Journal of VLSI Signal Processing, [Volumn Number], 109–124 (1998)
© 1998 Kluwer Academic Publishers, Boston. Manufactured in The Netherlands.

# Automatic Recognition of Human Walking in Monocular Image Sequences

JIA-CHING CHENG AND JOSÉ M. F. MOURA
*Deptartment of Electrical and Computer Engineering, Carnegie Mellon University,*
*5000 Forbes Ave., Pittsburgh, PA 15213*

moura@ece.cmu.edu

*Received ??; Revised ??*

**Abstract.** In numerous content-based video applications, it is important to extract from a video sequence a representation for humans in motion. This task is difficult, because humans are not rigid objects and they are capable of performing a wide variety of actions. However, often, human movements can be categorized into repetitive and rhythmic patterns of motion. Identifying the motion pattern of a human significantly alleviates the task of construction of its representation. We propose here a model-based recognition of the generic posture of human walking in dynamic scenes. We model the human body as an articulated object connected by joints and rigid parts, and model the human walking as a periodic motion. The recognition task is to fit the model walker sequence to the walker in the live video (data walker sequence). We achieve this by determining the period of the data walker sequence and finding its phase with respect to the model walker sequence. We present promising results of how our system performs with a live video sequence.

## 1. Introduction

Video representation is a key element in a wide variety of applications such as video coding and video manipulation (indexing, editing, composition, and retrieval [2, 9, 10, 11, 20]]. Content-based representations describe a video according to its contents such as motion, shape, and texture. Jasinschi and Moura developed a content-based video manipulation scheme which they referred to as generative video (GV) [9, 10]. GV is a video meta representation. GV decomposes a video sequence into constituent entities according to the relative motion of these entities in the video. For a video with a dominant background, GV generates a background world image (or background template) for the background, and a figure world image (or figure template) for each moving object. Other approaches to video representation can be found in [2, 19].

The success of GV in decomposing video in terms of motion relies on powerful motion estimation and segmentation algorithms. Since motion estimation for 3-D scenes and for non-rigid objects is still among the most difficult challenges in the research area of computer vision and image processing, GV in [9, 10] is restricted to videos with rigid objects moving in very constrained environments.

We explore the representation of *non-rigid* motion in the framework of GV. The domain of interest will be restricted to human movements. Our goal is to create a high-level representation scheme of human movements in a 3-D dynamic scene. It involves solving problems of action recognition, part tracking, part decomposition,and texture recovery. Currently, we study human walking. We focus on the recognition of human walking in this work.

We propose a model-based approach to recognize human walking in *live* videos. We adopt generic models for the human body and human walking. The modeling provides a sequence of walking patterns which we refer to as model walking sequence. The recognition task is now to fit the model walking sequence to the walker in the live video. This is done by searching the model walking sequence by using a contour-based matching method to find the pattern in the model walking sequence that best resembles the walker in each frame.

In Section 2 we illustrate the concept of GV. In Section 3 we describe our approach and review related work in tracking and recognition of human movements. We describe in detail our system in Sections 4-6. In Section 7 we provide experimental results. Finally, in Section 8 we conclude the paper.

## 2. Generative Video

Generative video (GV) [9, 10] describes video according to its contents including motion, shape, and texture. Figure 1 illustrates GV. Suppose that there is a video sequence of 4 frames as shown in the left side of Figure 1. The background is a house. The house moves relative to the camera from right to left. The foreground is a car. The car moves from left to right.

GV decomposes the video into objects with coherent motions; thus there are two objects as shown in the right side of Figure 1. Each object is described by a template, referred to as a world image in [9, 10], and a motion script. A template characterizes the shape and texture of an object. A motion script describes how an object moves in the video. This representation is sufficient to reproduce the original video. Furthermore, it's more compact and meaningful than the original raw video.

Jasinschi and Moura's experiments in [9, 10] show that GV is capable of analyzing videos with 2-D rigid motions. Their method requires the computation of the image velocity (optical flow) field. They identify the moving objects, including the background, from the velocity histograms. The background motion is assumed to dominate, corresponding to the dominating peaks of the velocity histograms. After registering the background, which compensates the background motion, the residual motions correspond to the moving objects. In [9, 10] they identify the moving objects sequentially using this same dominance criterion of the velocity histograms. Finally, they find and determine if a moving object is rigid by using an image correlation method.

For a video with a car moving in the foreground of a street scene as shown in Figure 2 (a), GV extracts the car and assigns to it a template, as shown in Figure 2 (b). The techniques in [9, 10] are restricted to rigid body motions. Figure 3 (a) shows a video with a human holding a mug walking front-and-parallel to a camera. Because the upper body is generally rigid, GV extracts and assigns to it a template, as shown in Figure 3 (b). The legs exhibit nonrigid motion and are self-occlusive. GV in [9, 10] cannot analyze non-rigid motion; as a result, the moving region corresponding to the lower limbs is labeled as a model failure region, see Figure 3 (b), which has to be treated differently.

The goal of this paper is to extend GV to represent *articulated* human motion. We focus on human walking. The challenge lies on the tracking and recognition of human motion.

## 3. Approach

Tracking and recognition of humans and their actions is not a new task in computer vision. Previous work in this area includes [7, 8, 12, 16, 17]. Most systems in this domain resort to model-based approaches. They either adopt an apriori model of the human body [7, 8, 12, 16, 17] or make assumptions on the types of motion of the human [8, 17]. An early attempt to recognize human movements is reported by O'Rourke and Badler [16]. Their system tracks human motion based on constraint propagation. They adopted a 3-D geometrical model of the body, but used synthesized images to simplify low-level feature extraction.

Hogg [8] considered human walking recognition in live videos. He modeled both the human body and the human motion. The human body is de-

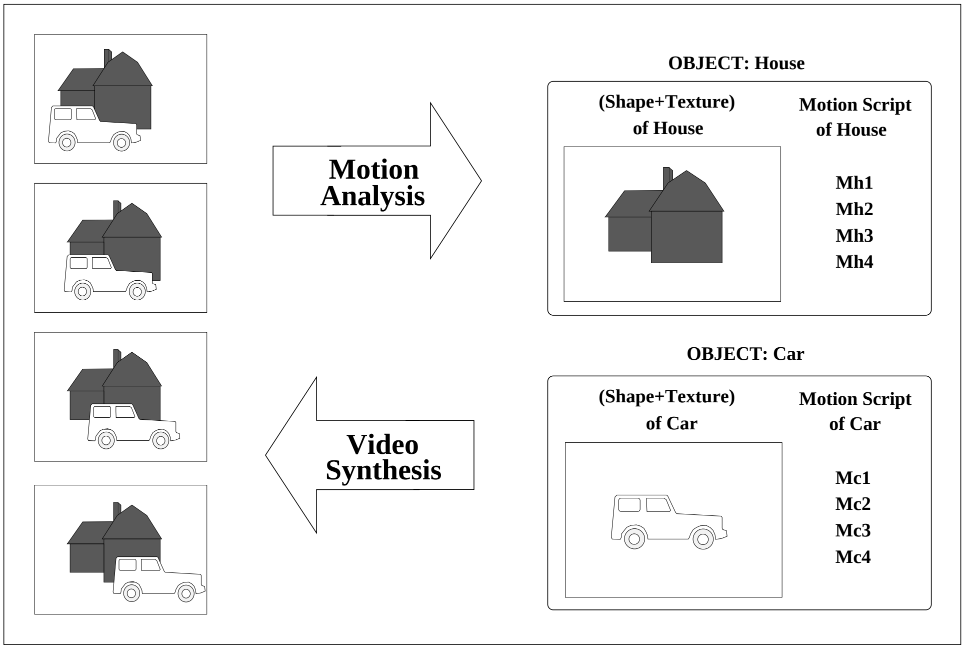


*Fig. 1.* Generative video (GV).

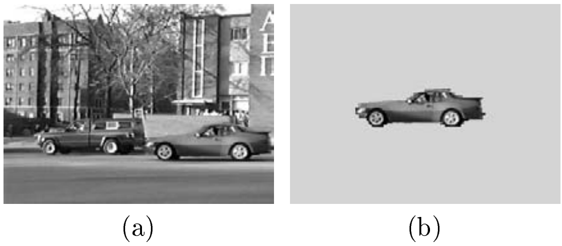


*Fig. 2.* GV's representation of a rigid object.

scribed as a set of elliptical cylinders; the motion model is acquired interactively from a prototype image sequence. A similar approach is taken by Rohr [17]. Rohr also adopted a cylindrical model for the human body. However, Rohr modeled the motion through a time series, averaging the kinematic data provided by the medical motion studies conducted by Murray [15].

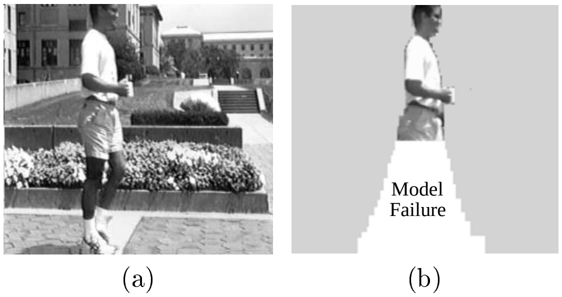


*Fig. 3.* GV's representation of a non-rigid object.

In recently years, several researchers investigated tracking high degree-of-freedom human motion [7, 12]. Gavrila and Davis [7] studied tracking human movements based on a multi-view 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 in order to simplify the extraction of the contours of the human body. Kakadiaris and Metaxas [12] presented a similar multi-view approach for large number of DOF tracking of human body, yet they modeled the body parts of a human as deformable contours.

Due to its complex nature, the human body is non-rigid, it is capable of performing a wide variety of actions, and can be highly self-occlusive. 3-D systems for tracking and recognition of human movements are operated either in very controlled environments or by applying constraints on the movements. The systems of Gavrila and Davis [7] and Kakadiaris and Metaxas [12] can track unconstrained actions, yet they need known initial posture as a start-up and several static cameras to provide sufficient views. On the other hand,

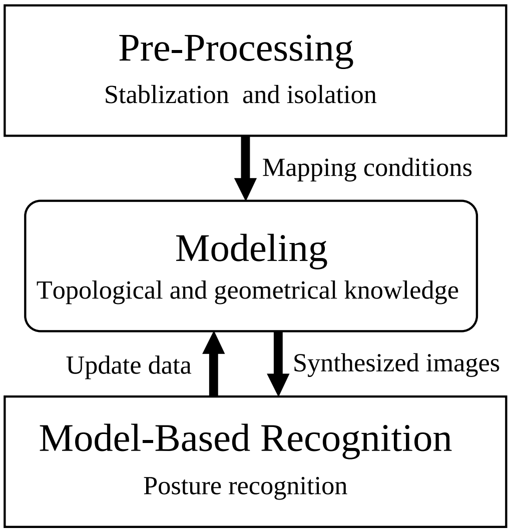


*Fig. 4.* Block diagram for the recognition system.

Hogg's [8] and Rohr's [17] systems can track and recognize human movements in monocular image sequences, but they constrain the domain of human motion to walking, and they require the camera to be stationary.

We propose here a model-based approach to recognize human walking. Our system is capable of recognizing the posture of a walking human in a complex scene. Functionally, it is most closely related to the work of Hogg [8] and Rohr [17]. However, the systems of Hogg [8] and Rohr [17] require a stationary camera, and the human subject walks front-and-parallel to the camera. Our system allows for camera motion during video capturing. The task is made more complicated by the camera mobility.

Our system consists of three components: preprocessing, modeling, and recognition, as shown in Figure 4. The pre-processing stage isolates the walker from the background and estimates the position of the walker. The modeling block contains knowledge about the human body and human walking. It generates useful measurements for the recognition step. The recognition step is the most essential part for analyzing human walking. It recognizes human walking with assistance from the modeling block. It estimates the walking posture by matching edge information extracted from the real image with edge information derived from the model.

We describe each of the three components in detail in the following sections.

## 4. Pre-Processing

The pre-processing component isolates the walker from the background and estimates the position of the walker. Its goal is to extract the walker and track its torso in every frame of a live video sequence.

The implementation is based on a detection-and-pursuit strategy: first, we detect a moving object with a motion-based segmentation method; then, we pursue the detected object.

### *4.1. Detecting Moving Objects*

We assume that the background motion between two image frames is parameterized accurately by a 2-D projective transformation. We estimate the motion of the background for every two consecutive frames. The computation framework is based on an iterative multiscale approach as described in Appendix A.

Once the image background motion has been determined, we register consecutive images using this motion. As a result, we null the image background motion; the remaining motion is due to moving objects. Figure 5 (a) shows a motion detection image. The brighter a pixel is in this motion detection image, the more likely the pixel is to belong to a moving object, and vice-versa. By choosing an appropriate threshold value, we extract regions corresponding to moving objects from the motion detection image. As shown in Figure 5 (b), the result is a binary image, which we refer to as motion detection template. The white areas are moving regions.

### *4.2. Pursuing A Moving Object*

After detecting a moving object, we track the object to obtain its motion information. The object of interest here is a walking human. Experimental evidence reveals that the motion between the head and torso of a walking human is negligibly small; thus we treat these two parts as a single rigid object. We locate the approximate area of

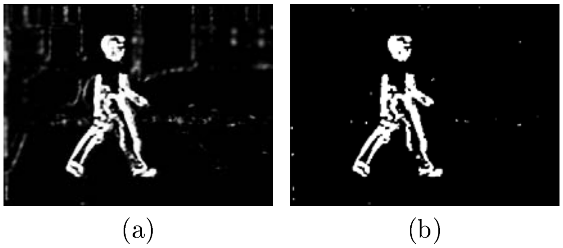


*Fig. 5.* Detecting a walking subject.

the head-and-torso; then track it to obtain its motion information.

First, we adopt an intuitive method to find a rectangular bounding box for the walker, see Figure 6. We assume that the ground is parallel to the horizontal axis of the image plane. We locate the two lateral boundaries by projecting vertically the motion detection template, see the pixel histogram in the horizontal direction at the bottom of Figure 6. This vertical projection is followed by a horizontal projection, see the pixel histogram in the vertical direction at the left of Figure 6. We locate the bounding box from the boundaries of the histograms of Figure 6. We then assume that the head-and-torso are approximately confined to within the upper half of the rectangular box. We estimate the 2-D affine motion of the head-and-torso between two consecutive frames. This gives us the evolution of the 2-D position of the walker between frames.

The output of the pre-processing step consists of two parts:

- The first part is a sequence of motion detection templates. A motion detection template provides the moving regions corresponding to the walker in each frame. As can be seen from the motion detection template shown in Figure 5 (b), the template is not a complete or accurate segmentation of the walker. These templates however will suffice as inputs to the recognition stage described in Section 6. The recognition stage uses a contour-based method. This method requires only information about the edges of the walker which reside within the white regions in the motion detection template.
- The second part is the motion information of the background and the motion information of the head-and-torso of the walker. These motions constitute the mapping conditions which are vital for the synthesis of the model walker.

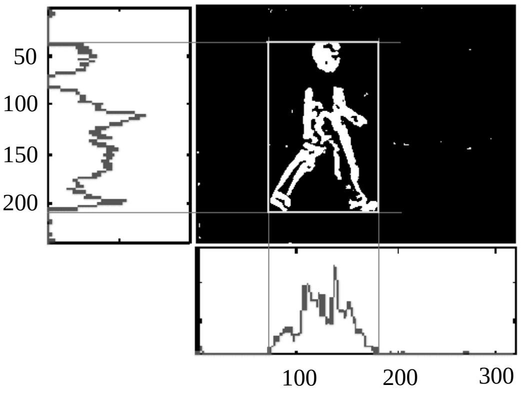


*Fig. 6.* Identifying the position of a walking subject.

## 5. Human Modeling

Human models facilitate the recognition described in Section 6. There are two major components to setting up a model for the human walker: (1) the model of the human body, which provides the geometrical knowledge about the walker; (2) the model of the walking, which provides the topological knowledge about the walker. We assume that these two types of knowledge are known apriori. We use them to synthesize the walker.

### *5.1. Modeling the Human Body*

3-D Graphical modeling of the human body generally consists of two elements: a representation of the skeletal structure (or so-called stick figure), and a representation of the surface surrounding the structure. The stick figure is a collection of segments and joint angles used to specify the position and the configuration of a human body. The surface structure describes the outlook, i.e., physical shape and texture of the human body. A variety of models have been proposed to represent the human body in human animation [18]. An elaborate model with more articulated parts and degrees of freedom as well as a more complicated surface structure generates a more realistic human body; the price paid is that it requires a large number of parameters to represent the human body, increasing the complexity of the estimation of the human movements.

The purpose of our modeling scheme is to generate the contour information of the walker. The 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 [14], which was adopted by Hogg [8] and Rohr [17] in their work. The human body, represented as a stick figure in Figure 7 (a), is considered to be composed of 12 rigid 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, as shown in Figure 7 (b). Each part is therefore described by six parameters, see Figure 7 (b): one for the length of the truncated cone, $l_{cy}$, two for the lengths of the major axes of the ellipses, $l_{el1}$ and $l_{el2}$, one for the ratio of the major axis to the minor axis of the ellipse, $r_{el}$, and another two for the heights of the semi-oval spheres attached to the ends of the truncated cone, $l_{sp1}$ and $l_{sp2}$.

### 5.2. Modeling Human Walking

Our goal is to recognize human movements in live video. The agreement with actual movements is important. For describing human locomotion, there are two basic approaches: kinematic and dynamic [4]. The kinematic approach utilizes general biomechanical knowledge or observations on human locomotion. The dynamic approach regards the human body as a linked structure. Dynamics like Newton-Euler Mechanics are used to simulate the movements governed by muscles. This method requires knowledge of the internal forces and torques for stimulating the movements. The kinematic approach is much simpler than the dynamic approach, yet it still provides an adequate representation for human movements. It is therefore a more plausible choice for modeling human motion in live videos.

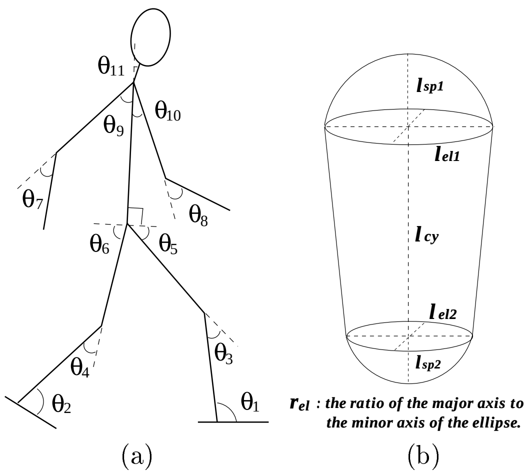


*Fig. 7.* Model of the human body.

We adopt the kinematic approach in modeling the human movements. Previous research in the biomechanics of human locomotion [15] provides useful measurements for modeling human walking. Murray [15] conducted experiments on measuring gaits of males and females in a wide range of ages and heights. These results reveal that the movement patterns of different body parts are similar for different people. Rohr [17] used the average measurements of the movement patterns [15] in his work. Encouraged by his results, we adopt the same set of measurements in modeling the human walking.

The stick model shown in Figure 7 (a) has 11 joints and joint angles $\theta_i$, $(i = 1, 2, \cdots, 11)$. Murray [15] considered human walking as an articulated motion with 10 DOF, which are $\theta_1$ to $\theta_{10}$. In our model, we add an extra DOF, a joint angle between the neck and the torso, i.e., $\theta_{11}$. Due to the symmetry of walking, Murray only measured five joint angles: $\theta_1$, $\theta_3$, $\theta_5$, $\theta_7$, and $\theta_9$. The other five joint angles are derived by symmetry from those five measured joint angles. Reference [15] presents the time series for one cycle, averaged over 30 normal individuals, for each of these five joint angles.

We sample these averaged time series at equally-spaced time instants. We collect these equally-spaced samples at time instant $p$ in the model posture vector

$$\Theta_M(p) \overset{df}{=} \left[\, \theta_{M1}(p) \;\; \theta_{M2}(p) \;\; \theta_{M3}(p) \;\; \cdots \right. \\ \left. \cdots \;\; \theta_{M9}(p) \;\; \theta_{M10}(p) \;\; \theta_{M11}(p) \,\right]^T \tag{1}$$

The time index $p \in [0, 1)$ of the time series of the joint angles is usually referred to as the pose.

The joint angle time series are periodic with period of 1. Figure 8 shows time series of the joint angles for the hip and knee, $\theta_{M3}$ and $\theta_{M5}$, respectively. The joint angle time series are our motion model (prior knowledge). For live videos with different walking subjects, these series need to be adjusted. For example, different people, even the same human with different walking speed, produce different stride sizes. To make this model more realistic, we modulate the set of joint angle time series according to the stride of the walker. We compute the stride for the data walker in the $k$-th frame, $w_D(k)$, from the width of the histogram obtained by vertical projection of the motion detection template, see Section 4. We determine the stride $w_M(p)$ of the model walker at pose $p$ by a similar procedure. First, we synthesize the model walker sequence using the body model introduced in Section 5.1 and the walking model provided by the joint angle time series described in this subsection. For each model walker, we isolate the model walker from the background to obtain a body template. We then project vertically the body template to obtain the stride of the model walker. It turns out that the model walker of pose $p = 0$ has the largest stride.

We define the ratio of the widest stride of the data walker to the widest stride of the model walker as the stride ratio, $r_s$,

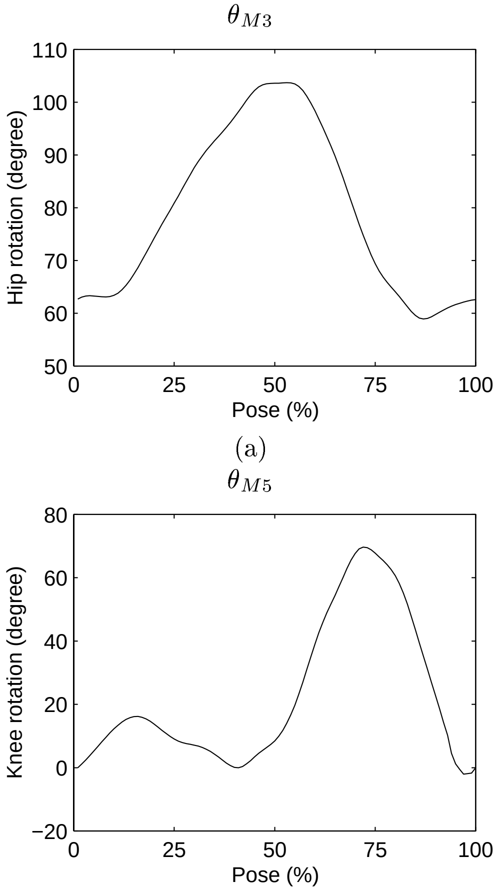
$$r_s = \frac{\max\limits_{1\leq k\leq K} w_D(k)}{\max\limits_{0\leq p<1} w_M(p)} = \frac{\max\limits_{1\leq k\leq K} w_D(k)}{w_M(p=0)} \tag{2}$$

We modify the joint angle time series $\theta_{Mi}$ using the stride ratio $r_s$. The resulting modified joint angle time series $\tilde{\theta}_{Mi}(p)$ are given by

$$\tilde{\theta}_{Mi}(p) = \begin{cases} r_s(\theta_{Mi}(p) - \frac{\pi}{2}) + \frac{\pi}{2} \\ \qquad\qquad \text{if } i \in \{1, 2, 5, 6\} \\ r_s\theta_{Mi}(p) \\ \qquad\qquad \text{if } i \in \{3, 4, 7, 8, 9, 10\} \end{cases} \tag{3}$$

The modified posture $\tilde{\Theta}_M(p)$ is defined similarly to equation (1).

The net result of the modeling block is the synthesis of the model walker sequence. Figure 9 (b) shows three model walkers at poses 0.3, 0.6, and 0.9. The model walker sequence will be used in conjunction with the data walker sequence in the recognition stage of next section.



(a)

(b)

*Fig. 8.* Two joint angle time series for walking.

## 6. Recognition of Human Walking

The task of recognition is to fit the sequence modeling the walking to the walker in a live video. We achieve this by determining the period of the data walker sequence and finding its phase with respect to the model walker sequence.

We defined two types of walkers: the data walker and the model walker. The data walker, $W_D(k)$, is detected in the pre-processing step described in Section 4 from the live video, where $k$ is the corresponding frame number. The model walker, $W_M(p)$, is synthesized in the modeling block using the modified posture, $\tilde{\Theta}_M(p)$, where $p \in [0, 1)$ is the pose. Figure 9 (a) shows some frames of the data walker, $W_D(k), k = 10, 20, 30$, in a live video sequence. Figure 9 (b) shows the model walker at different poses, $W_M(p), p = 0.30, 0.60, 0.90$. The challenge is now to determine for each data walker frame $W_D(k)$ what is the pose of the corresponding model walker $W_M(p)$.

Our recognition algorithm is a contour-based method. Following are two important reasons why we use a contour-based method: (i) we only have apriori shape information about the walker and lack information representing the texture of the walker; (ii) the edge information is usually more reliable and robust than the texture information.

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 motion detection template, see Section 4. We refer to these edge images shown in Figure 10 as edge maps, $E_D(k)$. The edge maps $E_D(k)$ are generated by first applying the Canny edge detector [5] to the data walker, $W_D(k)$, and then by removing the clutter using the motion detection template.

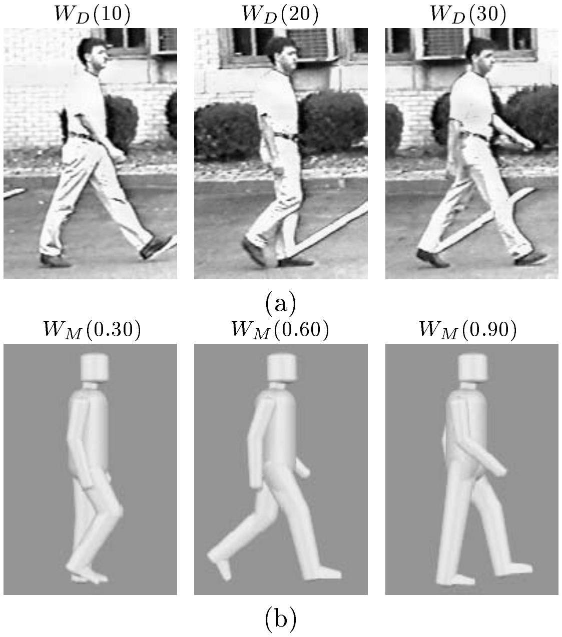


*Fig. 9.* Frames in data and model walker sequences.

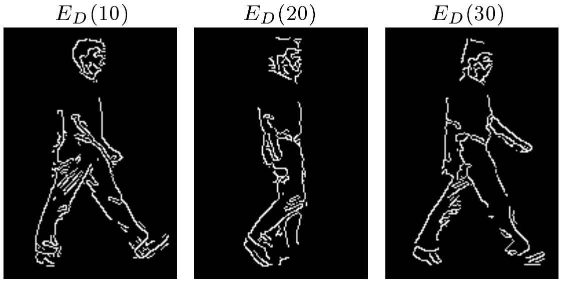


*Fig. 10.* Edge maps for the data walker.

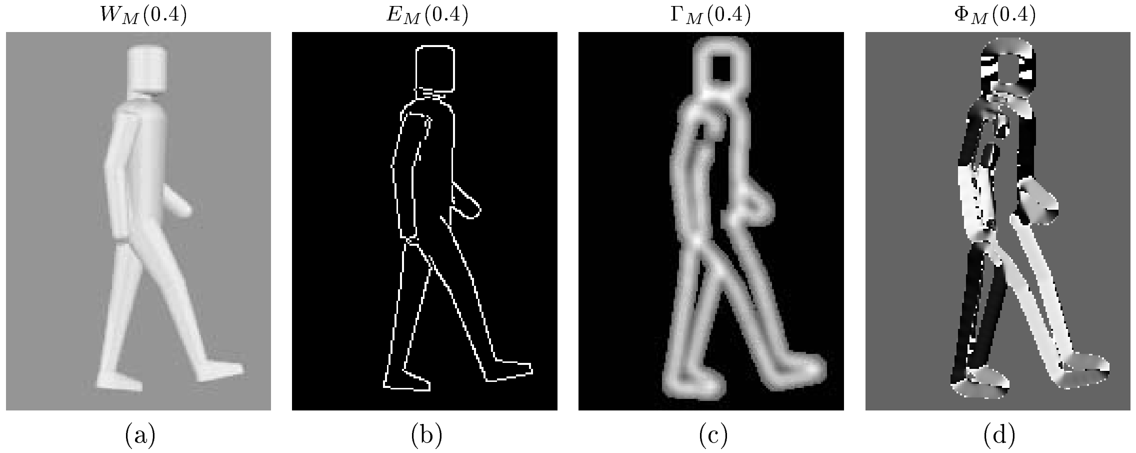
Our model of the human body is a generic cone-shaped model; therefore, the edge maps of the model walker basically consist of piece-wise straight lines. The edge maps of the data walker generated from live video usually have a large number of cluttered edges and broken edges. It is difficult to extract straight lines from the edge maps of the data walker accurately. Therefore, instead of adopting a line-fitting matching method as in [17], for instance, we propose a matching method which does not require extraction of high-level features from the edge maps. Our method modifies the chamfer matching in [3] as we explain below.

Given a data walker $W_D(k)$, we estimate the pose of the corresponding model walker $W_M(p)$ by matching the edge information of the data walker with the edge information of the model walker. This is done by a generate-and-test approach. Starting with the data walker $W_D(k)$, we search the pose space of the model walker, i.e., the model walker sequence from $p = 0$ to $p = 1$, to estimate the pose $p_{sim}(k)$ corresponding to the model walker posture that is closest to the data walker. Closest is defined in terms of maximizing a similarity measure, $s(W_D(k), W_M(p))$, which quantifies how close the data walker $W_D(k)$ is to a model walker $W_M(p)$.

The similarity measure $s(W_D(k), W_M(p))$ requires knowledge of an edge map $E_D(k)$ and a phase map $\Phi_D(k)$ for the data walker $W_D(k)$, as well as of an edge map $E_M(p)$, a phase map $\Phi_M(p)$, and a distance map $\Gamma_M(p)$ for the model walker $W_M(p)$ as described next.

### 6.1. *Distance and Phase Maps*

For the model walker with pose $p$, $W_M(p)$, we create the edge map $E_M(p)$ by using the Canny edge detector [5]. From the edge map $E_M(p; x, y)$, we then construct the distance map $\Gamma_M(p; x, y)$, which indicates the distance of a pixel to its clos-


$W_M(0.4)$ $E_M(0.4)$ $\Gamma_M(0.4)$ $\Phi_M(0.4)$

(a) (b) (c) (d)

*Fig. 11.* Model walker, and its edge, distance, and phase maps.

est edge pixel. The distance map is defined by

$$\Gamma_M(p;x,y) = \begin{cases} (1-\alpha) + \frac{\alpha}{\delta_\Gamma}(\delta_\Gamma - \min\limits_{e\in E_M} \|e,(x,y)\|) \\ \qquad \text{if } \min\limits_{e\in E_M} \|e,(x,y)\| \leq \delta_\Gamma \\ 0 \quad \text{otherwise} \end{cases} \tag{4}$$

where $(x,y)$ is the position of a pixel, $\alpha \in [0,1]$, $e$ is the position of an edge pixel in $E_M(p)$, and $\delta_\Gamma$ is a given threshold.

The phase map $\Phi_M(p;x,y)$ collects the orientation of the edges in the edge map $E_M(p;x,y)$. It is defined by

$$\Phi_M(p;x,y) = \begin{cases} \tan^{-1}\frac{\nabla_y(W_M*G)}{\nabla_x(W_M*G)} \\ \qquad \text{if } \min\limits_{e\in E_M} \|e,(x,y)\| \leq \delta_\Gamma \\ 0 \quad \text{otherwise} \end{cases} \tag{5}$$

where $\nabla_x$ and $\nabla_y$ are the components of the gradient operator and $G$ is a Gaussian lowpass filter.

Figures 11 (b), (c) and, (d) show in gray level the edge map, the distance map, and the phase map, respectively, for the model walker with pose $p = 0.4$, $W_M(0.4)$, in Figure 11 (a). Note that the pixel values of the distance map and the phase map have been linearly converted from $[-\pi, \pi]$ to $[0, 255]$ and from $[0, 1]$ to $[0, 255]$, respectively, for visualization purposes.

The distance map indicates the distance of a pixel to its nearest edge. The brighter the gray level of a pixel is, the smaller is its distance to its closest edge. The distance map is modified from the chamfer image [3], which is virtually the same as our distance map when $\alpha = 1$ and $\delta_\Gamma$ is sufficiently large. The parameter $\delta_\Gamma$ is used to regulate the potential region, which is defined as the region consisting of pixels with distances less than or equal to $\delta_\Gamma$ to the contour. Only pixels inside the potential region have non-zero values. The parameter $\alpha$ is defined as the sensitivity of the distance map. The sensitivity is at its lowest when $\alpha = 0$. In this case, the intensity of every pixel inside the potential region has the same value, regardless of its actual distance to the contour. The sensitivity increases as $\alpha$ increases.

The phase map is derived from the gradient of a blurred model walker; it possesses the orientation information of the edge map. Dark pixels have a phase close to $-\pi$, indicating that their orientations point upward, while bright pixels have a phase close to $\pi$, indicating that their orientations point downward. The other edge pixels have orientations in between.

We use these two maps as geometry filters to measure the geometric similarity between the model walker and a data walker. As mentioned, functionally, our distance map is similar to the chamfer image [3] used for measuring the similarity between two sets of edges. The chamfer matching method in [3] computes the similarity between two sets of edges by only measuring the distance between them. It doesn't consider the orientation information between these edges, which we believe is as important as the distance information. Our phase map provides this information by measuring the orientation between these two sets of edges.

Similarly, from the edge map $E_D(k)$ and the data walker $W_D(k)$, the phase map $\Phi_D(k;x,y)$ for the data walker is defined by

$$\Phi_D(k;x,y) = \begin{cases} \tan^{-1}\frac{\nabla_y(W_D * G)}{\nabla_x(W_D * G)} & \text{if } (x,y) \in E_D \\ 0 & \text{otherwise} \end{cases} \tag{6}$$

### 6.2. *Similarity Measure*

For the data walker in frame $k$, $W_D(k)$, we search the pose space of the model walker, $W_M(p)$, $p \in [0,1)$, to estimate the pose, $p_{sim}(k)$, of the posture which is closest to the data walker $W_D(k)$, by maximizing a similarity measure $s(W_D(k), W_M(p))$

$$p_{sim}(k) = \arg\max_{p \in [0,1)} s(W_D(k), W_M(p)) \tag{7}$$

The similarity measure $s(W_D(k), W_M(p))$ indicates for each pair $(k,p)$, where $k$ is the frame index and $p$ is the pose, the geometric resemblance in terms of edges between $W_D(k)$ and $W_M(p)$. It is defined as

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

where

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

where $\delta_\Phi$ is a given threshold. We call the procedure defined by $S_M(k,p;x,y)$ in the equation above phase filtering. For each edge pixel of the data walker $W_D(k)$, phase filtering compares the phase of each edge pixel of the data walker, $\Phi_D(k;x,y)$, with the phase of its corresponding pixel of the model walker, $\Phi_M(p;x,y)$. As a result, phase filtering only preserves those edges in the data walker that have phase (orientation) similar to the phase of their corresponding pixels of the model walker. We then sum the distance values of those preserved edges to obtain the similarity measure.

We illustrate the matching process in Figure 12. Figure 12 (a) is the edge map of a data walker. We match it to the model walker whose distance map is shown in Figure 12 (b). Figure 12 (c) shows the edge map of the data walker superimposed to the distance map of the model walker in Figure 12 (b). Since the two postures differ from each other, only a few edges of the data walker fall within the nonzero region of the distance map. Also we need to verify if the phase of those edges falling in the overlap region is similar to the phase of their counterpart points in the model walker. Figure 12 (d) shows the result after phase filtering. Since most of the edges of the data walker are removed during phase filtering, see Figure 12 (d), the similarity measure is very small. In other words, the result reveals that the model walker of Figure 12 (b) is not a good match to the data walker of Figure 12 (a). Note that phase filtering has eliminated almost all edges in the intersection of the legs of the model walker and data walker, though the distance of those eliminated edges of the data walker to the edges of the model walker is very small. If we just used the distance map as the chamfer matching method does in [3], these edges would contribute erroneously to the similarity measure.

### 6.3. *Fittest Posture*

We search the pose space of the model walker to find the closest pose, $p_{sim}(k)$, for the data walker in a number of consecutive frames, $W_D(k), k = 1, 2, \cdots, K$, by maximizing the similarity measure defined in (8); then, determine the period, $T_p \overset{df}{=} {f_p}^{-1}$, (in frames/cycle) from the fundamental frequency $f_p$, and the phase, $\phi_p$, (or the pose of the walker in the first frame of the video) by a line fitting algorithm

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

We designate $p_{fit}(k) \overset{df}{=} {f_p}^*(k-1) + {\phi_p}^*$ to be the fittest pose of the data walker $W_D(k)$, and $\Theta_{fit}(k) \overset{df}{=} \Theta_M(p_{fit}(k))$ the fittest posture.

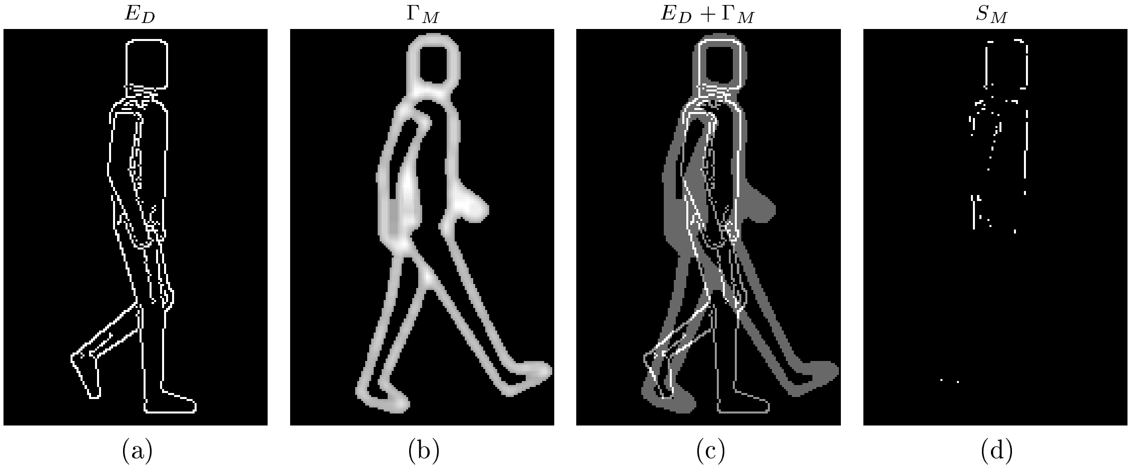


*Fig. 12.* Illustrating the result of phase filtering.

## 7. Experiments

We present results on recognizing the posture of a walker in the *Pedro* sequence. The *Pedro* sequence is a live video of an outdoor scene. We first apply the pre-processing stage described in Section 4 to extract from each image the walking human, which we refer to as the data walker. Then, we apply our recognition algorithm described in Section 6 to the first 30 frames of the *Pedro* sequence. For each frame of the data walker, the recognition step generates a phase map for the data walker; then searches the pose space to: synthesize a model walker; generate a distance map and a phase map for the model walker; maximize the similarity measure to find the closest pose; and, finally, line-fit the data of the estimated closest pose for all these 30 frames of the data walker to determine the frequency and phase of the walking posture. To test the robustness of our approach, we estimate the closest pose for the data walker in each of the 30 frames by searching the entire pose space with a pose increment of 0.01, in other words, by best matching to a posture in a sequence of 100 model walker frames.

Figure 13 shows the results of matching the data walker of Frame 2. The horizontal axis is the value of the pose indicated as a percentage of the period in a walking cycle. The vertical axis is the similarity measure defined in (8). The result suggests that the model walker with pose of 0.77 is the best match to the data walker. We may notice in Figure 13 that there is another peak centered around the 0.26 pose, which is about half a period apart from the major peak. This is due to the symmetric characteristic of walking. This large secondary peak may cause large errors (or outliers), see below.

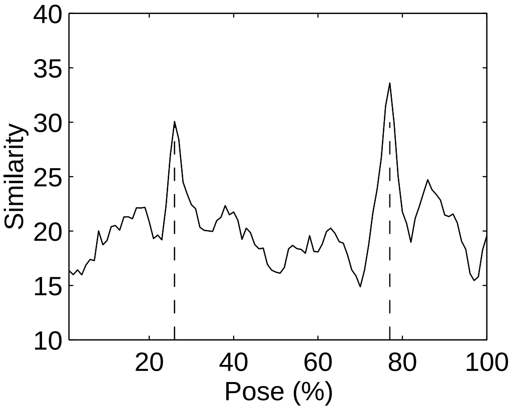


*Fig. 13.* Similarity measure for the data walker frame 2 in *Pedro* sequence.

We perform the matching mentioned above on the data walkers for Frame 1 through Frame 30. The result is shown in Figure 15. As can be seen, most of the data points scatter around a straight line except for three outliers, the three data points corresponding to Frames 14, 18, and 23. The outliers are due to the symmetric characteristic of walking as discussed above. These three data points will fall within the desired range if we compensate them by ±0.50. We then determine

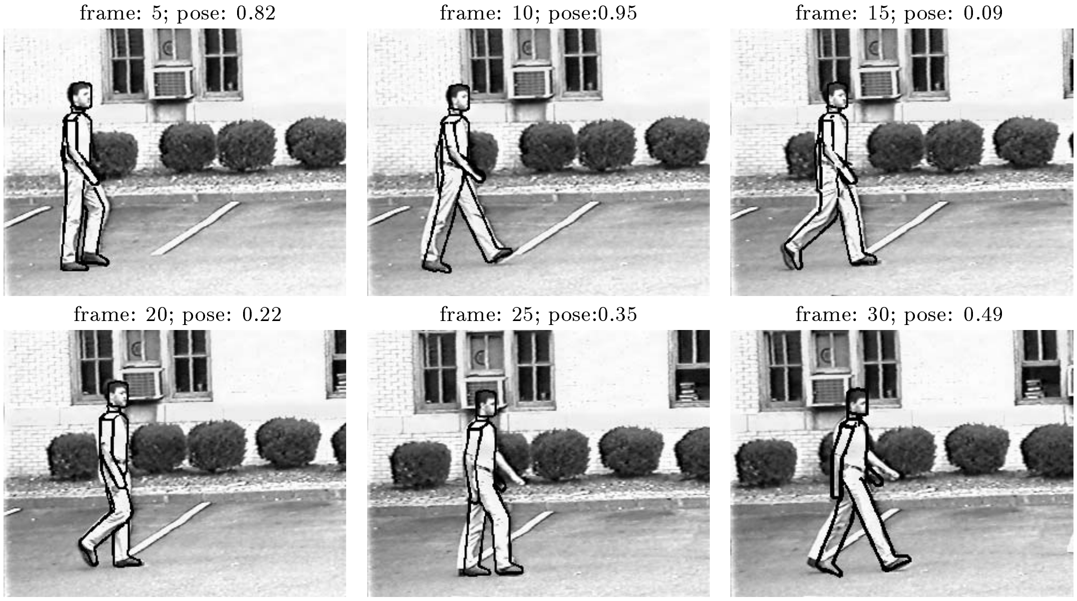


*Fig. 14.* Recognition results for *Pedro* sequence.

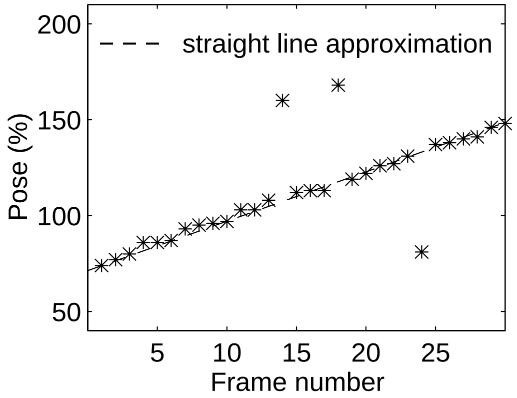


*Fig. 15.* Line-fitting to obtain frequency and phase for data walker sequence in *Pedro* sequence.

the period and the phase of the posture for the data walker by applying equation (9). We obtain $f_p = 0.0267$ and $\phi_p = 0.7129$. This result shows that the fittest posture of the walker in frame $k$ of the *Pedro* sequence is $\Theta_{fit}(k) = \Theta_M(p_{fit}(k))$ where $p_{fit}(k) = 0.0267(k-1) + 0.7129$. This indicates that the fittest pose for Frame 1 is $p_{fit}(1)$ $= 0.7129$, and that the period of the walking cycle is $T_p \cong 37.45$ frames/cycle. We then superimpose the contour of the approximate model walkers to their corresponding data walkers. Some of the resulting images are shown in Figure 14. These results represent very fair recognition of the walking posture.

## 8. Conclusions

Content-based representation of human movements in live videos describes video according to the motion, shape, and texture of the human subject. It involves solving the problems of action recognition, part tracking, part segmentation, human modeling, and texture recovery. In this paper, we focus on action recognition. We propose a model-based recognition scheme for estimating the posture of a walking human. We model the human body as an articulated object connected by joints and rigid parts, and the human walking as a periodic motion. We estimate the posture by matching the edge information obtained from the live video sequence with the edge information derived from the model. We test the algorithm on live video with promising results. In [6], we extend this work by using the estimated posture of the action recognition stage described in this paper as the initial posture for tracking the body parts of the walking human.

## Appendix A

### Motion Estimation: Projective Model

Under perspective projection, the projective transformation

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} \frac{p_1 x \, + \, p_2 y \, + \, p_3}{p_7 x \, + \, p_8 y \, + \, 1} \\ \frac{p_4 x \, + \, p_5 y \, + \, p_6}{p_7 x \, + \, p_8 y \, + \, 1} \end{bmatrix} \tag{A1}$$

provides exact parameters to account for all the possible camera motions. However, this set of parameters is difficult to compute. Assume that the camera's field of view is relatively narrow [1], the projective transformation can be approximated by low-order polynomials such as the pseudo-perspective transformation

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} q_1 + (1+q_2)x + q_3 y + q_7 x^2 + q_8 xy \\ q_4 + q_5 x + (1+q_6)y + q_7 xy + q_8 y^2 \end{bmatrix} \tag{A2}$$

or the affine transformation

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} a_1 + (1+a_2)x + a_3 y \\ a_4 + a_5 x + (1+a_6)y \end{bmatrix} \tag{A3}$$

When the motion is small, the pseudo-perspective transformation is a good approximation and it is more stable to compute than the projective transformation; we therefore estimate the pseudo-perspective transformation as an intermediate step in our algorithm to estimate the projective transformation parameters.

Figure A.1 depicts the algorithm for estimating the projective transformation parameters. The approach is similar to that of Mann and Picard [13]. First, we estimate the 8-parameter pseudo-perspective transformation by a gradient-based method, then we determine the 8-parameter projective transformation from the parameters of the pseudo-perspective transformation by using a simple conversion method, and finally we register the frames with the projective motion parameters. The three processes are performed iteratively until a satisfactory result is obtained. We implement this algorithm in a multiscale strategy. We describe below the estimation algorithm for pseudo-perspective motion parameters, the conversion algorithm, and the multiscale implementation strategy.

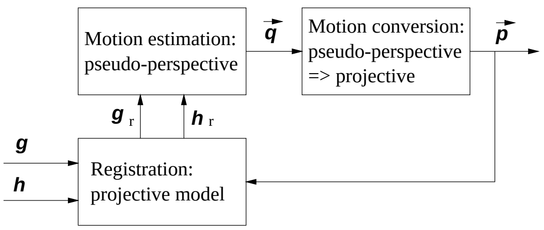

Motion estimation: pseudo-perspective
$\vec{q}$
Motion conversion: pseudo-perspective => projective
$\vec{p}$
$g_r$
$h_r$
$g$
$h$
Registration: projective model

*Fig. A.1.* Algorithm for estimating the parameters of the projective transformation.

#### *Estimation of Pseudo-Perspective Motion*

The pseudo-perspective transformation describes the motion between two consecutive frames as

$$I(x,y,t') \overset{df}{=} I(x+u(\mathbf{x},\vec{q}), y+v(\mathbf{x},\vec{q}), t) \tag{A4}$$

where $[u(\mathbf{x},\vec{q}) \; v(\mathbf{x},\vec{q})]^T$ is the motion field

$$\begin{aligned} \begin{bmatrix} u \\ v \end{bmatrix} &\overset{df}{=} \begin{bmatrix} x'-x \\ y'-y \end{bmatrix} \\ &= \begin{bmatrix} q_1 + q_2 x + q_3 y + q_7 x^2 + q_8 xy \\ q_4 + q_5 x + q_6 y + q_7 xy + q_8 y^2 \end{bmatrix} \end{aligned} \tag{A5}$$

The model requires the estimation of 8 parameters. We estimate the image velocity $\vec{q} \overset{df}{=} [q_1 \; q_2 \; q_3 \; q_4 \; q_5 \; q_6 \; q_7 \; q_8]^T$ by minimizing the cost function

$$C(\vec{q}) \cong \sum_{(x,y)\in R} (I(x,y,t') - I(x+u, y+v, t))^2 \tag{A6}$$

Under the assumption of small motion, we omit the higher order terms of the Taylor series expansion of $I(x+u, y+v, t)$. Then, we have

$$\begin{aligned} C(\vec{q}) &= \sum_{(x,y)\in R} (I_t - uI_x - vI_y)^2 \\ &= \sum_{(x,y)\in R} (I_t - \mathbf{m}(x,y)\vec{q}) \end{aligned} \tag{A7}$$

where $\mathbf{m}(x,y) \overset{df}{=} [I_x \; xI_x \; yI_x \; I_y \; xI_y \; yI_y \; (x^2 I_x + xyI_y) \; (xyI_x + y^2 I_y)]$, $I_x \overset{df}{=} \frac{\partial I(x,y,t)}{\partial x}$, $I_y \overset{df}{=} \frac{\partial I(x,y,t)}{\partial y}$, and $I_t \overset{df}{=} \frac{\partial I(x,y,t)}{\partial t}$.

By denoting the vectors $\mathbf{M}$ and $\mathbf{I}_t$ as

$$\mathbf{M} = \begin{bmatrix} \mathbf{m}(x_1, y_1) \\ \mathbf{m}(x_2, y_2) \\ \vdots \\ \mathbf{m}(x_N, y_N) \end{bmatrix}, \; \mathbf{I}_t = \begin{bmatrix} I_{t_1} \\ I_{t_2} \\ \vdots \\ I_{t_N} \end{bmatrix} \tag{A8}$$

we then rewrite equation (A7) as

$$C(\vec{q}) = \|\mathbf{M}\vec{q} - \mathbf{I}_t\|^2 \tag{A9}$$

The least squares solution to equation (A9) is

$$\vec{q} = -(\mathbf{M}^T\mathbf{M})^{-1}\mathbf{M}^T\mathbf{I}_t \tag{A10}$$

provided that the matrix $\mathbf{M}^T\mathbf{M}$ is invertible.

*Conversion from Pseudo-Perspective to Projective*

After obtaining a set of pseudo-perspective motion parameters, we establish the correspondence between the images. We choose corresponding pairs of points by applying the pseudo-perspective motion parameters determined earlier

$$\begin{bmatrix} x_k \\ y_k \end{bmatrix} \overset{T_{\vec{q}}}{\Longrightarrow} \begin{bmatrix} x'_k \\ y'_k \end{bmatrix} \qquad k = 1, \cdots, K \tag{A11}$$

where $T_{\vec{q}}$ denotes the pseudo-perspective transformation of equation (A5), and $k$ is the index for the corresponding pairs of points. We assume that these pairs of points also comply with the projective motion model, i.e.,

$$\begin{bmatrix} x'_k \\ y'_k \end{bmatrix} = \begin{bmatrix} x_k & y_k & 1 & 0 & 0 & 0 & -x_k x'_k & -y_k x'_k \\ 0 & 0 & 0 & x_k & y_k & 1 & -x_k y'_k & -y_k y'_k \end{bmatrix} \vec{p} \tag{A12}$$

where $\vec{p} \overset{df}{=} [p_1 \; p_2 \; p_3 \; p_4 \; p_5 \; p_6 \; p_7 \; p_8]^T$, $p_i, i = 1, \ldots, 8$, are the projective motion parameters defined in equation (A1). Since there are 8 unknowns to be determined, we need at least four corresponding point-pairs. By choosing four corner points from the image at time $t$, and finding their corresponding points in the image at time $t'$, we obtain $\vec{p}$ by solving a system of 8 equations in 8 unknowns.

*Multiscale Implementation*

We implement the algorithm to estimate the motion parameters in a multiscale manner. We construct a Gaussian pyramid for each of the two images $g = I(t)$ and $h = I(t')$. In this strategy, the projective motion parameters are determined at each level of the pyramid and propagated down to the next finer level. We summarize the whole procedure as follows:

1. *Initialization*: Construct an $L$-level Gaussian pyramid for each of the two images, $g$ and $h$.
2. *Iteration*: Start from the coarsest level, $l = 1$. Perform the following steps in sequence, $l = 1, \cdots, L$.
   (a) **Estimation**: Estimate $\vec{q_l}$, the 8-parameter pseudo-perspective motion between the two images, $g_l$ and $h_l$.
   (b) **Conversion**: Convert $\vec{q_l}$ to $\vec{p_l}$, the 8-parameter projective motion.
   (c) **Iteration**: Verify the fitness of the newly determined projective motion. Repeat the two steps above till a satisfactory result is obtained.

## References

1. G. Adiv, "Determining 3D motion and structure from optical flow generated by several moving objects," *IEEE Trans. Pattern Anal. Machine Intell.*, **7**, pp. 384-401, July 1985.
2. P. Anandan, M. Irani, R. Kumar, and J. Bergen, "Video as an image data source: efficient representations and applications," in *Proc. IEEE Conf. on Image Processing*, Washington, DC, **1**, pp. 318-321, 1995.
3. H. G. Barrow, J. M. Tenenbaum, R. C. Bolles, and H. C. Wolf, "Parametric correspondence and chamfer matching: two techniques for image matching," in *Proc. of the 5th Annual Int. Joint Conf. on Art. Intell.*, pp. 659-663, Aug. 1977.
4. N. Badler, C. B. Phillips, and B. L. Webber, *Simulating Humans: Computer Graphics Animation and Control*, Oxford University Press, New York, 1993.
5. J. F. Canny, "A computational approach to edge detection," *IEEE Trans. Pattern Anal. Machine Intell*, **8**(6), pp. 679-698, 1986.
6. J. C. Cheng and J. M. F. Moura, "Tracking human walking in dynamic scenes," in *Proc. IEEE Conf. on Image Processing*, Santa Barbara, CA, **1**, pp. 137-140, Oct. 1997.
7. D. M. Gavrila and L. S. Davis, "3-D model-based tracking of human in action: a multi-view approach," in *Proceedings of IEEE CVPR*, pp. 73-80, June 1996.

8. D. Hogg, "Model-based vision: a program to see a walking person," *Image and Vision Computing*, **1**(1), pp. 5-20, 1983.
9. R. S. Jasinschi, Generative Video: a Meta Video Representation, *Ph.D. dissertation, Dept. of Elec. and Comp. Engineering, Carnegie Mellon University*, July 1995.
10. R. S. Jasinschi and J. M. F. Moura, "Generative Video: a meta video representation," **note:** submitted to *IEEE Trans. Image Processing*, June 1996, 30 pages.
11. R. S. Jasinschi and J. M. F. Moura, "Content-based video sequence representation," in *Proc. IEEE Conf. on Image Processing*, Washington, DC, **2**, pp. 229-232, 1995.
12. I. A. Kakadiaris and D. Metaxas, "Model-based estimation of 3D human motion with occlusion based on active multi-viewpoint selection," in *Proceedings of IEEE CVPR*, pp. 81-87, June 1996.
13. S. Mann and R. W. Picard, "Video orbits of the projective group: a new perspective on image mosaicing," *Technical Report No. 338, MIT Media Lab*, 1995.
14. D. Marr and H. K. Nishihara, "Representation and recognition of the spatial organization of three-dimensional shapes," in *Proc. R. Soc. London B*, **200**, pp. 269-294, 1978.
15. M. P. Murray, "Gait as a total pattern of movement," *American Journal of Physical Medicine*, **46**(1), pp. 290-332, 1967.
16. J. O'Rourke and N. I. Badler, "Model-based image analysis of human motion using constraint propagation," *IEEE Trans. Pattern Anal. Mach. Intell.*, **2**(6), pp. 522-536, Nov. 1980.
17. K. Rohr, "Toward model-based recognition of human movements in image sequences," *CVGIP: Image Understanding*, **59**(1), pp. 94-115, 1994.
18. N. M. Thalmann and D. Thalmann, *Models and Techniques in Computer Animation*, Springer-Verlag, Tokyo, 1993.
19. J. Y. A. Wang and E. H. Adelson, "Representing moving images with layers," *IEEE Trans. Image Proc.*, **3**(5), pp. 625-638, Sep. 1994.
20. H. Zhang and Q. Tian, "Digital video analysis and recognition for content-based access," *ACM Comput. Surv.*, **27**(4), pp. 643-644, 1995.

**Jia-Ching Cheng** was born in Taipei, Taiwan. He is currently a research assistant working towards the Ph.D. degree in Electrical Engineering at the Department of Electrical and Computer Engineering, Carnegie Mellon University, Pittsburgh.

His research interests are in image and video signal processing, computer vision, and multi-media signal processing.

**José M. F. Moura** received the engenheiro electrotécnico degree in 1969 from Instituto Superior Técnico (IST), Lisbon, Portugal, and the M.Sc., E.E., and the D.Sc. in Electrical Engineering and Computer Science from the Massachusetts Institute of Technology (M.I.T.), Cambridge, in 1973 and 1975, respectively.

He is presently a Professor of Electrical and Computer Engineering at Carnegie Mellon University (CMU), Pittsburgh, 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 had visiting appointments at several Institutions, including M.I.T. (Genrad Associate Professor of Electrical Engineering and Computer Science, 1984–1986) and the University of Southern California (research scholar, Department of Aerospace Engineering, Summers 1978–1981). His research interests include statistical signal processing, digital communications, image and video processing, radar and sonar, and multiresolution techniques. He has organized and codirected two international scientific meetings on signal processing theory and applications. He has over 190 published technical contributions, including invited ones, and is co-editor of two books.

Dr. Moura is currently an elected member of the Board of Governers of the *IEEE Signal Processing Society*, the *Editor in Chief* for the *IEEE Transactions in Signal Processing* and a member of the IEEE Signal Processing Society Publications Board, of the *IEEE Signal Processing Society Underwater Accoustics Technical Committee*, and of the *IEEE Signal Processing Society MultiMedia Signal Processing Technical Committee*. He was a member of the *IEEE Press Board* (1991-95), a technical Associate Editor for the *IEEE Signal Processing Letters* (1993-95), and an Associate Editor for the *IEEE Transactions on Signal Processing* (1988-92). 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 is a Fellow of the IEEE and corresponding member of the *Academy of Sciences of Portugal* (Section of Sciences). He is affiliated with several IEEE societies, Sigma Xi, AMS, IMS, and SIAM.