Document Title:	Real Time Computer Surveillance for Crime Detection
Author(s):	Larry S. Davis
Document No.:	192734
Date Received:	February 22, 2002
Award Number:	1999-LT-VX-K019

192734

c.1

Real Time Computer Surveilance for Crime Detection

Larry S. Davis
Computer Vision Laboratory
University of Maryland

Sponsored by:
National Institutes of Justice

September 2001

Chapter 1

Introduction

The Computer Vision Laboratory at the University of Maryland has been designing and developing real time computer vision algorithms for visual surveillance systems. A visual surveillance system must be able to detect and track people under a wide variety of environmental and imaging conditions, and then must analyze their actions and interactions with one another and with objects in their environment to determine when "alerts" should be posted to human security officers. Our research addressed three fundamental problems in the development of such systems:

1. Robust algorithms for detection of people in outdoor environments. There are many factors that complicate the problem of detecting people from a stationary camera against "fixed" backgrounds, including changes in illumination conditions (either sudden changes due to cloud cover changes or gradual changes as the sun moves across the sky), background movement due to wind load, or changes in weather such as rainfall or snow. We developed a novel approach to background modeling and model adaptation that deals effectively with these sources of variation and implemented a real time version of the algorithm that can detect people against complex backgrounds and under changing environmental conditions.

2. Models for tracking multiple people using multiple cameras - for surveillance over large areas it is unlikely that the surveillance system would have sufficient cameras to monitor the entire surveillance area at high resolution at all times. Instead, the cameras must be multiplexed to obtain that coverage - i.e., scanned over either regular or activity dependent paths to detect, track and analyze human activity. We developed a control model similar to those used for resource

allocation in computer systems to determine when and where cameras should look to maximize the number of targets that can be detected and tracked.

3. Finally, once a person is detected and tracked we must analyze that person's behavior. Sometimes this is as simple as determining whether or not a person enters a "prohibited" area, but often it requires analyzing the interactions that a person has with other people and with objects. In particular, it is important to determine if a person is carrying an object and to be able to visually separate the object from the person carrying it so that it can be analyzed by other vision algorithms (e.g., is it a gun or a broom?). We have extended and improved upon previous research we have conducted on determining whether or not a person is carrying an object so that we can deal with a wider variety of objects than before.

Each of these three research areas is amplified, with examples, in the remaining sections of the report.

Chapter 2

Nonparametric Background Subtraction

2.1 Introduction

In video surveillance systems, stationary cameras are typically used to monitor activities at outdoor or indoor sites. Since the cameras are stationary, the detection of moving objects can be achieved by comparing each new frame with a representation of the scene background. This process is called background subtraction and the scene representation is called the background model. Typically, background subtraction forms the first stage in automated visual surveillance systems. Results from background subtraction are used for further processing, such as tracking targets and understanding events.

Typically, in outdoor environments with moving trees and bushes, the background of the scene is not completely static. For example, one pixel can be the image of the sky at one frame, a tree leaf at another frame, a tree branch on a third frame and some mixture subsequently; in each situation the pixel will have a different intensity (color). This research focuses on how to construct a statistical representation of the scene background that supports sensitive detection of moving objects in hard outdoor situations.

2.2 Background Modeling

The model keeps a sample of intensity values for each pixel in the image and uses this sample to estimate the probability density function of the pixel intensity. The density function is estimated using kernel density estimation technique. Since this approach is quite general, the model can approximate any distribution for the pixel intensity without any assumption about the underlying distribution shape. Figure 2.1-b shows the estimated background probability where brighter pixels represent lower background probability pixels.

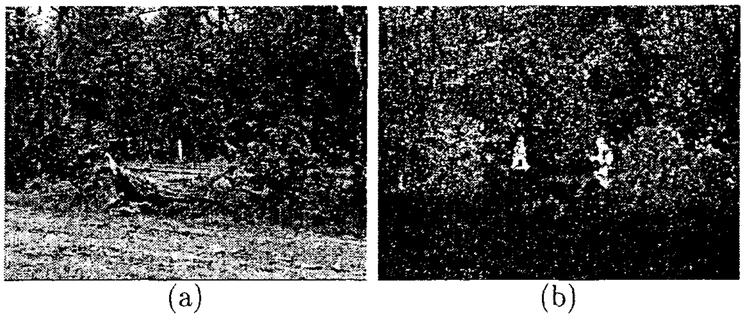


Figure 2.1: Background Subtraction. (a) original image. (b) Estimated probability image.

The model can handle situations where the background of the scene is cluttered and not completely static but contains small motions that are due to moving tree branches and bushes. The model is updated continuously and therefore adapts to changes in the scene background. The approach runs in real-time.

2.3 Probabilistic Suppression of False Detection

In outdoor environments with fluctuating backgrounds, there are two sources of false detections. First, there are false detections due to random noise which are expected to be homogeneous over the entire image. Second, there are false detections due to small movements in the scene background that are not represented by the background model. This can occur locally, for example, if a tree branch moves further than it did during model generation. This can also occur globally in the image as a result of small camera displacements caused by wind load, which is common in outdoor surveillance and causes many false detections. These kinds of false detections are usually spatially clustered in the image and they are not easy to eliminate using morphological techniques or noise filtering because these operations might also affect detection of small and/or occluded targets.

The second stage of detection aims to suppress the false detections due to small and unmodelled movements in the scene background. If some part of the background (a tree branch for example) moves to occupy a new pixel, but it was not part of the model for that pixel, then it will be detected as a foreground object. However, this object will have a high probability to be a part of the background distribution at its original pixel. Assuming that only a small displacement can occur between consecutive frames, we decide if a detected pixel is caused by a background object that has moved by considering the background distributions in a small neighborhood of the detection.

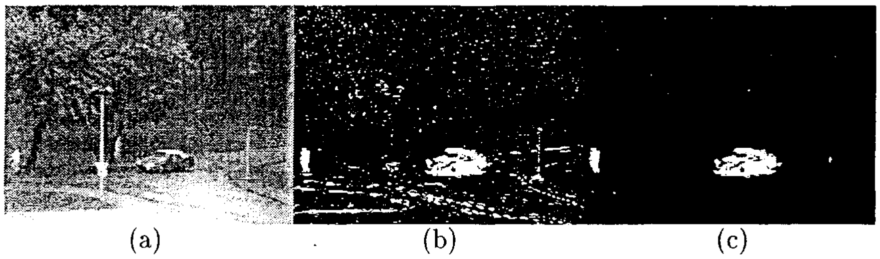


Figure 2.2: b) Result after the first stage of detection. (c) Result after the second stage

Figure 2.2-b shows results for a case where as a result of the wind load the camera is shaking slightly, resulting in a lot of clustered false detections especially on the edges. After probabilistic suppression of false detection (figure 2.2-c) most of these clustered false detection are suppressed, while the small target on the left side of the image remains.

2.4 Shadow Suppression

The detection of shadows as part of the foreground regions is a source of confusion for subsequent phases of analysis. It is desirable to discriminate between targets and their shadows. Color information is useful for suppressing shadows from the detection by separating color information from lightness information. Figure 2.3 shows the detection results for an indoor scene using both the (R, G, B) color space and the (r, g) color space after using the lightness variable, s, to restrict the sample set to relevant values only.

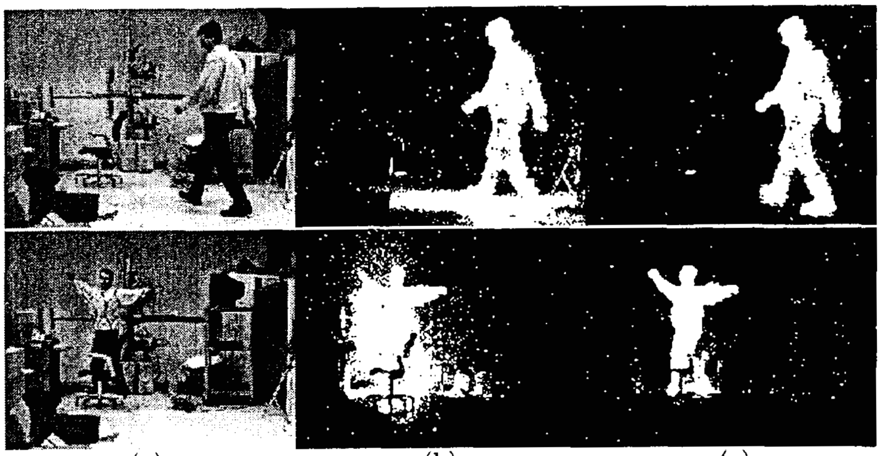


Figure 2.3: (b) Detection using (R, G, B) color space (c) detection using chromaticity coordinates (r, g) and the lightness variable s.

2.5 Detection Results

Figures 2.4 shows three frames from three sequences with different environments. Top figure shows detection results for a target in a wooded area where the tree branches are heavily moving and the target is highly occluded. Figure 2.4-middle shows the detection results using an omni-directional camera for camouflaged targets walking through woods. Figure 2.4-bottom shows the detection result for a rainy day where the background model adapts to different rain conditions and successfully detect the moving vehicle.

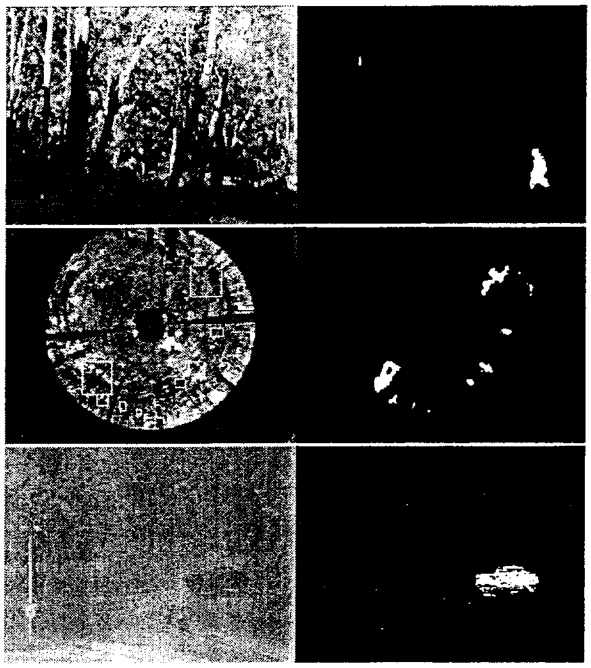


Figure 2.4: Top:Detection result for an omni-directional camera. Bottom:Detection result for a rainy day.

Chapter 3

Multiplexing a Single Camera to Track Multiple Targets

3.1 Introduction

We studied the problem of designing a surveillance system, equipped with a single camera, to track multiple moving targets in the camera's field of regard (FOR) in real time. The main objectives are to accommodate as many targets as possible, and to maintain each tracked target for as long as possible.

We assume that the camera is stationary except for pan/tilt rotation, and that its internal calibration parameters and position (in some world frame) are known. We also assume that an external mechanism, such as a Moving Target Indicator (MTI), does the initial detection of moving targets in the camera's FOR and cues our system with their initial 3D positions. Finally, we assume the targets are moving on a known surface and their motions are sufficiently modelled with first-order dynamics.

The camera's field of view indeed might only cover a small fraction of the entire field of regard (FOR) at any time. Furthermore, the camera is to be shared by targets moving anywhere within the entire field of regard. Hence the problem involves two main challenges:

- To manage the time-allocation, or multiplexing, of the camera among different areas of its field of regard, taking into account the varying 'needs' of the targets in each area.

- To maintain knowledge of a target's motion trajectory through only short and intermittent periods of tracking, so as to be able to constantly re-acquire a target after a period of not tracking it.

We propose a system architecture consisting of two independent modules that operate in a cyclical loop: *a planning module* that manages the high-level multiplexing issue, and *a control-tracking module* that deals with the low-level frame-to-frame detection and tracking of targets. This is illustrated by the block diagram of Figure 3.1(a).

We model the planning module as a queuing system that schedules the access of multiple contending users (the targets) to a scarce resource (the camera), as shown in Figure 3.1(b). Our scheduling scheme is a function of two key parameters to be defined for each target: (i) the length of each target's time slot, and (ii) the time between any two consecutive time slots allocated to any one target.

The control/tracking module is modelled as a recursive data filter (or stochastic estimator) that estimates the target's motion based on noisy measurements of its position in the image, and controls the orientation of the camera to make sure the target stays within the camera's field of view during tracking.

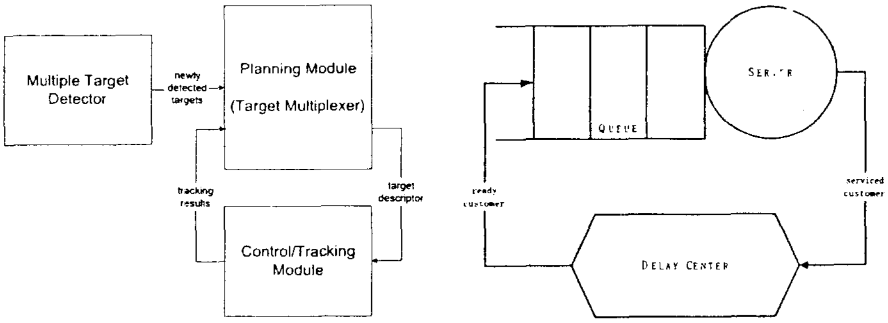


Figure 3.1: (a) Overall system architecture. (b) Queueing model of planning module.

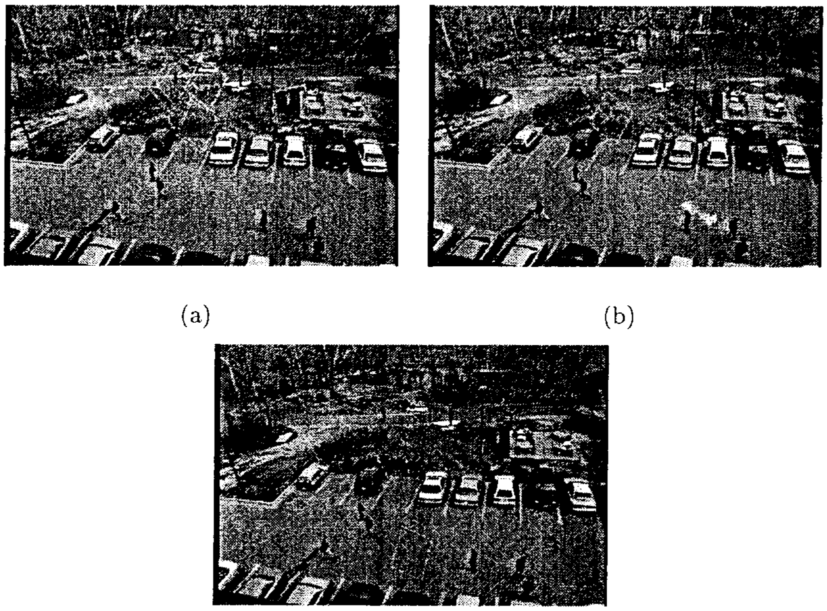
3.2 Simulations

The system was simulated using monocular video sequences taken from a stationary camera[1]. These sequences depict a typical urban outdoor surveillance setting, wherein eleven people are walking randomly, in different directions, at a various paces in a more or less linear fashion. The camera used has a wide-angle field of view (a focal

[1] A controllable pan/tilt camera was initially not available for real-time testing

length of 67mm and image size of 720x480 pixels), and is kept stationary throughout the sequence. The actual target trajectories in the sequence were determined via a simple 2D detection tracking algorithm. In order to simulate the switching of the camera viewpoint, we let the actual field of view be the (virtual) camera FOR, and the image plane be 50x30 pixels (instead of the actual 720x480) corresponding to a virtual narrow field of view. Figure 3.2(a) shows the virtual fields of view (the yellow quadrangles), with their centers connected by the red poly-line, over the first 100 frames of the sequence.

Figure 3.3 below shows a plot of the degree of multiplexing throughout the sequence, and Figures 3.2(b) and 3.2(c) shows the trajectories of two of the tracked targets. The red line represents the actual trajectory and the green line represents the filter-estimated trajectory.



(a)

(b)

(c)

Figure 3.2: (a) Virtual field of view of the camera for the first 100 frames of the sequence. (b) Estimated and measured trajectories of two of the multiplexed targets.

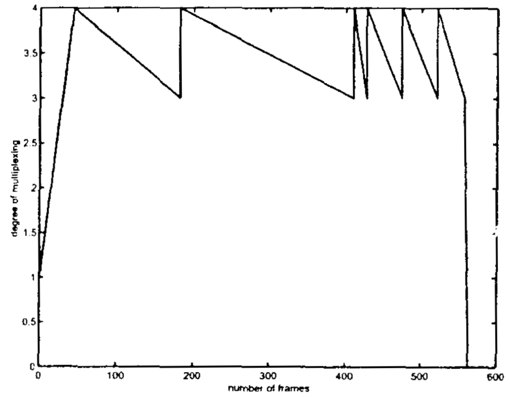


Figure 3.3: Virtual field of view of the camera for the first 100 frames of the sequence.

Chapter 4

Detection of Load-carrying People for Gait and Activity Recognition

4.1 Introduction

The detection of whether a walking person carries an object is of interest in human activity recognition. In many surveillance applications, an important class of human activities are those involving interactions of people with objects in the scene, which include depositing an object, picking up an object, and the exchange of an object between two people. Given the time intervals during which objects are carried by any one person, we would expect that a temporal logical reasoning system will be able to infer events of object pickup, object deposit and object exchange.

Carried object detection is also of interest to gait recognition because carried loads are considered a gait-altering factor (i.e. they alter the the dynamics of walking). Moreover, some gait recognition algorithms are appearance-based, and hence the presence of a large carried object that distorts the silhouette shape of the person is very likely to 'break' such algorithms. Thus, it is essential to determine whether a person is carrying an object before attempting gait recognition.

We limit the scope of the problem by making the following assumptions:

- The camera is stationary. This simplifies the foreground detection procedure, and helps decouple detection problem from the problem at hand.

- The person is walking in upright pose. This is a reasonable assumption for a person to carry an object.

- The person walks with a constant velocity for a few seconds (i.e throughout the analysis for carried object detection).

The method consists of three modules operating in tandem. First, we detect and track the person for some N frames in the video sequence and obtain N binary blobs of the person. Then, we classify the person as *naturally-walking* or *object-carrying*, based on spatiotemporal analysis of certain features of binary silhouette of the blobs. Finally, we segment the object via static shape analysis of a select frame (though work on the latter is in progress and will not be reported here).

The differences between natural walking gait and load-carrying gait may be attributed to any of the following (this list does not claim to be exhaustive):

- The manner by which the person carries the object; e.g. when holding a box with both hands, the arms no longer swing.
- Occlusion of part of the silhouette, such as when a handbag or suitcase held on the side occludes the legs.
- Protrusion of the object outside silhouette, hence distorting its contour shape.
- The sheer weight of an object; a heavy object will most likely cause a person not to swing his arms as much.

We capture these differences between natural gait and load-carrying gait via temporal behavior of correspondence-free binary shape features, consisting of the bounding box widths of horizontal segments of the silhouette, as shown in Figure 4.1. Specifically, we formulate constraints on the periodicity and amplitude of these features, and claim that these constraints are typically violated when the person is carrying an object.

For a naturally-walking person, the width time series of the upper and lower body are periodic with the same period, and the average amplitude of the upper body is less than that of the lower body. This is illustrated in Figure 4.2. The topmost plot contains the width series of the upper and lower body (denoted by U and L respectively), and bottom plot contains their respective autocorrelation functions. The peaks of the latter are used to compute the periodicity.

The presence of a carried object causes the width series along some body region to be aperiodic. Figure 4.3 shows a person carrying a bucket in each arm. He is hardly swinging his arms, perhaps because the buckets must be heavy. This explains why the upper body series is not periodic, while lower body's is. Figure 4.4 illustrates

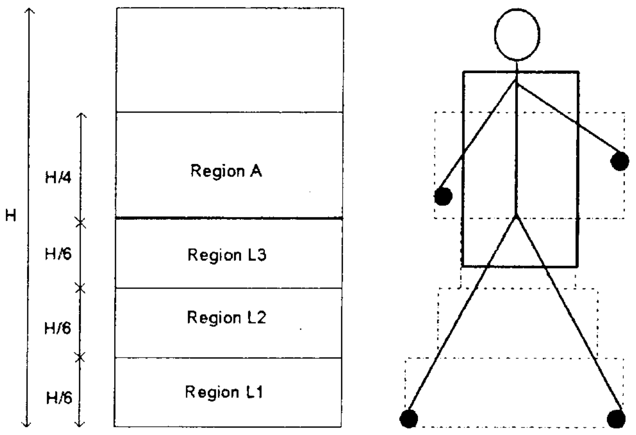
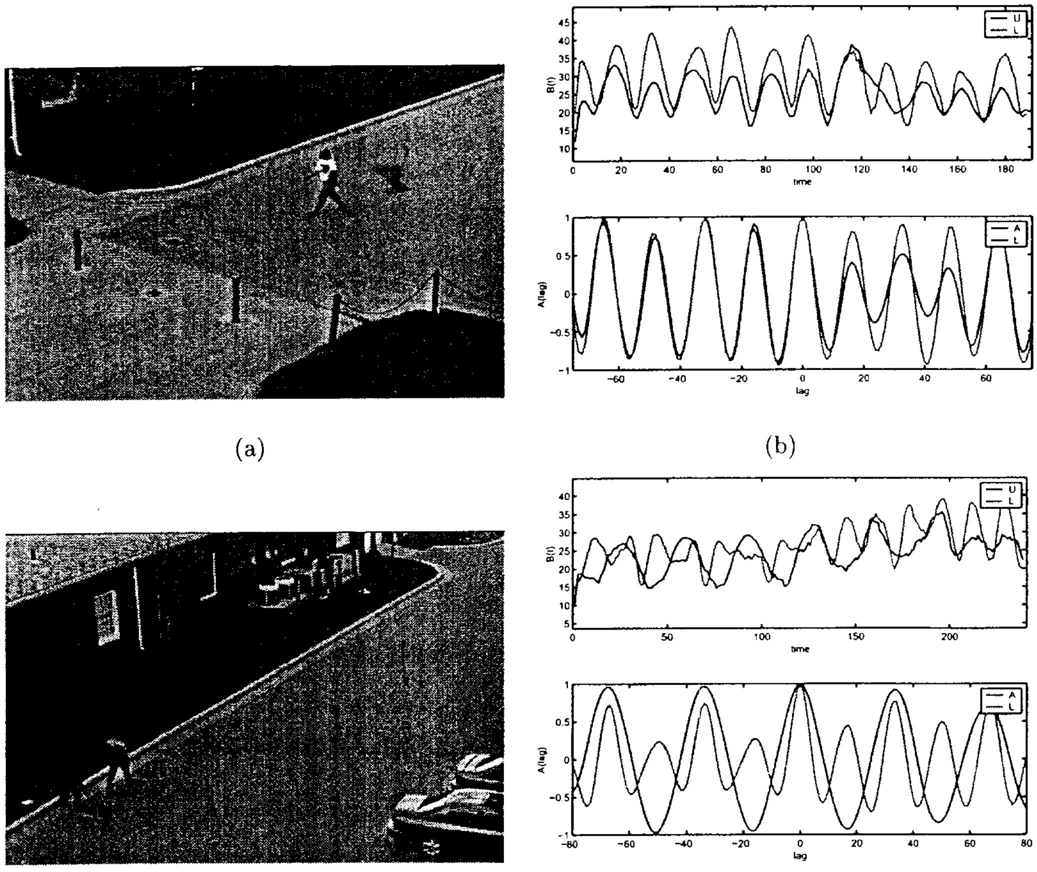


Figure 4.1: Subdivision of body silhouette into 5 segments for shape feature computation.

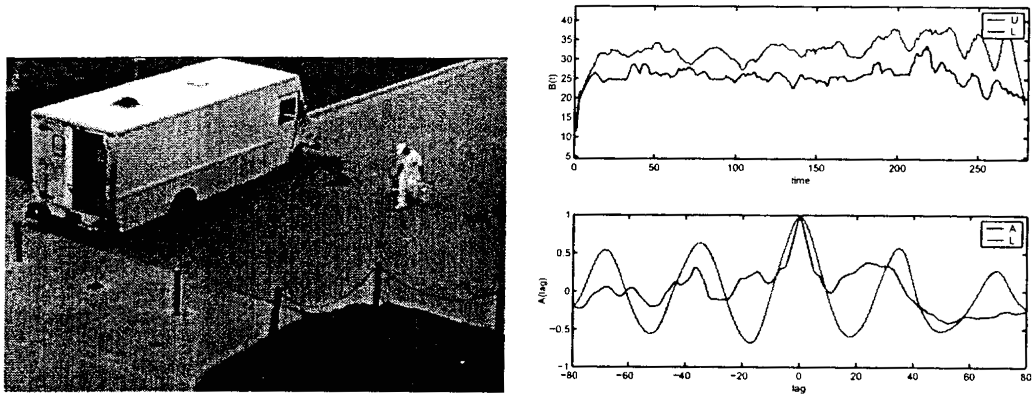
the same case with a different person. Note here that the upper body series seems to oscillate at a higher frequency than the legs, which maybe due to independent oscillation of the carried handbag (particularly if it's lightweight). Figures 4.5 and 4.6 illustrate the case when the lower body's series is periodic while upper body's is not. Both examples involve a person an object with both hands, hence no arm swinging.



(a) (b)

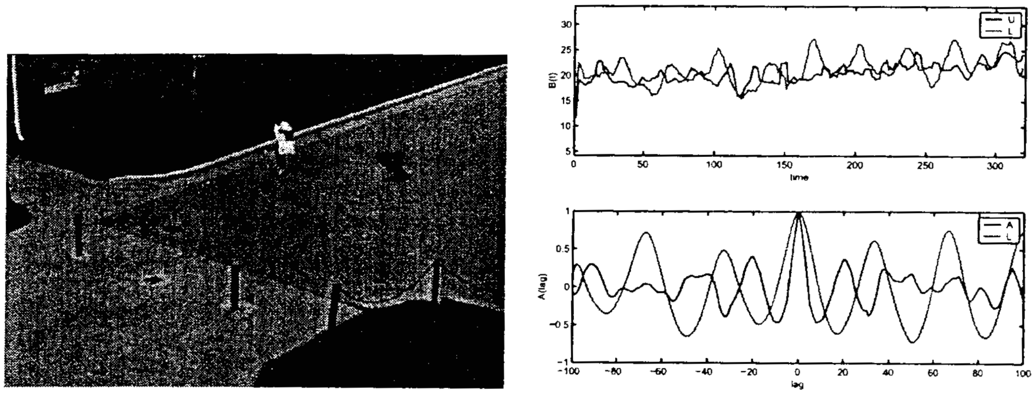
(c) (d)

Figure 4.2: (a,c) A natural-walking person. (b,d) Corresponding width series and autocorrelation functions.



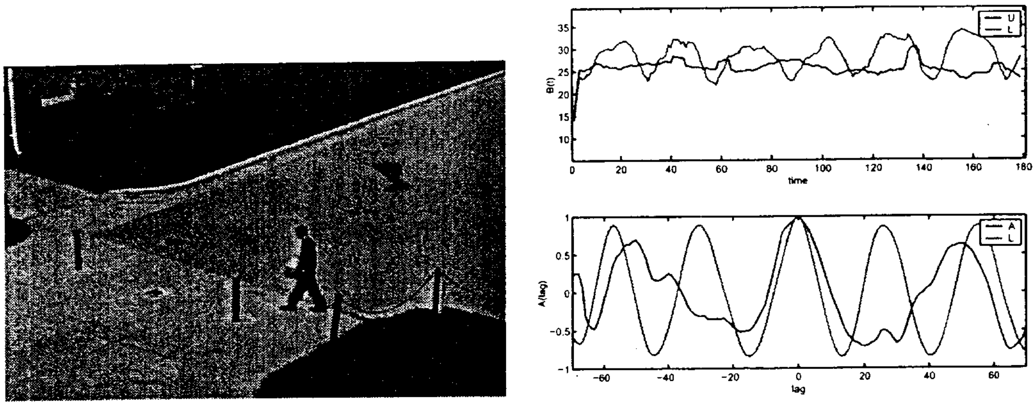
(a) (b)

Figure 4.3: Person carrying two objects on the side. Width series of lower body region is periodic, while that of upper body is not.



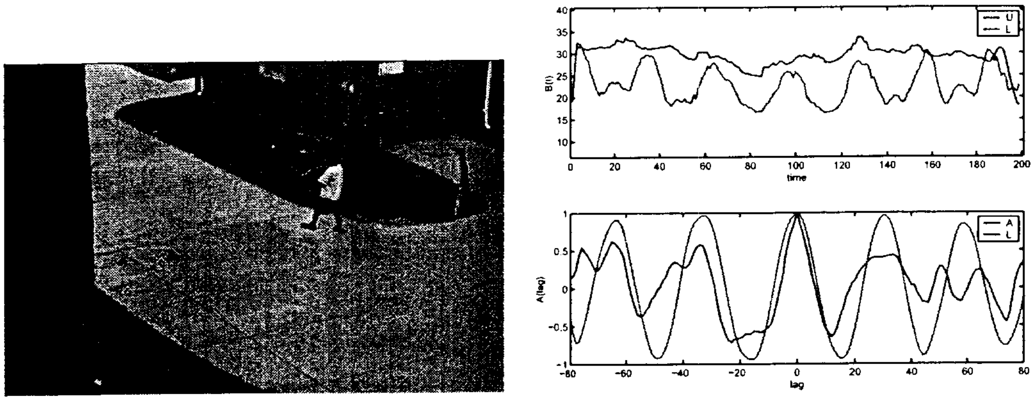
(a) (b)

Figure 4.4: Person carrying a handbag on the side. Width series of lower body region is periodic and of upper body region is aperiodic.



(a) (b)

Figure 4.5: Person carrying a box in front with two hands. Width series of lower body region is periodic and of upper body region is aperiodic.



(a) (b)

Figure 4.6: Person carrying a box in front with two hands. Width series of lower body region is periodic and of upper body region is aperiodic. Furthermore, average width of upper body region is larger than that of lower body region.

Publications

[1] A. Elgammal, D. Harwood, and L. S. Davis, "Nonparametric background model for background subtraction," in *Proc. of 6th European Conference of Computer Vision*, 2000.

[2] C. BenAbdelkader, P. Burlina, and L. Davis, "Single camera multiplexing for multi-target tracking," in *ICIAP*, 1999.

[3] C. BenAbdelkader, P. Burlina, and L. Davis, "Single camera multiplexing for multi-target tracking," in *Multimedia Video-based Surveillance Systems*, pp. 130–142, Kluwer Academic Publishers, 2000.

[4] C. BenAbdelkader, P. Burlina, and L. Davis, "Gait as a biometric for person identification in video sequences," Tech. Rep. 4289, University of Maryland College Park, 2001.