Capture and Representation of Human Walking in Live Video Sequences

Jia-Ching Cheng and José M. F. Moura, *Fellow, IEEE*

***Abstract*— Extracting human representations from video has vast applications. In this paper, we present a knowledge-based framework to capture metarepresentations for real-life video with human walkers. The system models the human body as an articulated object and the human walking as a cyclic activity with highly correlated temporal patterns. We extract for each of the body parts its motion, shape, and texture. Once available, this structural information can be used to manipulate or synthesize the original video sequence, or animate the walker with a different motion in a new synthesized video.**

***Index Terms*— Capture, cyclic motion, human walkers in real video, motion, recognition, stick model, video contents.**

I. Introduction

VIDEO significantly increases the perceived level of interactivity in many multimedia applications, ranging from video conferencing to immersive and collaborative environments to the entertainment industry. To enhance its efficiency and versatility, it is important to develop metarepresentations that describe digital video in terms of its structure rather than in terms of pixels and frames. For example, in a video sequence, if we can capture a human automatically in each frame of the sequence, it is then possible to highlight the human while dimming or replacing the background. We consider recovering these metarepresentations for real-life video sequences with a walking human. The major tasks are to determine the camera motion, to reconstruct the background as viewed across the sequence, and to capture the human and the human motions.

Extracting humans and their representations has also found wide application in computer graphics, animation, and virtual reality [16]. Chromo keying is a technique commonly used in these contexts. However, extracting human representations from *real-life* video remains a challenge.

In this paper, we describe a system that develops metarepresentations for real-life videos with humans in action. This extends generative video (GV), described in [10], [11], and [14]. GV is a framework that represents video in terms of structural components: informational units and ancillary data. The informational units capture the spatial information. They describe the shape and the texture of individual parts in the video experiencing coherent motion. Examples of informational units include the background, humans, cars, or other moving objects. The ancillary data describes the temporal information, like the motion of the informational units and the motion of the camera, plus additional auxiliary information. In the digital video community, these structural components are often referred to as *video contents* and provide a compact representation for the original video sequence. Once available, the GV representation lends itself to simple video annotation, access, manipulation, indexing, nonlinear editing, or synthesis. By analyzing its motion and shape, each individual information unit can be labeled, for example, as a car, with additional attributes such as color. These annotations facilitate retrieval from video databases. The original or a subset of the informational units can be recombined, with the same motions or different motions, with the same or a new background, regenerating the original or a different video sequence.

Due to the complex nature of the human body, which is nonrigid and capable of performing a wide variety of actions, and of the real-life video, which is dynamic and has cluttered background, it is a difficult task to capture the human and its motion in a real-life video sequence. It requires solving the following problems.

- Detecting the moving human in a dynamic scene.
- Approximately locating the human in each frame. We refer to this as the recognition step, since it recognizes the posture of the human in each frame. The posture is defined roughly by the relative position of the four limbs, the head, and torso. Details are in Section III.
- Tracking the human body parts to determine their motions.
- Recovering the shape and texture for the human.

We develop a knowledge-based approach to solve the above problems. Section III overviews the main tasks of our system and briefly reviews the literature related to tracking and recognition of human motion. Section III describes the *knowledge database*. Sections IV–VII consider in more detail the components of the *capture* block. Section IV details the preprocessing, Section V the posture recognition, Section VI the tracking, and Section VII the texture recovery. Section VIII illustrates with real-life videos our experimental results. Finally, Section IX concludes the paper.

II. System Overview

Fig. 1 shows a diagram of our video representation system. Functionally, it is decomposed into three blocks:

- *Knowledge Database Block:* The knowledge database describes the human body and the walking movement, and the shape and motion of the individual body parts. It

Manuscript received September 1, 1998. The associate editor coordinating the review of this paper and approving it for publication was Prof. Wayne Wolf.

The authors are with the Electrical and Computer Engineering Department, Carnegie Mellon University, Pittsburgh, PA 15213-3890 USA (e-mail: moura@ece.cmu.edu).

Publisher Item Identifier S 1520-9210(99)04213-3.

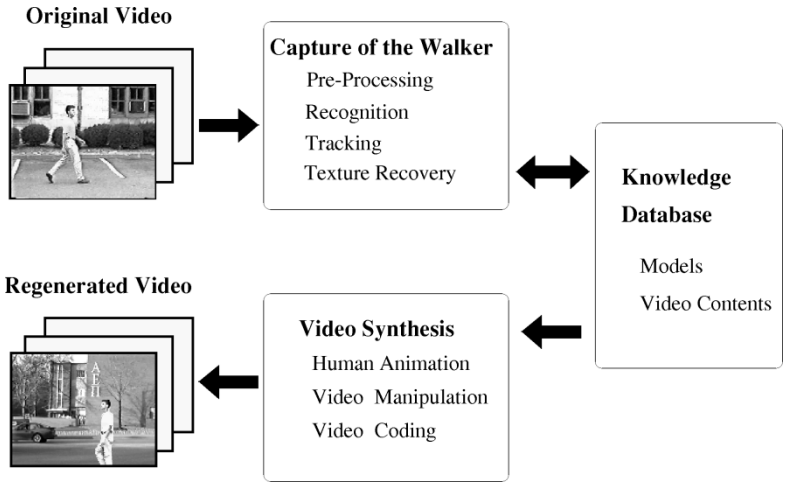


Fig. 1. Diagram of the video representation system.

contains models for the human body, the body parts, and the human motions, as well as possibly for other video contents.

- *Capture of the Human Walker Block:* The capture block combines the body and walking models with real live videos to extract the human and human motion from the video. It includes algorithms to extract the motion, shape, and texture for each rigid part of the human walker and for any other object exhibiting a distinct motion, including the background. No prior model is assumed for the texture. The texture is recovered solely from the video sequence.
- *Video Synthesis Block:* The synthesis block uses the metarepresentation of the original video to resynthesize it or to manipulate it, for example, by altering the texture, replacing the background, or modifying the motions to regenerate a different video sequence.

The capture block is the key component of our system. We elaborate on our approach to capturing a human walker below.

Capturing the motion of human walkers from live video is essentially a localization problem. We adopt a two-stage procedure: global stage and local stage. In the global stage, we use the *preprocessing* block to locate the human in each frame of the video sequence and the *recognition* block to find the approximate posture of the walking human. The preprocessing block resorts to low-level vision techniques to detect and isolate the walker in a dynamic scene. The recognition algorithm is guided with *a priori* models for the walking motion. These are crude walking models obtained by averaging over tens of walking patterns of normal male adults. The output of the global stage is a fair approximation of the posture. This localization information, if used to extract the walker and to resynthesize the original sequence, leads to visible artifacts and low quality video. These artifacts are overcome by the next processing stage—the tracking block.

The output of the global stage is input to the local stage—the tracking block—which tracks the articulated human motions. The tracking algorithms are gradient-based search methods initialized by the model posture recognized in the global stage. They provide accurate positioning and motion estimation for each individual body part.

The fine-tuned tracking results enable the extraction of the texture for the body parts by the *texture recovery* block.

This two-pronged approach—global localization (preprocessing and recognition) and local localization (tracking)—is successful in capturing the human walkers and results in high quality resynthesized video.

A. Related Work

Tracking and recognition of humans and their actions is not a new task in computer vision. Previous work in this area includes [8], [9], [12], [17], and [18]. Due to its complex nature, most systems resort to model-based approaches, control the video capturing devices, or constrain the environment. These approaches are not practical in multimedia applications when it is desired to recover the human from real-life videos with little human intervention with as few constraints as possible.

Most existing techniques fall into one of two categories.

1) *Single View and Constrained Motion*: See Hogg [9], Rohr [18], and Yacoob and Black [19]. In [9], Hogg presented work on recognizing human walking in real images. He modeled both the human body and the human motion. The human body is described as a set of elliptical cylinders; the motion model is acquired interactively from a prototype image sequence. A similar approach is taken by Rohr [18]. Rohr also adopted a cylindrical model for the human body. However, Rohr modeled the motion through time series, averaging the kinematic data provided by the medical motion studies conducted by Murray [15].

   Recently, Yacoob and Black [19] presented a system for tracking human body parts from a monocular image sequence. They model the body parts as articulated links with rectangular patches. They have demonstrated tracking human legs using a three-patch model. Their system needs initial positions for the corner of each patch, and it only tracks nonoccluded body parts.
2) *Multiview and Unconstrained Motion*: See Gavrila and Davis [8] and Kakadiaris and Metaxas [12]. Gavrila and Davis [8] presented a system for tracking human movements based on a multiview approach. Their model

of a human body is constructed with super-quadrics and a large number of degrees of freedom (DOF). The human subjects can perform unconstrained actions, but need to wear tight clothes with plain colors. Kakadiaris and Metaxas [12] presented a similar multiview approach for high DOF tracking of the human body. They modeled the body parts of a human as deformable contours. These high-DOF systems can track unconstrained actions, but they need to be operated in very controlled environments with several static cameras to provide sufficient views, and need known initial poses as start-up conditions.

All these systems require stationary video capturing devices. The system we describe belongs to the category of *Single View and Constrained Motion*, yet it allows for camera motion during video capturing. The task is made more complicated by the camera mobility. It significantly improves upon our previous work [4]–[6].

III. Knowledge Database

Our primary concern is to capture humans and their motions from live video. The problem could be reduced to an exhaustive search over each frame of the sequence attempting to match templates that represent the possible different configurations of the human. The templates themselves are not known, which makes this search clearly a daunting task. To simplify the task, while still producing good capture results, the initial global stage roughly locates the walker in each frame. This step is guided strongly by prior models, which are collected in a knowledge database that groups models for the human body and the human motions. The human body is described as an articulated object with 12 three-dimensional (3-D) rigid body parts. Each body part is a generalized truncated cone with semi-oval spheres attached at each end.

We focus on human walking, which is highly structured and constrained. The constraints provide strong cues for capturing the walkers from live video. We incorporate two constraints. 1) Physical and kinematic constraints that simplify the human body to 12 body parts and reduce human walking to 14 DOF, and 2) dynamic constraints that restrict the search space for the estimation of the motion parameters.

In the following, we first discuss general issues related to human modeling, then describe our articulated model for the human body, and, finally, describe the models for human walking.

A. Human Modeling

To reduce the complexity of the capture phase and the synthesis phase, we adopt models for the human body and for the human walking. To motivate the models, we analyze the major tasks of the capture and synthesis phases. The goal of the capture phase is to recover for each frame of the live video sequence the human and its motion. Capture requires posture recognition, also referred to as action recognition, body parts tracking, and texture recovery. The goal of the synthesis phase is to regenerate the original video sequence or a modified video sequence by manipulating the video representation obtained in the capture phase.

We briefly discuss each of the tasks of the system and their impact on the human body model and motion model.

- *Recognition* of the posture or action of the human walker in each frame of the video sequence. We achieve this by matching the contour of the human walker in a given frame of the real video with the contour of a synthesized human walker in a model sequence.
- *Tracking* of each of the articulated body parts of the human across the video sequence. This recovers accurate postures for the walker in each frame of the video. We use a gradient-based method that requires an articulated model providing an accurate geometrical resemblance to each of the body parts in the human.
- *Recovery* of the texture from the image sequence. It needs accurate geometrical information for each of the body parts.
- *Synthesis* of the human from the metarepresentation that includes the motion, shape, and texture as recovered from the original video. Human synthesis is also used to generate the synthetic humans used in the action recognition stage. This task requires a human model enabling the manipulation of its geometrical and topological characteristics.

To accomplish these tasks, we adopt a kinematic model for the human body and use a set of predefined time series to characterize the temporal patterns of human walking. These time series are adequate, yet simple, models. We describe the model in detail in the following two subsections.

B. Articulated Human Body Model

One of the primary purposes of our modeling scheme is to generate the contour information of the walker. Shape differs from one human to another. It suffices for our purposes to adopt an articulated cone-shaped model. This model is similar to that of Marr and Nishihara [13], which was adopted by Hogg [9] and Rohr [18] in their work. The human body, represented as a stick figure in Fig. 2(a), is considered to be composed of 12 rigid body parts (head, torso, plus two primitives of arms, and three primitives of legs). Each part is represented by a truncated cone with elliptical cross section and a semi-oval sphere attached to each end of the truncated cone (see [6] for the details). More general models could be used that adjust the dimensions of the body parts.

The stick model shown in Fig. 2(a) is a hierarchical model. The root originates from the torso. Each stick is linked with its parent at a joint with 3 DOF, i.e., in general, a stick can rotate with respect to the three axes in the joint-centered coordinate system. With 11 joints, there are then 33 rotational DOF. We assign to each joint j a rotational vector $\Omega_j = [\theta_j \phi_j \psi_j]^T, 1 \leq j \leq 11$.

C. Human Walking

As mentioned earlier, we focus on human walking and consider a single walker. These are not very restrictive assumptions, and we will discuss them in Section IX. A stick model with 12 independently moving body parts has a total of 72 DOF in 3-D. Articulating the body parts through 11 joints as in Fig. 2(a) reduces the number of DOF to 33 rotational

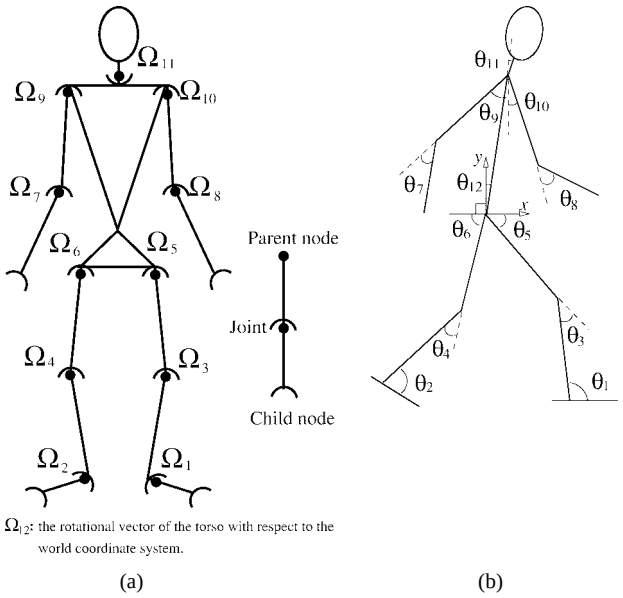


Fig. 2. Articulated human body kinematic model. (a) General model. (b) Walking model.

DOF plus 6 DOF for the human body as a whole. We further reduce this number by assuming that the human walks on a plane (the torso plane) with a small angle with respect to the camera plane. We refer to this as *front and parallel* walking. The 3 DOF of each joint are now reduced to 1, and the 6 DOF of the human body to 2 translational DOF on the plane and 1 rotational DOF (for the torso), leading to a total of 14 DOF. The 2 translational DOF fix the global position of the walker on the walking plane; in particular, we choose the coordinates of the center of the torso u and v. The 12 rotational DOF correspond to the angles θ_i in Fig. 2(b). The rotational vectors Ω_i become simply $\Omega_i = [0, 0, \theta_i], i = 1, 2, \cdots, 12$.

The posture of the walker is defined by the 12-dimensional vector[1]

$$\Theta \triangleq [\theta_1\ \theta_2\ \theta_3 \cdots \theta_{10}\ \theta_{11}\ \theta_{12}]^T. \tag{1}$$

This vector describes the relative position of the different body parts.

1) Data and Model Walkers: In the sequel, we need to distinguish between two walkers: the walker that is to be captured from the video and the walker that is synthesized with the assumed body model and the model assumed posture. We refer to the former as the "data" walker and to the latter as the "model" walker. The postures of each of these walkers are accordingly indexed as Θ_D and Θ_M, where the index D stands for data walker, and the index M stands for model walker. The components of each of these vectors are similarly indexed by D or M, as the case may be. Similarly, we index with D or M the torso coordinates u and v.

2) Frame Number k and Pose p: The motion of the walker is defined by the time evolution of the vector posture angle Θ and the torso center coordinates. With the data walker, time is indexed by the frame number k. With the model walker, the time index will be referred to as the pose and represented by p. The pose is normalized; i.e., p is restricted to the interval $[0\ 1]$. It is usually given in percentage, for example, $p = 50\%$. So, the walking motions will be described by vectors like $\Theta_D(k)$ or $\Theta_M(p)$.

To capture the walking motion, we structure further the motions. We adopt two models for walking: the first is deterministic and is used in the recognition stage; the second is stochastic and is used in the tracking stage. When needed, we distinguish between the models by indexing the quantities of interest by "det" or "st," respectively. Usually, this will not be required and understood from the context. We now detail these models.

3) Deterministic Walking Model—Recognition: In the recognition phase, we assume that walking is a *periodic* motion with a constant (unknown) period T. Murray [15] conducted experiments on measuring gaits of males and females in a wide range of ages and heights. His results reveal that the movement patterns of different body parts as defined by the posture angles θ_i are similar for different people. Rohr [18] used the average measurements of the movement patterns given in [15] in his work. Further, the walking is assumed symmetric so that $\theta_i, i = 2, 4, 6, 8, 10$ can be obtained from the odd posture angles $\theta_j, j = 1, 3, 5, 7, 9$. As a compromise between simplicity and accuracy, in the recognition phase, we adopt Rohr's approach to model the human movements, and use these average joint angle time series as our motion model (prior knowledge). Note that in our model we do not restrict the angles θ_{11} and θ_{12}.

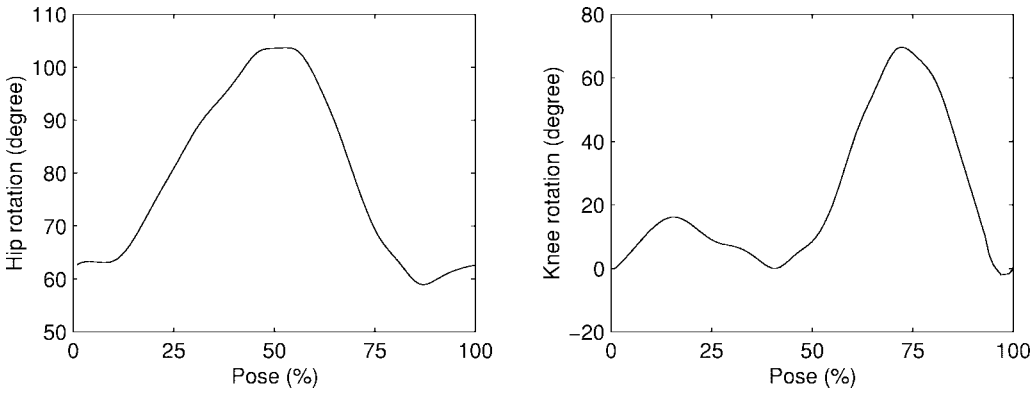
Fig. 3 shows the fundamental period of the (periodic) time series of the joint angles for the hip and knee, θ_{M3} and θ_{M5}, respectively. Reference [18] provides similar time series for θ_{M1}, θ_{M7}, and θ_{M9}.

4) Stochastic Walking Model—Tracking: In the tracking stage, we keep the same articulation shown in Fig. 2(b) to represent the data walker, but generalize the motion model of the previous paragraph.

5) Walking Cycle—Anchor Frames and Complement Anchor Frames: The deterministic model assumes that the walking is periodic with a constant period. Walking, in reality, is better described as a cyclic motion where the duration of each cycle is not constant. We consider the uneven characteristic of the duration of the walking cycle by modeling this duration as a random sequence. Denote the walking cycle m by $WC_m, m = 1, 2, \cdots$ (see Fig. 4). The walking cycles WC_m and WC_{m+1} are defined by special frames, which we refer to as *anchor frames* A_m with index i_{A_m}. Walking cycle WC_m starts at anchor frame A_m and ends at the frame immediately prior to the next anchor frame A_{m+1}.

The anchor frames have poses near 0. We also introduce for each walking cycle WC_m a *complement anchor frame* A_m^c. The complement anchors are the center frames of the walking cycles and have poses approximately equal to 50%. The anchor frames and the complement anchor frames are close to being the frames with the least self-occlusion. Because of this, they are used in Section VI when extracting reference templates for the body parts. The templates are then used for tracking the body parts in walking cycle WC_m.

[1] In (1), $\triangleq$ stands for definition.



(a) (b)

Fig. 3. Temporal walking curves: (a) hip angle θ_{M3}; (b) knee angle θ_{M5}.

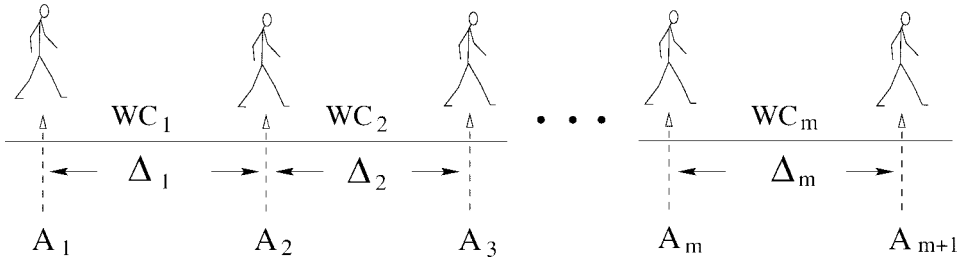


Fig. 4. Anchor frames.

The duration of the walking cycle WC_m, represented by Δ_m, is modeled by

$$\Delta_m = \Delta_{m-1} + \xi_m \quad \text{for } m \geq 1. \tag{2}$$

Equation (2) is a first order difference equation, autoregressive (AR) model for $\{\Delta_m\}$. We could adopt a higher order AR model, or, more generally, an autoregressive moving average (ARMA) model to describe the $\{\Delta_m\}$ sequence. For simplicity, we choose an AR model of first order. We assume that the sequence $\{\xi_m\}$ is a sequence of independent, identically distributed (iid) random variables having uniform probability density function $p(\xi_m)$ in the interval $-\alpha_0 \leq \xi_m \leq \alpha_0$, where α_0 is a constant.[2]

The initial condition of the recursion in (2) is the period of the walking estimated in the recognition stage.

6) Walking Posture: The recognition stage adopts a generic walking model to characterize the walking pattern. In the real world, of course, the walking of an arbitrary individual will significantly depart from this average pattern. We also need to accommodate the walking cycles.

Let $\Theta_{\text{st } M}(m,p)$ be the posture of the model in cycle WC_m at pose p. We adopt the following model:

$$\Theta_{\text{st } M}(m,p) = \Theta_{\text{st } M}(m-1,p) + \eta(m,p), \qquad m \geq 1 \tag{3}$$

$$\Theta_{\text{st } M}(0,p) = \Theta_{\text{det } M}(p) \tag{4}$$

where $\Theta_{\text{det } M}(p)$ is the posture vector for the deterministic model introduced above for the recognition stage. The vector $\eta(m,p)$ accounts for the departure from the generic model.

[2]In practice, ξ_m are discrete random variables, and we assume they have a uniform point masses distribution over a maximum integer range $\pm\alpha_0$.

IV. Preprocessing

Recall Section III-C, where we assume there is a single walker walking in front of a moving camera. The preprocessing estimates the 3-D camera motion, the position of the walker across the sequence, and the orientation of the head and torso. It consists of four steps.

- *Stabilizing the camera motion* by estimating the background motion. We model this motion with a two-dimensional (2-D) eight-parameter projective model. We assume that the scene is a planar surface and that most of the points of the scene satisfy this constraint. Due to lack of space we do not detail the estimation of the projective motion parameters.
- *Detecting the walker* isolates the human walker from the background. This is done by low-level vision techniques that include background registration and motion based detection algorithms.
- *Pursuing the walker* estimates the motion for the walker's head-and-torso. We consider a 2-D four-parameter affine model: a rotational parameter θ_k, a scaling parameter s_k, and two translational parameters u_k and v_k, where k is the frame index.
- *Estimating the position of the walker* recovers 3-D background motion from 2-D motion.

The output of the preprocessing block includes the 3-D camera motion and the position and the orientation of the walker across the video sequence.

V. Posture Recognition

The preprocessing localized the walker in each frame of the sequence by pinpointing the center of the torso and determined the head and torso rotational angles. Regarding the other ten

posture angles $\theta_{Di}(k)$, they are assumed to be periodic with the fundamental period equal to the fundamental period of the model posture θ_{Mi}; see the time series in Fig. 3. To recognize the posture, we are left only with determining the period T_p and with determining the phase ϕ_p of the posture of the data walker. The phase ϕ_p realigns the data walker posture angle time series with the model walker posture angle time series. Once these two time series are realigned, the pose p corresponding to the frame k is given by

$$p(k) = \frac{k-1}{T_p} + \phi_p.$$

To estimate T_p and ϕ_p, we determine for each data walker $W_D(k)$ the best matching model walker $W_M(p)$. This establishes for each k the corresponding matching pose $p(k)$. We then fit a straight line to the scattering plot of $p(k)$ versus k.

We describe briefly each of these two steps of the posture recognition algorithm, see also [6].

A. Contour Matching

To match the data walker in each frame with a model walker, we match the contour of the data walker $W_D(k)$, where k is the corresponding frame number, with the contour of the model walker synthesized from the model $W_M(p)$, where $p \in [0\ 1]$ is the pose. Since we track a walker in a dynamic scene, we expect the edges to be cluttered. To reduce the noise introduced by these cluttered edges, we consider only edges falling within the region corresponding to the data walker extracted by the motion detection process described in Section IV.

We estimate the posture by matching edge information of the data walker with edge information of the model walker. We introduce below a similarity measure that quantifies how close a data walker $W_D(k)$ is from a model walker $W_M(p)$. This similarity measure involves a phase filtering operation. This is based on constructing a distance map and a phase map.

1) Distance and Phase Maps: For the model walker with pose $p, W_M(p)$, we create the edge image $E_M(p)$ by using the Canny edge detector [3]. We construct the distance map $\Gamma_M(p;x,y)$

$$\Gamma_M(p;x,y) = \begin{cases} \alpha + \beta(\delta_\Gamma - \min\limits_{\boldsymbol{e}\in E_M} \|\boldsymbol{e},(x,y)\|), & \text{if } \min\limits_{\boldsymbol{e}\in E_M} \|\boldsymbol{e},(x,y)\| \le \delta_\Gamma \\ 0, & \text{otherwise} \end{cases} \tag{5}$$

where

(x,y)	pixel position;
α, β	positive constants;
$\boldsymbol{e}$	position of an edge pixel in $E_M(p)$;
δ_Γ	given threshold.

Then we construct the phase map $\Phi_M(p;x,y)$

$$\Phi_M(p;x,y) = \begin{cases} \tan^{-1} \dfrac{\nabla_y(W_M * G)}{\nabla_x(W_M * G)}\Big|_{(x,y)}, & \text{if} \min\limits_{\boldsymbol{e}\in E_M} \|\boldsymbol{e},(x,y)\| \le \delta_\Gamma \\ 0, & \text{otherwise} \end{cases} \tag{6}$$

where ∇_x and ∇_y are the components of the gradient operator and G is a Gaussian lowpass filter.

The distance map indicates the distance of a pixel to its closest edge pixel. The phase map is derived from the gradient of a blurred model walker; it possesses the orientation information of the edge image. We use these two maps as geometry filters to measure the geometrical similarity between the model walker and the data walker. Functionally, our distance map is similar to the chamfer image [2] used for measuring the similarity between two sets of edge pixels. The chamfer matching method in [2] computes the similarity between two sets of edge pixels by only measuring the distance between them. It doesn't consider the orientation information between these edges, which, in practice, seems to be actually more important than the distance information. Our phase map provides this information by measuring the orientation between these two sets of edge pixels.

Similarly, we construct for the data walker $W_D(k)$ an edge image $E_D(k)$ and a phase map $\Phi_D(k)$. In this step, we choose in (5) and (6) $\delta_\Gamma = 0$ so that there is no need to generate $\Gamma_D(k;x,y)$.

2) Similarity Measure: For the data walker in frame $k, W_D(k)$, we determine its closest pose in the model by

$$p_{\text{sim}}(k) = \arg \max_{p\in[01]} s(W_D(k), W_M(p)) \tag{7}$$

where $s(W_D(k), W_M(p))$ is the similarity measure

$$s(W_D(k), W_M(p)) = \frac{\sum\limits_{(x,y)} S_M(k,p;x,y) \cdot \Gamma_M(p;x,y)}{\sum\limits_{(x,y)} S_M(k,p;x,y)} \tag{8}$$

where

$$S_M(k,p;x,y) = \begin{cases} 1, & \text{if } (x,y) \in E_D \text{ and} \\ & \quad |\Phi_D(k;x,y) - \Phi_M(p;x,y)| \le \delta_\Phi \\ 0, & \text{otherwise} \end{cases}$$

where δ_Φ is a given threshold. We call the procedure defined by $S_M(k,p;x,y)$ as phase filtering; see [6] for details.

B. Posture Fitting

After finding the closest pose $p_{\text{sim}}(k)$ for each of the data walkers in a number of consecutive frames $W_D(k), k = 1, 2, \cdots, K$ by using the approach described in Section V-A, we determine the period $T_p \triangleq f_p^{-1}$ in frames/cycle and the phase ϕ_p (or the pose of the walker in the first frame of the video) by a line fitting algorithm

$$[f_p \quad \phi_p] = \arg \min_{f_p,\phi_p} \sum_k \|p_{\text{sim}}(k), f_p(k-1) + \phi_p\|. \tag{9}$$

We designate $p_{\text{fit}}(k) \triangleq f_p(k-1) + \phi_p$ to be the fittest pose of the data walker $W_D(k)$ and $\Theta_{\text{fit}}(k) \triangleq \Theta_M(p_{\text{fit}}(k))$ to be the fittest posture.

VI. Tracking Articulated Human Motions

The goal of the tracking stage is to accurately locate the position of all the body parts of the human walker across all the frames of the video sequence. The preprocessing block localized the (center of the) torso and the orientation of the head and torso across all the frames of the sequence—it resolved the two translational degrees and two of the rotational degrees. It also determined the scaling factor associated with the camera. The recognition block assumed the generic model about the human body and human walking described in Section III. The output of the recognition stage is the period T_p and the phase ϕ_p of the walking cycle. Using these, the recognition stage roughly identifies the posture of the walker, i.e., the vector $\Theta_{\text{fit}}(k)$ in each of the frames, resolving all remaining degrees of freedom.

When contrasting walker sequences synthesized using the posture sequence $\Theta_{\text{fit}}(k)$ with the data walker sequences, we observe that there is a certain level of mismatch, as shown in Fig. 8(a). This section describes the tracking algorithm that follows the recognition stage and significantly reduces the artifacts that persist after recognition has been accomplished. The output of the tracking is an improved estimate of the walking posture, i.e., of the data posture $\Theta_D(k)$; hence a fine tuning of the rotational degrees of freedom.

In Section VI-A, we describe the tracking algorithm, referred to as the Human Walking Tracking Algorithm (HWTA). In Sections VI-B–D, we describe the building modules in the HWTA.

A. Human Walking Tracking Algorithm

Fig. 5 shows a block diagram of the HWTA. An early implementation has been reported in [5]. The HWTA has three main modules.

- *Localization of Anchor Frames:* This module locates for each walking cycle WC_m the anchor frame A_m and the complement anchor frames A_m^c. The anchor frames have poses close to 0 and the complement anchors poses close to 50%.
- *Registration of Frames:* This module registers accurately the postures of the anchor frames A_m and the complement anchor frames A_m^c by determining the position of all body parts.
- *Tracking of the Body Parts:* This module estimates for each walking cycle WC_m the true posture of the walker in each frame. The recognition results of Section V provide initial reference templates for the tracking of the body parts.

The tracking algorithm is iterative on the cycles WC_m into which we divide the walking sequence. Before entering the iterative loop, the first block of the algorithm has an initialization step that *registers* a reference frame, which we call the anchor frame A_1, i.e., determines the position of all the body parts in the first anchor frame A_1. This provides starting templates for all the body parts. After this block, the HWTA enters the iterative loop; see Fig. 5. The loop iteration is divided into five main blocks. The first block locates the reference frames—the anchor frames A_{m+1} and the complement anchor frames $A_m^c, m \geq 1$. This step is followed by a body part tracking block that tracks the head and torso (H/T), the front leg (FL), and the front arm (FA) across all the frames within the current cycle WC_m. The next block registers the anchor frames A_{m+1} and the complement anchor frames A_m^c, i.e., determines the position of the body parts in these frames. The block provides possibly several complementary body part reference templates. The templates for the back leg (BL) and back arm (BA) are used to track these two body parts on the current cycle; see the second block labeled Tracking. The final block simply tests the end of the loop.

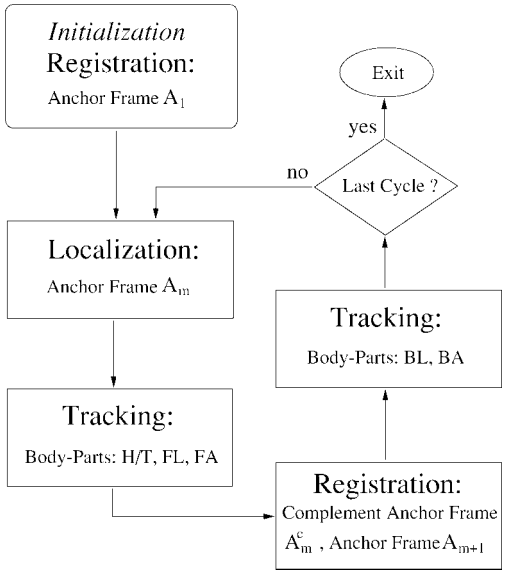


Fig. 5. Block diagram of the Human Walking Tracking Algorithm (HWTA).

We now describe in detail the building modules in the HWTA.

B. Frame Registration

As explained in Section VI-D, in each walking cycle, the body parts are tracked by a template matching technique. This template matching uses texture cues to track each of the body parts. This requires that we extract from the data walker the texture templates of the body parts. The texture templates are then used as initial reference templates for the subsequent tracking of the corresponding body parts. To extract accurately the texture of the body parts, we fine tune the posture parameters of the body parts. This is the goal of the frame registration module that we now describe in detail.

The first issue regards choosing the initial reference templates. The ideal frames for providing initial reference templates are the ones with least occlusion. The data for the temporal walking patterns suggests that frames with walkers with poses close to 0 and 50%, whose arms and legs are widely open, are good candidates. The anchor frames A_m and the complement anchor frames A_m^c are exactly these ideal frames.

We break the tasks of frame registration into three groups according to the types of body parts that need to be registered.

- *Initialization/Registration of All Body Parts in the First Anchor Frame* A_1: All body parts of the human walker in A_1 are registered to provide initial reference templates for tracking.

- *Registration of the Back Leg in Anchor Frames A_{m+1} and Complement Anchor Frames $A_m^c, m \geq 1$:* This task establishes reference templates for the thigh, the shank, and the foot of the back leg that are used for tracking the back leg.
- *Registration of the Back Arm in Anchor Frames A_{m+1} and Complement Anchor Frames $A_m^c, m \geq 1$:* This establishes reference templates for the upper arm and for the forearm of the back arm that are used for tracking the back arm.

The above registration tasks are similar. We estimate the data postures of these frames by using a hybrid matching method, which incorporates image intensity, contour, motion cues. Due to lack of space, we omit the description of the method.

C. Anchor Frame Localization

To apply the dynamic constraints to the tracking of human walking, we need to section accurately the image sequence into walking cycles. This is accomplished by identifying the indices of the anchor frames which by definition mark the beginning of each walking cycle. This is the goal of the anchor frame localization. We describe the method for localization of the anchor frames in the next paragraph.

1) Localization Method: Due to lack of space, we do not detail how the body parts in the first anchor frame are registered, but assume that this task has been accomplished, and that the posture of the data walker of the first anchor frame A_1 has been fine-tuned to a new posture estimate $\Theta_D(1)$; see [7] for details. The registered data walker in anchor frame A_1 is then used as a reference template for locating the second anchor frame A_2, as the algorithm enters the loop in the block diagram of Fig. 5. After $m-1$ times around the loop, the algorithm has located the first m anchor frames, i.e., $A_i, i = 1, 2, \cdots, m$, and has registered the posture of the data walker $\Theta_D(i)$ in these anchor frames $A_i, i = 1, 2, \cdots, m$. We consider the mth cycle in the loop by explaining the localization of the next anchor frame A_{m+1}. The durations Δ_m of consecutive walking cycles follow the first-order AR model described by (2). The predicted index $\hat{\imath}_{A_{m+1}}$ for anchor frame A_{m+1} is

$$\hat{\imath}_{A_{m+1}} = i_{A_m} + \Delta_m. \tag{10}$$

We now correct this predicted estimate of $\hat{\imath}_{A_{m+1}}$. We minimize the sum of squared differences (SSD) between the data walker in frame i_{A_m} and the data walker in frames with frame numbers within $\hat{\imath}_{A_{m+1}} \pm \alpha_0$. The quantity α_0 was introduced in model (2); see footnote 2 on p. 148.

D. Body Parts Tracking

The tracking of the body parts fine tunes the posture parameters of the walking model, i.e., the data posture vector sequence $\{\Theta_D(k)\}$. We adopt a gradient-based method. Dynamic constraints regarding the walking patterns and kinematic constraints and physical constraints regarding the articulation are incorporated to improve stability and reliability. We decompose the human body into five groups: head and torso and four limbs. Each part therefore consists of two or three rigid segments. We develop algorithms to track these multiple-segmented articulated objects.

The registration of the anchor frames and the complement anchor frames identifies the positions of the head and torso and the positions of the joints of the limbs with respect to the torso, so it takes care of the translational motion of the shoulder joints and hip joints. We are left with determining the rotational motion of the articulated body parts. Each articulated body part is modeled as a planar manipulator, which is a robot arm with joints on the same plane and revolutes rotating around the same axis. In the following paragraphs, we first describe the kinematic model for the articulated objects that we adopt and then develop a least squares solution for tracking an articulated body part.

1) Tracking Articulated Objects: For a K joint manipulator working in an N-DOF space, the joint angle vector $\boldsymbol{\Phi} \in R^K$ is determined by inverting the nonlinear relation

$$\boldsymbol{x} = f(\boldsymbol{\Phi}). \tag{11}$$

Equation (11) expresses what is known as the forward kinematic problem [1]. Differentiating with respect to time, one obtains

$$\dot{\boldsymbol{x}} = \frac{\partial \boldsymbol{x}}{\partial \boldsymbol{\Phi}} \dot{\boldsymbol{\Phi}} = \boldsymbol{J}(\boldsymbol{\Phi})\dot{\boldsymbol{\Phi}} \tag{12}$$

where $\boldsymbol{J}(\boldsymbol{\Phi}) \in R^{N \times K}$ is the Jacobian matrix. A manipulator working in 3-D space with K revolute joints rotating around the z-axis, i.e., where each joint has two DOF, is referred to as a planar manipulator. Assume that each segment of the planar manipulator is modeled as a cylinder-like rigid object, as shown in Fig. 6. For a given point $\boldsymbol{x}_{k,i} = (x_{k,i}, y_{k,i}, z_{k,i})$ on the surface (and in the inner body) of the kth segment, it can be shown that

$$\begin{aligned}
\frac{\partial x_{k,i}}{\partial \phi_j} &= \begin{cases} (-y_{k,i} - y_{j,0}), & k \geq j \\ 0, & k < j \end{cases} \\
\frac{\partial y_{k,i}}{\partial \phi_j} &= \begin{cases} (x_{k,i} - x_{j,0}), & k \geq j \\ 0, & k < j \end{cases} \\
\frac{\partial z_{k,i}}{\partial \phi_j} &= 0
\end{aligned} \tag{13}$$

and the entries of the Jacobian $\boldsymbol{J}$ at the point $\boldsymbol{x}_{k,i}$ are specified in (14), shown at the bottom of the page, where $\boldsymbol{\Phi} = [\phi_1 \phi_2 \cdots \phi_{K-1} \phi_K]^T$. The last $K-k$ columns and the last row of $\boldsymbol{J}(\boldsymbol{x}_{k,i}, \boldsymbol{\Phi})$ are zero.

$$\boldsymbol{J}(\boldsymbol{x}_{k,i}, \boldsymbol{\Phi}) = \begin{bmatrix} -(y_{k,i} - y_{1,0}) & -(y_{k,i} - y_{2,0}) & \cdots & -(y_{k,i} - y_{k,0}) & \mathbf{o} \\ x_{k,i} - x_{1,0} & x_{k,i} - x_{2,0} & \cdots & x_{k,i} - x_{k,0} & \mathbf{o} \\ 0 & 0 & \cdots & 0 & \mathbf{o} \end{bmatrix} \tag{14}$$

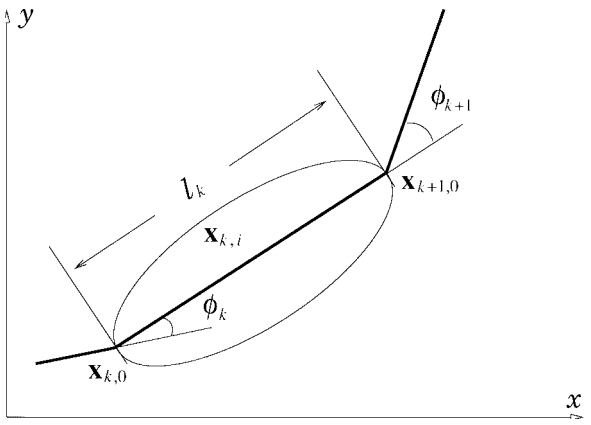


Fig. 6. Structure of a K-DOF manipulator.

2) Least Squares Solution: At time t, assume we know the rotation $\boldsymbol{\Phi}(t)$. We determine by least squares the incremental rotation $\Delta\boldsymbol{\Phi}_t$ at time t that leads to the rotation at time $t+1$, i.e., $\boldsymbol{\Phi}(t+1) = \boldsymbol{\Phi}(t) + \Delta\boldsymbol{\Phi}_t$. The object surface specified by the manipulator is defined in an object centered coordinate system, denoted as (x^c, y^c, z^c). An actual object surface must be expressed in the world coordinate system, namely, (x^w, y^w, z^w). For the human walker in 3-D, we assume that Ω_y, the angle between the orientation of the walker and the image plane of the camera, is a known constant; $\Omega_x = 0$, i.e., the $x^c - z^c$ plane is parallel to the $x^w - z^w$ plane; and $\Omega_z = 0$, i.e., the $y^c - z^c$ plane is parallel to the $y^w - z^w$ plane. The relationship between the (x^c, y^c, z^c) and the (x^w, y^w, z^w) coordinate systems is defined by a homogeneous coordinate transform $\boldsymbol{R}_y$, i.e.,

$$\begin{bmatrix} x^w \\ y^w \\ z^w \end{bmatrix} = \boldsymbol{R}_y \begin{bmatrix} x^c \\ y^c \\ z^c \end{bmatrix} \tag{15}$$

$$\boldsymbol{R}_y = \begin{bmatrix} \cos\Omega_y & 0 & \sin\Omega_y \\ 0 & 1 & 0 \\ -\sin\Omega_y & 0 & \cos\Omega_y \end{bmatrix}. \tag{16}$$

At this point, we do not estimate the 3-D motion parameters of the perspective projection model. As we recall from Section III-C, we constrain the walker to move in front of the camera with a limited view angle. This enables us to approximate the perspective projection by an orthographic projection. Under the orthographic projection, a point $\boldsymbol{x}_o^c = [x_o^c y_o^c z_o^c]^T$ in the object coordinate system is projected to the image location $\boldsymbol{x}_{\rm im} \triangleq [x_{\rm im} y_{\rm im}]^T$ given by

$$x_{\rm im} = \begin{bmatrix} x_{\rm im} \\ y_{\rm im} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \cdot \boldsymbol{R}_y \cdot \boldsymbol{x}_o^c. \tag{17}$$

Note that we ignore a scaling factor in the equation above. Taking the time derivative of (17), the velocity field for a pixel on the image plane is

$$\boldsymbol{u}(\boldsymbol{x}, \boldsymbol{\Phi}) = \begin{bmatrix} u(\boldsymbol{x}, \boldsymbol{\Phi}) \\ v(\boldsymbol{x}, \boldsymbol{\Phi}) \end{bmatrix} \triangleq \dot{\boldsymbol{x}}_{\rm im} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \cdot \boldsymbol{R}_y \cdot \dot{\boldsymbol{x}}_o^c \tag{18}$$

where the $\dot{()}$ stands for time derivative. Assuming that the changes in image intensity between two consecutive video frames t and $t+1$ are described by

$$I(x, y, t+1) \triangleq I(x + u(\boldsymbol{x}, \boldsymbol{\Phi}), y + v(\boldsymbol{x}, \boldsymbol{\Phi}), t). \tag{19}$$

The first-order Taylor series expansion of (19) leads to the well-known gradient formulation

$$I_t = [I_x I_y] \cdot \begin{bmatrix} u(\boldsymbol{x}, \boldsymbol{\Phi}) \\ v(\boldsymbol{x}, \boldsymbol{\Phi}) \end{bmatrix} \tag{20}$$

where $I_x \triangleq (\partial I(x,y,t)/\partial x)$ and $I_y \triangleq (\partial I(x,y,t)/\partial y)$ are the components of the spatial image gradient at location (x, y), and $I_t \triangleq (\partial I(x,y,t)/\partial t)$ is the temporal image gradient.

Substituting (12) and (18) into (20) yields[3]

$$\begin{aligned} I_t(\boldsymbol{x}_{k,i}) &= [I_x(\boldsymbol{x}_{k,i})\, I_y(\boldsymbol{x}_{k,i})] \cdot \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \\ &\quad \cdot \boldsymbol{R}_y \cdot \boldsymbol{J}(\boldsymbol{x}_{k,i}, \boldsymbol{\Phi}) \cdot \Delta\boldsymbol{\Phi}_t \\ &= [I_x I_y] \cdot \begin{bmatrix} \cos(\Omega_y) & 0 & \sin(\Omega_y) \\ 0 & 1 & 0 \end{bmatrix} \cdot \boldsymbol{J}(\boldsymbol{x}_{k,i}, \boldsymbol{\Phi}) \cdot \Delta\boldsymbol{\Phi}_t \\ &= [I_{\phi^1_{k,i}} I_{\phi^2_{k,i}} \cdots I_{\phi^K_{k,i}}] \cdot \Delta\boldsymbol{\Phi}_t \qquad (21) \\ &= \boldsymbol{m}_{k,i} \cdot \Delta\boldsymbol{\Phi}_t \qquad (22) \end{aligned}$$

where $I_{\phi^l_{k,i}} = -I_x \cos(\Omega_y)(y_{k,i} - y_{l,0}) + I_y(x_{k,i} - x_{l,0})$.

Choose for the kth segment N_k points[4] from the surface of this segment, which are projected to the image plane. This results in the following system of N_k linear equations:

$$\boldsymbol{M}_k \cdot \Delta\boldsymbol{\Phi}_t = \boldsymbol{I}_{k,t} \tag{23}$$

$$\boldsymbol{M}_k = \begin{bmatrix} \boldsymbol{m}_{k,1} \\ \boldsymbol{m}_{k,2} \\ \vdots \\ \boldsymbol{m}_{k,N_k} \end{bmatrix}, \quad \boldsymbol{I}_{k,t} = \begin{bmatrix} I_t(\boldsymbol{x}_{k,1}) \\ I_t(\boldsymbol{x}_{k,2}) \\ \vdots \\ I_t(\boldsymbol{x}_{k,N_k}) \end{bmatrix}. \tag{24}$$

Combining the K systems of linear equations (23) corresponding to the K revolute joints into a single vector equation

$$\boldsymbol{M} \cdot \Delta\boldsymbol{\Phi}_t = \boldsymbol{I}_t \tag{25}$$

$$\boldsymbol{M} = \begin{bmatrix} \boldsymbol{M}_1 \\ \boldsymbol{M}_2 \\ \vdots \\ \boldsymbol{M}_K \end{bmatrix}, \quad \boldsymbol{I}_t = \begin{bmatrix} \boldsymbol{I}_{1,t} \\ \boldsymbol{I}_{2,t} \\ \vdots \\ \boldsymbol{I}_{K,t} \end{bmatrix}. \tag{26}$$

The least squares solution of (25) is

$$\Delta\boldsymbol{\Phi}_t = -(\boldsymbol{M}^T \cdot \boldsymbol{M})^{-1} \cdot \boldsymbol{M}^T \cdot \boldsymbol{I}_t. \tag{27}$$

See [5] for the special case of this algorithm for tracking a two-segment articulated object.

[3] Going from (12) to (20), we assume that time is discrete and $\dot{\boldsymbol{\Phi}}$ is replaced by $\Delta\boldsymbol{\Phi}_t$.

[4] These are usually referred to as feature points.

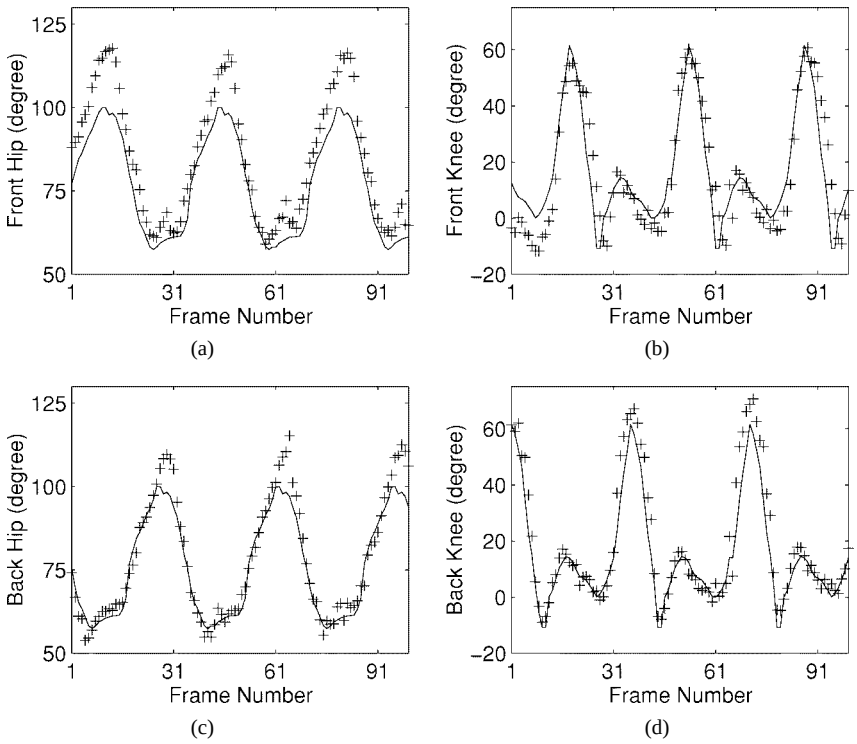


Fig. 7. Estimation results of posture parameters: recognition results are in dashed line; tracking results are in solid lines.

VII. Texture Recovery

The last block of the capture component is the texture recovery block. It extracts from the video the texture for the human walker.

Occlusion and the dynamics in the scene may result in a body part to remain partially unseen in several image frames. To recover the texture, it is necessary to integrate texture patches from several frames and possible different views using a Venn diagram of the view and of the occlusion.

Since we assume that the orientation of the walker with respect to the image plane stays at a constant angle, the projections of the human body parts on the image plane remain almost unchanged during a walking cycle. Therefore, in each walking cycle, we can assign a 2-D template to each body part. This simplifies the 3-D texture recovery to a 2-D problem. For each walking cycle, the 3-D texture is then obtained by integrating the recovered 2-D templates.

Instead of arbitrarily choosing frames from an image sequence as measurements from which to recover the texture for the body parts, we choose the frames based on the data posture vector sequence $\{\Theta_D(k)\}$ obtained from the tracking stage in Section VI. The data posture vector sequence $\{\Theta_D(k)\}$ determines the occlusion of each body part. Observation of human walking patterns suggests that, to recover the texture for occluded body parts, it is sufficient to integrate the texture from the two frames with least occlusion. The frames with the least occlusion are located in the neighborhood of the anchor frames and of the complement anchor frames. We work with the anchor frames A_m and the complement anchor frames A_m^c. Due to lack of space, we do not provide the details of the recovery of the texture.

VIII. Experiments

The capture block of the system has four task modules: preprocessing, recognition, tracking, and texture recovery modules. In addition the system has the synthesis block. The knowledge database supports all these tasks.

The preprocessing module processes the raw video and outputs the motions of the background and of the head-and-torso of the walker. The output motion vectors enable the extraction of the data walkers from the video and the synthesis of their corresponding model walkers. The recognition block then estimates the generic walking parameters, i.e., the period and the phase of the walking, by matching the contours between the model walkers and the data walkers. The resulting walking parameters provide a crude estimate to the walker's posture. The tracking module fine tunes this posture estimate. The fine-tuned posture estimate enables the texture recovery module to extract texture templates for each individual body part from the real video. Finally, the synthesis block synthesizes video by manipulating the video metarepresentation.

We present experimental results for two real-life video sequences: *Pedro* sequence and *Juhn* sequence. For the *Pedro* sequence, Fig. 7(a) and (b) show the rotation angle of the hip joint and the rotation angle of the ankle joint of the front leg, respectively, and Fig. 7(c) and (d) show the rotation angle of the hip joint and the rotation angle of the ankle joint of the back leg, respectively. The dashed lines are the result from the recognition stage in Section V and the solid lines are after the tracking stage in Section VI. There is a significant difference between the dashed lines and the solid lines, which means that the tracking stage significantly modifies the estimation of the angles and of the posture.

(a)

(b)

Fig. 8. *Pedro* sequence. Recognition and tracking results.

Fig. 9. *Juhn* sequence. Tracking results.

(a)

(b)

Fig. 10. Synthesized articulated human sequences.

To judge the quality of these estimates, we contrast the visual look of the patterns recovered from the live video after the recognition stage and tracking stage and compare these patterns with the original video. This comparison is shown in Fig. 8 for the *Pedro* sequence. This figure displays a number of frames of the original video where we superimpose to the original data walker in the frame the contour of the model walker synthesized with the posture recovered either by the recognition or the tracking stages. In the images in Fig. 8(a), the model walker is animated using the posture $\Theta_{\text{fit}}(\cdot)$ de-

termined by the recognition stage described in Section VI. We observe a significant level of mismatch between the data walker and the contour of the model walker. The frames in Fig. 8(b) are generated by animating the model walker using $\Theta_D(\cdot)$, the posture determined by the tracking stage described in Section VI. Contrasting Fig. 8(a) with Fig. 8(b), we conclude that the mismatches after the recognition stage are basically eliminated by the tracking stage.

Fig. 9 illustrates similar results, but only for the tracking stage for the *Juhn* sequence. With this sequence, there is a significant zoom out effect, but the quality of the tracking is again very good. The experiments with these two sequences, as well as other experiments not shown due to lack of space, demonstrate that the system can provide very accurate tracking of human walkers.

With the motion, shape, and texture recovered from the live video, we can synthesize different articulated human sequences. Fig. 10(a) shows frames from a synthesized sequence using the *motion* and texture recovered from the *Pedro* sequence. Fig. 10(b) shows frames from a second sequence synthesized with the texture recovered from the original video but animated with different postures.

IX. Conclusion

In this paper, we presented a system that captures automatically a walking human from a monocular real video sequence. The result is a representation that includes the camera motion, the extended background as seen across the video sequence, the human walker shaped by a stick model with 12 cone shaped body parts, the human walking given by a vector time series that defines the posture of the human in each frame, i.e., the position of each of the body parts, and the body parts texture. The system is a significant tool that can be used in numerous multimedia applications. Extracting the video contents simplifies indexing and retrieval in video databases. The nature of the representation adds functionality to nonlinear video editors like generating articulated human sequences that may differ from the original sequence. Reducing significantly the amount of data describing the video significantly facilitates video transmission and manipulation; see [14], where a similar metarepresentation was used for transmission of video over wireless links. The current system implementation is all in MatLab, so it is slow and far from being real time. It takes currently about 20 min on a alpha 233 MHz station to process each frame. We believe that by coding with a compiled language, with algorithmic optimizations, and with faster processors, the system could become close to being real time. The framework can be generalized beyond the simplifying assumptions underlying the work: single walker, the body parts dimensions stay fixed, and the motion is restricted to a planar walking motion. Including several walkers is easy to handle if the individual walkers appear unoccluded over several frames in the sequence. Body parts with variable dimensions can be handled by increasing the dimensionality of the problem, for example, by including the dimensions of each body part as additional degrees of freedom. To extend the framework to other cyclic human actions like running and jogging, one needs a similar generic model to guide the recognition stage. These models can be obtained by analysis of running or jogging sequences. We are currently exploiting some of these extensions.

References

[1] H. Asada and J.-J. E. Slotine, *Robot Analysis and Control*. New York: Wiley, 1986.

[2] H. G. Barrow, J. M. Tenenbaum, R. C. Bolles, and H. C. Wolf, "Parametric correspondence and chamfer matching: Two techniques for image matching," *Proc. 5th Annu. Int. Joint Conf. Artificial Intelligence*, Aug. 1977, pp. 659–663.

[3] J. F. Canny, "A computational approach to edge detection," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. PAMI-8, pp. 679–698, June 1986. '

[4] J. C. Cheng and J. M. F. Moura, "Model-based recognition of human walking in dynamic scenes," in *Proc. IEEE First Workshop Multimedia Signal Processing*, 1997, pp. 268–273.

[5] ——, "Tracking human walking in dynamic scenes," in *Proc. of IEEE Int. Conf. Image Processing, ICIP'97*, Santa Barbara, CA, vol. 1, pp. 137–140, 1997.

[6] ——, "Automatic recognition of human walking in monocular image sequences," *J. VLSI Signal Process.*, vol. 20, no. 1/2, pp. 107–120, Oct. 1998.

[7] J. C. Cheng, "Capture and representation of human walking in live monocular video," Ph.D. dissertation, Dept. Elect. Comput. Eng., Carnegie Mellon Univ., Pittsburgh, PA, Nov. 1998.

[8] D. M. Gavrila and L. S. Davis, "3-D model-based tracking of human in action: A multi-view approach," in *Proc. IEEE Int. Conf. Computer Vision Pattern Recognition*, CVPR'96, June 1996, pp. 73–80.

[9] D. Hogg, "Model-based vision: A program to see a walking person," *Image Vis. Comput.*, vol. 1, no. 1, pp. 5–20, 1983.

[10] J. M. F. Moura and R. S. Jasinschi, "Content-based video sequence representation," in *Proc. IEEE Int. Conf. Image Processing*, ICIP'95, vol. 2, pp. 229–232, 1995.

[11] ——, "Content based video compression system," U.S. Patent 5 854 856, Dec. 1998.

[12] I. A. Kakadiaris and D. Metaxas, "Model-based estimation of 3D human motion with occlusion based on active multi-viewpoint selection," in *Proc. IEEE Int. Conf. Computer Vision Pattern Recog.*, CVPR'96, June 1996, pp. 81–87.

[13] D. Marr and H. K. Nishihara, "Representation and recognition of the spatial organization of three-dimensional shapes," *Proc. R. Soc. Lond. B*, vol. 200, pp. 269–294, 1978.

[14] J. M. F. Moura, R. S. Jasinschi, H. Shiojiri, and J. C. Lin, "Video over wireless," *IEEE Personal Commun. Mag.*, vol. 3, pp. 44–54, Feb. 1996.

[15] M. P. Murray, "Gait as a total pattern of movement," *Amer. J. Phys. Med.*, vol. 46, no. 1, pp. 290–332, 1967.

[16] J. Ohya, Y. Kitamura, F. Kishino, N. Terashima, H. Takemura, and H. Ishii, "Virtual space teleconferencing: Real-time reproduction of 3D human images," *J. Vis. Commun. Image Represent.*, vol. 6, no. 1, pp. 1–25, 1995.

[17] J. O'Rourke and N. I. Badler, "Model-based image analysis of human motion using constraint propagation," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. PAMI-2, pp. 522–536, Nov. 1980.

[18] Y. Yacoob and M. J. Black, "Parameterized modeling and recognition of activities," in *Sixth Int. Conf. Computer Vision, ICCV'98*, Mumbai, India, Jan. 1998, pp. 120–127.

Jia-Ching Cheng was born in Taiwan, R.O.C. He received the B.S.E. and M.S.E. degrees in electrical engineering from Tatung Institute of Technology, Taipei, Taiwan, in 1987 and 1989, respectively. From 1993 to 1998, he was with Carnegie Mellon University (CMU), Pittsburgh, PA, where he received the Ph.D. degree in electrical and computer engineering from in December 1998.

From 1989 until 1993, he was a Lecturer, Department of Electrical Engineering, Tatung Institute of Technology. Since January 1999, he has been with the Faculty of the Tatung Institute of Technology.

José M. F. Moura (S'71–M'75–SM'90–F'94) received the engenheiro electrotécnico degree in 1969 from Instituto Superior Técnico (IST), Lisbon, Portugal, and the M.Sc., E.E., and the D.Sc. degrees in electrical engineering and computer science from the Massachusetts Institute of Technology (MIT), Cambridge, in 1973 and 1975, respectively.

He is presently a Professor of Electrical and Computer Engineering at Carnegie Mellon University (CMU), Pittsburgh, PA, which he joined in 1986. Prior to this, he was on the faculty of IST, where he was an Assistant Professor (1975), Professor Agregado (1978), and Professor Catedrático (1979). He has held visiting appointments at several Institutions, including MIT (Genrad Associate Professor of Electrical Engineering and Computer Science from 1984 to 1986) and the University of Southern California, Los Angeles, (Research Scholar, Department of Aerospace Engineering, Summers 1978 to 1981). His research interests include statistical signal processing and telecommunications, wavelets and time-frequency transforms, image processing, video representation, video editing, and manipulation. He has over 200 published technical contributions, is the co-editor of two books, holds three patents in the areas of image and video processing, and digital communications with the U.S. Office of Patents and Trade, and has given numerous invited seminars at U.S. and European Universities and Laboratories.

Dr. Moura was elected Vice-President for Publications for the IEEE Signal Processing Society (SPS) in February 1999 for the period 2000–2002 and is an elected member of the Board of Governors of the same Society. He has been on the Editorial Board of the PROCEEDINGS OF THE IEEE since January 1999. Since October 1995, he has been the Editor-in-Chief for the IEEE TRANSACTIONS ON SIGNAL PROCESSING and a member of the IEEE SPS Publications Board. He is currently a member of the Multimedia Signal Processing Technical Committee and a member of the Sensor Array and Multichannel Processing Technical Committee. He was a member of the Underwater Accoustics Technical Committee of the SPS until 1997, a member of the IEEE Press Board from 1991 to 1995, a technical Associate Editor for the IEEE SIGNAL PROCESSING LETTERS from 1993 to 1995, and an Associate Editor for the IEEE TRANSACTIONS ON SIGNAL PROCESSING from 1988 to 1992. He was a Program Committee member for the IEEE International Conference on Image Processing (ICIP'95) and for the IEEE International Symposium on Information Theory (ISIT'93). He has organized and codirected two international scientific meetings on signal processing theory and applications and has been a member of the organizing committee for several other signal processing international technical meetings. He is a corresponding member of the Academy of Sciences of Portugal (Section of Sciences) and a Fellow of the IEEE. He is affiliated with several IEEE societies, Sigma Xi, AMS, IMS, and SIAM.