α is chosen diagonal because, for each class, motion in the x axis is independent from motion in the y axis. Also, S_α depends of the ω_α class, since the cinematics of each class are different from one another.

  Note that instead of using the crucial point position in the previous frame as the mean of the Gaussian, we could use a predicted position $\hat{z}_t$ using a first order dynamic system. However, in our experiments, we have observed that human motion does not follow a constant velocity model, which makes the random walk more adequate in this case.
- As for the third factor $P(\omega_\alpha)$, it reflects the prior knowledge on class ω_α. In our case for example, the classes head and feet have a higher $P(\omega_\alpha)$ than hands since hands are much more often occluded than head and feet. Moreover, they are likely to have more irregular and fast movement patterns. This a priori knowledge directly translates in the way the algorithm is computed. This aspect is commented in section 3.2.2.

3.2.1 Mahalanobis Distance and Gating

As it will be detailed in the next section, we have chosen a sequential approach in order to tackle the crucial point candidates classification problem. Let us introduce firstly the Mahalanobis distance and the gating approach it introduces, two concepts that have proved to be useful in this sequential context.

In order to increase the speed and robustness of the tracking, we use a validation volume, or *gating* approach. This way, measurements with low probabilities of assignment are excluded from the classification, thereby reducing the combinatorics of the corresponding problem.

This approach uses the Mahalanobis distance in order to define a volume around a CP at time $(t-1)$. This volume defines the area where we allow that same CP to be at time t. Or in other words, a CP candidate with coordinates $z_t^{(i)}$ at time t can be assigned to $z_{t-1}^{(j)}$ only if $z_t^{(i)}$ falls into the validation volume.

As described in [Cox 1993], the Mahalanobis distance is obtained by looking for contours of constant probability in a Gaussian density distribution. Indeed, being a Gaussian distribution given by

$$\begin{aligned}\mathcal{N}[z_t] &\triangleq \mathcal{N}[z_t; z_{t-1}, S] \\ &= (2\pi)^{-d/2}|S|^{-1/2} \\ &\times exp\left(-\frac{1}{2}\{[z_t^{(i)} - z_{t-1}^{(j)}]^T \cdot S^{-1}[z_t^{(i)} - z_{t-1}^{(j)}] \ \}\right)\end{aligned} \tag{3.5}$$

where S represents the covariance of the Gaussian distribution and d is the dimension of the measurement vector z_t (in our case $d = 2$).

Contours of constant probability density are thus given by

$$\left(z_t^{(i)} - z_{t-1}^{(j)}\right)^T S^{-1}\left(z_t^{(i)} - z_{t-1}^{(j)}\right) = \gamma \tag{3.6}$$

The left side of Equation (3.6) is called the Mahalanobis distance. The right side of the equation is a constant value.

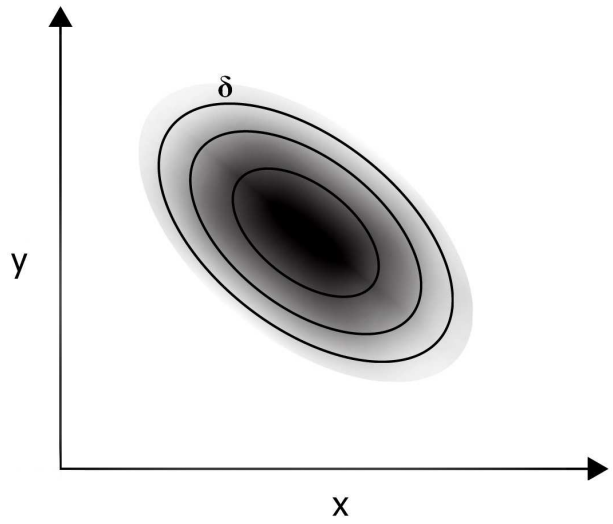


Figure 3.1: *Contours of equal probability in a two dimensional Gaussian probability distribution. Validation volume V is defined inside threshold δ. Darker shades correspond to the more probable areas to find a measurement.*

Hence, a *validation volume* V can be defined by

$$V(\gamma) = \{z : [z_t - z_{t-1}]^T S^{-1} [z_t - z_{t-1}] \leq \gamma\} \tag{3.7}$$

It can be shown that the Mahalanobis distance is chi-squared distributed. The probability that the distance is inferior to the parameter γ can therefore be obtained from χ^2 distribution tables. For example, if we want to establish a validation volume with a 95% probability of finding the measurement, then $\gamma = 5.99$. Hence, if a measurement fails the inequality test of Equation (3.7) with $\gamma = 5.99$, there is a 5% or less probability that it is associated with the geometric feature.

In our approach, we have also used a gating technique by creating a *validation volume* V. Following the same reasoning as introduced for the Mahalanobis distance, yet keeping the Bayesian formalism, we impose a gating by choosing a threshold δ (Fig. 3.1). Nevertheless, as in our case we perform tracking and classification at the same time, we define the validation volume using both the a priori probability and the random walk Gaussian

density. Note that from this point the volume is not an ellipsoid anymore. However, this fact does not affect the algorithm, since we do not need to compute the whole volume but only to compare the value of

$$p(z_t|\omega_\alpha)p(z_t|z_{t-1},\omega_\alpha) \tag{3.8}$$

to the threshold δ. If the probability of a Crucial Point candidate computed with respect to a certain class ω_α returns

$$p(z_t|\omega_\alpha)p(z_t|z_{t-1},\omega_\alpha) \leq \delta \tag{3.9}$$

then, it implies (argument in Eq. (3.1))

$$P(\omega_\alpha|z_t,z_{t-1}) = 0 \tag{3.10}$$

and thus that particular Crucial Point candidate can not be classified in the ω_α class.

3.2.2 Sequencing versus Global Classification Approach

Concerning the classification problem, we have favored a sequential approach to tackle it (details about its implementation are presented in Section 3.4). We have seen that for each class, only CPs that are inside a certain probability region are taken into account for classification. Moreover, Crucial Points that are already labelled in a certain class are also removed from the CP candidates set for the rest of the classification process.

Another approach would have been to take into account all available CP candidates for each class. This way a more global classification would have been accomplished since in the sequential approach each labelling influences the remaining classifications by sequentially removing possible candidates.

We have nevertheless chosen the sequential approach because combined with the gating and the particular context of our problem it performs faster and with at least as much robustness as the global approach.

As we have already commented when introducing factor $P(\omega_\alpha)$ of Equation 3.2, for the kind of applications we focus on, good a priori information is indeed available concerning classes behavior. In our case for instance, it is known that head and feet (in that order) are likely to have a better a priori reliability than hands. As it will be shown in the performance quantification tables of Section 3.5, this hypothesis is verified.

Hence, by performing the classification process following a decreasing *reliability* order, robustness is increased. At each classification step a class more reliable than the next is indeed being processed, while at the same time combinatorics are decreased for the following -less reliable- classes. Moreover, classification is performed faster, as only one pass is needed for each class.

3.3 Detection Step

In the previous section we described how we classify crucial candidates in points that have a corresponding candidate in the previous frame, and those that have not. In this section we focus in the latter case. In the detection step, we look for new crucial points, if any, that were occluded or not detected before. Since human features are unique, the class labels which have already been assigned to crucial point candidates in the tracking step are removed from the list of possible class labels Ω. Therefore the remaining candidate points, i.e. those that were not labelled in the tracking step, have to be classified into the remaining classes. Nevertheless, a point that has no candidate in the previous frame is very likely to be either a new crucial feature that has just appeared or noise. In order to discriminate noisy errors from legitimate features we apply the following rationale: the more prominent the point, the more likely to be a crucial feature. In this case, the prominence information is extracted from the intensity value $I_t^{(i)}$, computed for each crucial candidate from the $g(p)$ function (Section 2.3.3).

Since no information is available from the previous frame, the class label is determined with the same MAP technique, using the prior probability map and the intensity of the crucial point candidates. This leads to the following class probability to maximize

$$P(\omega_\alpha|I_t, z_t) \propto p(z_t|\omega_\alpha)P(\omega_\alpha)p(I_t|\omega_\alpha) \tag{3.11}$$

where $p(I_t|\omega_\alpha)$ is modelled by a uniform density with different parameters for noise and crucial point classes:

- The choice of the parameters m_a and m_b is done so that CP generate large values of I, while noise generates any values of I inside the admissible range. Only CP candidates with higher intensities than m_a will be taken into consideration for classification (Eq. 3.13). As for m_b parameter, it is not an important one, since only m_a is needed for the sake of classification. It can thus be set to a very high value, keeping in mind that it is limited to half the silhouette height in geodesic distance (i.e. the highest possible intensity value for a given silhouette would be the intensity of the local maxima representing either the hand or the feet in a 2 CP configuration).
- For simplicity, we choose the same probability density function for all classes (see Figure 3.2), i.e.

  $$p(I_t|h) = p(I_t|lf) = ... = p(I_t|rh) \tag{3.12}$$

  We have favored this hypothesis since due to self-occlusions, intensities can very much vary, and these parameters would thus not be not robust enough as a priori classification parameters. Furthermore, different probability density functions would also have implied two new m_{ai}, m_{bi} parameters for each class.

Note that, for the first frame in the process, the Bayesian approach only performs the detection step since there are no previous references to crucial points. In contrast to many other tracking approaches, this algorithm performs detection at each time step. As a result, it does not need the person to adopt a specific posture at the beginning of the image sequence. For the first frames of a sequence, the system simply works in pure detection mode until crucial points are found (Fig 3.3).

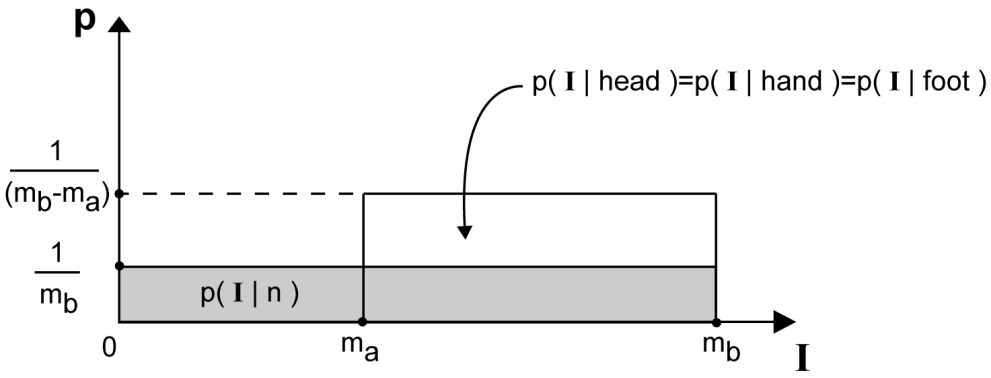


Figure 3.2: *Uniform density probabilities for $p(I_t|\omega_\alpha)$ in detection mode.*

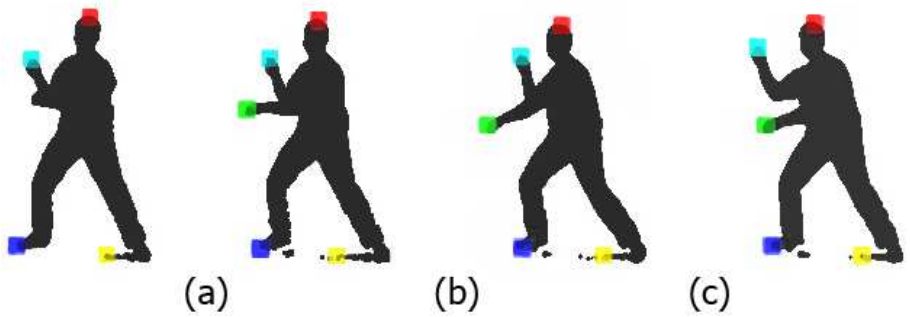


Figure 3.3: *Detection and tracking modes. Transition (a) illustrates a detection mode concerning CP 'left hand' (in green). Transitions (b) and (c) illustrate a tracking mode for the same CP.*

3.3.1 Prior probability maps

In this section we describe the probability maps used for computing the factor $p(z_t|\omega_\alpha)$ in equations (3.3) and (3.11). The map reflects our prior knowledge about possible location z_t of the crucial point in the ω_α class. For each frame and for each label, a generic map is adapted to the observed user position using a limited number of measurements extracted from this silhouette. These measurements, i.e. the silhouette height, its center of gravity and torso angle, are sufficient to construct a simple human model inspired by Da Vinci's *vitruvian* model. We thus apply the canonical proportions depicted on Fig. 3.4. This simple model allows the adaptation of each probability map to the silhouette position and user morphology on each frame. Figure 3.6 reflects this adaptation, depicting two particular frames in the same sequence, and how three prior probability maps adapt around the silhouette (the corresponding left hand and foot maps have been omitted for the sake of clarity, since the maps associated to the hand and feet pairs are symmetrical). Also, maps have been rendered in a translucent manner for illustration, however they do not actually ever interfere with one another since each prior probability is computed independently.

In practice, each generic map is defined using a 2D Gaussian distribution, positioned with respect to the CoG using our *vitruvian* model. We have chosen to work with Gaussian distributions since their covariance can easily be adapted to each different class. As it will be presented and illustrated hereafter, each Gaussian is then truncated in order to conform to each class points possible positions and limits inside the class.

The construction parameters are:

- The maximal possible head height. This parameter is quite straightforward to extract, by simply noting the highest silhouette pixel. We have nevertheless introduced a small variation that has proven to improve quite importantly the algorithm robustness. Before the extraction of the silhouette highest point we accomplish an erosion with the same structuring element employed in order to extract the torso angle, but with half its size. This way the arms disappear from the silhouette and we can extract the actual overall silhouette height,

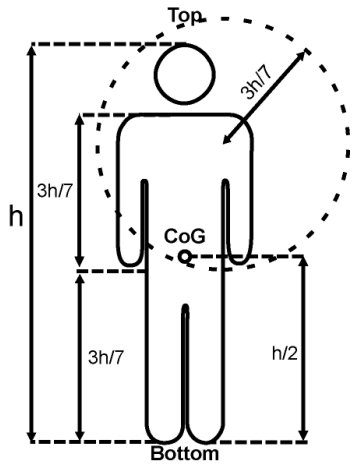


Figure 3.4: *Coarse vitruvian model. A dashed circle illustrates the left arm region.*

even when the user is raising his/her arms above the head. This fact has proven to have a positive impact in the detection phase, since the top of the silhouette will also be used for imposing the upper limit of the head map. This way, a hand placed above the head (and thus above the 'top' value of the silhouette) will not be labelled as a head, since its prior probability will be set to zero. Figure 3.9 in Section 3.5 depicts a sequence that illustrates a situation in which this head a priori probability map avoids a Head/Hand inversion. See also Figures 3.6 and 3.5(a) for other illustrations of head prior probability maps.

- The angle of the torso. It is used in order to place the head prior probability map with respect to the CoG. Its computation has been detailed in Section 2.4.3.

- The Center of Gravity. It is an essential parameter for the prior maps positioning, since it is taken as half the total height. It is, for example used to set the upper limit of the feet prior map, or the overall arm length (Fig. 3.4). Its computation has been detailed in Section 2.3.1.

- The arm and leg lengths. As depicted in Figure 3.4, the arm length is approximated by $\frac{3h}{7}$, h being the total silhouette height. The leg length is approximated by $\frac{h}{2}$.

More precisely, for the head prior map, depicted in Figure 3.5 (a), a 2D Gaussian is centered at the intersection between the angle of the torso in that frame, and the maximal head height limit. The Gaussian is then truncated in order to only allow spacial values that are contained inside an angle range area around the angle of the torso. This gives the map a 'pie slice' look.

Concerning the hands and feet classes, the same approach is followed: a Gaussian is centered on the most probable class point position with respect to the CoG. These most probable positions are fixed a priori according to the specific applications. Even if these positions are fixed a priori, note that, as described previously, these positions are expressed with respect to parameters that are extracted from the silhouette at each frame, and thus they adapt to the user's silhouette morphology constantly. These points, that will be denoted μ_α in the sequel, are thus updated at each frame.

These μ_α could have been additionally influenced by previous positions of the different classified CPs. We could indeed have chosen to update the most probable position for each class with each new classification. In order to compare the impact of this approach, we carried several tests where each μ_α was updated using the information on the position of the previous detected Crucial Points in each class. We experienced that whether the average position is fixed by an a priori information about the context and the application or updated at each frame by a learning procedure did not influence much the results, this latter requiring however more computational time.

Each Gaussian is then truncated. Concerning the feet, the acceptable area for the class is limited by the length of the legs, or in our case by its approximation: CoG_{down} (Fig. 3.5 (b)). Equivalently, the hand class acceptable area is limited by the arm length approximation ($\frac{3}{7}$ of the total silhouette height), as depicted in Figure 3.7.

For example, concerning the Head class probability map, we have

$$p(z_t|\omega_\alpha) \propto \begin{cases} e^{-\frac{1}{2}(z_t-\mu_\alpha)^T \cdot S_\alpha^{-1} \cdot (z_t-\mu_\alpha)} & \text{if } \theta \geq (\theta_T - \frac{\phi}{2}) \\ & \text{and } \theta \leq (\theta_T + \frac{\phi}{2}) \\ & \text{and } r \leq CoG_{up} \\ 0 & \text{otherwise} \end{cases}$$

where we have noted θ_T as being the angle of the torso, θ as the angle of the CP candidate, ϕ as the overall acceptable angle range (typically it is chosen $= \frac{\pi}{8}$), r as the distance from the CP candidate to the CoG, CoG_{up} as the distance from the CoG to the head height limit.

Figures 3.7 and 3.5 show all prior probability maps used for a biped standing application. Figure 3.7 depicts the right hand rh map adaptation for a particular frame. Again, only one map is depicted for each pair of hands and feet.

Note that the map can be easily adapted for analyzing different motion types. This enables a fast generalization of the method to other applications. This is illustrated in Section 3.5, where the technique is adapted to a wheelchair user and also a tennis player.

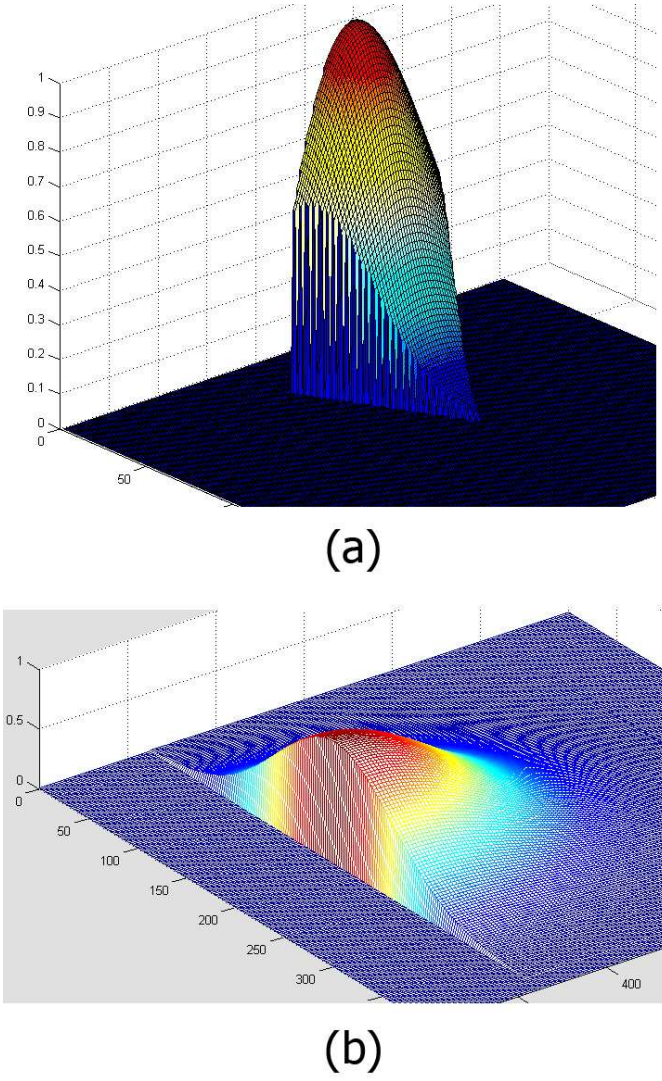


Figure 3.5: *Prior probability maps, corresponding to the head class (a), and the right foot class (b), respectively. In (a), the 'pie slice' look translates the angle range around the angle of the torso where the head is expected to be found. In (b), the lower clean cut corresponds to the lowest point of the silhouette, below which no Crucial Point candidate can be found.*

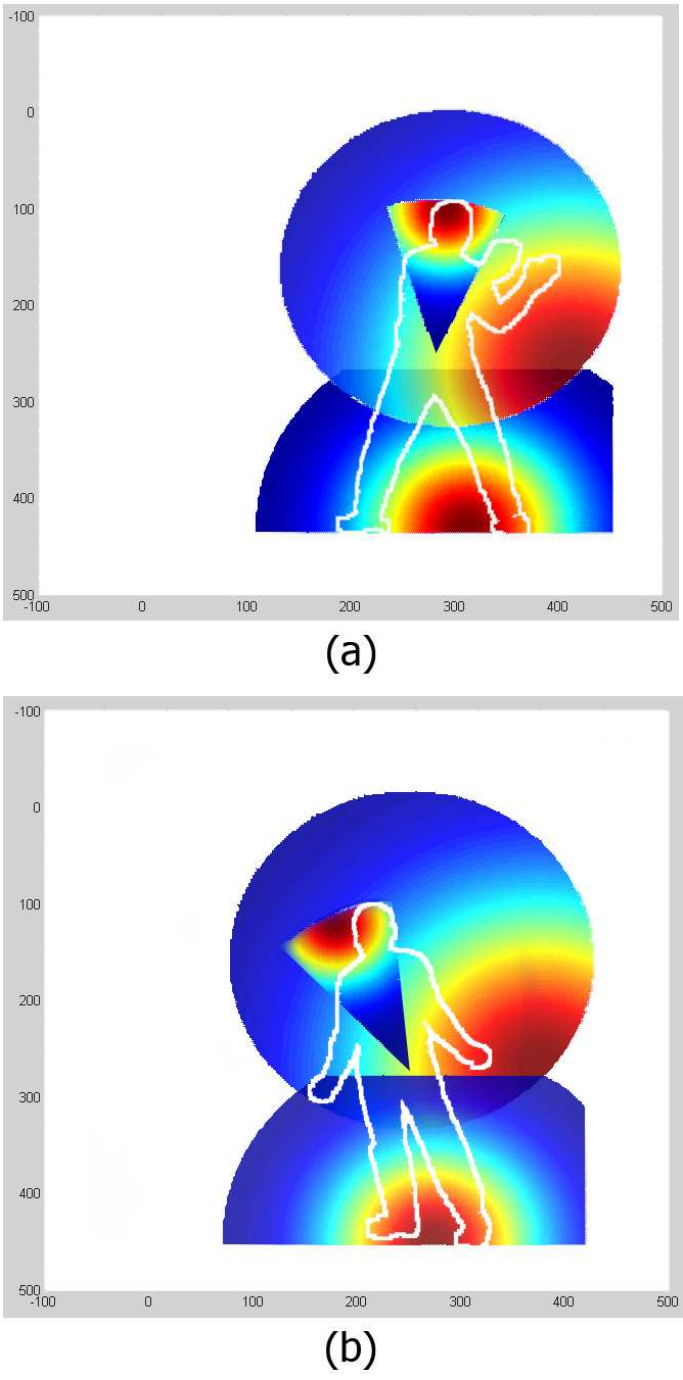
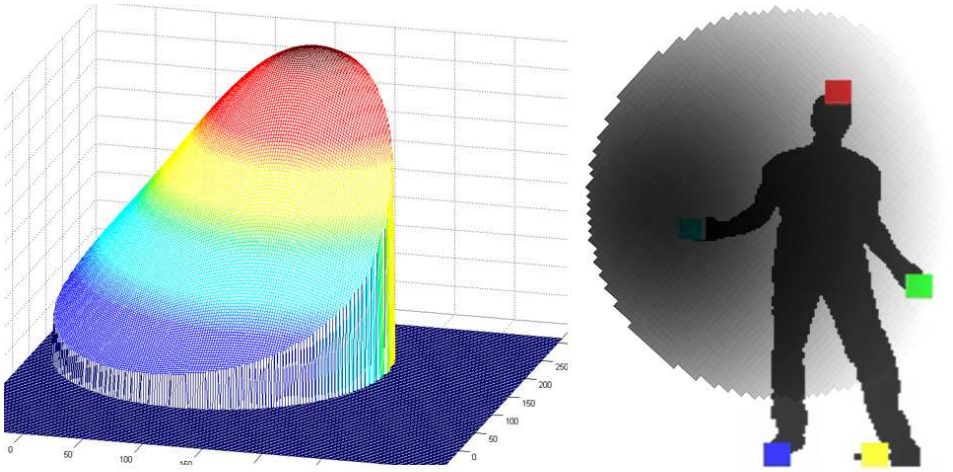


Figure 3.6: *Prior probability map adaptation around two particular positions in the same sequence. Head, left hand and left foot maps are depicted, around the user particular positions on each frame. Warmer colors correspond to higher probabilities.*



(a) (b)

Figure 3.7: *Prior probability map for crucial point label $\omega_\alpha = rh$ (right hand) (a). Application to a particular frame (b). Increasing probabilities are illustrated with darker shades of gray.*

3.4 Implementation

In this section we describe some aspects of the algorithm implementation.

We have seen in Section 3.2.1 that the gating introduced in the **tracking step** is actually performed on the tracking factor $p(z_t|z_{t-1}, \omega_\alpha)$ and on the a priori information factor $p(z_t|\omega_\alpha)$ at the same time, by multiplying both probabilities before comparing the result to threshold that defines the validation volume V. Moreover, in order to reduce the possible classes matching combinatorics and also to avoid the cases where one of the two factors would have an overall biassing effect on the product of Equ. 3.9, the gating is performed in practice in two steps with different thresholds.

- The first one, that we call the *hard threshold* , has a very selective validation volume, leading to only match candidates that have a high average tracking / a priori probability (Eq. 3.9). This way, only very reliable candidate-class matches are classified at first, highly reduc-

ing the combinatorics for the second gating.

- The second gating, uses a *softer* threshold,and aims at classifying those candidates that were not matched in the previous gating, even though they have an important intensity I_t and an average overall probability. This second step allows the classification of candidates that would have otherwise been classified as noise due to a very low value of only one of the two factors taken into account in Eq. (3.9). This can indeed happen when a feature has either a very low $p(z_t|z_{t-1}, \omega_\alpha)$ factor because it was performing a very fast movement, or either a very low $p(z_t|\omega_\alpha)$ factor because it was located in a very low probability area for that class (typically the right hand found somewhere quite far on the left side of the body, for example).

Moreover, concerning the Gaussian distributions, neither the Gaussian used in the Gauss-Markov random walk of Eq. 3.4 nor the Gaussian distributions used to construct the a priori probability maps are normalized, for three main reasons. Firstly, because since we are always comparing different classes values, normalized measures are not necessary. Also, this approach gives us the possibility of weighting differently one of the two parameters of Equation 3.8. This way we could favor either the temporal information (the feature tracking), or the known a priori information (via the a priori probability maps). In our case we have always obtained good results giving the same weight to both parameters, yet in some particular (or ill-conceived) applications, giving priority to one of the two aspects could prove to be useful. Finally, it is faster.

Concerning the **detection step**, let us go back to Equation (3.11):

$$P(\omega_\alpha|I_t, z_t) \propto p(z_t|\omega_\alpha)P(\omega_\alpha)p(I_t|\omega_\alpha)$$

This probability has to be maximized in order to classify a new appearing CP candidate, only taking into account its intensity $p(I_t|\omega_\alpha)$ and its a priori location probability $p(z_t|\omega_\alpha)$.

Firstly a gating is also performed via the intensity threshold m_a value (Fig. 3.2). Every CP candidate that would not satisfy the condition:

$$I_t < m_a \tag{3.13}$$

would not be taken into consideration for the detection step.

Also, if we rewrite Equation (3.11) during the classification of crucial candidate cp into either one of the cp classes $\in \Omega$ or the noise (n) class we have:

$$P(n)p(z_t|n)p(I_t|n) \underset{cp}{\overset{n}{\gtrless}} P(cp)p(z_t|cp)p(I_t|cp) \tag{3.14}$$

which can be rewritten as

$$\frac{P(n)p(z_t|n)p(I_t|n)}{P(cp)p(I_t|cp)} \underset{cp}{\overset{n}{\gtrless}} p(z_t|cp) \tag{3.15}$$

In this last equation, $p(I_t|cp)$ and $p(I_t|n)$ are constant in the range of I_t ($m_a \leq I_t \leq m_b$) (see Fig. 3.2), and so do $P(n)$ and $P(cp)$. Besides, by hypothesis $p(z_t|n)$ has a constant value. The left hand side of Equ. 3.15 can therefore be replaced by a parameter δ_2:

$$\delta_2 = \frac{p(z_t|n)P(n)p(I_t|n)}{P(cp)p(I_t|cp)} \tag{3.16}$$

Hence in practice a detection threshold δ_2 is fixed.

$$p(z_t|cp) \leq \delta_2 \tag{3.17}$$

Finally, note that for each frame only five exponentials are computed (one for each class map), and not the whole truncated Gaussian functions. This inter-image phase has thus a very low computational complexity, and in any case lower than the intra-image CP extraction.

3.5 Results

For performance quantification purposes we have analyzed many different sequences with various kinds of movements and segmentations. In this section we present four different sequences that help illustrate the average error rates of the method, as well as its limits. Firstly we present two synthetic -and thus perfectly segmented- sequences. Also a longer real sequence, that presents very general types of postures, which quantification results are compared with the results of the same sequence using a perfect segmentation, performed manually. This allows to separate pure algorithm errors from errors induced by the segmentation. Finally, a sequence of a wheelchair user illustrates one of the possible extensions of the method.

In order to quantify the error rate of each different label we assume that the region of the hand extends from the tip of the fingers to the wrist, the feet from the ankle to the tip of the toes, and the head region corresponds to the skullcap. An error is counted each time a label is detected outside these regions, or not detected although it is distinguishable by visual inspection. In Fig. 3.12 for example, the right hand is not detected in the third frame, yet being slightly visible: it is thus counted as a missed detection error. We thus divide the possible errors in two different sub-classes: erroneous label and missed detection. These errors are reported in the following tables. The sum of missed detection and label errors divided by the total number of frames gives the global average error rate.

Note that all these sequences (original and processed ones) are available at:
`http://www.tele.ucl.ac.be/~pedro/gestural/`

3.5.1 Synthetic results

These first two sequences, *dodge* and *barbecue*, are 182 and 306 frames long respectively. The first one depicts a virtual person dodging bullets by performing fast and athletic movements. The other one presents the same character trying to light a barbecue with gasoline and setting fire to its clothes, with the subsequent abrupt movements.

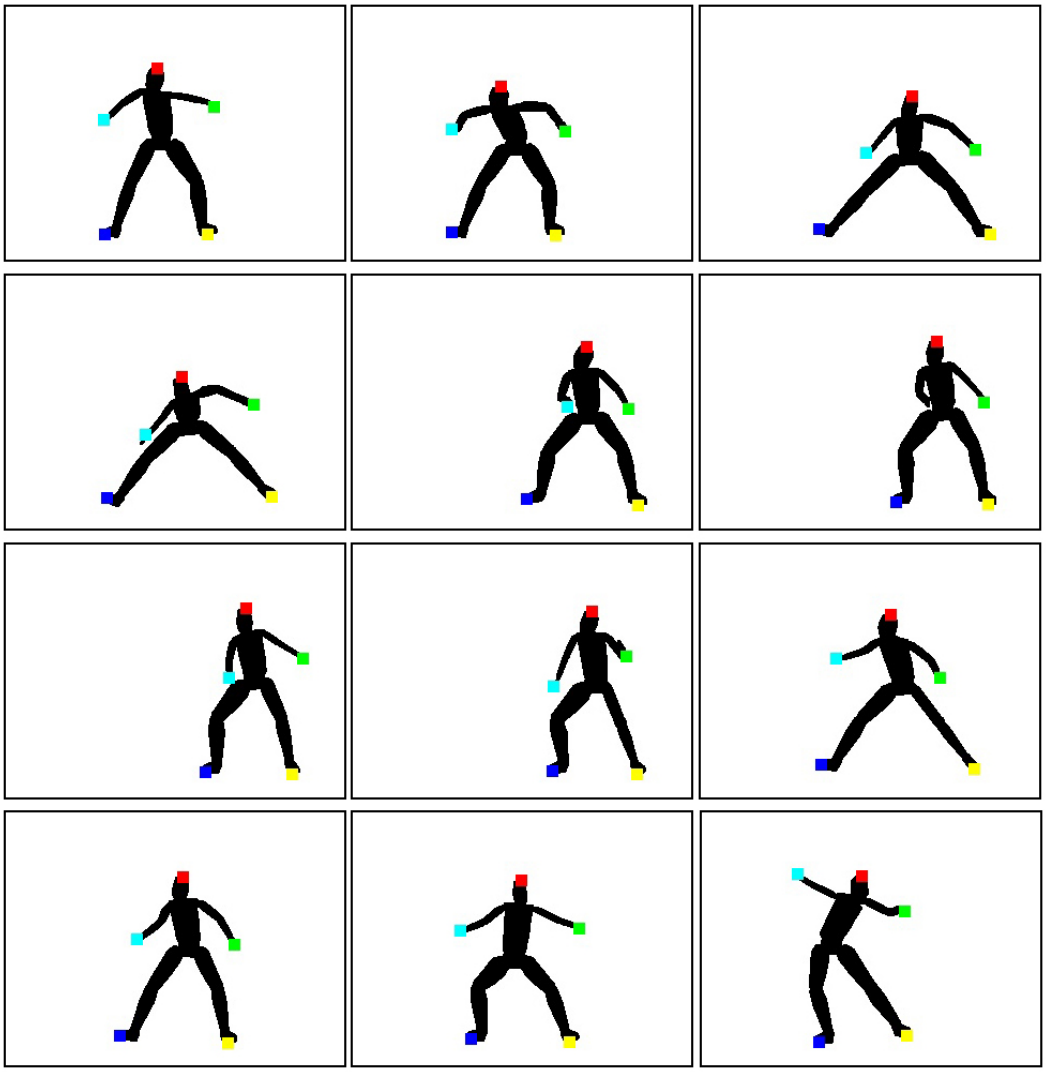


Figure 3.8: *'Dodge' sequence key frames. Total length: 182 frames. Part I.*

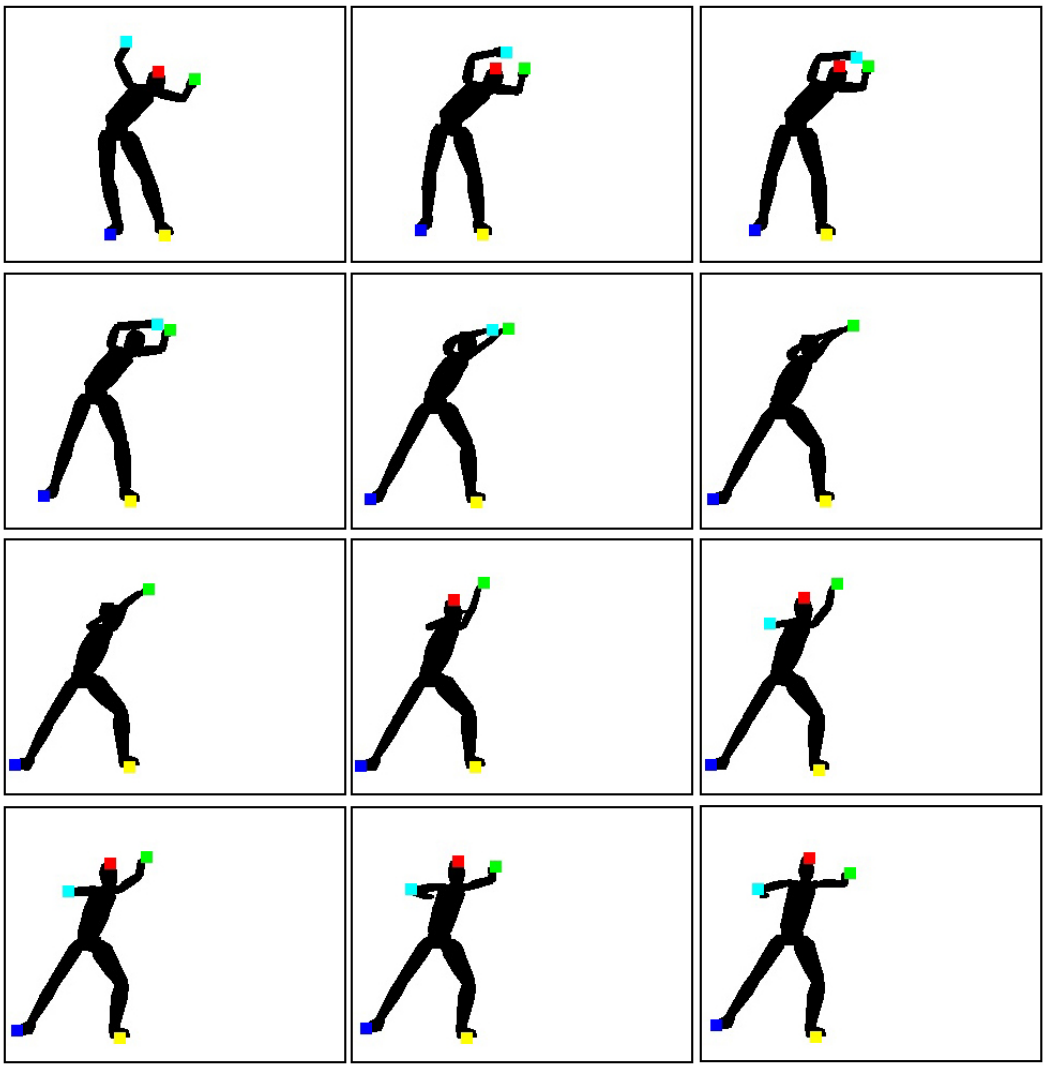


Figure 3.9: *'Dodge' sequence key frames. Total length: 182 frames. Part II.*

	Left Hand	Right Hand	Head	Left Foot	Right Foot
Label Error	5	5	9	0	0
Missed Detect.	3	3	1	0	0
Error Rate (%)	4.4	4.4	5.5	0	0

Table 3.1: *Error rates, sequence dodge. Average Error Rate:* ***2.85%***

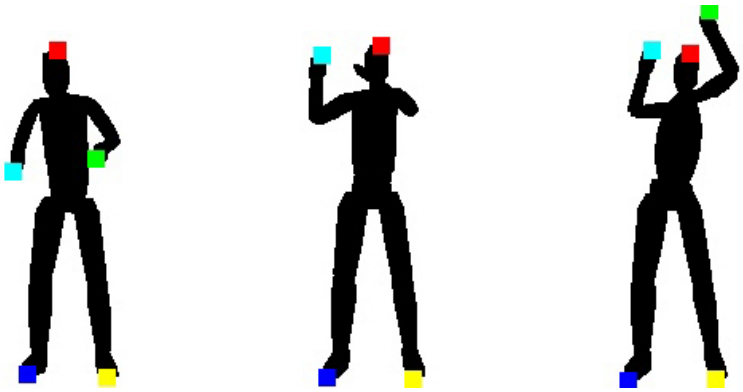


Figure 3.10: *Barbecue sequence sample. Total length: 306 frames*

Error rates are overall quite low. However, the barbecue error rate table shows a much higher hand labelling error than normal. This can be explained by one of the algorithm limitations, which is to be unable to discriminate between a hand and an elbow in a frontal bent arm posture (Fig. 3.11). This configuration is particularly present in this sequence, resulting in a unusually high error rate.

Finally, we can at this point also verify the hypothesis introduced via the factor $P(\omega_\alpha)$ in Section 3.2. This factor has been presented as the representation of the *a priori* information available on the different classes. In our case, this information is namely that Crucial Points belonging to the head class are more *reliable* than those belonging to the feet class, and these latter more reliable than those of the hands class. In practice this assumption lead us to the choice of a sequential treatment of the classification problem. At this point, it is important to note that Tables 3.1 and 3.3 verify this hypothesis. Average error rates are indeed the lowest for the head class, and

	Left Hand	Right Hand	Head	Left Foot	Right Foot
Label Error	44	16	1	0	0
Missed Detect.	2	6	1	0	0
Error Rate (%)	15	7.1	0.6	0	0

Table 3.2: *Error rates, sequence barbecue. Average Error Rate:* ***4.57%***

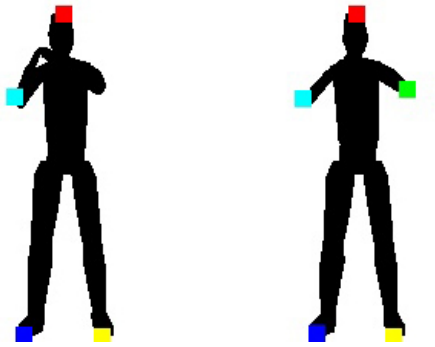


Figure 3.11: *Hand/Elbow misclassification error.*

the highest for the hands classes. Although all following quantification tables verify it as well, it is particularly visible in these two first sequences where results are independent of segmentation noise.

3.5.2 Real segmentation results

General movement range

The test sequence is 380 frames long and displays a vast scope of human gestures: jumps, walks, hand gesticulation in all directions and hand and feet inversions and occlusions (see Figure 3.12. It also represents a good sample of realistic working conditions.

The average error rate has a value of **5.5%** (**1.9%** with perfect segmentation). Table 3.3 shows also error rates obtained using the same sequence

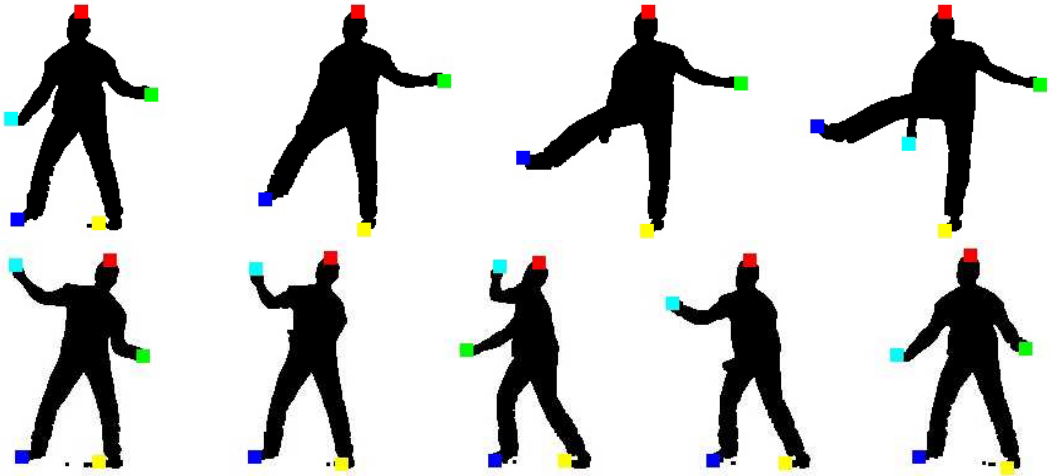


Figure 3.12: *Typical results on the sequence with real segmentation. On the third frame of the first sequence sample, the hand is already visible but not yet detected by the algorithm.*

Figure 3.13: *Segmentation errors due to residual shadows lead to small foot location bias.*

	Left Hand	Right Hand	Head	Left Foot	Right Foot
Automatic Segmentation					
Label Error	4	6	1	47	22
Missed Detect.	8	4	3	2	4
Error Rate (%)	3.3	2.7	1.0	13.3	7.0
Manually Corrected Segmentation					
Label Error	4	6	1	1	2
Missed Detect.	8	4	3	2	4
Error Rate (%)	3.3	2.7	1.0	0.8	1.6

Table 3.3: *Error rates with general sequence. Average Error Rate:* ***5.5%*** *(****1.96%*** *with perfect segmentation)*

but manually segmented. The higher error rates found during the feet tracking using automatic segmentation are due to residual shadows segmented below the user. In those cases feet are labelled very near to the exact foot location, but still outside the region we defined previously (see Fig. 3.13). This also explains the dramatic drop in error rates concerning the feet when comparing real segmentation to perfect segmentation. Note that the errors are more severe for the left foot since due to scene illumination it creates stronger cast shadows in this sequence.

Possible application: Virtual aerobic home training

The second test sequence is 706 frames long and displays a first possible application of real-time cheap markerless motion capture algorithms: virtual aerobic home training (see figure 3.14).

The average error rate has a value of **5.86%** . Table 3.4 shows the detailed error quantification.

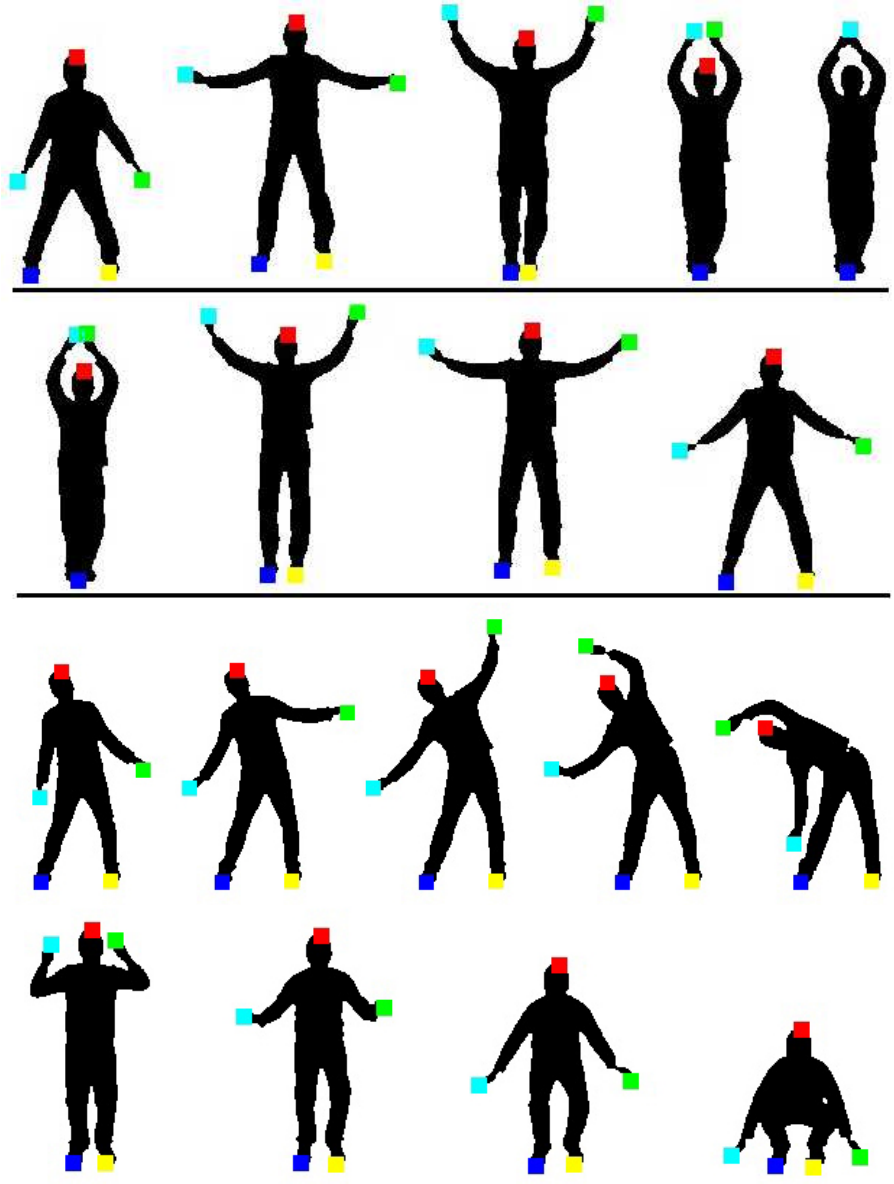


Figure 3.14: *Representative keyframes on the Virtual Aerobic sequence. A black line representing the floor level has been depicted in the first two rows in order to illustrate the small jumping actions.*

	Left Hand	Right Hand	Head	Left Foot	Right Foot
Label Error	47	25	2	37	32
Missed Detect.	13	10	22	9	12
Error Rate (%)	8.4	4.9	3.3	6.5	6.2

Table 3.4: *Error rates of the Virtual Aerobic sequence. Average Error Rate:* ***5.86%***

	Left Hand	Racket	Head	Left Foot	Right Foot
Label Error	4	20	2	4	6
Missed Detect.	39	3	11	8	5
Error Rate (%)	5.6	1.7	3.3	1.5	1.4

Table 3.5: *Error rates of the Tennis sequence. Average Error Rate:* **2.7%**

Testing the algorithm flexibility. Application: Virtual Tennis game

In order to illustrate the adaptability of the algorithm to different contexts and configurations, we analyzed the case of another possible application of the motion capture algorithm: virtual sports training, i.e. tennis. Note how the user silhouette morphology changes with respect to previous tests, due to the inclusion of the racket in the silhouette. Our goal here was thus to extract, label and track the extremity of the racket and the other 4 extremities. To do so only the right hand probability map needed to be adapted. The adaptation consisted on doubling the diameter of the possible positions for that extremity, as well as adapting the class covariance S_α in order to take into account its faster movements.

The *Tennis* sequence is 758 frames long. The average error rate has a value of **2.7%** . Table 3.5 shows the detailed error quantification.

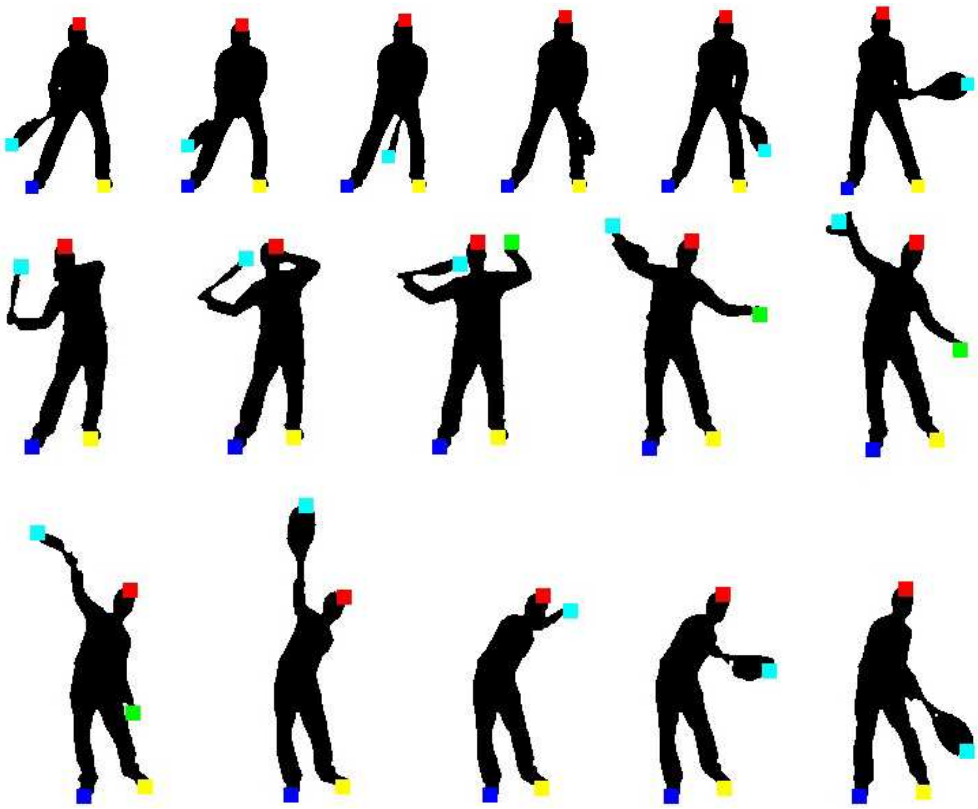


Figure 3.15: *Representative keyframes of the Tennis sequence.*

Testing the algorithm flexibility: Wheelchair user

We also tested the method within the context of a wheelchair user (Fig. 3.16), by changing the a priori maps . As presented in Table 3.6, results are quite similar with those found in the biped-stand up position. Within this context we obviously disregarded the feet.

	Left Hand	Right Hand	Head	Left Foot	Right Foot
Automatic Segmentation					
Label Error	0	2	1	-	-
Missed Detect.	10	4	1	-	-
Error Rate (%)	5.5	3.3	1.1	-	-

Table 3.6: *Error rates, wheelchair user sequence. Average Error Rate:* **2%**

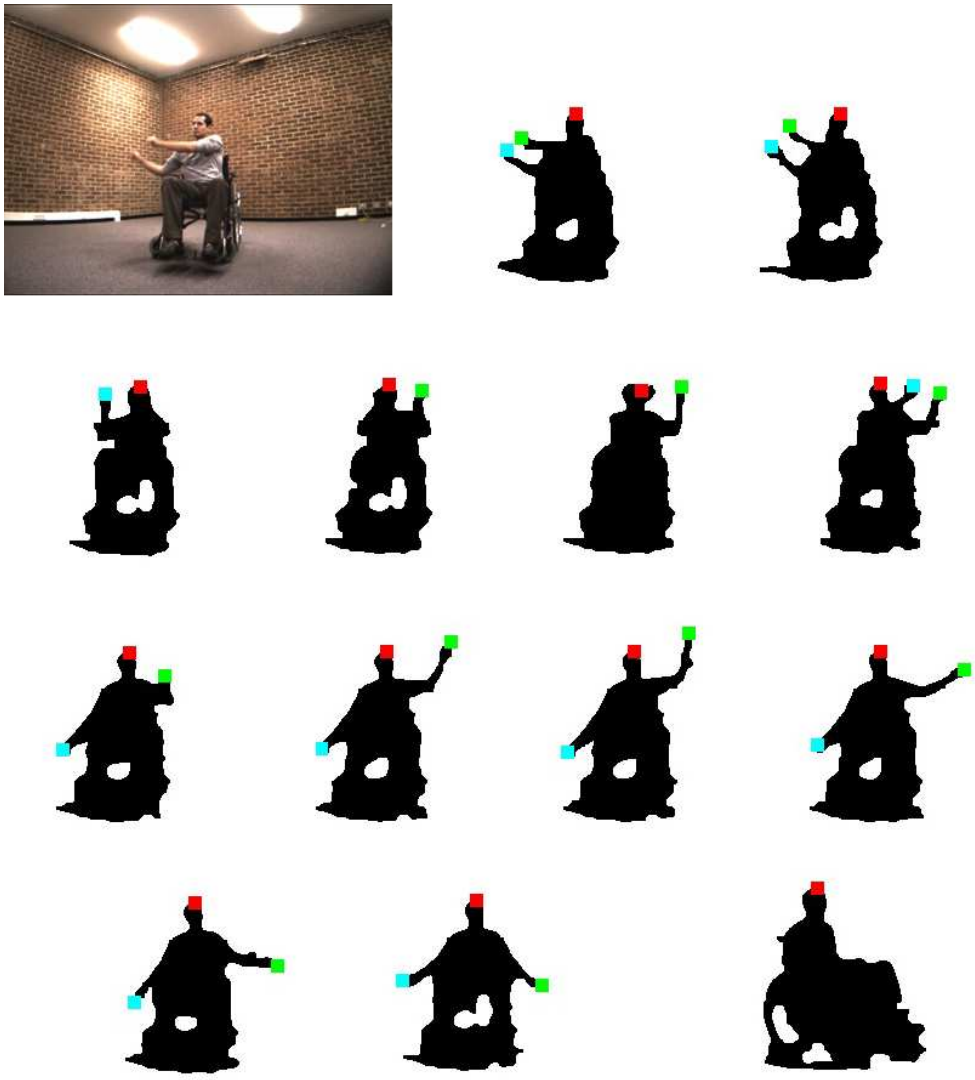


Figure 3.16: *Wheelchair sequence sample. Total sequence length: 180 frames.*

Figure 3.17: *The poorly segmented Navigation sequence environment.*

Testing the algorithm robustness limits: Segmentation. Application: Gestural navigation

Moreover, in order to test the algorithm robustness against segmentation noise we have conducted several experiments. We illustrate this with a third possible application simulation: gestural navigation. The main difference with previous sequences consist in that instead of using a well controlled scenario with an advanced segmentation algorithm, the sequences were segmented using a naive background substraction algorithm, in a random space without any lighting fine-tuning (Fig. 3.17). The presented sequence is 726 frames long and was the very first recorded and segmented sequence. See figure 3.18 for a sample sequence of the results.

The average error rate has a value of **6.76%** . Table 3.5 shows the detailed error quantification.

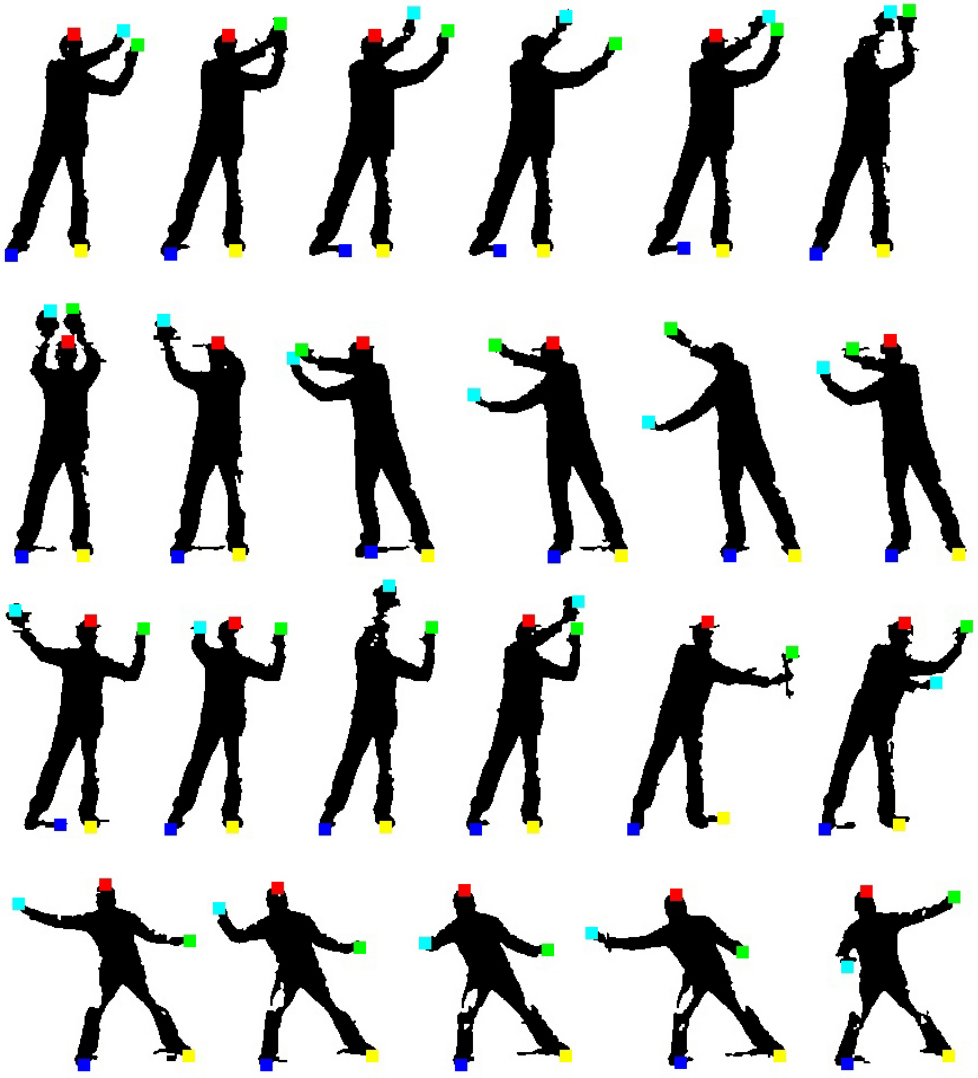


Figure 3.18: *Representative keyframes of the poorly segmented Navigation sequence.*

	Left Hand	Right Hand	Head	Left Foot	Right Foot
Label Error	45	38	10	27	49
Missed Detect.	11	11	16	19	21
Error Rate (%)	7.7	6.7	3.5	6.3	9.6

Table 3.7: *Error rates of the Poor Segmentation Navigation sequence. Average Error Rate:* ***6.76%***

Testing the algorithm robustness limits: Challenging postures

Finally, we have experimented with particularly challenging postures for the type of applications we target, in order to test the algorithm performances towards specially difficult (and even awkward) body movements. From a series of particularly chosen sequences we have depicted three different examples that summarize the algorithm performances in this context. The first one (Fig.3.19) illustrates some intricate movements, aiming at losing or confuse head and hands classes. The two next sequences illustrate the tracking issues when features of the same class fuse and then split up (Figures 3.20 and 3.21).

These three sequences illustrate that the algorithm behaves quite robustly even in the presence of awkward and difficult body movements, when these movements imply different feature types (i.e. head and hands, or feet and hands -Fig.3.12-). On the contrary, due to the unpredictable nature of human gestures and the subsequent lack of prediction of our model, in cases where same features split and fuse (two hands or two feet), a tracking inversion between the two features can occur (Fig.3.20). In those situations however, thanks to the a priori models all the feature classes are correctly assigned, there is only a subclass inversion (left/right foot or hand), which eventually gets automatically corrected.

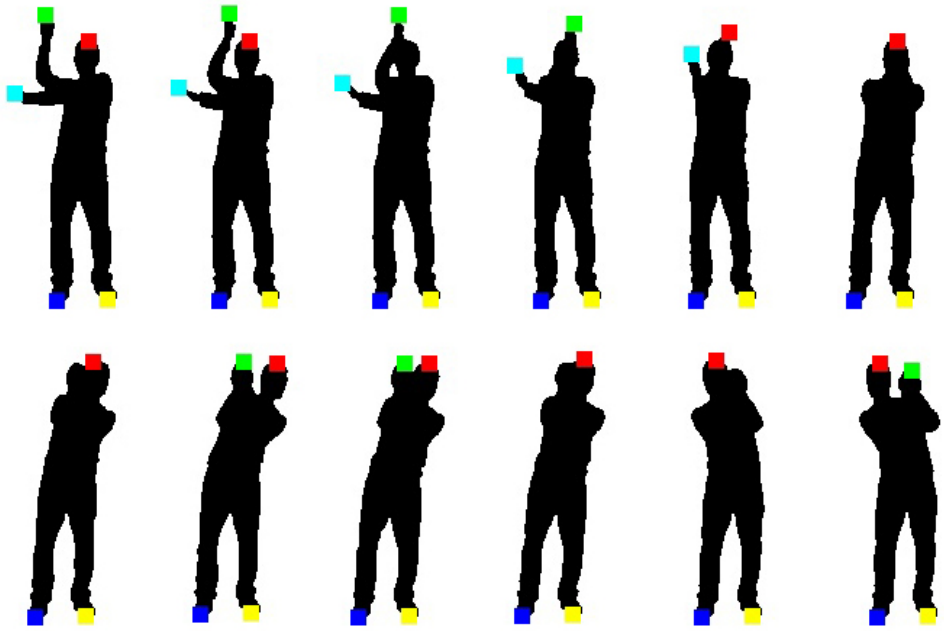


Figure 3.19: *Representative keyframes of the first challenging body movement. Despite the complicated body movement, all features are correctly classified and tracked.*

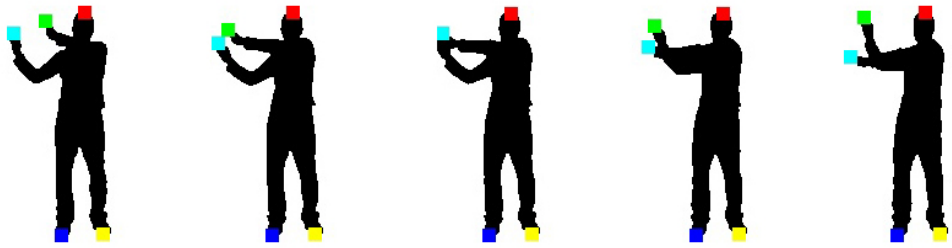


Figure 3.20: *Representative keyframes of the second challenging body movement. There is an inversion in hand classes after the fusion of the two hands.*

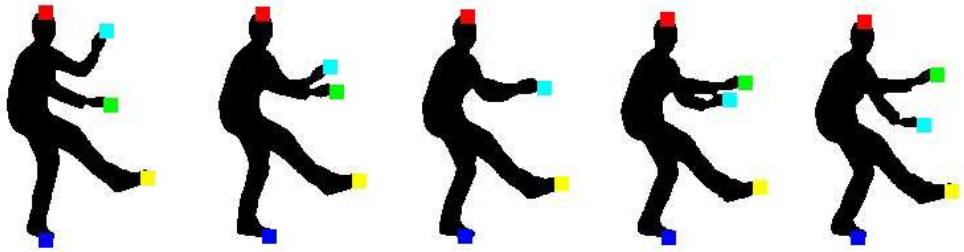


Figure 3.21: *Representative keyframes of the third challenging body movement. Hands are correctly classified and tracked.*

3.6 Conclusions

A new technique for robust real-time 2D human posture estimation has been proposed. It has been presented as a hybrid approach using both image processing tools and probabilistic processes. It uses indeed the whole intra-image algorithm presented in Chapter 2 concerning the CP extraction phase, mostly based on image processing tools. On the other hand, a Bayesian framework has been presented concerning the inter-image classification and tracking steps.

The overall method has proven to have an average error rate range in realistic conditions from 2% to 7% in very noisy contexts. Due to these quantitative results, as well as for the adaptability and less heuristic approach, this inter-image approach is presented as an improvement with respect to results obtained with the pure intra-image algorithm.

For the tracking and pose estimation steps, a simple probabilistic 2D model and a Bayesian approach help labelling and tracking the different human features while allowing adaptation to different contexts of human capture (other than the usual biped stand-up configuration), without imposing a specific initial posture to the user.

Hence, we consider this whole chain, composed of an intra-image CP extraction module, and an inter-image CP classification and tracking, as the main contribution of the present thesis.

Let us indeed compare the synthetic results presented in the previous section to the ones presented in Table 2.3 that summarized the performances of the all intra-image approach. We must although bare in mind that they are computed without any kind of temporal information (and thus tracking). The average error rate of the complete extraction and labelling intra-image, presented in Section 2.4.4 is 4% for synthetic images and 8.44% for real ones, whereas the average of the three rates of the latter hybrid intra (for CP extraction) and inter-image (for labelling and tracking) approach for perfectly segmented silhouettes is 2.83%, and around 6% for real ones. We can thus state that this latter technique produces more accurate results with less computational complexity. The all-intra image approach presents indeed two bottlenecks: firstly the creation of the geodesic distance map, that is $\mathcal{O}(N^3)$ with standard approaches and $\mathcal{O}(N)$ with the presented optimization approach, and secondly the creation (at each frame) of the geodesic skeleton, that again can be computed with linear computational complexities with optimized approaches. The hybrid approach shares with the all-intra method the geodesic distance map creation, however, the classification and tracking module, thanks to the sequential implementation by gating described higher has an insignificant computational complexity compared to the bottleneck. Hence, it has an overall $\mathcal{O}(N)$ complexity.

Even though the morphological skeletons produce very dense and relevant information about the user morphology and pose at each frame, at this stage that approach is not viable for the kind of applications we focus on. Its relevance to the perspectives of this work is presented in Section 5.3.

Experimenting with possible extensions and perspectives

4

We now present several possible extensions of our 2D motion capture algorithm. Even though they appear at an experimentation stage (and we thus do not consider them as 'contributions'), they produce encouraging results and represent good illustrations of some of the possible perspectives and future work of this thesis. We first present an extension of the pure intra-image CP extraction and classification module presented in Chapter 2, that examines the extension of the method to 3D. Secondly, in Section 4.2 we briefly introduce some preliminary results on a possible application of our 2D motion capture approach. Using inverse kinematics, we are indeed able to approximate a whole body pose only by taking as input the extracted movement of the extremities produced by our inter-image CP extraction and classification algorithm.

4.1 Stepping into 3D

LET us now present a possible extension of the intra-image Crucial Point extraction and labelling procedure presented in Chapter 2, using a two non calibrated camera configuration, respectively positioned in a frontal and lateral view, in an orthogonal disposition. Each camera extracts and labels the Crucial Points visible in the frontal and lateral view respectively. After triangulation of the information extracted in the two views, 3D information is subsequently retrieved. Finally, a tracking step is presented. Contrary to the tracking method presented in Chapter 3, that combines classification and tracking in a single

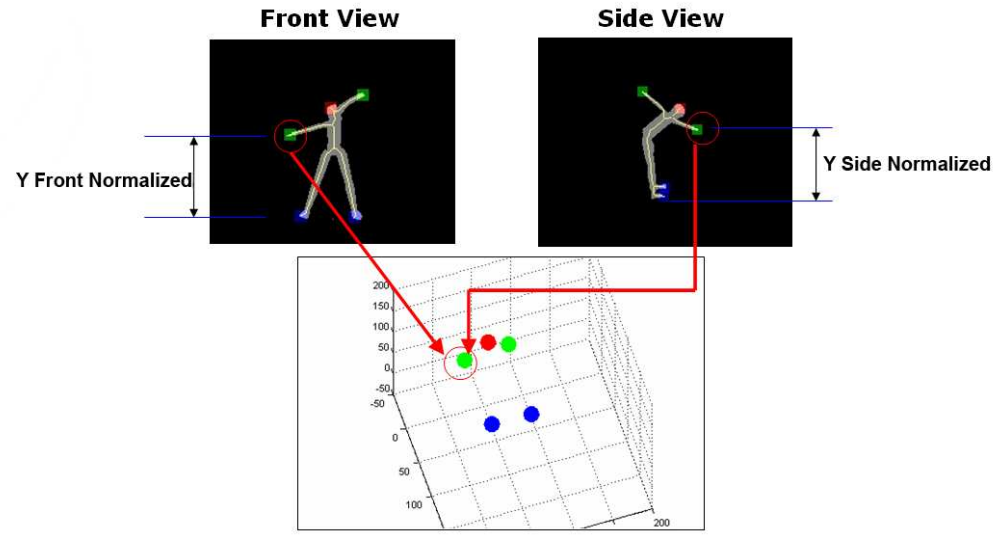


Figure 4.1: *Non calibrated 3D triangulation using normalized axes. Previously extracted and labelled CP positions (in red, green and blue) are matched using the common normalized vertical axis.*

module, this tracking step performs an additional tracking layer on 3D positions, after the two view information has been merged.

4.1.1 Triangulation

Bearing in mind the kind of 2D applications we try to tackle with this work, we nevertheless stepped into 3D as a natural extension of the 2D CP extraction achieved so far. This step firstly intended to produce a three-dimensional relative movement reproduction, yet also to create an additional tracking layer that would increase the extraction robustness and decrease the number of auto-occlusion configurations. Concerning this latter point, we aimed at having for each frame two images that would display a maximum disparity of the scene, in order to minimize the redundant information. Hence the orthogonal camera setting. Nevertheless, the redundant information (visible Crucial Points in both images) allows the CP

matching between the two views. In addition, these 3D features are temporally tracked.

Since camera calibration has not been an issue so far, as well as to ease the hardware manipulation and equipment installation in real case scenarios, the use of of non calibrated cameras has been privileged.

To achieve triangulation we need a common measure that allows comparing relative distances in order to match corresponding Crucial Points in the two views. Since the frontal and lateral view share the same vertical axis, distances along that axis are used in order to compare and match the different crucial point positions. In order to make the comparison possible, y axes are at each frame normalized with respect to the total silhouette height, so as to take into account the scaling factor between views. We thus adopt the orthographic projection hypothesis [Bray et al. 2004]. In the sequel every vertical coordinate (y) will refer to a normalized one.

3D triangulation is obtained in practice by linking for each CP its y frontal coordinate with its y lateral coordinate. The x frontal coordinate corresponds to the final 3D x coordinate, and the lateral x coordinate represents the depth axis in the 3D domain (Figure 4.1).

Moreover, when no matching is possible because the CP is not available in one of the two views, its 3D position can nevertheless be inferred using the available information in the other orthogonal view. Let us illustrate this with an example: in Figure 4.2, only the frontal coordinates of the hands are visible. The y position of those points in the frontal axis are then transposed to the y axis of the lateral view in order to approximate the area of the actor's region where that crucial point is occluded. The hands depth (the x axis in the lateral view) is then taken as being the middle point of the actor's region's section at the transposed relative height. Finally, when a CP is not available in none of the two views, no information is fused (and there is thus no information displayed in the final 3D domain for that point).

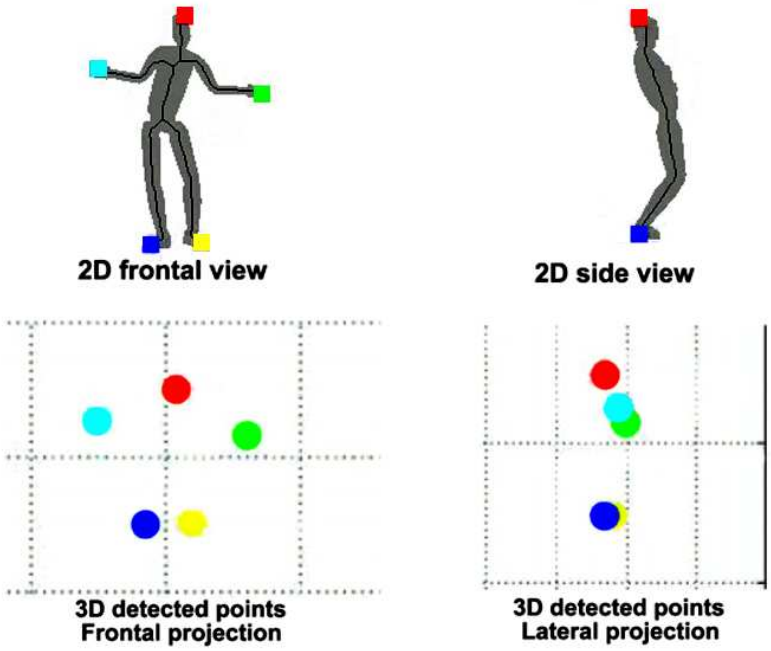


Figure 4.2: *3D triangulation example. Complementary views help solving auto-occlusion situations. The lower row displays the 3D results, projected in the frontal and lateral plane respectively, in order to ease the comparison with the original results. In this case, both hands are visible in the frontal view yet occluded in the side view. Their depth coordinates are computed by approximating their occlusion area in the side view.*

4.1.2 Reliability coefficient

This fusion stage provides the three-dimensional information, while also performing a preliminary verification of the previous classification process. Indeed, each 2D classified crucial point is assigned with a reliability coefficient, which is set for each view, and is inversely proportional to the probability of having a 2D classification error in that frame.

The proposed reliability coefficient is used in cases where the matching process links together two Crucial Points that are classified into different

classes in the two orthogonal views (Figure 4.3). In those cases the emerging 3D point will keep the label of the corresponding 2D point that has a higher reliability coefficient.

As we have seen in Section 2.4.3, when the number of extracted CP is lower than 5 the intra-image feature classification needs to introduce some heuristics. The number of extracted CP is thus the main factor concerning intra image classification robustness. Nevertheless, as we have presented in Section 2.4.3, the existence of loops in the morphological skeleton can sometimes lead as well to variations in the feature classification. Hence, the reliability coefficient depends on these two criteria:

- Number of detected crucial points (N_c): The most influent parameter in the robustness of the crucial point extraction and classification is the number of visible points. As previously commented, the classification strategy can correctly handle situations where less than five points have been detected, as a result of the hypothesis-verification stage previously described. However, its robustness decreases when the number of extracted points is lower than 5.
- Number of loops in the robust skeleton (N_l): Another parameter that decreases the classification accuracy is the presence of loops in the robust skeleton. Loops create different possible paths to communicate between two crucial points. This fact may sometimes lead to geodesic distance miscalculations.

The reliability coefficient RC combines the information of the number of altogether detected crucial points (N_c), and of the presence of loops in the robust skeleton (N_l) (a maximum number of 5 loops is assumed, which is assumed to be the maximum number of loops the segmented silhouette can produce):

$$RC = 5 + N_c + N_l \leq 10 \tag{4.1}$$

Figure 4.3 shows an example of this situation in which the five crucial points are correctly detected in the 2D frontal view whereas the self occlusions lead to only detecting two crucial points in the 2D lateral view. In

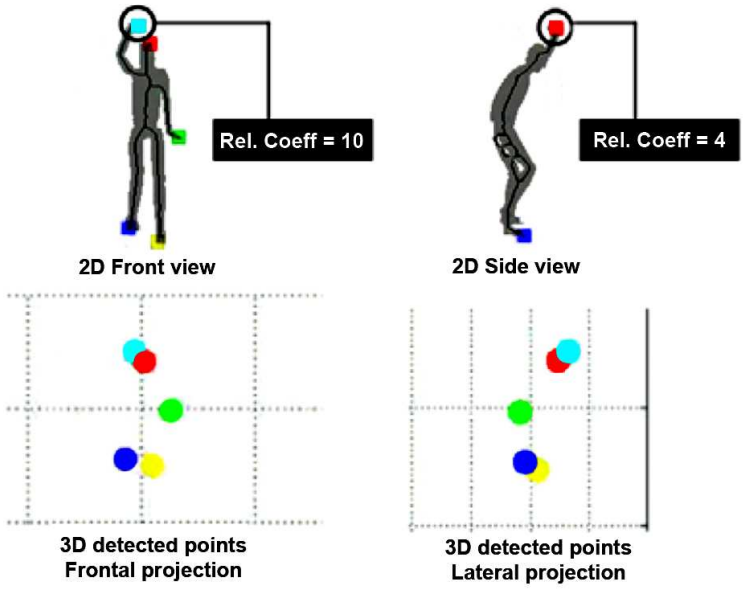


Figure 4.3: *Example of a 3D matching correcting a 2D misclassification, via the reliability coefficient. The frontal configuration presents a very high reliability coefficient, and all CP are extracted and labelled correctly. The lateral view presents a lower reliability coefficient and an misclassification. The fused CP in 3D endorses the most reliable class label.*

this latter, the hand has been incorrectly classified since it partially hides the head and is furthermore aligned with it and hence with the torso angle. Although the left hand is misclassified in the side view, the algorithm is able to correct the label relaying on the reliability coefficients. This is shown in the 2D representation of the 3D detected and labelled crucial points.

Points occluded in both views are omitted in the 3D representation and will be taken into account during the inter-image tracking phase.

4.1.3 3D Tracking

To confer more robustness to the whole chain, we also use temporal information to track the positions of crucial points from one frame to the next.

This final step will link 3D coordinates of each point with their closest correspondent in the previous frame, using the Hungarian matching method [Kuhn 1955]. The Hungarian method optimizes the matching between two sets of points; that is, it minimizes the total matching cost. In our case the two sets of points are the 3D points of the present frame and the 3D points of the previous frame. The costs to minimize are represented by the distances between inter-image corresponding points. A maximal inter-image matching distance threshold is imposed for each 3D crucial point. This allows identifying the points that do not have a correct spatial continuity, resulting from either an incorrect match during the triangulation phase, or a double occlusion leading to a non 3D display. Since CPs that are not available in any of the two views are not displayed in the 3D domain, this later case results into a temporary flickering of the feature class assigned to this specific point. Temporal matching will help creating a temporal continuity of the CP classes by keeping the last known position of the unmatched 3D point for a short period of time (typically 5 frames), until the point reappears (Figure 4.5).

Moreover, **non predetermined key frames** are used as tracking milestones. These are frames in which the reliability coefficients of all Crucial Points (in both views) are very high (and have thus a very low probability of including a classification error). In those cases, resulting 3D coordinates are taken as a reference for subsequent tracking.

Hence this 3D tracking stage:

- Introduces temporal coherence and stabilizes missing CP in the 3D domain due to occlusions in one of the two orthogonal views, or in both.
- Avoids brief erroneous switching between labels, introduced by intra-image classification stage, as well as those that may be introduced by the 3D fusion stage. Intra-image misclassification produce

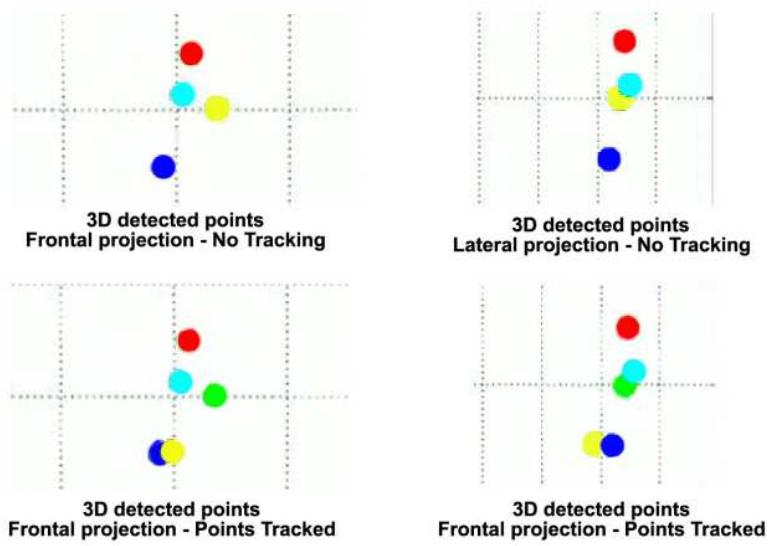


Figure 4.4: *3D tracking example. It provides an additional temporal coherence. In this case it corrects a left hand - left foot label inversion, keeping the hidden left foot position of the previous frame.*

fast inversions between classes, whereas the 3D fusions stage introduces inversions in the depth axis.

- Allows estimating the 3D position of a CP when self occlusions do not permit the direct estimation of a coordinate (occlusion in only one of the two views).

4.1.4 Results

The technique has been only tested with several synthetic sequences. Firstly because synthetic images allow us to test the robustness against self occlusions and complicate body postures, yet ignoring the segmentation imperfections. Moreover, since we have considered this approach as a *perspective* experiment, we favored the theoretical approach, leaving software

and hardware difficulties aside temporarily. Working with synthetic sequences indeed allows avoiding problems of camera synchronization for instance, yet permitting to evaluate this approach possibilities and limits, under ideal conditions.

Furthermore, we present a set of figures illustrating the whole computational chain and the different key notions of the algorithm. Figure 4.5 presents a whole 3D movement sequence. For the sake of legibility, frames are presented at a 5Hz pace. It keeps the same label color code as the previous presented figures (red for the head, cyan for the right hand, green for the left hand, blue for the right foot and yellow for the left foot). It presents the two views 2D extraction and classification (upper side of each frame), and the 3D fusion and temporal tracking (the lower side of each frame). Concerning the 3D fusion illustration, the three-dimensional reconstructed points have been projected into a frontal and lateral plane respectively in each frame, in order to facilitate spatial visualization by comparing the features positions with the original CPs displayed on the silhouettes.

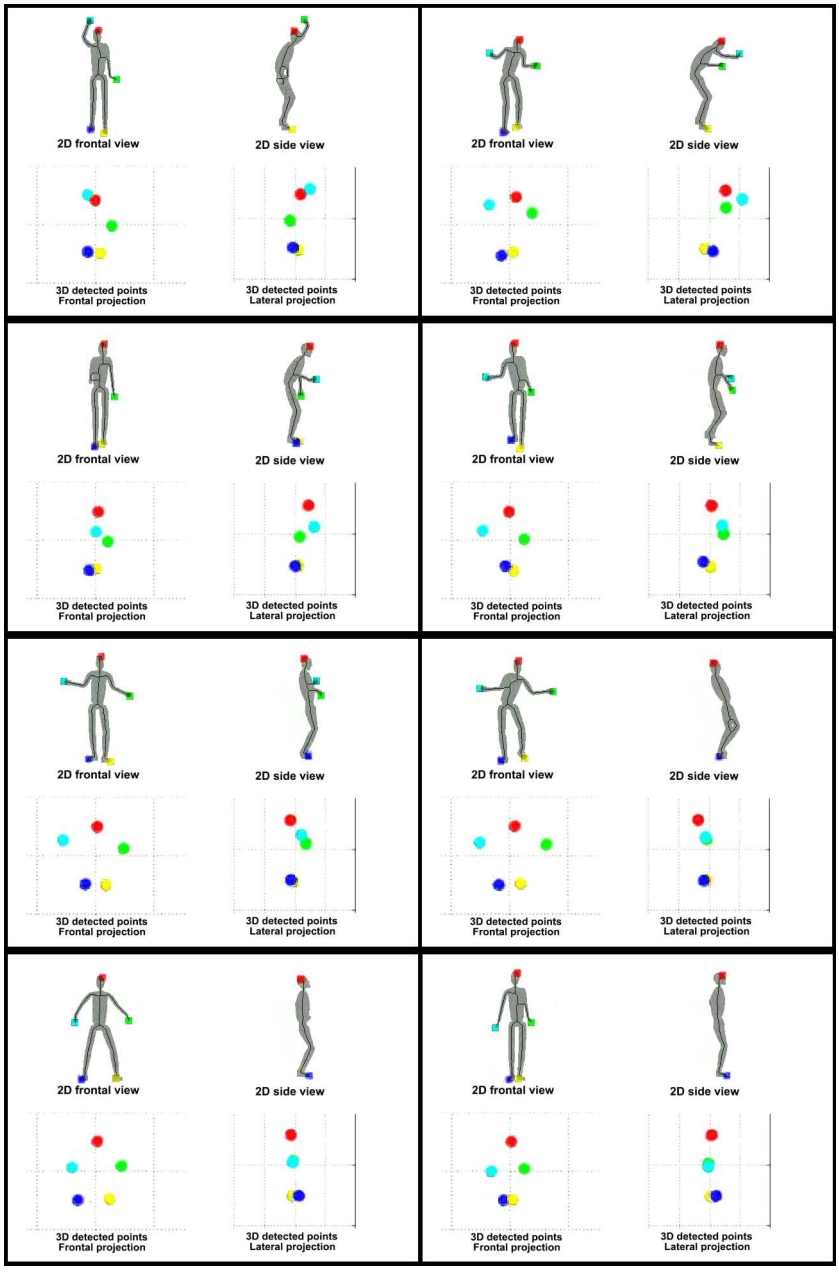


Figure 4.5: *Crucial Point 3D fusion results, using synthetic sequence chacha, displayed at 5Hz. For each frame, the upper row depicts the 2D intra results, for the two cameras. The lower row displays the 3D results, projected in the frontal and lateral plane respectively, in order to ease the comparison with the original results.*

4.2 Taking Crucial Points as input for animation

When considering perspectives for a certain research, another important issue is the kind of algorithms that are likely to be placed following on from the results of that research. In our case, these are the algorithms that are likely to take our CP positions and labels as an input.

An interesting field that could take our CPs as input is computer animation. This way Crucial Points could animate (or help animating) real-time social agents or Computer Graphics (CG) character animation. Hereafter we describe our preliminary research in this area.

4.2.1 Inverse kinematics

The planar forward kinematics problem is a trivial one. Figure 4.7 represents a scheme of a simple joint lying in the X-Y plane. The system contains two links of lengths l_1 and l_2. Two joints (the black dots) connect the links. The angles at each of these joints are θ_1 and θ_2. The forward kinematics problem is stated as follows: given the angles at each of the system joints, where do we find point $P\ (X_p\ ,\ Y_p)$?

In our case however, we need to find the joint angles knowing the location of point P (each one of our Crucial Points). This reverse problem is at the basis of the inverse kinematics, mostly used nowadays in robotics [Guez and Ahmad 1988, Wang and Chen 1991] and computer animation [Hodgins et al. 1995, Rose et al. 1996].

To aid the illustration of the problem, we define an imaginary straight line that extends from the first joint to the last one. This way we denote:

B: length of imaginary line.

α: angle between X-axis and imaginary line.

β: interior angle between imaginary line and link l_1.

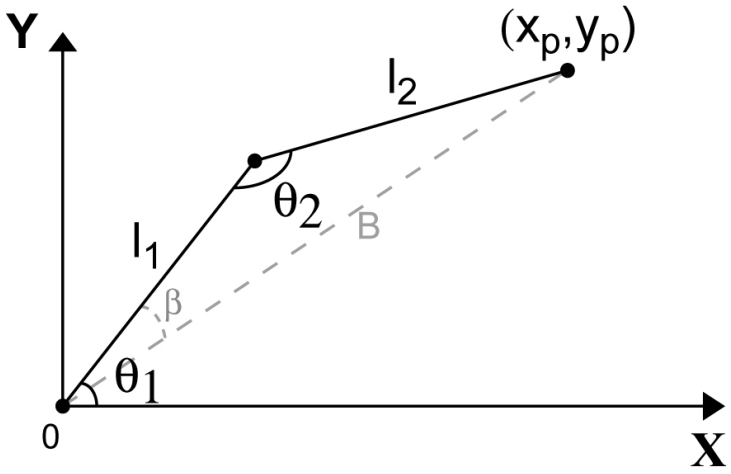


Figure 4.6: *Forward and inverse kinematics problem illustration. System joints are represented by black dots. System links lengths are respectively l_1 and l_2. θ_1 and θ_2 represent the system angles at the joints. Point P (X_p , Y_p) is placed at the extremity of link l_2. Imaginary line B links the joint placed at the system origin with point P. β denotes the interior angle between imaginary line and link l_1.*

And we thus have:

$$B^2 = X_p^2 + Y_p^2 \tag{4.2}$$

$$\alpha = \arctan^2\left(\frac{Y_p}{X_p}\right) \tag{4.3}$$

$$\beta = \arccos\left(\frac{l_1^2 - l_2^2 + B^2}{2 \cdot l_1 \cdot B}\right) \tag{4.4}$$

$$\theta_1 = \alpha + \beta \tag{4.5}$$

$$\theta_2 = \arccos\left(\frac{l_1^2 + l_2^2 - B^2}{2 \cdot l_1 \cdot l_2}\right) \tag{4.6}$$

Respectively using the Pythagorean theorem (Equ. (4.2)) and the law of cosines (Equations (4.4) and (4.6)).

4.2.2 2D animation model

Using inverse kinematics, we have seen how it is possible to infer intermediate limb angles once the extremities positions (our Crucial Points) are given. Nevertheless, even in 2D, for most CP positions there are 2 different solutions. Using an a priori and very simple human 2D joint model, we can however set joint angle limits that will help reducing possible system solutions. We have extracted the following table with human joint types and angle ranges (Table 4.1) from a digital character animation work [Maestri 1996]. Even though it is known nowadays that joint angle ranges are interdependent [Urtasun and Fua 2004], simpler models (and animators know-how) are sufficient in the animation domain. The model that we have created as a first approach to this field is a very simple one, in a 2D plane. It basically translates the simple vitruvian model of Figure 3.4 from the morphological information extracted from the silhouette (i.e. the angle of the torso, the CoG and the top of the head). Inverse kinematics subsequently link the features of the model to the crucial points positions that are imposed at each frame, within the possible angle ranges. For example, having the angle of the torso and the length of the upper body (distance from the CoG to the top of the head), the position of the shoulders are fixed thanks to the previous information. The orientation of the biceps and the forearm will thus be approximated using inverse kinematics from the anchor point *shoulder* to the end point *CP hand*. An analogous approach is followed in order to approximate the orientation of the thigh and the shin.

Segment	**Joint**	**Type**	**X** (in degrees)	**Y** (in degrees)	**Z** (in degrees)
Foot	Ankle	Rotational	65	30	0
Shin	Knee	Hinge	135	0	0
Thigh	Hip	Ball & Socket	120	20	10
Spine	Hip & Spine	Rotational	15	10	0
Shoulder	Spine	Rotational	20	20	0
Biceps	Shoulder	Ball & Socket	180	105	10
Forearm	Elbow	Hinge	150	0	0
Hand	Wrist	Ball & Socket	180	30	120

Table 4.1: *Simplified human angle joint ranges for digital character animation. Columns X, Y and Z represent each segment joint angle ranges. It is assumed the Z axis is oriented along the bone of the joint.*

4.2.3 Results

In order to test this simple algorithm, we have injected the CP results extracted from the synthetic *dodge* sequence (see Section 3.5.1) into our inverse kinematics model. Figure 4.7 presents a series of keyframes from the resulting sequence.

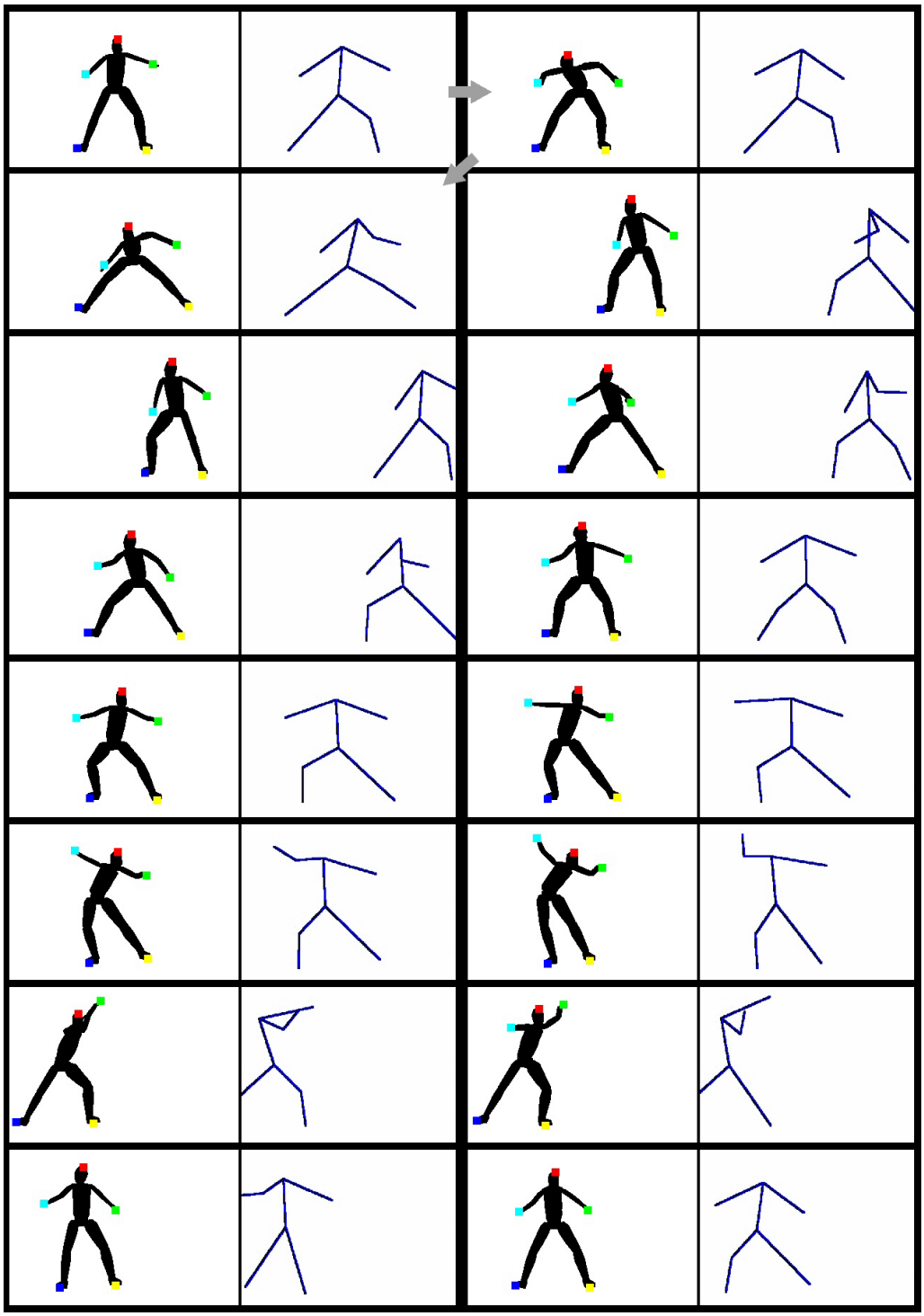


Figure 4.7: *Inverse kinematics results, computed on the dodge sequence. Key frames extracted from a 182 frames sequence.*

Conclusion 5

5.1 Conclusions and contributions

In this thesis we have addressed the problem of non invasive motion capture in real-time interactive contexts. We have presented all the components of our algorithm and its results, in two main branches: an intra-image step and an inter-image one (see Figure 1.4). Hereafter, along with overall conclusions, main contributions appear in bold.

In the intra-image phase we have presented how to take advantage of well known image processing tools in order to analyze the human silhouette and extract as much anatomical information as possible. This way we are able to extract a series of crucial features (what we have called Crucial Points, or CP). These tools allow us to extract crucial human features without expensive material (such as multiple or stereovision cameras) nor skin detection, that has been proven to be very unstable. We have adopted **a novel approach in order to extract human extremities: to use geodesic distance maps with respect to the user's Center of Gravity**. Furthermore, we have presented a **geodesic distance map computation optimization** as a novel alternative to the standard algorithms presented in Section 2.3.2. This alternative allows a geodesic distance map creation in a single pass, hence having a computational complexity of $\mathcal{O}(N)$.

We are able to extract Crucial Points that are visible in the silhouette with a 94% average accuracy rate (Section 2.3.4). The main drawback of this approach is that it is very sensitive to auto-occlusions. Even though a

minimum number of crucial features is always extractable (i.e. head and foot/feet), crucial features inside the silhouette will remain invisible to our algorithm. We have tried to overcome this issue by incorporating an additional camera, in an orthogonal configuration. This way, 3D information could be retrieved, and the cases of auto-occlusions reduced. We have presented some preliminary results of this extension in Section 4.1.

Also, morphological skeletons have been introduced (Section 2.4.1). We have seen that they condense the morphological information of a whole silhouette in a few branches. Unfortunately, standard (and even faster optimized skeletonisation algorithms) always produce *noisy* skeletons. These morphological skeletons contain indeed numerous branches, where a branch is created for each single silhouette border irregularity, disregarding the intensity of this latter. To counter this problem, we have imagined a useful tool, **the morphological skeleton selective pruning** (Section 2.4.2). This algorithm takes the extracted CP and the standard morphological skeleton as input and produces clean skeletons, that only contain branches that connect to an actual CP. Thus these *robust* skeletons introduce a very convenient measurement space between crucial features. As presented in Section 2.4.3, this domain will allow both geodesic and Euclidian measurements, in order to classify the extracted CP into head, hands or feet. This classification is achieved with an average accuracy rate of 95%. This average accuracy rate, combined with the 96% CP extraction accuracy rate, produce a **global extraction and labelling accuracy rate of 91%**.

The inter-image part has been presented inside the context of a global algorithm chain, that provides **a novel technique for robust real-time and non invasive 2D human motion capture**. This technique has been presented as a hybrid approach using both image processing and stochastic tools. It uses indeed the whole intra-image algorithm presented in Chapter 2 concerning the CP extraction phase, yet the inter-image classification and tracking steps are performed inside a Bayesian framework. **The main qualities of this latter approach are its robustness, its adaptability to different contexts, the fact that it does not need to impose a specific initial posture to the user, that it does not use color information nor expensive hardware and its computational lightness. Finally, the whole extraction chain has proved to have an average accuracy rate of 94.5% in noisy contexts**. Due to these strong points this inter-image approach is presented as an improvement with respect to results obtained with the pure intra-

image algorithm, and the **main achievement of this thesis**. However, the CP extraction being the first phase of the algorithm, it presents the same auto-occlusion drawback described above.

5.2 Publications

Journal Papers

1. P. Correa Hernández, J. Czyz , T. Umeda, F. Marqués, X. Marichal, B. Macq: *Bayesian Approach for Morphology-based 2D Human Motion Capture*. IEEE Transactions on Multimedia. To be published.

2. P. Correa Hernández, F. Marqués, X. Marichal, B. Macq: *3D Human Posture Estimation Using Geodesic Distance Maps*. Invited paper to Multimedia Tools and Applications International Journal special issue, Springer US. To be published.

Conference Papers

1. P. Correa Hernández, J. Czyz , F. Marqués, T. Umeda, B. Macq: *Silhouette-based 2D Motion Capture For Real-Time Applications*. IEEE's ICIP 2005, Genoa (Italy). September 2005.

2. P. Correa Hernández, F. Marqués, X. Marichal, B. Macq: *3D Human Posture Estimation using geodesic distance maps*. Joint AMI / PASCAL / IM2 / M4 Workshop on Multimodal Interaction and Related Machine Learning Algorithms, Martigny, Switzerland. June 2004.

3. K. Gaitanis, P. Correa Hernández, B. Macq: *Modelization of Limb Coordination for Human Action Analysis*. IEEE International Conference on Image Processing (ICIP) 2006, Atlanta, USA. October 2006.

4. K. Gaitanis, P. Correa Hernández, B. Macq: *Human Action Recognition Using Silhouette based Feature Extraction and Dynamic Bayesian Networks*. 7th International Workshop on Image Analysis for Multimedia Interactive Services, South Korea. April 2006.

5. U. Toshiyuki, P. Correa Hernández, F. Marqués, X. Marichal: *A Real-Time Body Analysis for Mixed Reality Application*. Tenth Korea-Japan Joint Workshop on Frontiers of Computer Vision. Fukuoka, Japan. February 2004.

6. D. Douxchamps, X. Marichal, T. Umeda, P. Correa Hernández, F. Marqués: *Automatic Body Analysis for Mixed Reality Applications*. WIAMIS 2003 - 4th European Workshop in Image analysis for Multimedia Interactive Services, London UK. April 2003.

5.3 Perspectives

Pose recognition

In addition to what we consider the main contribution of this work, a purely intra-image approach has been presented. It focuses on the use of clean morphological skeletons, that condense to a certain extent the whole body posture in a very small amount of pixels. In this work they have been used in order to classify Crucial Points in the intra-image approach. However, their pertinence can be highly enhanced with an extended analysis of each skeleton morphology, in order to retrieve the whole user body pose. A very fruitful research on this topic is already being followed in collaboration with the Signal Processing Department of the UPC (Barcelona), lead by Prof. F. Marqués.

Gesture recognition

Also, when considering perspectives for a certain research, an important issue is the kind of algorithms that are likely to be placed following on from the results of that research. In our case, these are the algorithms that are likely to take our CP positions and labels as an input. In Section 4.2 we have seen that via Inverse Kinematics, one of these fields could be CG character animation. Another field, with an even more direct possible application is action recognition. These algorithms could take the positions and labels of extracted Crucial Points to a semantic level. We are beginning a very fruitful research in that sense [Gaitanis et al. 2006]. This research takes Crucial Points as input, and aims at detecting simple actions such as walking, jumping and displacing objects. In this context, Crucial Points are considered as cooperative agents forming a cooperative team (the whole body). Movements are analyzed at an individual level and at team level using a hierarchical structure. A novel framework for online probabilistic plan recognition in cooperative multiagent systems (the MultiAgent AHMEM) is used in order to model the human body and detect the actions that are performed. Knowledge of the high-level team actions (*walking* action) improves the pertinence of our predictions on the low-level individual actions (hand is moving back and forth) and allows us to compensate for missing or erroneous data.

3D animation

Another important field that could process our output is computer animation. Our Crucial Points are likely to be taken as input in order to animate (or help animating) real-time avatars and social agents or CG character animation. See Section 4.2 for details and preliminary results.

3D motion capture

Also, as commented in Section 4.1 we have experimented in how to extend our motion capture algorithms to 3D, by applying the same method to several cameras and fusing the resulting information.

Another alternative to this set up is the use of stereovision cameras. This approach will rule out synchronization issues between cameras, while getting rid of the 3D fusion step and having to compute only one image at each frame instead of two. On the other hand, depth maps obtained with this stereovision cameras still leave a lot to be desired.

Bibliography

A. Agarwal and B. Triggs. 3d human pose from silhouettes by relevance vector regression. In *Computer Vision and Pattern Recognition*, pages II: 882–888, 2004.

S. Aukstakalnis and D. Blatner. *The Art and Science of Virtual Reality*. Berkeley, CA, Peachpit Press, 1992.

A. M. Baumberg and D. C. Hogg. Learning spatiotemporal models from training examples. In *Technical Report 95.9*, University of Leeds School of Computer Studies, 1995.

H. Blum. An associative machine for dealing with the visual field and some of its biological implications. In *Proc. Second Annual Bionics Symposium, Cornell University*, pages 244–260, 1961.

R. A. Bolt. put-that-there: Voice and gesture at the graphics interface. In *International Conference on Computer Graphics and Interactive Techniques*, pages 262–270, 1980.

M. Brand. Shadow puppetry. In *International Conference on Computer Vision*, pages 1237–1244, 1999.

M. Bray, E. Koller-Meier, Schraudolph N., and L. Van Gool. Stochastic meta-descent for tracking articulated structures. In *IEEE Workshop on Articulated and Nonrigid Motion (ANM04) Washington D.C.*, June 2004.

L. Calabi and W. Harnett. Shape recognition, prairie fires, convex deficiencies and skeletons. In *Technical Report I, Parke Math. Lab. Inc., One River Road, Carlisle MA*, 1966.

I. Cohen and M. W. Lee. 3d body reconstruction for immersive interaction. In *Second International Workshop on Articulated Motion and Deformable Objects. Palma de Mallorca, Spain*, September 2002.

J. Cortadellas, J. Amat, and F. de la Torre. Robust normalization of silhouettes for recognition applications. In *Pattern Recognition Letters archive*, volume 25(5), pages 591–601, April 2004.

J. C. Cox. A review of statistical data association techniques for motion correspondence. In *International Journal of Computer Vision*, volume 1, pages 53–66, 1993.

O. Cuisenaire. Distance transformations: fast algorithms and applications to medical image processing. In *PhD Thesis, Université catholique de Louvain, Belgium*, October 1999.

J. Czyz, B. Ristic, and B. Macq. A color-based particle filter for joint detection and tracking of multiple objects. In *International Conference on Acoustics, Speech and Signal Processing, Philadelphia, USA*, March 2005.

J. W. Davis and A. F. Bobick. The representation and recognition of action using temporal templates. In *IEEE Transactions on Pattern Analysis and Machine Intelligence*, volume 23, pages 257–267, 2001.

J. W. Davis and A. F. Bobick. A robust human-silhouette extraction technique for interactive virtual environments. In *Workshop on Modelling and Motion Capture Techniques for Virtual Environments, Geneva, Switzerland*, number 1, pages 12–25, November 1998.

Q. Delamarre and O. D. Faugeras. 3d articulated models and multi-view tracking with silhouettes. In *International Conference on Computer Vision, Washington, DC, USA*, pages 716–721, September 1999.

D. Demirdjian and T. Darrell. 3-d articulated pose tracking for untethered diectic reference. In *Int. Conf. on Multimodal Interfaces, Pittsburg, USA*, October 2002.

J. Deutscher, A. Blake, and I. Reid. Articulated body motion capture by annealed particle filtering. In *IEEE Conference on Computer Vision and Pattern Recognition, Columbia, USA*, volume 2, pages 126–133, 2000.

E. W. Dijkstra. A note on two problems in connexion with graphs. In *Numerische Mathematik*, volume 1, pages 269–271, 1959.

A. Doucet, N. de Freitas, and N. Gordon. *Sequential Monte Carlo Methods In Practice*. Springer-Verlag, 2001.

D. Douxchamps, X. Marichal, T. Umeda, P. Correa Hernández, and F. Marqués. Automatic body analysis for mixed reality applications. In *WIAMIS 2003 - 4th European Workshop in Image analysis for Multimedia Interactive Services, London UK*, April 2003.

W. D. Ellis. *A Source Book of Gestalt Psychology*. New York: Harcourt, Brace, 1938.

S. S. Fels and G. E. Hinton. Glove-talk: A neural network interface between a data-glove and a speech synthesizer. In *IEEE Trans. Neural Networks*, volume 4, pages 2–8, 1993.

P. F. Felzenszwalb and D. P. Huttenlocher. Efficient matching of pictorial structures. In *Computer Vision and Pattern Recognition, Hilton Head Island, South Carolina, USA*, pages 66–73, June 2000.

D. A. Forsyth and M. M. Fleck. Body plans. In *Computer Vision and Pattern Recognition, San Juan, Puerto Rico*, pages 678–683, June 1997.

P. Fua, A. Gruen, N. D'Apuzzo, and R. Plankers. Markerless full body shape and motion capture from video sequences. In *International Archives of Photogrammetry and Remote Sensing*, volume 34, pages 256–261, May 2002.

H. Fujiyoshi and A. Lipton. Real-time human motion analysis by image skeletonization. In *IEEE Workshop on Applications of Computer Vision (WACV)*, pages 15–21, October 1998.

K. Gaitanis, P. Correa Hernández, and B. Macq. Modelization of limb coordination for human action analysis. In *IEEE International Conference on Image Processing (ICIP) 2006, Atlanta, USA*, October 2006.

D. M. Gavrila. The visual analysis of human movement: A survey. In *Computer Vision and Image understanding*, volume 73, pages 82–98, January 1999.

A. Guez and Z. Ahmad. Solution to the inverse kinematics problem in robotics by neural networks. In *IEEE Int. Conf. of Neural Networks*, volume 2, pages 617–623, August 1988.

R. Haralick and L. Shapiro. *Computer and Robot Vision*, volume 1. Addison-Wesley, 1992.

I. Haritaoglu, D. Harwood, and L. S. Davis. W4: Real-time surveillance of people and their activities. In *IEEE Transactions on Pattern Analysis and Machine Intelligence*, volume 22, pages 809–830, August 2000.

J. K. Hodgins, W. L. Wooten, D. C. Brogan, and J. F. O'Brien. Animating human athletics. In *22nd annual conference on Computer graphics and interactive techniques (SIGGRAPH), Los Angeles, CA, USA*, volume 29, pages 71–78, August 1995.

S. Ioffe and D. A. Forsyth. Probabilistic methods for finding people. In *International Journal on Computer Vision*, volume 43, pages 45–68, June 2001.

G. Johannson. *Configurations in event perception*. Uppsala: Aimqvist and Wiksell, 1950.

G. Johansson. Visual perception for biological motion and a model for its analysis. In *Perception and psychophysics*, volume 14, pages 201–211, February 1973.

S. Ju, M. J. Black, and Y. Yacoob. Cardboard people: A parameterized model of articulated image motion. In *IEEE International Conference on Automatic Face and Gesture Recognition, Killington, VT, USA*, volume 34, pages 38–44, 1996.

S. Kettebekov, M. Yeasin, and R. Sharma. Prosody based audiovisual coanalysis for coverbal gesture recognition. In *IEEE Transactions On Multimedia*, volume 7, 2005.

M. W. Krueger. *Artificial Reality II*. Addison-Wesley, 1991.

M. W. Krueger. Videoplace - an artificial reality. In *SIGCHI'85 Proc. ACM, New York, USA*, pages 35–40, 1985.

H. Kuhn. The hungarian method for the assignment problem. In *Naval Research Logistics Quarterly*, volume 2, pages 83–97, 1955.

C. Lantuejoul and S. Beucher. On the use of the geodesic metric in image analysis. In *Journal of Microscopy*, volume 121, pages 39–49, january 1981.

C. Lantuejoul and F. Maisonneuve. Geodesic methods in quantitative image analysis. In *Pattern Recognition*, volume 17(2), pages 177–187, 1984.

G. Maestri. *Digital Character Animation*. New Riders Publishing, 1996.

F. Maisonneuve and M. Schmitt. An effcient algorithm to compute the hexagonal and dodecagonal propagation function. In *5th European Congress For Stereology*, pages 515–520, September 1989.

P. Maragos and R. Schafer. Morphological skeleton representation and coding of binary images. In *IEEE Transactions on Signal Processing*, volume 34, pages 1228–1244, 1986.

X. Marichal, B. Macq, D. Douxchamps, and T. Umeda. Real-time segmentation of video objects for mixed-reality interactive applications. In *VCIP 2003 - SPIE Visual Communication and Image Processing Int'l Conference*, pages 41–50, Lugano, Switzerland, July 2003.

F. Meyer. Digital euclidean skeletons. In *Visual Communications and Image Processing*, pages 251–262, October 1990.

A. Micilotta and R. Bowden. Real-time upper body detection and 3d pose estimation in monoscopic images. In *European Conference on Computer Vision, Graz, Austria*, May 2006.

A. S. Micilotta and R. Bowden. View-based location and tracking of body parts for visual interaction. In *Proc. of British Machine Vision Conference*, pages 849–858, 2004.

K. Mikolajczyk, C. Schmid, and A. Zisserman. Human detection based on a probabilistic assembly of robust part detectors. In *European Conference on Computer Vision*, pages Vol I: 69–82, 2004.

P. Milgram and F. Kishino. A taxonomy of mixed reality visual display. In *IEICE Transactions on Information and Systems*, volume E77-D(12), pages 1321–1329, 1994.

T. B. Moeslund and E. Granum. A survey of computer vision-based human motion capture. In *Computer Vision and Image Understanding*, volume 81, pages 231–268, 2001.

G. Mori and J. Malik. Estimating human body configurations using shape context matching. In *European Conference on Computer Vision, Copenhagen, Denmark*, page 666, May 2002.

D. D. Morris and J. M. Rehg. Singularity analysis for articulated object tracking. In *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 289–297, 1998.

J. Ohya, A. Utsumi, and J. Yamato. *Analyzing Video Sequences of Multiple Humans: Tracking, Posture Estimation and Behavior Recognition*. Kluwer, March 2002.

V. I. Pavlovic, R. Sharma, and T. S. Huang. Visual interpretation of hand gestures for human-computer interaction: A review. In *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1997.

A. Pentland. Looking at people: Sensing for ubiquitous and wearable computing. In *IEEE Transactions on Pattern Analysis and Machine Intelligence*, volume 22(1), pages 107–119, January 2000.

D. L. Quam. Gesture recognition with a dataglove. In *Proc. 1990 IEEE National Aerospace and Electronics Conf.*, volume 2, May 1990.

D. R. Urtasun, Fleet, and P. Fua. Monocular 3d tracking of the golf swing. In *Conference on Computer Vision and Pattern Recognition, San Diego, CA*, volume 1, pages 932–939, June 2005.

X. Ren, A. C. Berg, and J. Malik. Recovering human body configurations using pairwise constraints between parts. In *IEEE International Conference on Computer Vision, Beijing, China*, pages 824–831, October 2005.

R. Rosales and S. Sclaroff. Inferring body pose without tracking body parts. In *Computer Vision and Pattern Recognition*, pages II: 721–727, 2000.

C. Rose, B. Guenter, B. Bodenheimer, and M.F. Cohen. Efficient generation of motion transitions using spacetime constraints. In *23rd annual conference on Computer graphics and interactive techniques (SIGGRAPH), New Orleans, Louisiana, USA*, pages 147–154, August 1996.

M. Schmitt. Des algorithmes morphologiques a l'intelligence artificielle. In *PhD Thesis, Ecole des Mines, Paris*, February 1989.

editor Serra, J. *Image Analysis and Mathematical Morphology, Volume 2: Theoretical Advances*. Academic Press, London, 1988.

J. Serra. *Image Analysis and Mathematical Morphology Vol. I.* Ac. Press, London, 1982.

C. Sminchisescu and A. Telea. Human pose estimation from silhouettes: A consistent approach using distance level sets. In *International Conference in Central Europe on Computer Graphics, Visualization and Computer Vision, Bory, Czech Republic*, February 2002.

C. Sminchisescu, A. Kanaujia, Z. Li, and D. Metaxas. Discriminative density propagation for 3d human motion estimation. In *Computer Vision and Pattern Recognition, San Diego, CA, USA*, pages 390–397, June 2005.

P. Soille. *Morphological Image Analysis, Principles and Applications, Second Edition*. Springer-Verlag, 2003.

M. Soriano, B. Martinkauppi, S. Huovinen, and M. Laaksonen. Skin detection in video under changing illumination conditions. In *Proc.of the 15th International Conference on Pattern Recognition*, September 2000.

F. Sparacino, G. Davenport, and A. Pentland. Media in performance: Interactive spaces for dance, theater, circus, and museum exhibits. In *IBM Systems Journal. Issue Order No. G321-0139*, volume 39, page 3, 2000.

D. J. Sturman and D. Zeltzer. A survey of glove-based input. In *IEEE Computer Graphics and Applications*, volume 14, page 30, 1994.

K. Sukel, G. Brostow, and I. Essa. Towards an interactive computer-based dance tutor. In *Concept Paper, Georgia Institute of Technology, Atlanta*, 1998.

M. Turk. Visual interaction with lifelike characters. In *IEEE International Conference on Automatic Face and Gesture Recognition, Killington, VT*, page 368, October 1996.

T. Umeda, P. Correa Hernández, J. Czyz, F. Marqués, and X. Marichal. A real-time body analysis for mixed reality application. In *Tenth Korea-Japan Joint Workshop on Frontiers of Computer Vision, Fukuoka, Japan*, October 2004.

R. Urtasun and P. Fua. 3d human body tracking using deterministic temporal motion models. May 2004.

L. van Vliet and B. J. Verwer. A contour processing method for fast binary neighbourhood operations. In *Pattern Recognition Letters*, volume 7, pages 27–36, january 1988.

L. Vincent. Efficient computation of various types of skeletons. In *SPIE, Medical Imaging V, San Jose, California*, volume 1445, 1991a.

L. Vincent. Morphological transformations of binary images with arbitrary structuring elements. In *Signal Processing*, volume 22, pages 3–23, 1991b.

L. Wang and C. C. Chen. A combined optimization method for solving the inverse kinematics problems of mechanical manipulators. In *IEEE Transactions on Robotics and Automation*, volume 7(4), pages 489–499, August 1991.

C. Wren, A. Azarbayejani, T. Darrell, and A. Pentland. Pfinder: Real-time tracking of the human body. In *IEEE Transactions on Pattern Analysis and Machine Intelligence*, volume 19, pages 780–785, July 1997.

Y. Wu, Y. Hua, and T. Yu. Tracking articulated body by dynamic markov network. In *Ninth IEEE International Conference on Computer Vision, Nice, France*, volume 2, page 1094, October 2003.

A. Yilmaz and M. Shah. Actions sketch: A novel action representation. In *Computer Vision and Pattern Recognition, San Diego, CA, USA*, pages 984–989, June 2005.

T. Zhao and R. Nevatia. Bayesian human segmentation in crowded situations. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 2, pages 459–466, June 2003.