W^4: Real-Time Surveillance of People and Their Activities

Ismail Haritaoglu, *Member*, *IEEE*, David Harwood, *Member*, *IEEE*, and Larry S. Davis, *Fellow*, *IEEE*

Abstract—W^4 is a real time visual surveillance system for detecting and tracking multiple people and monitoring their activities in an outdoor environment. It operates on monocular gray-scale video imagery, or on video imagery from an infrared camera. W^4 employs a combination of shape analysis and tracking to locate people and their parts (head, hands, feet, torso) and to create models of people's appearance so that they can be tracked through interactions such as occlusions. It can determine whether a foreground region contains multiple people and can segment the region into its constituent people and track them. W^4 can also determine whether people are carrying objects, and can segment objects from their silhouettes, and construct appearance models for them so they can be identified in subsequent frames. W^4 can recognize events between people and objects, such as depositing an object, exchanging bags, or removing an object. It runs at 25 Hz for 320×240 resolution images on a 400 Mhz dual-Pentium II PC.

Index Terms—Surveillance, people tracking, activity detection, real-time vision, body part analysis.

1 Introduction

The objective of this paper is to present a set of techniques integrated into a low-cost PC based real-time visual surveillance system, called W^4, for simultaneously tracking people and their body parts, and monitoring their activities in monochromatic video. W^4 constructs dynamic models of people's movements to answer questions about **what** they are doing, and **where** and **when** they act. It constructs appearance models of the people it tracks so that it can track people with relative identity (**who**) through occlusion events in the imagery. W^4 has been designed to work with only monochromatic stationary video sources, either visible or infrared. While most of the previous work on detection and tracking of people has relied heavily on color cues, W^4 is designed for outdoor surveillance tasks, and particularly for night-time or other low light level situations. In such cases, color will not be available, and people need to be detected and tracked based on weaker appearance and motion cues.

The major features of W^4 are as follows: W^4

- Learns and models background scenes statistically to detect foreground objects, even when the background is not completely stationary (e.g., motion of tree branches).
- Distinguishes people from other objects (e.g., cars) using shape and periodic motion cues.
- Tracks multiple people simultaneously even when they are moving together, or interacting with each other (Fig. 1).
- Constructs an appearance model for each person during tracking that can be used to identify people after occlusion.
- Detects and tracks six main body parts (head, hands, feet, torso) of each person using a static shape model and second order motion tracking of dynamic appearance models.
- Determines whether a person is carrying an object, and segments the object so it can be tracked during exchanges.

The block diagram in Fig. 2 shows the system architecture of W^4. In the first stage (detection), W^4 detects foreground pixels using a statistical-background model; they are grouped into foreground blobs, and a set of global and local features of each blob are computed. In the second stage (silhouette analysis), each blob is classified into one of three predetermined classes using static silhouette shape and dynamic periodicity analysis: single-person, people in a group, and other objects.

If a blob is classified as single-person, then a silhouette-based posture analysis is applied to the blob to estimate the posture of the detected person. If a person is in the upright-standing posture, then a further dynamic periodic motion analysis and symmetry analysis are applied to determine whether the person is carrying an object. If the person is not carrying an object or is in a different generic posture than standing posture, then W^4 detects body parts using silhouette boundary shape analysis. If a blob is classified as a group, then W^4 cannot detect individual body parts, postures, or carried objects. Instead, W^4 assumes that all the people in the group are in an upright standing posture and it segments the group into individuals. If a blob is classified as an object other than a person, W^4 does not do any further silhouette analysis; it simply attempts to track the object through the video. After the silhouette-based analysis is completed, a tracker computes the correspondence between previously tracked blobs and currently detected blobs, constructs appearance and motion models, and recovers the trajectories of the tracked blob.

- *I. Haritaoglu is with the IBM Almaden Research Center, 650 Harry Road, San Jose CA 95120. E-mail: ismailh@almaden.ibm.com.*
- *D. Harwood and L.S. Davis are with the Computer Vision Laboratory, University of Maryland, College Park, MD 20742. E-mail: {harwood, lsd}@umiacs.umd.edu.*

Manuscript received 21 Apr. 1999; revised 24 Feb. 2000; accepted 28 Mar. 2000.
Recommended for acceptance by R. Collins.
For information on obtaining reprints of this article, please send e-mail to: tpami@computer.org, and reference IEEECS Log Number 109645.

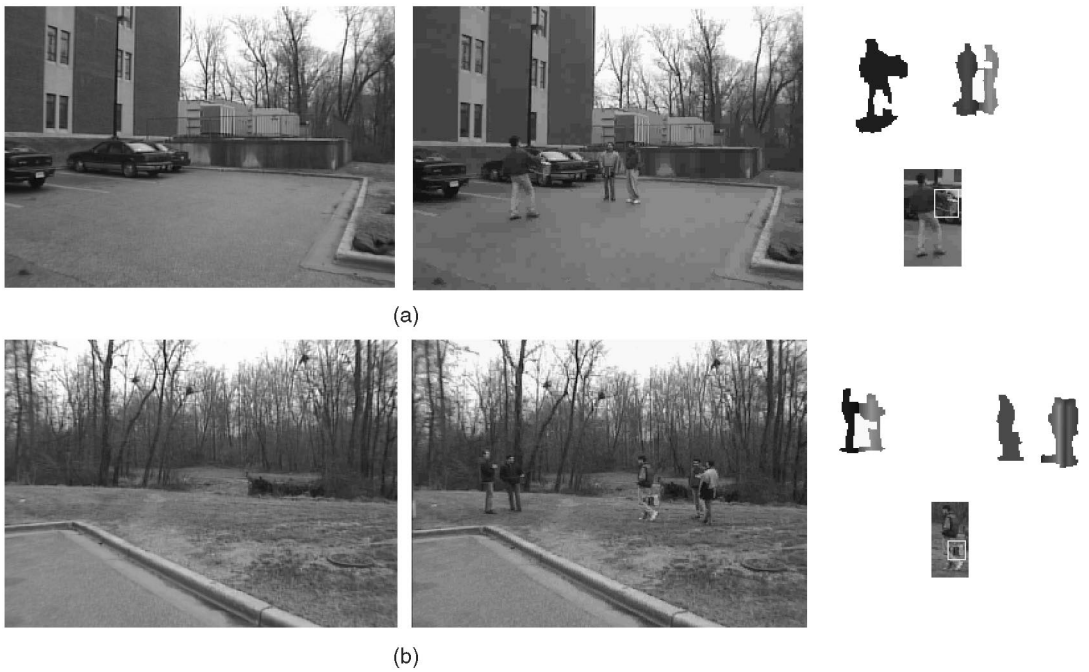


Fig. 1. Example of detection of people.

The overall performance of the silhouette-based techniques used in W^4 depends on how accurately foreground blobs are detected. If the foreground detection modules fails to segment people, then our silhouettes based methods will clearly fail. We developed a fast background subtraction based algorithm to detect people in outdoor scenes which handles many outdoor scene problems; however, as it is an intensity based method, a drastic change in scene illumination, such as clouds blocking the sun, can lead to gross detection failures. However, one could replace it with a more robust but slow detection algorithm such as in [14], [10].

Another problem that plagues W^4 is shadows. W^4 does not attempt to segment shadows from people's silhouettes. When shadows are large, W^4 has substantial difficulties in obtaining meaningful results from its silhouette-based analysis.

While camera orientation does not critically effect our detection or tracking algorithms, our silhouette-based methods requires a fairly oblique view, and they assume that complete body silhouettes are detected so that people are not occluded by stationary objects; so W^4 cannot detect body parts when a person is standing behind a desk and the lower half of the body is occluded. W^4 can automatically adjust to a wide range of scales, but its detailed body shape analysis requires that the blob is comprised of at least 250 pixels (e.g., 25×10). Generally, our body part analysis, multiple person tracking, and carried object detection algorithms performs better when people appear bigger than 75×50 pixels.

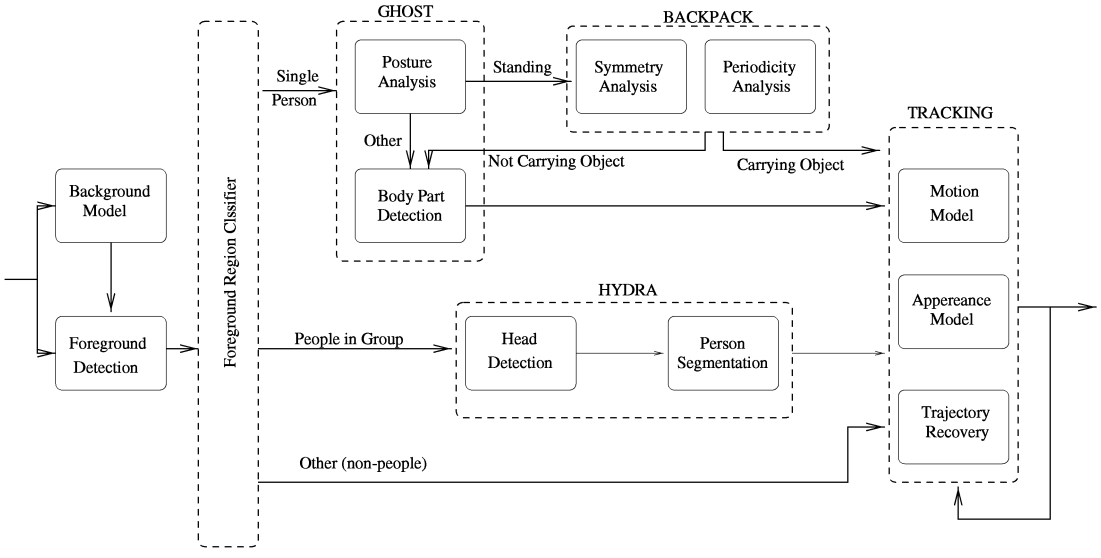


Fig. 2. The system architecture of W^4.

TABLE 1
Classification of Previous People Tracking Systems According to Their Sensor Type and Detection and Tracking Functions

	Area	Sensor	Camera	Detection	Tracking People
System	Indoor (I) Outdoor (O)	Color (C) Grayscale (G)	Single (S) Stereo (O) Multiple (M)	Single Gaussian (S) BiModal (B) Mixture of Gaussian (M)	Single Isolated(S) Multiple Isolated (M) Multiple in Group (G)
Pfinder [34]	I	C	S	S	S
CMU [25]	O	C	M	S	M
LOTS [6]	O	G	S	S	M
MIT [14]	O	C	M	M	M
TI [29]	I	G	M	S	S
SRI [3]	I	G	O	S	M
W^4	O	G	S	B	M,G
KidRooms [5]	I	C	M	S	M
S. Kiosk [31]	I	C	S,O	S	S
M.Mirror [9]	I	C	O	S	M
EasyLiving [37]	I	C	M,O	S	M

The remainder of this paper is organized as follows: After a brief literature review (Section 1.1), Section 2 describes algorithms to detect people in outdoor scenes. First, a statistical background model is computed and foreground regions are detected in each frame. Then foreground regions are classified as either human or "other" using silhouette-based analysis. Section 3 focuses on tracking an isolated person using silhouette-based methods and describes the use of dynamic appearance models used for identifying people. Section 3.2 describes the silhouette-based models to detect and track the six main body parts. In Section 3.4, an efficient method is described to determine whether or not a person is carrying an object and monitoring their activities. Section 4 considers the situation when a foreground region contains multiple people. Finally, Section 5 concludes with a summary of the advantages, real-time performance, and limitations of W^4.

1.1 Related Work

There has been a significant number of recent projects on detecting and tracking people. We can classify those systems into categories, according to their sensor types (single or multiple camera, color or gray scale), and their functionality (track single person, multiple people, handle occlusion), as shown as in Table 1.

Pfinder [34] has evolved over several years and has been used to recover a 3D description of a person in a large room size space. Pfinder has been used in many applications. It solves the problem of person tracking in complex scenes in which there is a single unoccluded person and fixed camera. Pfinder utilizes a 2D image analysis architecture with two complementary procedures for 2D tracking and initialization. Spfinder [2] is a recent extension of Pfinder in which a wide-baseline stereo camera is used to obtain 3D models. Spfinder has been used in a smaller desk-area environment to capture accurate 3D movements of head and hands.

Both Pfinder and W^4 use a statistical background model to locate people. However, Pfinder uses a single Gaussian distribution of color at each pixel, while W^4 uses a bimodal distribution of intensity at each pixel. Both systems use silhouettes to detect body parts; however, Pfinder assumes that there is only a single person in the scene and in an upright standing posture. W^4 allows multiple person groups and isolated people in different postures.

KidRooms [5] is a tracking system based on "closed-world regions." These are regions of space and time in which the specific context of what is in the regions is known. These regions are tracked in real-time domains where object motions are not smooth or rigid and where multiple objects are interacting.

Smart Kiosk [31] is an application to detect and track people in front of a kiosk. It uses both color information, face detection, and stereo information for detection. However, when people are very close to the kiosk, it can only track a single person.

TI's system [29] is a general purpose system for moving object detection and event recognition where moving objects are detected using change detection and tracked using first-order prediction and nearest-neighbor matching. Events are recognized by applying predicates to a graph formed by linking corresponding objects in successive frames. TI's system also uses background subtraction to find people and objects using a single Gaussian distribution of intensity of each pixel. It is designed for indoor surveillance and it cannot handle small motions of background objects. It is a single person tracking system.

CMU developed a system [25] that allows a human operator to monitor activities over a large area using a distributed network of active video sensors. Their system can detect and track multiple people and vehicles within cluttered scenes and monitor their activities over long periods of time. They developed robust routines for detecting moving objects using a combination of temporal differencing and template tracking. Detected objects are classified into categories such as human, human groups, car, and truck using shape and color analysis, and these labels are used to improve tracking using temporal consistency constraints.

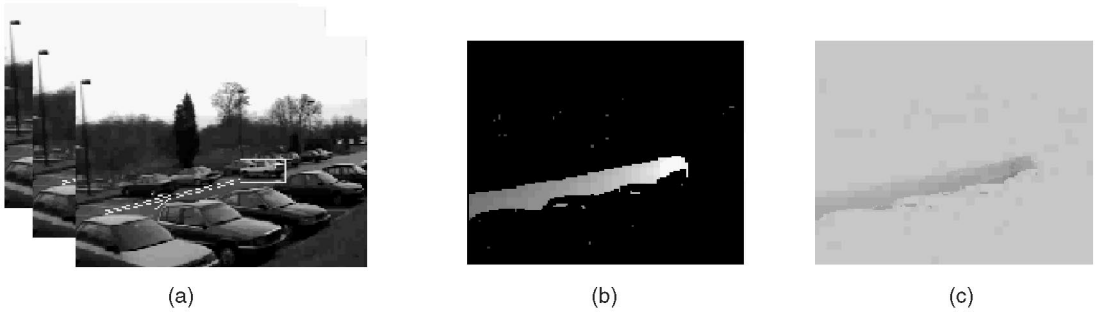


Fig. 3. An example of change-map used in background model computation: (a) input sequence, (b) motion history map, and (c) detection map.

MIT's system [13], [14] uses a distributed set of sensors, and adaptive tracking to calibrate distributed sensors, classify detected objects, learn common patterns of activity for different object classes, and detect unusual activities. Lehigh's omnidirectional tracking system [6] has been used successfully to detect people in camouflage. It has been designed to handle multiple independent moving bodies undergoing nonrigid motion using background subtraction followed by connected component labeling. Oliver et al. [28] and Morris and Hogg [27] describe statistical modeling of human and object interactions.

2 Background Scene Modeling and People Detection

This section focuses on detection of people in a single image using shape cues. We describe computational models that

- build a statistical model for a background scene that allows us to detect foreground regions even when the background scene is not completely stationary,
- classify those detected foreground regions as people or "other" objects, and
- determine whether a foreground region contains multiple people.

2.1 Background Scene Modeling

A simple and common background modeling method involves subtracting each new image from a model of the background scene and thresholding the resulting difference image to determine foreground pixels. The pixel intensity of a completely stationary background can be reasonably modeled with a normal distribution [34], [29], and it can adapt to slow changes in the scene by recursively updating the model. However, those approaches have difficulty in modeling backgrounds in outdoor scenes because they cannot handle the small motions of background objects such as vegetation (swaying tree branches). In this case, more than one process may be observed over time at a single pixel. In [11], a mixture of three normal distributions was used to model the pixel value for traffic surveillance applications to model road, shadow, and vehicle. In [13], pixel intensity is modeled by a mixture of K Gaussian distributions (typically, K is three to five). [10] uses a nonparametric background model by estimating the probability of observing pixel intensity values based on a sample of intensity values for each pixel. W^4 uses a model of background variation that is a bimodal distribution constructed from order statistics of background values during a training period. The background scene is modeled by representing each pixel by three values; its minimum $m(x)$ and maximum $n(x)$ intensity values and the maximum intensity difference $d(x)$ between consecutive frames observed during this training period. Other methods developed in our laboratory [21], [10] have more sensitivity in detecting foreground regions, but are computationally more intensive.

2.1.1 Learning Initial Background Model

W^4 obtains the background model even if there are moving foreground objects in the field of view, such as walking

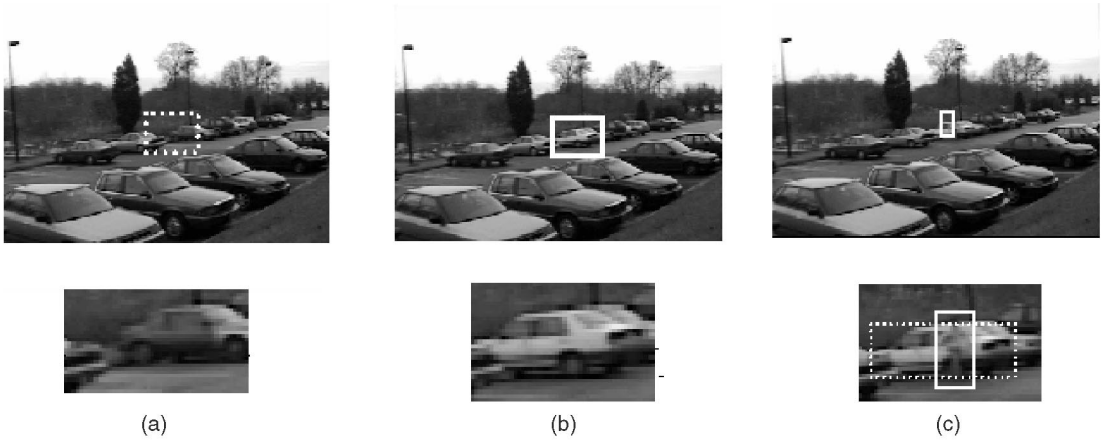


Fig. 4. A car which has been parked for a long time is added to background models (a) and (b), so the person getting off the car is detected (c).

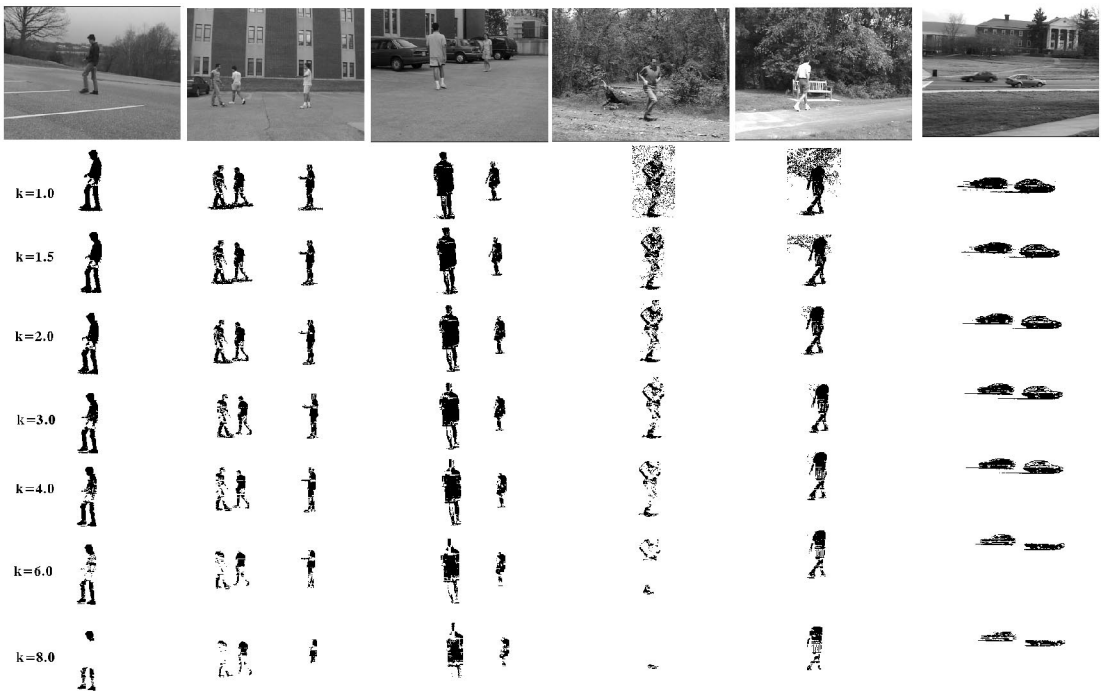


Fig. 5. An example of foreground region detection for different threshold values.

people, moving cars, etc. It uses a two stage method based on excluding moving pixels from background model computation. In the first stage, a pixelwise *median filter* over time is applied to several seconds of video (typically 20-40 seconds) to distinguish moving pixels from stationary pixels. In the second stage, only those stationary pixels are processed to construct the initial background model. Let V be an array containing N consecutive images, $V^i(x)$ is the intensity of a pixel location x in the ith image of V. $\sigma(x)$ and $\lambda(x)$ are the standard deviation and median value of intensities at pixel location x in all images in V. The initial background model for a pixel location x, $[m(x), n(x), d(x)]$, is obtained as follows:

$$\begin{bmatrix} m(x) \\ n(x) \\ d(x) \end{bmatrix} = \begin{bmatrix} min_z\{V^z(x)\} \\ max_z\{V^z(x)\} \\ max_z\{|V^z(x) - V^{z-1}(x)|\} \end{bmatrix}, \quad \text{where } |V^z(x) - \lambda(x)| < 2 * \sigma(x). \tag{1}$$

Here, $V^z(x)$ is classified as stationary pixels.

2.1.2 Updating Background Model Parameters

The background model cannot be expected to stay the same for long periods of time. There could be illumination changes, such as the sun being blocked by clouds causing changes in brightness, or physical changes, such as a deposited object. As W^4 uses an intensity-based background model, any changes in illumination can cause false positives. Additionally, any foreground object detected for a long time without any motion (a parked car) can cause false negatives (a person would not be detected while he is getting out of the car). W^4 uses two different methods to update the background.

- A **pixel-based update** method updates the background model *periodically* to adapt to illumination changes in the background scene.
- An **object-based update** method updates the background model to adapt to physical changes in the background scene. A deposited/removed object, or a parked car would be added into the background scene if it does not move for a long period of time.

W^4 uses the following method to update the background model: During tracking, W^4 dynamically constructs a change map to determine whether a pixel-based or an object based update method applies. The change map consists of three main components:

- A *detection support map* (gS) which represents the number of times a pixel location is classified as a background pixel in the last N frames.

$$gS(x,t) = \begin{cases} gS(x,t-1)+1 & \text{if } x \text{ is background pixel} \\ gS(x,t-1) & \text{if } x \text{ is foreground pixel.} \end{cases} \tag{2}$$

- A *motion support map* (mS) which represents the number of times a pixel location is classified as a moving pixel. A pixel is classified as a moving pixel by subtracting three consecutive images.

$$mS(x,t) = \begin{cases} mS(x,t-1)+1 & \text{if } M(x,t) = 1 \\ mS(x,t-1) & \text{if } M(x,t) = 0, \end{cases} \tag{3}$$

where

$$M(x,t) = \begin{cases} 1 & \text{if } (|I(x,t) - I(x,t+1)| > 2 * \sigma) \wedge \\ & \quad (|I(x,t-1) - I(x,t)| > 2 * \sigma) \\ 0 & \text{otherwise.} \end{cases} \tag{4}$$

TABLE 2
True Detection Rate for Different Background Scenes Which Have Different Intensity Variations as Shown in Fig. 5

	k=2	k=3	k=4	k=6	k=8
seq 1	0.89	0.80	0.72	0.51	0.32
seq 2	0.52	0.40	0.29	0.03	0.02
seq 3	0.81	0.70	0.59	0.41	0.28
seq 4	0.85	0.77	0.70	0.50	0.40
seq 5	0.85	0.77	0.71	0.59	0.46
seq 6	0.87	0.77	0.72	0.60	0.48

- A *change history map* (hS) which represents the elapsed time (in frames) since the last time that the pixel was classified as foreground pixel.

$$hS(x,t) = \begin{cases} 255 & \text{if } x \text{ is foreground pixel} \\ hS(x,t-1) - \frac{255}{N} & \text{otherwise.} \end{cases} \quad (5)$$

W^4 uses gS to determine the parts of the background which are updated by the pixel-based method and mS, gS, and hS to determine the parts of the background which are updated by the object-based method. The change-maps are set to zero after the background model is updated. In Fig. 3, an example of a change map used in the background update method is shown.

During tracking, the background model is computed separately for all pixels which are classified as foreground pixels ($m^f(x), n^f(x), d^f(x)$) and for all pixels which are classified as background pixels ($b^b(x)$, $n^b(x)$, $d^b(x)$). Let $m^c(x)$, $n^c(x)$, $d^c(x)$ be the background model parameters currently being used; the new background model parameters $m(x)$, $n(x)$, $d(x)$ are determined as follows:

$$[m(x), n(x), d(x)] = \begin{cases} [m^b(x), m^b(x), d^b(x)] & \text{if } (gS(x) > k * N)(\text{pixel-based}) \\ [m^f(x), n^f(x), d^f(x)] & \text{if } (gS(x) < k * N \wedge mS(x) < r * N) \\ & \quad (\text{object-based}) \\ [m^c(x), n^c(x), d^c(x)] & \text{otherwise,} \end{cases} \quad (6)$$

where k and r are typically 0.8 and 0.1, respectively. In Fig. 4, a car which has been parked for a long time is added to background model, so the person getting out of the car is detected and tracked successfully.

Another important factor is how fast the background model adapts to change. Sudden changes in background illumination, such as clouds blocking the sun, make the detection fail. Occasionally, we encounter similar "catastrophes" with this updating procedure and large parts of the image are classified as foreground. So, when a large percentage (> 80 percent, e.g.,) of the image is detected as foreground, W^4 stops tracking and starts to learn new background model parameters, as described previously.

2.2 Foreground Region Detection

Foreground objects are segmented from the background in each frame of the video sequence by a four stage process: thresholding, noise cleaning, morphological filtering, and object detection. Each pixel is first classified as either a background or a foreground pixel using the background model. Giving the minimum $m(x)$, maximum $n(x)$, and the median of the largest interframe absolute difference d_μ images over the entire image that represent the background scene model $B(x)$, pixel x from image I^t is a foreground pixel if:

$$B(x) = \begin{cases} 0 \text{ background} & \begin{cases} (I^t(x) - m(x)) < kd_\mu \\ \vee I^t(x) - n(x)) < kd_\mu \end{cases} \\ 1 \text{ foreground} & \text{otherwise.} \end{cases} \quad (7)$$

We ran a series of experiments to determine the best threshold constant k using different background scenes while the background intensity variation is at different levels. The results are shown in Fig. 5. For image sequences where there is high intensity variation of background pixels (sequence 4 and sequence 5 in Fig. 5), our method yields a large number of false positives when $k < 2$. For other sequences, the false positive rate is very low. We generated the ground truth for the foreground region for each image in Fig. 5 and compared it with the detected foreground regions. Table 2 shows the true positive rates for different k. Note that when $k > 4$, the true detection rates drops under 70 percent. Our experiments show that $k = 2$ gives the highest true positive rates with the lowest false positives rates; we consistently use $k = 2$ in our system.

Thresholding alone, however, is not sufficient to obtain clear foreground regions; it results in a significant level of

Fig. 6. An example of foreground region detection while background has different intensity variation.

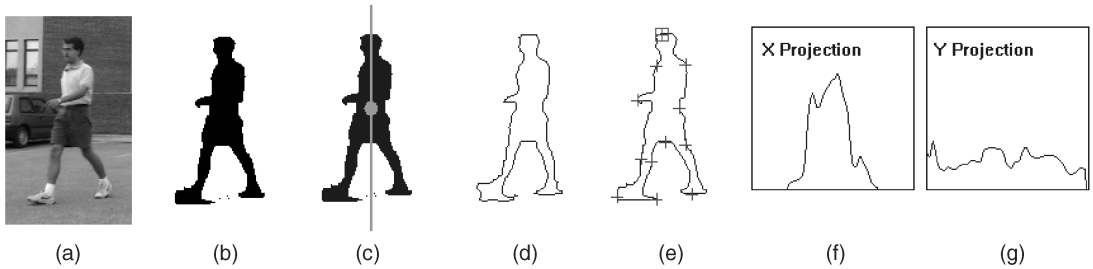


Fig. 7. Silhouette based shape features used in W^4: (a) input image, (b) detected foreground region, (c) its centroid and major axis, (d) contour of its boundary, (e) convex/concave hull vertices on its contour, (f) horizontal, and (g) vertical projection histogram.

noise. W^4 uses region-based noise cleaning to eliminate noise regions.

As the final step of foreground region detection, a binary connected component analysis is applied to the foreground pixels to assign a unique label to each foreground object (Fig. 6). W^4 then generates a set of shape and appearance features for each detected foreground object that are used to distinguish humans from other objects, detect people moving in groups, and detect people carrying objects. W^4 computes both global and local shape features of the silhouettes.

Global shape features. W^4 uses the median coordinate of each foreground region as an estimate of object position in the image coordinate system since it is not effected by the large motions of the extremities which tend to influence the centroid significantly. W^4 determines a *major axis* of the foreground region by applying a principal component analysis (PCA) to the foreground pixels. The major axis of a foreground region is used to compute the relative orientation of body parts and body posture. The best fitting axis which goes through the median coordinate is computed by minimizing the sum of squared perpendicular distances to the axis. The direction of the major axis is given by an eigenvector v associated with the largest eigenvalue of its covariance matrix (Fig. 7b).

The shape of a 2D binary silhouettes is represented by its *projection histograms*. W^4 computes the 1D vertical (horizontal) projection histograms of the silhouettes in each frame. Vertical and horizontal projection histograms are computed by projecting the binary foreground region on an axis perpendicular to the major axis and along the major axis, respectively (Figs. 7f and 7g). Projection histograms are normalized by rescaling projections onto a fixed length, and aligning the median coordinates at the center.

Local shape features. Because of the topology of the human body, it is likely that some body parts appear at extreme points or curvature maxima of the silhouette boundary. Therefore, W^4 analyzes the shape of the silhouette boundary to find "natural" vertices as a candidate set of locations for body parts. We implemented two methods to find points vertices on the silhouette boundary: a *recursive convex hull algorithm* (Graham scan) to find convex and concave hull vertices on the silhouette boundary and a *corner detector* based on local curvature of the silhouette boundary. The convex/concave hull algorithm gives better localization results but it is computationally expensive. Therefore, the corner detection algorithm is applied in every frame, but the convex/concave vertex algorithm is applied only when W^4 needs to detect the initial location of the body parts. Figs. 7d and 7e contains an example of a silhouette boundary and its corner points.

People have very distinctive shape, appearance, and motion patterns compared to other objects. One can use a static shape analysis , such as size perimeter, aspect ratio, or dynamic motion analysis, such as speed, or periodicity of the motion to distinguish people from other objects. W^4 combines static shape cues with a dynamic periodicity analysis to distinguish humans from other objects.

People exhibit periodic motion while they are moving. Cutler previously introduced a robust, image-correlation-based technique to compute the periodicity of a moving object [8]. A computationally inexpensive version of [8], which requires less memory, is employed by W^4 to determine shape periodicity. Periodic motion is determined by self-similarity of silhouettes over time using the silhouettes projection histograms. A detailed explanation of periodic motion computation can be found in [8], [19].

W^4 analyzes the vertical projection histogram of the silhouettes to determine whether the foreground region contains multiple people. A set of average normalized vertical projection histogram templates for a single person is precomputed experimentally from a database of different

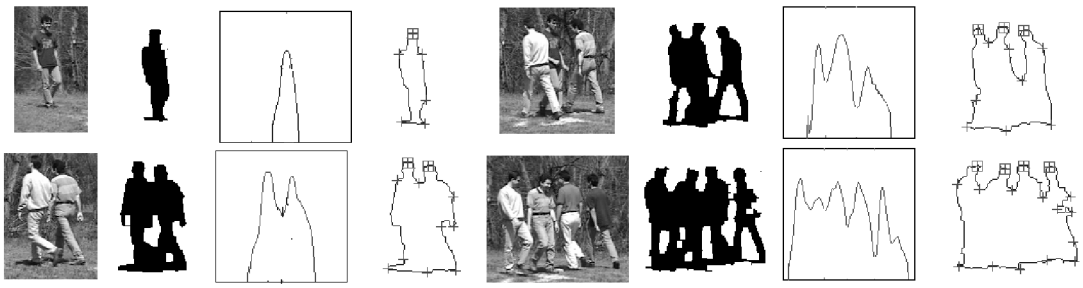


Fig. 8. Examples of silhouettes which contain different numbers of people, their blobs detected by background subtraction, their projection histograms, and their silhouette boundaries with detected head locations.

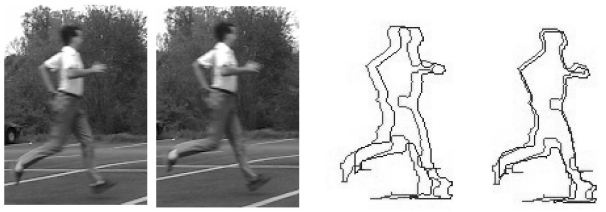


Fig. 9. Motion estimation of body using Silhouette Edge Matching between two successive frame: input images (first and second); alignment of silhouette edges based on difference in median (third); final alignment after silhouette correlation (fourth).

people in different postures. The similarities between the vertical projection of the current silhouette and those average single person vertical projection histogram templates are computed using the SAD method [17] and thresholds determined during training. An example of silhouettes which contain different number of people and their projection are shown in Fig. 8.

3 Tracking People: Isolated Person

Here, we consider the situation that a person continues to be tracked as a single blob. W^4 tracks people even in the event that its low-level detection algorithms fail to segment them as single foreground regions. This might occur because a foreground region becomes temporarily occluded (by some fixed object in the scene), or it splits into pieces (possibly due to a person depositing an object in the scene, or a person being partially occluded by a small object). Finally, separately tracked regions might merge into one because of interactions between people.

The goals of the person tracking stage are to:

- Determine when a new person enters the system's field of view, and initialize motion models for tracking that object.
- Compute the correspondence between the foreground regions detected by the background subtraction and the people currently being tracked by W^4.
- Employ tracking algorithms to estimate the position (of the torso) of each person and update the motion model used for tracking. W^4 employs second order motion models (including a velocity and, possibly zero, acceleration terms) to model both the overall motion of a person and the motions of its parts.
- Build an appearance model for each person that can be used to recognize people after occlusion.
- Detect and track body parts.
- Determine whether or not a person is carrying an object.

W^4 employs a second order motion model for each person to estimate its location in subsequent frames. The prediction from this model is used to estimate a bounding box location for each person. These predicted bounding boxes are then compared to the actual bounding boxes of the detected foreground regions. Given that a person is matched to a single blob (and the sizes of those blob are roughly the same), W^4 has to determine the current position of the person to update its motion model. Even though the total motion of an person is relatively small between frames, the large changes in the shape of its silhouette causes simple techniques, such as tracking the centroids of the foreground blob, to fail. Instead, W^4 uses a two stage matching strategy to update its global position estimate of a person. The initial estimate of displacement is computed as the motion of the **median** coordinate of the person, a robust estimate of a person's position. It allows us to quickly narrow the search space for the motion of the object. However, this estimate is not accurate enough for long term tracking. Therefore, after displacing the silhouette of the object from the previous frame by the median-based estimate, we perform a binary edge correlation between the current and previous silhouette edge profiles. This correlation is computed only over a 5×3 set of displacements. Typically, the correlation is

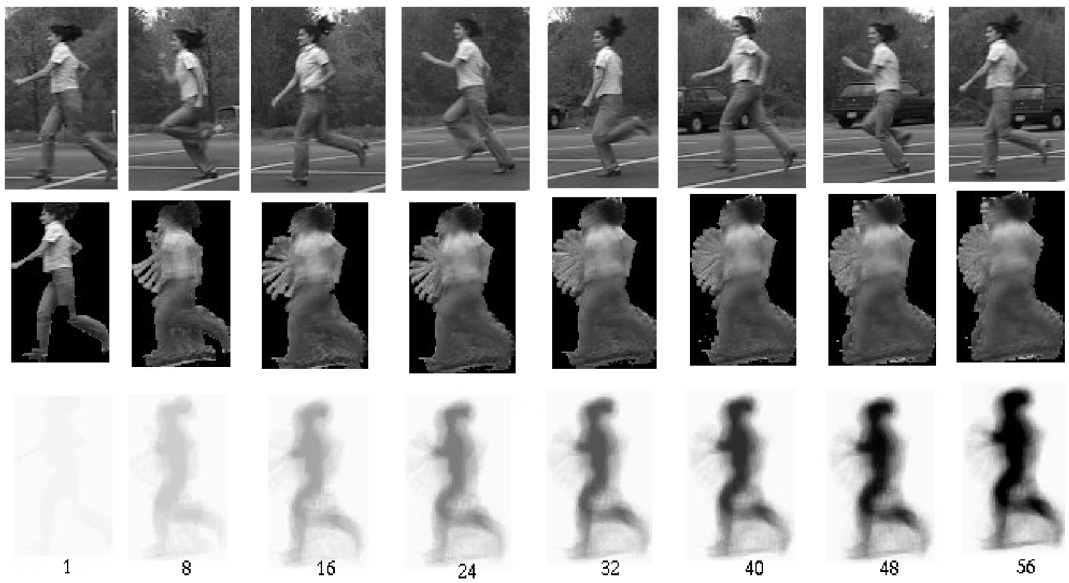


Fig. 10. An example of how textural (second row) and frequency (third row) components of temporal texture templates are updated over time.

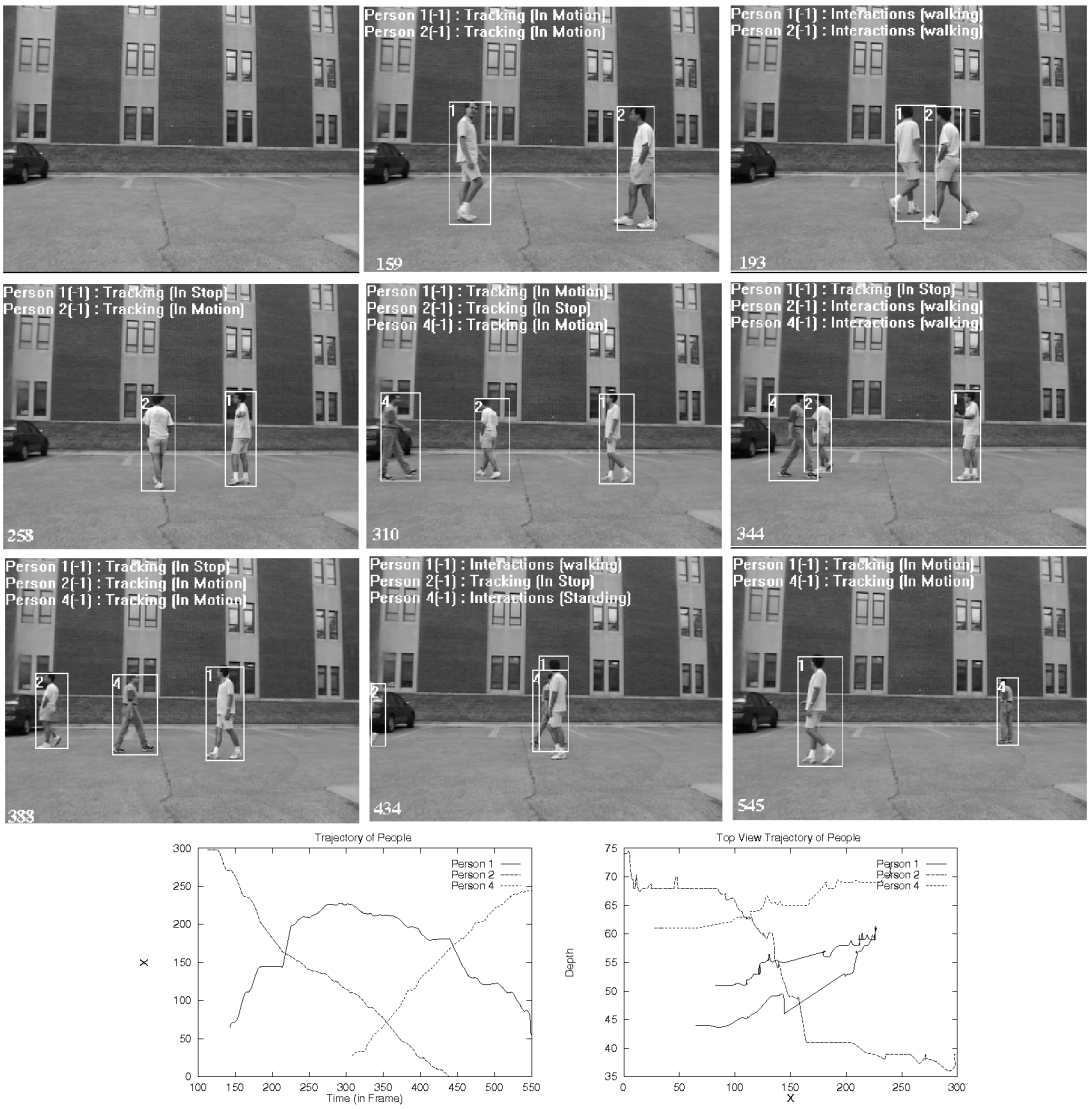


Fig. 11. An example of W^4 tracking: three people are entering, walking, meeting and leaving, and recovered ground plane trajectories of each people is given.

dominated by the torso and head edges, whose shape changes slowly from frame to frame. This tracking process is illustrated in Fig. 9.

W4's detection and tracking algorithms were tested on a ground truthed database of six hours of video taken in the front courtyard of our computer science research building. Three two-hour segments were taken under different environmental conditions, including snow, strong wind, and bright sunshine. During the six hours of video there are 338 isolated people who move through the camera's field of view. Out of those 338 people, 313 are detected and tracked correctly—i.e., they are tracked throughout the subsequence in which they are seen. Note that detection can fail intermittently during tracking, but the tracking can continue successfully; also, it might take several frames to detect and track a person once he/she enters the camera's field of view. In nine cases out of the 25 failures, the same person is detected twice in the same subsequence due to long detection failures. There were no false detections of people (although there are intermittent, instantaneous false detections of foreground regions). The system suffered 17 catastrophic failures of its background subtraction due to sudden illumination changes, changes in snow intensity, wind, etc. These were all identified automatically, and in all cases, the system was able to reinitialize its background model and successfully continue to track. The error rates do not include these periods in which the background model is being reinitialized. The system decides automatically when the background model restabilizes based on detection statistics.

3.1 Dynamic Appearance Model

A problem that arises when a merged region splits and the people "reappear," is determining the correspondence between the people that were tracked before the interaction and the people that emerge from the interaction. To accomplish this, W^4 combines the *gray-scale textural appearance* and *shape* information of person together in a 2D dynamic template called a textural temporal template, an extension to the temporal templates defined in [4] to

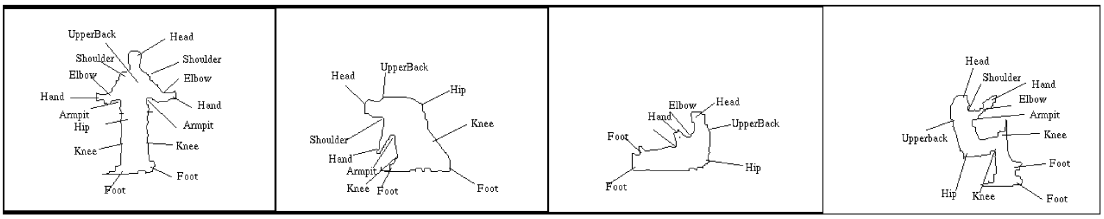


Fig. 12. Examples of the order of the body parts on the silhouette boundary.

recognize action using shape information. They not only contains shape information, but also gray-scale texture information.

The temporal texture template for an object is defined by:

$$\Psi^t(x,y) = \frac{I(x,y) + w^{t-1}(x,y) \times \Psi^{t-1}(x,y)}{w^{t-1}(x,y) + 1}. \quad (8)$$

Here, $I(x)$ refers to the intensity of pixel(x) which is classified as foreground pixel and all coordinates are represented relative to the *median* coordinate of the person. The w^t are the number of times that a pixel in Ψ_y is classified as a foreground pixel during the last N frame. The initial $w^t(x)$ of Ψ_y are zero and are incremented each time that the corresponding location (relative to the median template coordinate) is detected as a foreground pixel in the input image. Note that a temporal textural template has two components that can be used for subsequent identification: a *textural* component which represents the *gray-scale textural appearance* of the person (second row in Fig. 10); and a *shape component* (w^t) which represents shape information (third row in Fig. 10) of the human body for the last N frames. $w^t(x,y)$ is then normalized to a probability map $w_n^t(x,y)$. After normalization, $w_n^t(x,y)$ is treated as the probability that the pixel location (x,y) belongs to a person in the last N time frames. For example, while a person is walking, due to articulated motion of their arms and legs, the locations near the arms and legs have lower probability than the locations near the torso. In Fig. 10, third row, darker locations have higher shape probability (high $w_n^t(x,y)$ makes a stronger contribution to correlations used during reacquisition).

After separation or reappearance, each constituent object is matched with the separating people by correlating their temporal templates with the detected silhouettes over a small neighborhood search window. We compute the weighted similarity $C(p,r)$ between each detected separating person p and tracked person r at time t. Let $\mathbf{S}^t$ be the gray-scale silhouette of a separating person and Ψ^t be the temporal textural template of a person who has been tracked, but has disappeared for a while (due to either occlusion or leaving the scene).

$$C(p,r) = \frac{\sum_{(x,y)\in S^p} |S_p^t(x,y) - \Psi^t(x,y)| \times w^t(x,y)}{\sum w^t(x,y)}. \quad (9)$$

Those similarity values are normalized. If the normalized similarity is lower than a predetermined threshold, then the detected person is matched with the previously tracked person. The lowest $C(p,r)$ which is higher than a predetermined threshold indicates that a new person has entered the system's field of view, and the system gives it a new label and initializes motion models for tracking that person.

Fig. 11 illustrates W^4 tracking people; W^4 detects people, assigns unique labels to them and tracks them through occlusions and interactions.

3.2 Detection and Tracking of People's Body Parts Using Silhouettes

Detecting and tracking human body parts (head, hands, feet) is important in understanding human activities. In addition to tracking the body as a whole, we want to locate body parts such as the head, hands, torso, legs and feet, and track them in order to understand actions. In this section, we describe the computational models employed by W^4 to predict and track the locations of the six main body parts (e.g, head, hands(2), feet(2), and torso) while a person is in any of a number of postures.

Our system is motivated by two basic observations about the relative locations of body parts while people are in action.

- It is very likely that the head, hands, elbows, and feet lie on the silhouette boundary.
- The human body in any given posture has a topological structure which constrains the relative locations of body parts. The order of body parts along the silhouette boundary does not typically change when people perform an action while maintaining a generic posture (walking); however, the order does change when they change their generic posture (from walking to sitting).

W^4 uses a silhouette-based body model which consists of six primary body parts (head, hands(2), feet(2), and torso), which we want to locate, and ten secondary parts (elbows(2), knees(2), shoulders(2), armpits(2), hip, and upper back) which could be on the silhouette boundary and can help to locate the primary parts using the topology of the human body. The outline of the algorithm used in W^4 is as follows:

1. A hierarchical body posture analysis is applied to the silhouette to compute the similarities of horizontal and vertical projection histograms of the detected silhouette and the main postures. The body posture which yields the highest similarity measure is taken as the estimated posture.
2. A recursive convex-hull algorithm is applied to find possible body part locations on the silhouette boundary.
3. The location of the head is predicted using the major axis of the silhouette, the hull vertices, and the topology of the estimated body posture.
4. When the head location is determined, a topological analysis is applied to eliminate the hull vertices

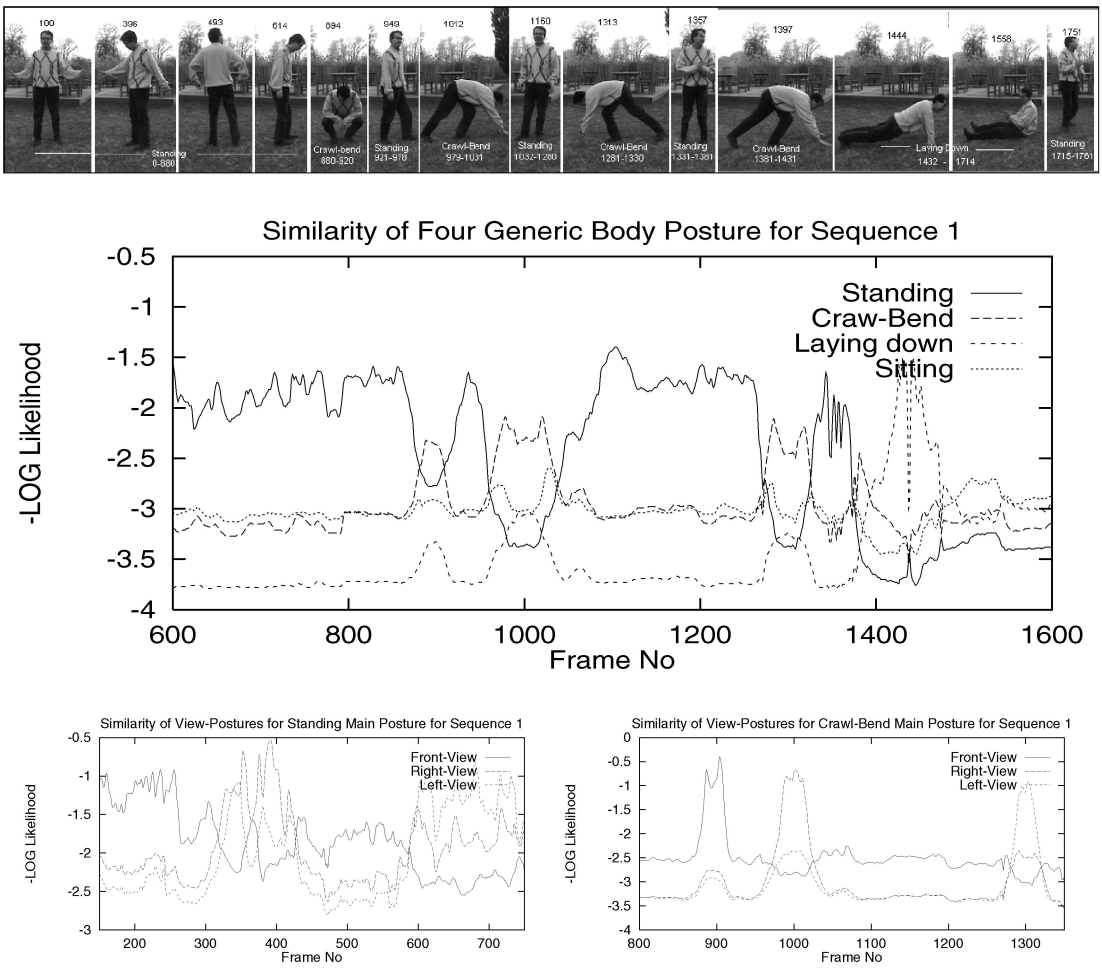


Fig. 13. The similarity of four main postures.

which won't be labeled as body parts, and to map the remaining hull vertices to the body parts using a topological-order preserving distance transform calculation.

3.2.1 2D Body Modeling Using Silhouettes

W^4 tries to locate body parts as long as they are on the silhouette boundary. The primary and secondary body parts should be consistent with the order of the main posture (with small variations). These orders are preserved as long as the body stays in the same main posture. For example, if we start from the head (Fig. 12) in clockwise order, the main order for the upright/standing pose is head-shoulder-elbow-hand-armpit-knee-foot-foot-knee-armpit-hand-elbow-shoulder-head. This order could vary for different viewpoints. Some parts could be missing on the silhouette boundary or some parts could be locally switched in the order (elbow-hand or hand-elbow) because of relative motion of the parts or local occlusion. However, the relative location of some parts (head, feet) should be preserved. For example, head-elbow-shoulder or hand-foot-knee are unacceptable partial orders for the standing posture. Any order of the body parts in the given silhouette should be generated from the order of the main posture by deleting the missing parts or switching the locations of some neighbor parts (elbow-hand). Therefore, if we know the posture of the given silhouette and the location of at least one body part, the labeling problem becomes one of mapping the set of body parts to the set of the silhouette segments without violating the expected order.

3.2.2 Estimation of Body Posture

People can be in many different postures while they are performing actions. Each posture has different appearances, varying with the point of view. Our system makes the assumption that the angle between the view direction and the ground plane is 0^o to $+60^o$. We collected examples of people over a wide range of views and extracted their silhouettes to discover the order of body parts on the silhouette for different postures. We observed that four different main postures (standing, sitting, crawling/bending, and lying down) have large differences in the order of body parts. The order in other postures is typically a variation of one of the main postures. W^4 classifies the observed human body posture in a *hierarchical* manner. Any body posture is classified into one of the four *main postures* (standing, sitting, crawling/bending, lying) and then each main posture is classified into one of three *view-based appearances* (front/back, left-side, and right-side).

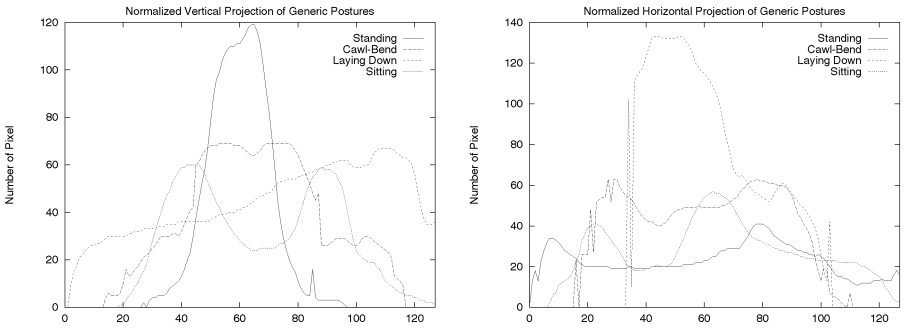


Fig. 14. The vertical and horizontal normalized projections of standing, crawling/bending, and lying down postures used in body posture estimation.

A body posture is represented by the normalized horizontal and vertical projection histograms, the median coordinate, and the major axis of its silhouette. Average normalized horizontal and vertical projection templates for each main posture (and for each view-based appearance of each main posture) were computed experimentally using 4,500 silhouettes of seven different people in three different views. These features are used to determine the similarity of the given posture to one of the four main postures. In Fig. 14, the normalized vertical and horizontal projection templates of the standing, crawling/bending, lying down, and sitting postures used for body posture estimation are shown.

W^4 compares the observed silhouette with the projection templates of the four main postures using the sum of absolute difference method to estimate the most similar main posture. Let S^i be the similarity between the detected silhouette and the ith main posture, H^i and V^i the horizontal and vertical projections of the ith main posture, and P and R the horizontal and vertical projections of the detected silhouette. S_i is calculated as

$$S_i = -log \sum_{h}^{128} \sum_{v}^{128} (H_h^i - P_h)^2 + (V_v^i - R_v)^2. \qquad (10)$$

W^4 determines the most similar main posture by using the highest score; it then applies the same method to determine the most similar view-based appearance for the estimated main posture. In Fig. 13, the results of main posture and view-based appearance estimation are shown for two sequences. In sequence 1 (1,750 frames), the person performed some simple work out actions. He was in the following postures (with frame numbers): standing (0-850), crawling/bending (850-910), standing (920-970), crawling/bending (left view) (970-1,080), standing (1,080-1,270), crawling/bending (right view) (1,270-1,320), standing (1,320-1,380), and lying down (1,400-1,470). The graph in Fig. 13b shows how the classification method is able to select the correct posture over time. Ninety-five percent of the postures were successfully classified in that sequence. Fig. 13c shows view-based appearance estimation for the standing (left) and crawling/bending (right) main postures for sequence 1. Note that when the body is in the crawling/bending posture (the peaks in Fig. 13c-right), the view-based appearance was successfully classified.

We ran another series of experiments to test our methods for posture estimation using 170 silhouettes taken from a variety of people. Silhouette samples consist of 52 standing, 49 bending, 33 lying down, and 39 sitting postures for four different people. The confusion matrix is shown in Table 3. Generally, misclassification occurs when a person is in transition from one posture to another posture.

3.2.3 Prediction of Body Part Locations

We take the head as a reference point to locate the other parts. The head is a stable body part compared to the others and its location can generally be easily predicted. W^4 tries to find a silhouette segment which includes the head by combining the constraints on the order of body parts on the silhouette for the expected main posture, the principal axis of the silhouette, and the ground plane information. Let p be the major axis of the silhouette, and let l_1 and l_2 be the two lines which intersect p at the median coordinate of the silhouette, with the angle between l_1 and p being α and the angle between l_2 and p being $-\alpha$. α varies between 22^o to 45^o according to the estimated posture and the aspect ratio of the silhouette. W^4 determines the silhouette segments whose starting and end points intersect points between the silhouette and lines l_1 and l_2. The ground plane is used to eliminate the silhouette segment which is on the opposite side of the head with respect to the median coordinate. Once we identify the silhouette segment for the head location, the hull vertices which are on that segment are prelabeled as head. Among those, the vertex which has maximum path distance to the median coordinate is selected as the head point.

After W^4 determines the head location, it tries to find the other primary body parts in the order of feet, hands, and torso using prior-knowledge about the topology of the estimated body posture.

Let M^i be a subset of the primary and secondary body parts which are predicted to be visible given the estimated

TABLE 3
Performance of the System on 170 Silhouettes

	Standing	Bending	Lying	Sitting
Standing	96	12	0	8
Bending	4	80	6	8
Lying	0	4	90	2
Sitting	0	4	3	82

Columns indicate percentage of the true posture and rows indicate the classification.

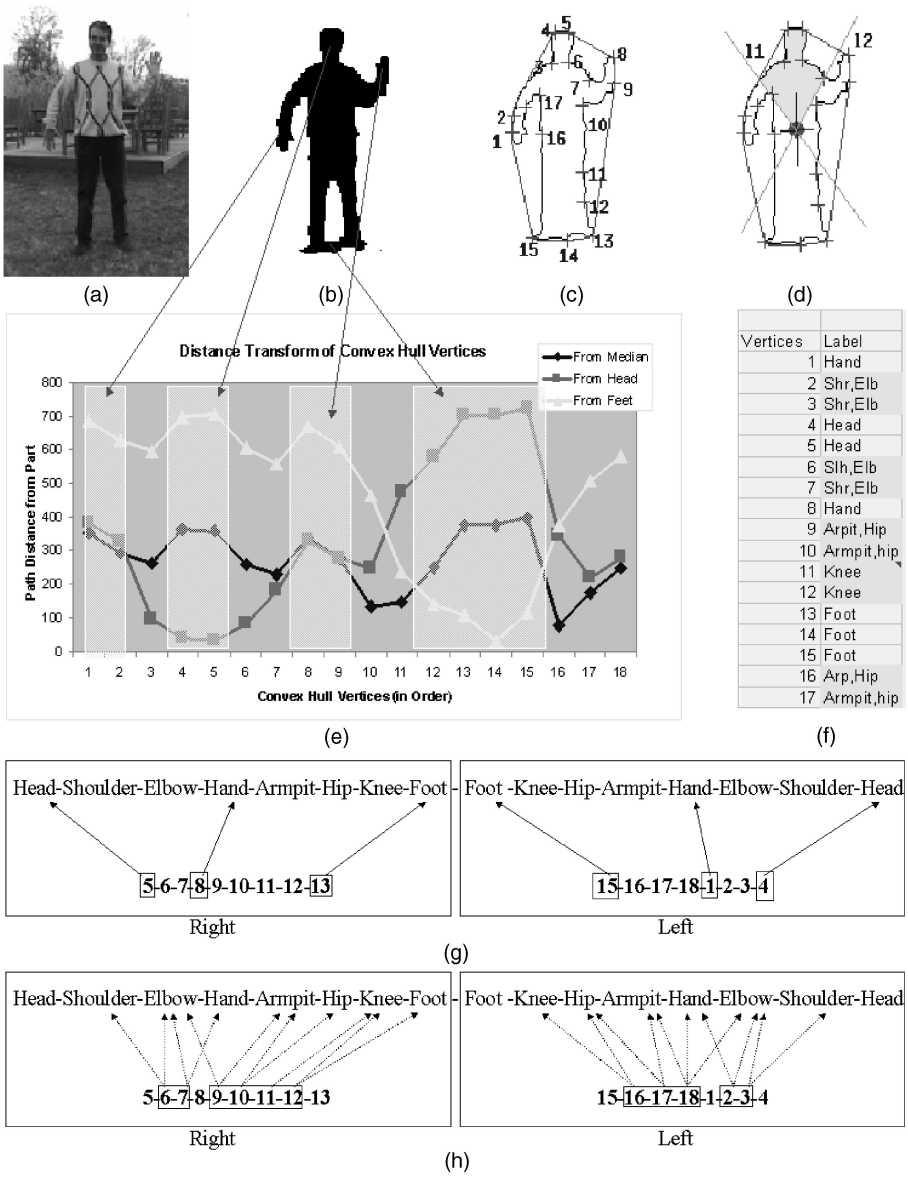


Fig. 15. An example showing how W^4 labels body parts: (a) original image, (b) detected silhouette, (c) detected convex and concave hull vertices, (d) silhouette segment for estimated head location, (e) distance transform measures for vertices from median, head and feet, and prelabeling of primary body parts (in shaded regions) after applying path distance constraints, (f) final labeling, (g) labeling primary body parts after applying the topological rules, (h) final labeling of secondary body parts after applying order constraints.

main posture i. Let V^t be the set of convex and concave hull vertices detected on the silhouette boundary. We need to find a partial mapping from M^i to V^t which should be consistent with the order of body parts for the estimated posture. W^4 uses path distance constraints (C^i), which contain information about the maximal relative path distances between body parts for each posture. Those constraints are applied to V^t to provisionally prelabel them. Then, W^4 uses topological rules (R) which are independent of posture to select the primary body parts among prelabeled vertices. At the final stage, W^4 uses topological order-constraints on body parts for the estimated posture for the final labeling of primary and secondary body parts. Constraints C^i on relative path distances and orders of the body parts for each main posture and view-based appearance of the main posture are computed experimentally. For example, C^{front}_{hand}, the relative distance from the head, hand to the median, should be bigger than some threshold. R_{hand} requires that a vertex has the maximal relative path distance from the median, labeled as hand among all the vertices which satisfy the hand constraint C^{front}_{hand}.

After the initial distance transform computation from the median and head are done, each hull vertex is prelabeled as one or more body parts according to whether it satisfies the path distance constraints. Fig. 15e shows the path distances from each convex hull vertex to the median, hand, and foot, and the prelabeling of vertices which satisfy the path distance constraints (shaded areas). Then, the topological rules are applied to the prelabeled vertices to locate the primary body parts; vertices 5, 8, 13, 15, and 1 are labeled as primary body parts, head, right hand, right foot, left foot, and left hand. This primary part mapping is shown in

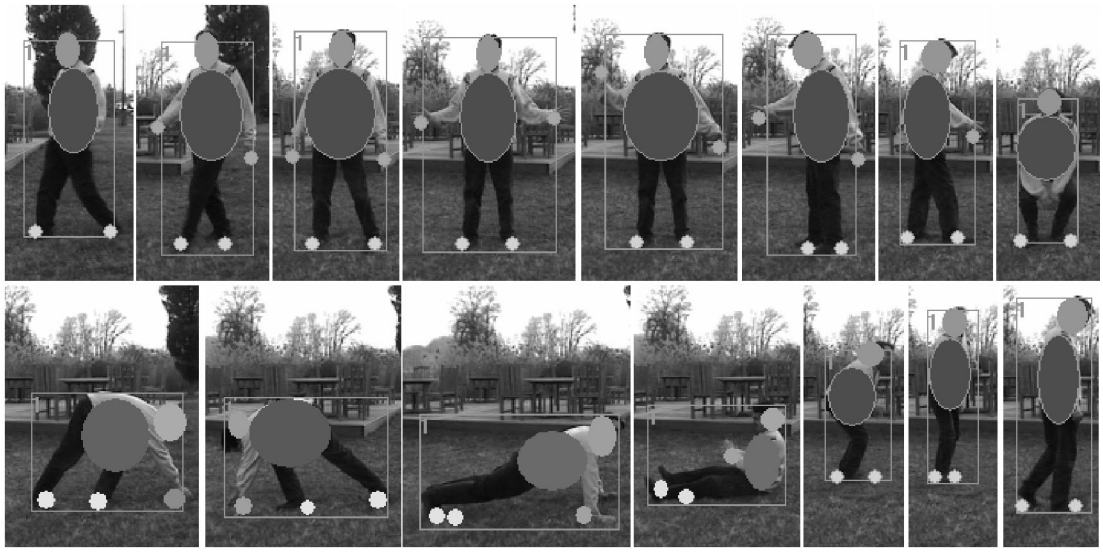


Fig. 16. Examples of using the silhouette model to locate the body parts in different actions.

Fig. 15g. Then, secondary body parts are labeled by applying the order constraints of the recognized posture (Fig. 15h). If more than one vertex is assigned to the same body part, another search is applied by narrowing the order constraints for those vertices; vertices 6, 7 were initially assigned to shoulder, but then finally vertex 6 is labeled as shoulder. While labeling a vertex, the distances to the previously labeled vertices should also be consistent. The torso is located between the median coordinate and the head along the major axis of the silhouette. Examples of body part labeling results are shown in Fig. 16.

3.3 Tracking of Body Parts

Gavrila [12] is a good survey on human body part tracking in 3D. We used a 2D approach to track body parts. W^4 uses template matching and motion prediction to track body parts for upright people. A second order motion model of body parts is employed by W^4. In each frame, after predicting the locations of the head and hands, their positions are verified and refined using temporal texture templates. These temporal texture templates are then updated as described previously, unless they are located within the silhouette of the torso. In this case, the pixels corresponding to the head and hand are embedded in the larger component corresponding to the torso. This makes it difficult to accurately estimate the median position of the part, or to determine which pixels within the torso are actual part pixels. In these cases, the parts are tracked using correlation, but the templates are not updated. The correlation results are monitored during tracking to determine if the correlation is good enough to track the parts correctly. Analyzing the changes in the correlation scores allows us to make predictions about whether a part is becoming occluded.

3.4 Detection of People Carrying Objects

Monitoring interactions between people and objects, and detecting unusual events such as depositing an object (unattended baggage in airports), exchanging bags, or removing an object (theft) requires an ability to detect people carrying objects, to segment the object from person, and to construct appearance model for the object so it can be identified in subsequent frames.

W^4 combines two basic observations to analyze people carrying objects: Human body shape is symmetric and people exhibit periodic motion while they are moving unencumbered. During tracking, the periodic motion of a person and his parts is estimated and the regions on the silhouette which systematically violate the symmetry-constraints are determined. Those results are combined to determine if a person is carrying an object and to segment the object from the silhouette. We construct an appearance

Fig. 17. (a) Example of people carrying an object and (b) the foreground regions detected by background subtraction.

model for each carried object, so that when people exchange objects, we can detect "who" carries "which" object via an analysis of the segmentation.

W^4 employs a global shape constraint derived from the requirement that the human body shape is symmetric around its body axis. W^4 uses that constraint to segment outlier regions from the silhouette. The expected shape model of a person is compared with the current person silhouette to determine the outlier regions (nonsymmetric region). One can observe that because of the motion of people's arms, legs, and hands, outliers are periodically detected in the vicinity of those body parts. However, outliers are detected continuously in the vicinity of a sufficiently large carried object because of continued symmetry constraint violations. Therefore, W^4 uses periodicity analysis to classify whether outlier regions belong to an object or a body part.

Symmetry Analysis. Silhouettes of humans are typically close to symmetric about the body axis while standing, walking, or running. Let l^s be the symmetry axis constructed for a given silhouette. Each pixel is classified as symmetric or nonsymmetric using the following simple procedure: Let p_l and p_r be a pair of pixels on the silhouette boundary such that the line segment from p_l to p_r is perpendicular to l^s and intersects with l^s at p_s (shown in Fig. 18). Let q_i^l and q_i^r be the length of line segment $[p_l, p_s]$ and length of line segment $[p_s, p_r]$, respectively. A pixel x lying on the line segment $[p_l, p_r]$ is classified as follows:

$$x = \begin{cases} \text{Nonsymmetric} & \text{if } q_s^x > min\{q_i^l, q_i^r\} + \epsilon \\ \text{Symmetric} & \text{otherwise,} \end{cases} \quad (11)$$

where q_s^x is the length of line segment from pixel x to p_s. Fig. 19 shows examples of symmetry-based segmentation results for people with and without an object by showing their detected head location, computed hypothetical symmetry axis, and nonsymmetric region segmentation.

3.5 Adding Periodicity

Nonsymmetric pixels are grouped together into regions and the shape periodicity of each nonsymmetric region is computed individually. The horizontal projection histogram segment bounded by a nonsymmetric region is used to compute the shape periodicity of the corresponding nonsymmetric region. A nonsymmetric region which does not exhibit significant periodicity is classified as an object

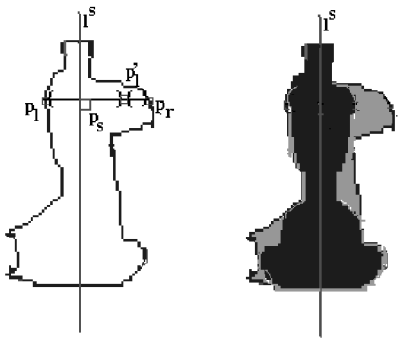


Fig. 18. Nonsymmetric region segmentation used in W^4.

carried by a person, while a nonsymmetric region which has significant periodicity is classified as a body part. In Fig. 20, the final classification results are shown for a walking person who is not carrying an object and a person who is carrying an object.

In the first example, a person is walking with 1 hz frequency (15 frames per half period with 100 percent confidence value); the similarity plot of the vertical projection histogram for the entire body is shown in Fig. 20a (right). Note that the legs and arms of the person violate the symmetry constraint periodically during walking. The pixels around the legs and arms are detected as nonsymmetric pixels and grouped into two nonsymmetric regions (region 1 around legs and region 2 around arms). Then, the similarity plots for region 1 and region 2 are obtained as shown in Fig. 20a. Note that the shape periodicity algorithm is applied only to the horizontal projection histogram segments bounded by regions 1 and 2. Periodicity is detected for region 1 at 1.1 hz and for region 2 at 1.03 Hz, which are very similar to the shape periodicity of the entire body. Therefore, those regions are classified as body parts (shown in green). In the second example, a person is walking and carrying a bag with 0.85 hz frequency (17.9 frame per half period with 98 percent confidence value); its similarity plot from the vertical projection histogram of the entire body is shown in the Fig. 20b. The legs of the person and the bag violate the symmetry constraint during walking and the regions around the legs (region 1) and the backpack (region 2) are grouped into nonsymmetric regions. Shape periodicity is detected for region 1 at 0.84 hz with high confidence and for region 2 at

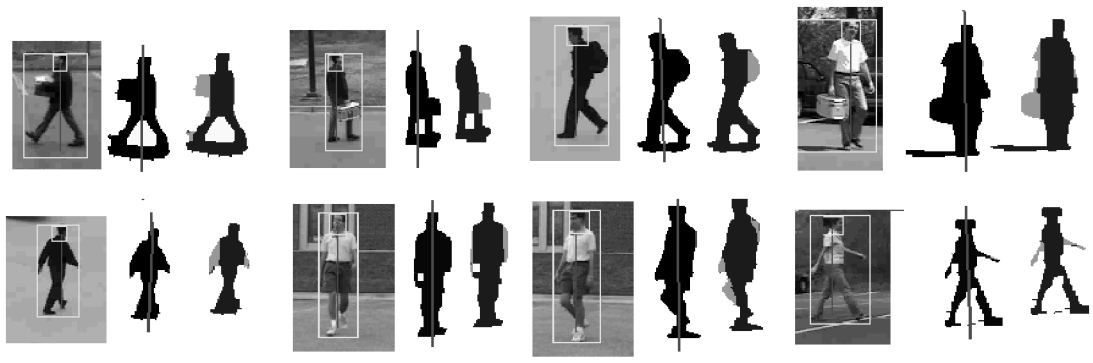


Fig. 19. Examples of symmetry-based segmentation results for people with and without an object by showing their detected head location, computed hypothetical symmetry axis, and final nonsymmetric region segmentation.

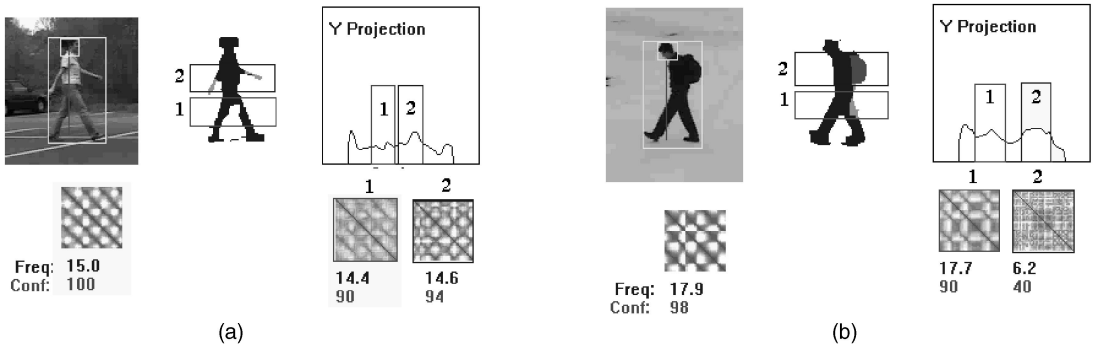


Fig. 20. Final object detection results based on nonsymmetric region segmentation and shape periodicity analysis of walking people without and with a bag.

2.5 Hz with low confidence. The periodicity of region 1 is very similar to the periodicity of the entire body and it is classified as a body part. However, region 2 does not have a significant fundamental frequency similar to the entire body, so it is classified as a carried object.

The symmetry and shape periodicity analysis used in W^4 are view-based techniques; the results depend on the direction of motion of the person, and location of the object on the silhouettes. Fig. 21 shows detection results where a person is carrying an object in his hand while moving in different directions. We ran a series of experiments using 100 sequences where a person is moving in different directions (people carry an object in 62 sequences, people do not carry an object in 38 sequences). We estimated the Receiver Operating Curve (ROC) which plots the probability of detection along y-axis and the probability of false detection along x-axis (Fig. 21 (right)). An ideal recognition algorithm would produce results near the top left of the graph (low false alarm and high detection probability). For different periodicity confidence thresholds, we computed the number of instances that are correctly classified as person-with-object (true positive) and the number of instances that are misclassified as people-with-object (false positive). For the optimal choice of thresholds, W^4 successfully determined whether a person is carrying an object in 91/100 sequences (Fig. 22). It generally failed on sequence where there is not a large nonsymmetric region (5/100) or insufficient shape changes (4/100) (causing low periodicity confidence value) e.g., when a person is moving toward to camera. In those cases, W^4 uses a nonglobal 2D-intensity based periodicity analysis [8] to compute periodicity to decrease the false positive rate (yielding a 95/100 success rate). W^4 uses appearances (shape, intensity, and position) information embedded into its temporal textural templates to track the objects if the object has been detected and its temporal textural template has been generated.

4 People in Groups

The shape analysis discussed in this section provides W4 with the ability to find and track people when there are small groups of people moving together, or interacting with each other. In these cases, individual people are not visually isolated, but are partially or total occluded by other people, as shown in Fig. 23. In this section, we describe a subsystem that detects multiple people when they move as groups and there is significant occlusion among them. It attempts to address the following problems:

- Given a foreground object corresponding to a collection of people, how many people comprise that object?
- Assuming that the W^4 has correctly established the number of people in the group, how can the group be segmented into its constituent individuals?

W^4 determines if an arbitrary foreground object corresponds to a group of people by analyzing its global shape and comparing it to shape models of individual people. It takes advantage of information obtained from prior views to analyze that group. For example, one could imagine a situation in which a person enters the field of view and joins an existing group; in this case, W^4 would have detected and

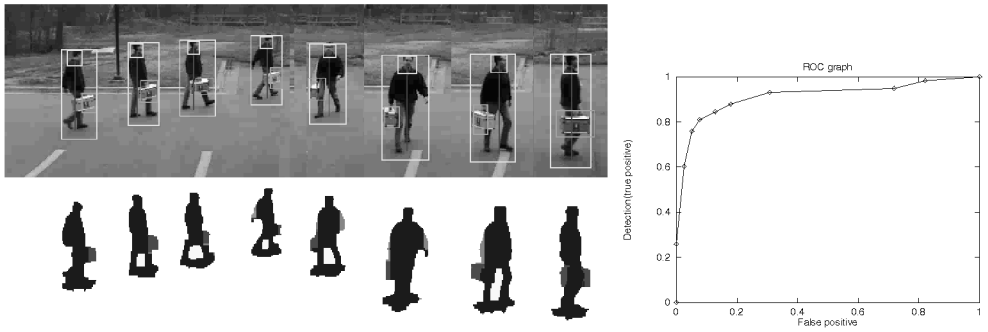


Fig. 21. Object detection results while a person is moving different directions and ROC curve for object detection.

Fig. 22. Object detection results while a person is carrying different types of object and moving different directions.

tracked that person and built an appearance model of that person; it employs these observations to update its model of that group.

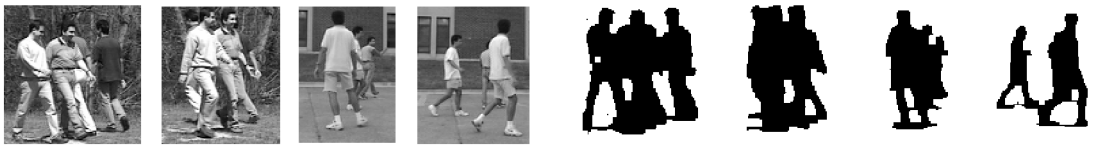
Intuitively, what types of information might be used to count people and segment groups?

- **Local shape information**. For example, by analyzing the boundary of the object, we might find pieces that look like heads, feet, hands, etc. In fact, W^4 operates by first attempting to locate heads based on local shape approximations to the object boundary.
- **Global shape information and constraints.** For example, people typically have two hands, arms, feet; their heads typically appear "above" the torso (i.e., along the axis of the torso), etc. W^4 employs a global shape constraint derived from the requirement that the head be aligned with the axis of the torso. In particular, by projecting the object region on an axis perpendicular to the assumed torso axis, one should observe a peak in the projection in the vicinity of a head. W^4 uses this constraint to eliminate false alarm heads detected by the local shape analysis. Additionally, one could use global shape information in segmentation. On the one hand, if pieces of the region can be identified as body parts, then in any hypothetical segmentation there are bounds on the number of each part type in each segment. Alternatively, there are path distance constraints among body parts and features (e.g., between the hand and the head), that could be used to assign pixels or segments of the object to individuals. W^4 performs this segmentation by creating a set of normalized distance maps—based on path distances to hypothesized body torso axes—that assign each object pixel to an individual in the group. In the future, we plan to extend this analysis to regions labeled as body parts so that we can use more informal models to perform the assignment, and combine this with discrete constraints on assignment of body parts to individuals.
- **Appearance information.** So, for example, a hypothesized head could be verified by matching the texture of the region to a prototypical face (as in face detection) assuming the face was visible from the camera. Or, one could use color and texture to segment the object into regions of uniformity that would be constrained to be assigned to a single individual in the group. W^4 currently makes no use of such appearance information in analyzing groups.

W^4 attempts to count the number of people in the group by identifying their heads. It is typical that people's heads lie on the silhouette boundary and most of the time they are visible (notable exceptions would be groups of football players competing to recover to fumble). Therefore, the presence of a head is a very useful cue to detect an individual person.

W^4 combines two methods based on geometric shape cues (convex hull-corner vertices on silhouette boundary) and the vertical projection histogram of the binary silhouette to detect heads (Fig. 24). Corner vertices are classified as to whether the shape in their vicinity is similar to the expected head shape, based on the curvature of a local set of the vertices. Vertices which have nonhead like curvature patterns are eliminated.

Significant peaks on the vertical projection histogram of the silhouettes are used to filter the results of the local shape analysis,—i.e., potential heads are retained only if there are significant projection histogram peaks in their vicinity. This



(a) (b)

Fig. 23. (a) Example of multiple people moving together and (b) the foreground regions detected by background subtraction.

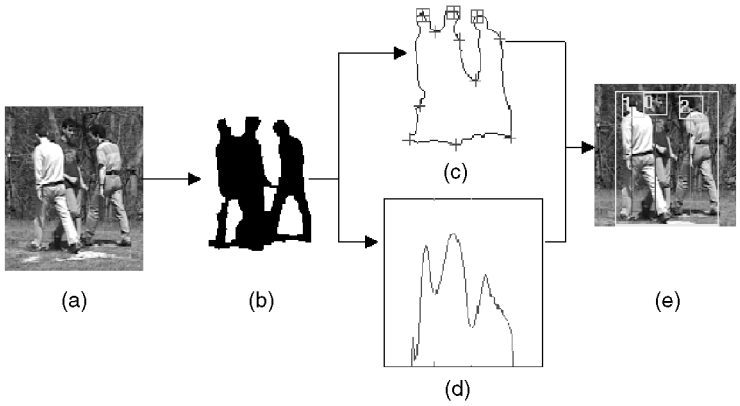


Fig. 24. (a) Silhouetted analysis, (b) silhouette extraction, (c) convex-hull/corner detection, (d) vertical projection histogram, and (e) final head detection.

step enforces the constraint that heads appears "above" the torso. The peak threshold value is selected as the mean value of the entire histogram. An example of how false positives are eliminated is shown in Fig. 25. The silhouette method detects three heads, because the shapes of the hands are similar to the expected head shape. However, the histogram method does not support the silhouette method at the location of the hands (there are no high peaks near those locations). Therefore, those false detections (labeled 1 and 3 in Fig. 25) are eliminated, and W^4 detects the head in the correct position.

W^4 next segments the foreground regions into subregions representing individual people using the local geometry of the silhouette. To do this, it first computes a path distances $d(x,y)$ from the hypothesized body torso axes for each person y to each pixel x in the region. This path distance is just the distance transform of the silhouette with respect to the torso axis—i.e., the minimum path length from pixel x to the torso axis of person y, where the path is constrained to be within the silhouette.

W^4 then normalizes the path distances $d(x,y)$ to normalized distance values , $n(x,y)$, as follows:

$$n(x,y) = \frac{\alpha(x,y)}{\sum_z(\alpha(x,z))}, \text{where } \alpha(x,y) = \frac{min_z(d(x,z))}{d(x,y)}. \quad (12)$$

In Fig. 26, an example of these normalized distance maps is shown. W^4 converts the normalized distance map into a binary support map $sM(y)$ which contains all the pixels in the blob whose normalized distance $n(x,y)$ is higher than a threshold. This "fuzzy" segmentation allows W^4 to detect each individual person in the group, as shown in Fig. 27. Note that depending on the threshold value, a pixel can either be assigned to a single person, or to multiple people. The Support maps, $sM(y)$, are also used to build the appearance model (temporal textural templates) for each person, as explained in the next section.

4.1 Tracking People in Groups

W^4 creates a dynamic intensity template for each detected head, updates them during tracking, and tracks them using correlation-based matching. The head-matching algorithm consists of two main parts. In the first part, W^4 determines a coarse-matching between tracked heads and detected heads by using a simple occupancy test between the bounding box of the estimated (via the motion model) head location and the bounding box of detected head locations in the current frame. In the second part, appearance similarity between matched pairs (via intensity-based template correlation) is computed. The correlation results are also monitored during tracking to determine whether a person is going to be occluded, or will occlude someone else. W^4 determines the best and worst correlation scores in a 5×3 search window and compares those values to determine if the correlation score is good enough for correct matching. During total occlusion, the location of a head is only predicted using a recursive least-square estimation method using the motion model; its template is not updated until the head reappears.

W^4 employs a second order motion model for each person in the group to estimate its location in subsequent frames similar to when W^4 tracks an isolated person. The prediction from this model is used to estimate a bounding box location for each head. These predicted bounding boxes are then

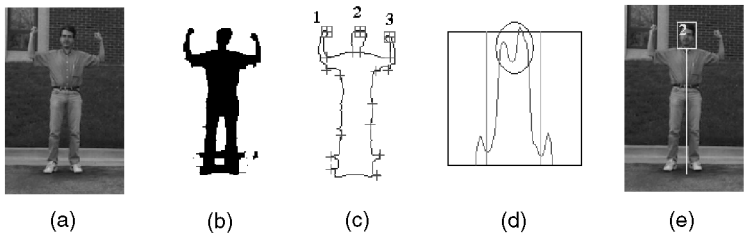


Fig. 25. (c) An example of head detection using silhouette boundary based only. (d) Vertical histogram-based only. (e) Final result after combining these two methods.

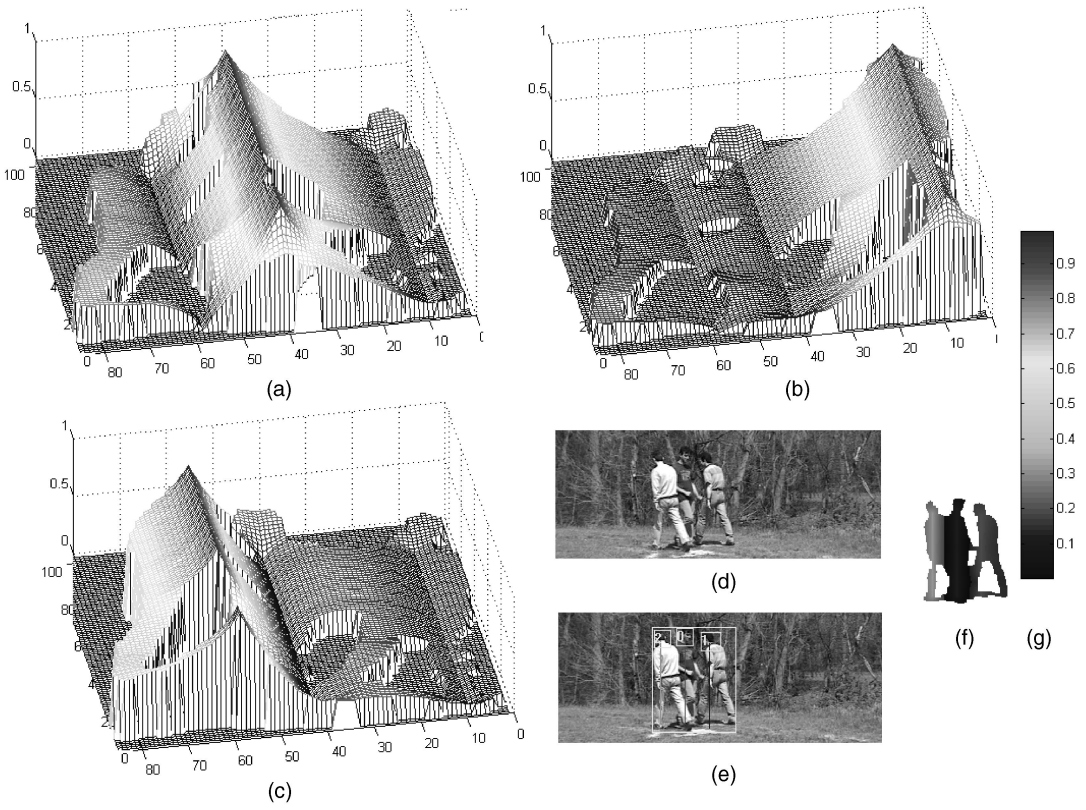


Fig. 26. An example of normalized distance maps for person 0, 1, and 2 are shown in (a), (c), and (b), respectively. The input image is in (d), head detection and tracking results in (f), and final segmentation based on normalized distances is in (e). (g) Shows the color coded normalization scale.

compared to the actual bounding boxes of the detected heads. The initial estimate of displacement is again computed as the motion of the median coordinate of the foreground region as we did for tracking isolated people, which allows us to quickly narrow the search space for the motion of the people. However, this estimate is not accurate enough for each person in that group. Therefore, after displacing the silhouette of the foreground region from the previous frame by the median-based estimate, the silhouette boundary is segmented into small edge-segments and each segment is assigned to a person according to their position in previous frames. Then, we perform a binary edge correlation between the current and previous silhouette edge segments for each person to obtain the local motion of the person (Fig. 28). As a final stage , the local motion of the head is calculated by correlating the head templates with the image in a 5×3 search window, centered at the predicted location of a person's head determined by the second stage estimation of body motion. The best correlation gives the motion of the head for that person.

During tracking, W^4 update its current tracking information (e.g., number of people currently being tracked). W^4 tracks both the foreground regions and individuals within each foreground region. When a person who has been tracked as single person joins a group whose individual's are being tracked by W^4, all appearance and motion information about that person is updated and W^4 starts to track that person as part of the group. Similarly, when a person leaves a group, W^4 tracks that person individually.

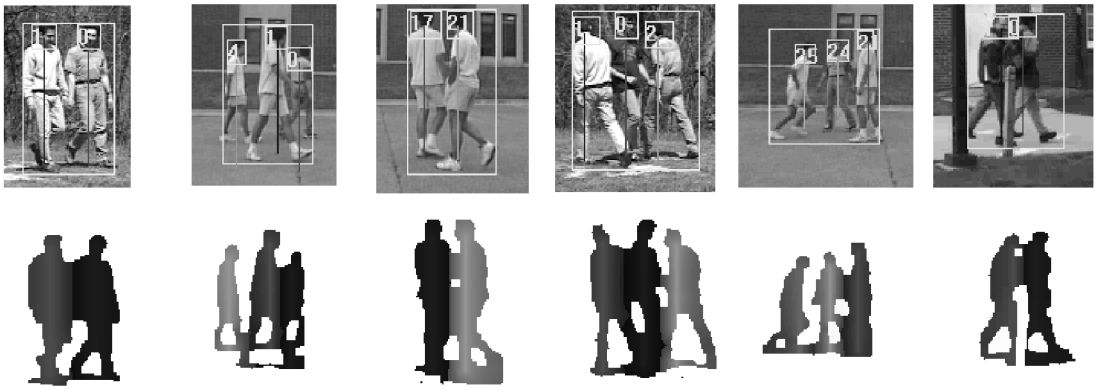


Fig. 27. Examples of person segmentation applied to foreground regions in a single image.

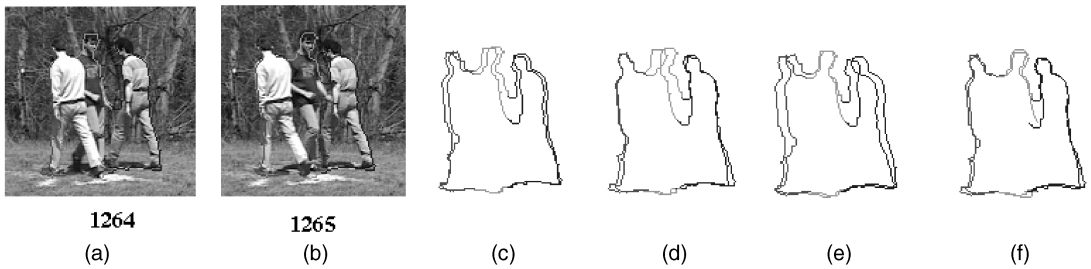


Fig. 28. Local motion estimation of each person based on binary silhouette boundary edge matching between two successive frame (a) and (b). Alignment of the silhouette edge segments without local motion correction (c), alignment of the silhouette edge segments based on only left-most person's local motion $(-2, 0)$ (d), based on only middle person's local motion $(3, 0)$ (e), and final alignment based on local motion of each person.

This two level tracking allows our system to track individual people even when multiple groups merge together or a big group splits into smaller groups.

W^4 constructs a temporal texture template while it is tracking and segmenting individual people. As the median coordinate of individual people cannot be correctly predicted while people move together, all coordinates are represented relative to the centroid of the head, instead of relative to the median coordinate of the body, as in (8).

We tested our method using 77 short image sequences where there are groups of people moving together causing total or partial occlusion. The number of people in each group is changing from two people to five people and they are moving in different directions in the scene. Fig. 29 illustrates tracking multiple people in some sample sequences. In 65 sequences, W^4 correctly counted the number of people in the scene. In eight sequences, W^4 detected fewer people than the actual number of people in the groups. The main reason for those failures (false negatives) is that the heads were occluded in the majority of images in these sequences. In four sequences, W^4 detected higher number of people (false positives) than the actual number of people in the scene. The main reason for those false positives is failures on foreground region detection that cause false head-like shapes on silhouettes.

5 Conclusion

We have described a real-time visual surveillance system, W^4, for detecting and tracking people and monitoring their activities in an outdoor environment. It operates on monocular gray-scale video imagery, or on video imagery from an infrared camera. Unlike many systems for tracking people, W^4 makes no use of color cues. Instead, W^4 employs a combination of shape analysis and tracking to locate people and their parts (head, hands, feet, torso) and to create models of people's appearance so that they can be tracked through interactions such as occlusions. W^4 is

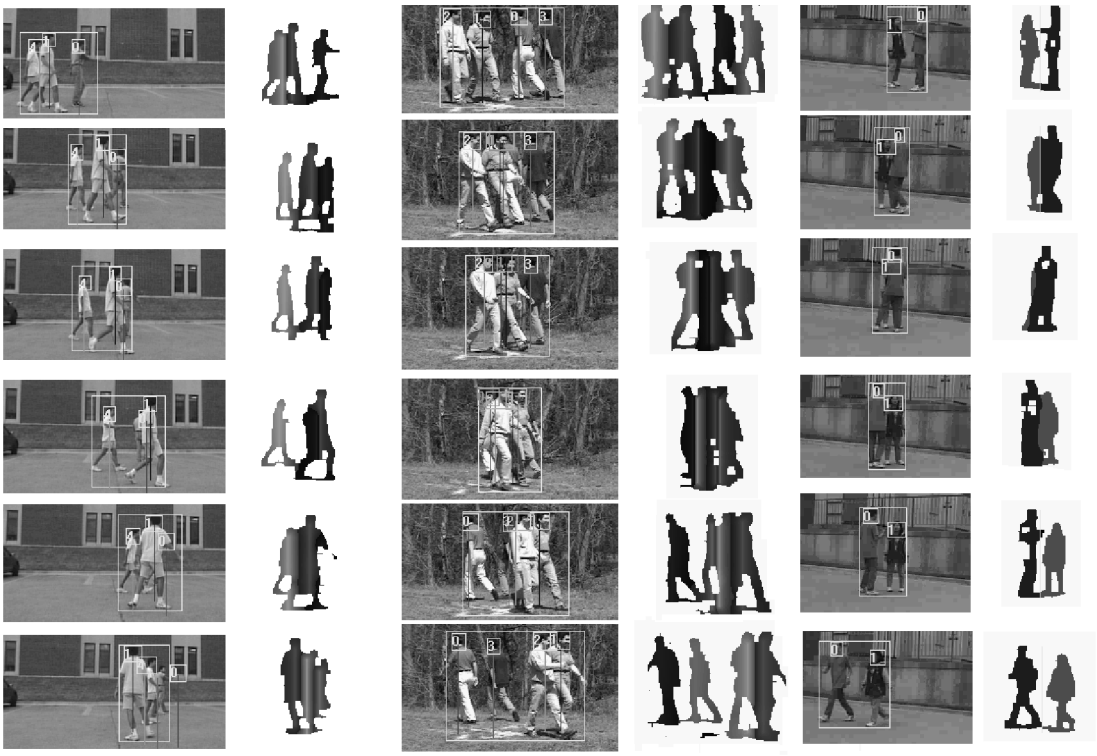


Fig. 29. Examples of detection and tracking multiple people.

Fig. 30. Images from example used in benchmarking.

capable of simultaneously tracking multiple people even with occlusion.

W^4 has been implemented in C++ and runs under the Windows NT operating system. Currently, for 320×240 resolution gray-scale images, W^4 runs at 20-30 Hz dual 400 Mhz Pentium PC, depending on the number of people in its field of view. SIMD type programming has been intensively used in the detection phase to achieve real time performance in a single processor. Table 1 gives the average execution times of each component of W^4 for four different image sequence where there are different numbers of people in the scene. W^4 utilized the second processor for overlapping the detection stage of image $n+1$ with the tracking stage of image n. By overlapping those two stage, W^4 achieves 65-80 percent improvement in execution time.

W^4 is primarily designed for outdoor surveillance; however, there are many other application areas for W^4. We have already used some parts of W^4 to develop a video-based motion capture system. A version of the W^4 system, extended to operate on color images, is run on each of multiple cameras observing a person. Its silhouette analysis and template matching achieve real-time 3D estimation of human postures. The estimated body postures are then reproduced in a 3D graphical character model by deforming the model according to the estimated data. The dynamics/kinematics model of human motion and Kalman filters [35] are utilized to help the tracking process, as well as to interpolate some 3D joint locations (i.e., elbows). The system runs on a network of Dual-Pentium 400 PCs at 28 frames per second. This project was successfully demonstrated in a demonstration in SIGGRAPH '98, Emerging Technology [20].

ACKNOWLEDGMENTS

The authors would like to thank Ross Cutler for discussions and comments on periodicity analysis and other system integration issues, and Esin Darici, Mehmet Altinel, Ugur Cetintemel, Tahsin Kurc, and Tolga Urhan and Thanarat Horprasert for their help and effort for video sequences used in this work.

TABLE 4
Execution Times in ms for Sample Image Sequences (Fig. 30) Where there Are One (a), Two (b), and (d) Three (c) People in the Current Scene Interacting with Each Other

	Examples in Figure 30	(a)	(b)	(c)	(d)
	% Foreground Region	6%	9%	15%	21%
Detection	Thresholding	3.69	3.73	3.79	3.85
	Connected components	2.01	2.40	2.98	2.72
	Noise removing	3.08	3.67	4.48	4.15
	Morphological op.	1.25	1.70	2.47	2.26
	Corner, projections	0.63	0.89	1.02	1.33
	Head detection	11.75	11.65	21.08	19.18
	People Segmentation	5.22	11.65	12.08	15.58
	Symmetric Analysis	1.77	2.77	1.54	3.42
Tracking	Forward Matching	0.02	0.024	0.05	0.03
	Motion Analysis	2.39	3.97	7.99	5.98
	Temporal Texture Tmp.	1.17	1.20	2.56	1.6
	Motion-Analysis	0.60	1.72	2.78	2.21
	Head Tracking	1.23	2.72	3.67	3.59
	Periodicity	3.98	1.61	2.49	4.72
Exec. Speed (fps)	Single Proc.	26.14	24.76	15.07	17.15
	Dual Proc.	39.98	37.54	26.07	28.86

REFERENCES

[1] K. Akita, "Image Sequence Analysis of Real World Human Motion, *Pattern Recognition,* vol. 17, no. 4, pp. 73-83, 1984.

[2] A. Azarbayjani, C. Wren, and A. Pentland, "Real-Time 3D Tracking of the Human Body," *Proc. IMAGE'COM,* 1996.

[3] D. Beymer and K. Konolige, "Real-Time Tracking of Multiple People Using Stereo," *Proc. IEEE Frame Rate Workshop,* 1999.

[4] A. Bobick and J. Davis, "Real-Time Recognition of Activity Using Temporal Templates," *Proc. IEEE Workshop Application of Computer Vision,* pp. 1,233-1,251, 1996.

[5] A. Bobick, J. Davis, S. Intille, F. Baird, L. Cambell, Y. Irinov, C. Pinhanez, and A. Wilson., "Kidsroom: Action Recognition in an Interactive Story Environment," Technical Report 398, M.I.T. Perceptual Computing, 1996.

[6] T. Boult, "Frame-Rate Multibody Tracking for Surveillance," *Proc. DARPA Image Understanding Workshop,* 1998.

[7] C. Bregler and J. Malik, "Tracking People with Twists and Exponential Maps," *Proc. Computer Vision and Pattern Recognition Conf.,* pp. 8-15, 1998.

[8] R. Cutler and L. Davis, "View-Based Detection and Analysis of Periodic Motion," *Proc. Int'l Conf. Pattern Recognition,* 1998.

[9] T. Darell, G. Gordon, M. Harville, J. Woodfill, "Integrated Person Tracking Using Stereo, Color, and Pattern Detection," *Computer Vision and Pattern Recognition,* 1998.

[10] A. Elgammal, D. Harwood, and L. Davis, "Non-Parametric Model for Background Subtraction," *Proc. IEEE Frame Rate Workshop,* 1999.

[11] N. Friedman and S. Russell, "Image Segmentation in Video Sequences: A Probabilistic Approach," *Uncertainty in Artificial Intelligence,* 1997.

[12] D. Gavrila, "The Visual Analysis of Human Movement: A Survey," *Computer Vision Image Understanding,* vol. 73, no. 1, 1999.

[13] E. Grimson, C. Stauffer, R. Romano, and L. Lee, "Using Adaptive Tracking to Classify and Monitoring Activities in a Site," *Proc. Computer Vision and Pattern Recognition Conf.,* pp. 22-29, 1998.

[14] E. Grimson and C. Stauffer, "Adaptive Background Mixture Models for Real Time Tracking," *Proc. Computer Vision and Pattern Recognition Conf.,* 1999.

[15] I. Haritaoglu, D. Harwood, and L. Davis, "W4: Who, When, Where, What: A Real Time System for Detecting and Tracking People," *Proc. Third Face and Gesture Recognition Conf.,* pp. 222-227, 1998.

[16] I. Haritaoglu, D. Harwood, and L. Davis, "W4S: A Real Time System for Detecting and Tracking People in 2.5D," *European Conf. Computer Vision,* 1998.

[17] I. Haritaoglu, D. Harwood, and L. Davis, "Ghost: A Human Body Part Labeling System Using Silhouettes," *Proc. Int'l Conf. Pattern Recognition,* 1998.

[18] I. Haritaoglu, D. Harwood, and L. Davis, "Backpack: Detecting People Carrying Object Using Silhouettes," *Proc. Int'l Conf. Computer Vision,* 1999.

[19] I. Haritaoglu, "W4: A Real Time System for Detection and Tracking of People and Monitoring Their Activities," PhD thesis, University of Maryland, Computer Science Dept., 1999.

[20] T. Horprasert, I. Haritaoglu, D. Harwood, L. Davis, C. Wren, and A. Pentland, "Real-Time 3D Motion Capture," *Proc. Second Workshop Perceptual Interfaces,* Nov. 1998.

[21] T. Horprasert, D. Harwood, and L.S. Davis, "A Robust Background Subtraction and Shadow Detection," *Proc. Asian Conf. Computer Vision,* Jan. 2000.

[22] S. Intille, J. Davis, and A. Bobick, "Real-Time Closed-World Tracking," *Proc. Computer Vision and Pattern Recognition Conf.,* pp. 697-703, 1997.

[23] S. Iwasawa, K. Ebihara, J. Ohya, and S. Morishima, "Real-Time Estimation of Human Body Posture from Monocular Thermal Images," *Proc. Computer Vision and Pattern Recognition,* 1997.

[24] K. Konolige, "Small Vision Systems: Hardware and Implementation," *Proc. Int'l Symp. Robotics Research,* 1997.

[25] A. Lipton, H. Fujiyoshi, and R. Patil, "Moving Target Detection and Classification from Real-Time Video," *Proc. IEEE Workshop Application of Computer Vision,* 1998.

[26] M. Leung and Y.H. Yang, "A Model Based Approach to Labeling Human Body Outlines," 1994.

[27] R Morris and D. Hogg, "Statistical Models of Object Interactions," *Proc. IEEE Workshop Visual Surveillance,* 1998.

[28] N. Oliver, B. Rosario, and A. Pentland, "Statistical Modeling of Human Interactions," *Proc. Workshop Interpretation of Visual Motion,* pp. 8-15, 1998.

[29] T. Olson and F. Brill, "Moving Object Detection and Event Recognition Algorithms for Smart Cameras," *Proc. DARPA Image Understanding Workshop,* pp. 159-175, 1997.

[30] J. O'Rourke and N. Badler, "Model-Based Image Analysis of Human Motion Using Constraint Propagation," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 2, pp. 522-536, 1980.

[31] J. Rehg, M. Loughlin, and K. Waters, "Vision for a Smart Kiosk," *Computer Vision and Pattern Recognition,* 1997.

[32] J. Segen and S. Pingali, "A Camera-Based System for Tracking People in Real-Time," *Proc. Int'l Conf. Computer Vision,"* 1996.

[33] Y.Y. Shanon, X. Ju, and M.J. Black, "Cardboard People: A Parameterized Model of Articulated Image Motion," *Proc. Second Int'l Conf. Automatic Face and Gesture Recognition,* 1996.

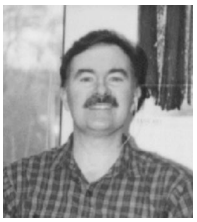
[34] C. Wren, A. Azarbayejani, T. Darrell, and A. Pentland, "Pfinder: Real-Time Tracking of the Human Body," *IEEE Trans. Pattern Analysis and Machine Intelligence* vol. 19, no. 7, July 1997.

[35] C. Wren and A. Pentland, "Dynamic Modeling of Human Motion," *Proc. Third IEEE Int'l Conf. Automatic Face and Gesture Recognition,* 1998.

[36] A. Selinger and L. Wixson, "Classfying Moving Object as Rigid or Non-Rigid without Correspondences," *Proc. DARPA Image Understanding Workshop,"* 1998.

[37] A. Shafer, J. Krumm, B Brumitt, B. Meyers, M. Czerwinski, and D. Robbins, "The New EasyLiving Project at Microsoft," *Proc. DARPA/NIST Smart Spaces Workshop,* 1998.

Ismail Haritaoglu received the BS and MS degrees in computer engineering and information science from Bilkent University, Turkey in 1992 and 1994, and the PhD degree in computer science from University of Maryland, College Park, in 1999. He studied real-time computer vision systems for monitoring human activities under the supervision of Dr. Larry Davis. Currently, he is a research scientist at the IBM Almaden Research Center, San Jose, California. His research interests are vision systems for human-computer interactions, perceptual user interfaces, interactive entertainment systems, and smart surveillance systems. He is a member of the IEEE.



David Harwood is a senior member of the research staff of the Institute for Advanced Computer Studies at the University of Maryland, College Park, with more than 40 publications in the field of computer image analysis. He is a member of the IEEE.

Larry S. Davis received the BA degree from Colgate University in 1970, the MS, and PhD degrees in computer science from the University of Maryland, College Park, in 1974 and 1976, respectively. From 1977-1981, he was an assistant professor in the Department of Computer Science at the University of Texas, Austin. He returned to the University of Maryland as an associate professor in 1981. From 1985-1994, he was the director of the University of Maryland Institute for Advanced Computer Studies. He is currently a professor at the Institute and the Computer Science Department, as well as chair of the Computer Science Department. He was named a fellow of the IEEE in 1997. Professor Davis is known for his research in computer vision and high performance computing. He has published more than 75 papers in journals and has supervised more than 15 PhD students. He is an associate editor of the *International Journal of Computer Vision* and an area editor for *Computer Models for Image Processor: Image Understanding*. He has served as program or general chair for most of the field's major conferences and workshops, including the Fifth International Conference on Computer Vision, the field's leading international conference. He is a fellow of the IEEE.