1-1-2005

# Real-time systems for moving objects detection and tracking using pixel difference method.

Bo Shen
*University of Windsor*

# Real-time Systems for Moving Objects Detection and Tracking Using Pixel Difference Method

by

Bo Shen

A Thesis
Submitted to the Faculty of Graduate Studies and Research
through the School of Computer Science
in Partial Fulfillment of the Requirements for
the Degree of Master of Science at the
University of Windsor

Windsor, Ontario, Canada
2005

*Your file Votre référence*
*ISBN: 978-0-494-34963-2*
*Our file Notre référence*
*ISBN: 978-0-494-34963-2*

# Abstract

Tracking of multiple moving objects from 'real world' in an image sequence is a very important task in Computer Vision. In particular, there are many problems in which objects undergoing motion must be detected and tracked. In this thesis, two tracking systems are introduced: Fish-Tracker System and Human-Counter System. In these two systems, the objects are tracked through a sequence of frames and the motion detection algorithm is used based on frame pixel difference. Fish-Tracker system is developed for department of Biological Sciences to determine spent time, path and velocity of moving fish. The direction and velocity of each object are calculated between each pair of frames and are used to predict the position of the object in the next frame. The calibration, reconstruction procedure and the problems due to the presence of noise and shadows in the images are also addressed. Human-Counter system is kind of surveillance system that is important for the office security or the marketing research. In this system, some basic image morphological operations such as opening and closing are employed to reduce noise. The adopted strategies such as a connected components algorithm and boundary extraction are detailed.

# Table of Contents

# List of Figures

# Chapter1

# Introduction

Tracking of multiple moving objects from 'real world' in an image sequence is a very important task in Computer Vision. In particular, there are many situations in which objects undergoing motion need to be detected and tracked. For instance, security and surveillance – to identify anomalous behavior in a parking lot or near an ATM; Medical therapy - to improve the quality of life for disabled people; Retail space instrumentation - to analyze shopping behavior of customers; Traffic management - to detect pedestrians and accidents; Interactive games – to provide interaction with intelligent systems and so on.

## 1.1 Definition

Object detection is the process of locating and segmenting a foreground object from background in image sequences for recognizing object type. Object tracking is used to match the target region during successive image sequences for estimating an object's spatial and temporal changes including its position, shape, motion and etc. These two processes are related because tracking usually starts with detecting objects, while detecting an object repeatedly in subsequent image sequence is often necessary for object tracking.

Detecting moving objects based on motion has very important significance in object detection and tracking. Compared with object detection without motion, motion detection complicates the object detection problem by adding object's temporal change requirements. A large variety of motion detection algorithms have been proposed. Pixel Difference technique is one important motion detection technique used in lots of real-time tracking systems. Pixel Difference technique relies on the detection of temporal changes at pixel level. The difference map is usually binarized using a predefined threshold value to obtain the motion/no-motion classification. Pixel Difference technique is a particularly efficient and sensitive method for detecting gray level changes between images.

## 1.2 Features used in object detection

Foreground objects can be detected from the background in image sequences according to different features. One or more features are extracted and the objects of interest are modeled in terms of these features. Then object detection and recognition can be transformed into a feature-matching problem. The features used in object detection are as follows:

- **Object detection based on shape**

Object detection based on shape is useful when it is difficult to extract reliable features for tracking. It is very complicated because not only we need to detect and determine the border of an object by edge detection and boundary-following algorithms but also need to remove noise by a preprocessing algorithm. A human body can be decomposed into approximate shapes such as an ellipsoid for the head and truncated cones for the limbs. The detection and shape characterization of the objects become more difficult for complex scenes where there are many objects with occlusions and shading. For example, W4 system introduced in ***[HHD98]*** is a real time system for tracking people and their body parts in monochromatic imagery. W4 employs a combination of shape analysis and robust techniques to detect people, and to locate and track their body parts (Figure1.1). This Figure shows that each detected foreground object can be represented by generating global shape and appearance features such as centroid (median coordinate of foreground region) and major axis. In addition, the shape of 2D silhouettes is represented by horizontal and vertical projection histogram.

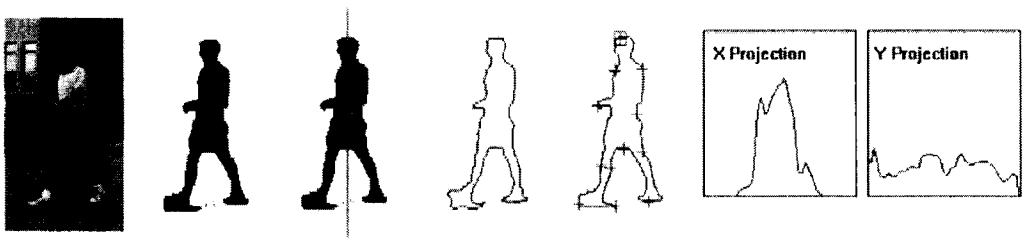



*Figure 1.1. Object detection based on shape in W4 system*

- **Object detection based on color**

Object detection based on color is relatively simple because color is constant and easy to be acquired. However, color is not always appropriate as the sole means of

detecting and tracking objects. For example, it is difficult to distinguish between a field of orange flowers and a tiger, because it lacks information about how the color is distributed spatially. It is important to group color in localized regions and to fuse color with textural properties. Heisele in ***[HKR97]*** develops an algorithm to detect and track vehicles or pedestrians in real-time using color cluster based technique. Each image is divided into given number of clusters by grouping pixels of similar color and position. Pfinder ("Person finder") ***[WAD97]*** is a real-time system for tracking people. The system uses a multi-class statistical model of color and shape to obtain a 2D representation of the head and hands. Each pixel in the background is associated with mean color value and a covariance matrix that describes the color distribution of each pixel. Meanwhile, Pfinder can detect a differently colored region as change in scene. Each blob that corresponds to the person's hands, head, feet, shirt and pants locations respectively has a spatial (x, y) and color (Y, U, V) component, and also have a detailed representation of its shape and appearance.

- **Object detection based on template**

Object detection based on template is a process of matching features between the template and the image sequence under analysis. There are two types of object template matching, fixed and deformable template matching. Fixed templates are useful when object shapes do not change with respect to the viewing angle of the camera. Deformable template matching approaches are more suitable for cases where objects vary due to either the deformation of the object or different object pose relative to the camera. Jain in ***[JZL96]*** propose a general object localization and retrieval scheme based on object shape using deformable templates. Prior knowledge of an object shape is described by a prototype template that consists of the representative contour/edges, and a set of probabilistic deformation transformations on the template. A scheme, which is based on this prior knowledge and the edge information in the input image, is employed to find a match between the deformed template and objects in the image. Their method has been applied to retrieve objects with a variety of shapes from images with complex background. The proposed scheme is invariant to location, rotation, and moderate scale changes of the template.

- **Object detection based on motion**

Object detection based on motion or detecting moving objects has very important significance in object detection and tracking. Compared with object detection without

motion, motion detection complicates the object detection problem by adding object's temporal change requirements. A large variety of motion detection algorithms have been proposed. They can be classified into the following groups.

1). Thresholding technique according to the frame difference
These approaches such as ***[DN90]*** rely on the detection of temporal changes either at pixel or block level. The difference map is usually binarized using a predefined threshold value to obtain the motion/no-motion classification.

2). Global frameworks --dense motion estimates
Many algorithms estimate the optical flow (velocity field in the image plane) that is the projection of the 3D-motion in the image plane. In subsequent motion segmentation, each image is divided into segments corresponding to objects with different motion properties. There are two important requirements for this approach: dense motion estimates must be calculated and motion discontinuities must be preserved. The main problem of this approach is that the optical flow cannot be reliably estimated in image parts with approximately uniform intensity and it is usually very time consuming.

## 1.2 The four elements in Tracking Systems

There are numerous real word applications of tracking system using different algorithms and procedures. However, as stated in ***[W]***, all tracking systems usually have four elements: *target representation, observation representation, hypotheses measurement and hypotheses generating.*

The ***target representation*** — such as size, color, shape, and motion — characterizes the target in a state space. For example, shapes in ***[HHD98]*** and color distributions in ***[HKR97]*** are often employed as target representations. Some methods employ both shape and color like in ***[WAD97].*** Sometimes motion could also be taken into account in target representations, since different objects can be segmented by the differences of their motions.

The ***observation representation*** defines the image features observed in the images. For instance, if the target is represented by its color appearance like in ***[HKR97]*** and ***[WAD97],*** certain color distribution patterns in the images could be used as the

observation of the target. If the target is characterized by its contour shape like in ***[HHD98],*** we should observe edges of the contour in the image.

The ***hypotheses measurement*** matches target state hypotheses with their image observations. In general, the question we often ask is that, given a certain image observation, which hypothesis will be most likely to produce such an image observation. For instance, in ***[JZL96],*** the template-matching tracking method takes SSD as the measurement. SSD is short for sum-of-squared-difference. The fewer the SSD measurement is, the higher the probability of the hypothesis will be.

The ***hypotheses generating*** produces new state hypotheses based on old estimation of target's representation and old observation. Two commonly used hypotheses generating algorithms in tracking are Kalman filtering in ***[K60], [LJH]*** and CONDENSATION algorithm in ***[IB98]***. Kalman filtering is a prediction-correction procedure under Gaussian assumptions, which the density could be characterized by its mean and covariance. By encapsulating the motion of the object into internal states, Kalman filtering aims at finding appropriate states that gives best-fit observations. Dynamic equation and measurement equation will be used in Kalman filter for representing the change in internal states and conversion from internal state to observation respectively. Although Kalman filter' approach is fast, it suffers from a few and yet serious problems. For instance, too much prior knowledge is required, dynamic model should be provided, the uni-modal Gaussian distribution is assumed and clutter background is not allowed. To overcome these problems, CONDENSATION (conditional density propagation) algorithm was developed. It aims at finding most probable area containing the feature or object based on sampling. By allowing more than one hypothesis, CONDENSATION algorithm can recover from false tracking of ambiguous feature or object. The larger the size of the samples, the more accurate the tracking result will be. However, larger sample size also implies higher computational time. As such, there is a tradeoff between time complexity and accuracy.

## 1.3 Thesis Overview

This thesis is organized as follows. Chapter 2 provides the theoretical background of computer vision techniques for object detection and object tracking. Chapter 3 gives

an overview about basic steps of general object tracking systems. Chapter 4 describes our interactive Fish-Tracker System used for biology department, including the implementation details and experiment results. Chapter 5 discusses different aspects of human identification and activity recognition, presents our Human-Counter System that is used for counting the number of people in the room. Chapter 6 is the conclusion and discussion for the future research direction.

# Chapter2

# Background for Vision-based object detection and tracking

Object detection and tracking are very important task in computer vision and many related techniques have been developed. The major goal of object tracking is to compute properties of objects in 3-D world from digital images. A space point in 3-D world and its 2D projection point are linked by the projection matrix that can be obtained from camera calibration process. After Camera calibration and 3D reconstruction, the properties of objects in 3-D world are straightforward. Foreground segmentation is a process used in object detection to partition a digital image into disjoint regions of interest based on some similarly criteria, such as intensity, region or boundary. Noise can have a dramatic effect on the magnitude of difference images and recognize wrong object, despite the fact that no interest object have been in that region. Some basic morphological operations such as: erosion, dilation, opening and closing, are particularly useful for the analysis of binary images and common usages including noise removal, edge detection. To establish the necessary background for object detection and tracking, some related techniques in computer vision mentioned above are discussed here.

## 2.1 Camera Calibration

It is straightforward to get some information about 2D point from image sequences, but our goal in object tracking is to compute properties of objects in 3-D world from these digital images. To that end, we need to know the projection matrix and the camera parameters. Camera calibration is a fundamental problem in computer vision and is a necessary step to extract metric information from 2D images. It is a process of determining the intrinsic and extrinsic parameters of the camera, or the process of estimating the projection matrix. A perspective projection matrix is the link between the 3D space points and the 2D pixel points.

Camera calibration techniques are roughly classified into two categories: pattern-based calibration and self-calibration ***[Z00]***. Pattern-based calibration uses a special calibration object, called calibration pattern. The calibration pattern's geometry in 3D space is known in advance; once a camera takes a digital image of the calibration

pattern, the correspondences between the 3D feature points and their 2D projections can be built. This set of data contains projection information and can be used to derive the intrinsic and extrinsic parameters. Pattern-based camera calibration can be done very efficiently. A well-known example is Tsai's calibration algorithm ***[T87].*** A recently improvement by Zhang's calibration algorithm in ***[Z00]*** makes it possible to use a planar calibration pattern, which considerably improves the flexibility. Self-calibration does not use the calibration pattern. The calibration process is to move the camera rigidly and take images with fixed intrinsic parameters; the correspondences between three images are sufficient to recover both intrinsic and extrinsic parameters. This approach is quite flexible, but it is not mature yet. Because there are many parameters to estimate, we cannot always obtain reliable results. A 3D point $P$ must undergo the following steps to be projected onto the image plane:

*1)* Euclidian transformation from world reference frame to camera reference frame. It can be expressed as a 3*4 matrix $D$, in which 6 extrinsic parameters are held, namely a rotation and a translation.

*2)* A 3D-2D perspective projection, projecting point $P$ onto the image plane within the camera reference frame. It can be expressed as a 3* 4 matrix $I$.

*3)* A 2D-2D transformation from camera reference frame to image coordinates. It can be expressed as a 3*3 matrix $A$, in which 4 intrinsic parameters are held.

Therefore, the projection matrix $M$ can be expressed as a 3 * 4 matrix $M$:

$$M = AID$$

The projection of a space point $P = (X, Y, Z, 1)$ onto the image point $p = (u, v, 1)$ is 3D-2D projection. (Figure2.1)

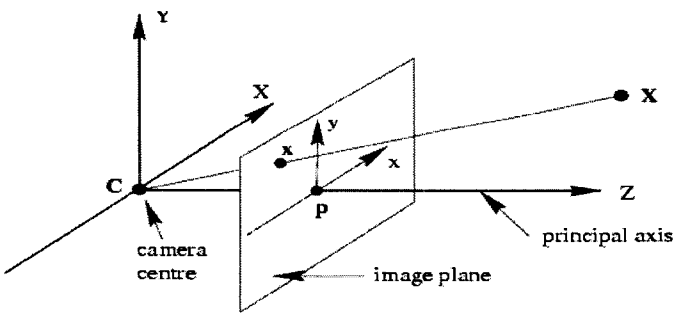



*Figure2.1. 3D-2D correspondence projection*

The projection can be expressed as:

$$\begin{pmatrix} u \\ v \\ 1 \end{pmatrix} = \lambda \begin{pmatrix} m_{11} & m_{12} & m_{13} & m_{14} \\ m_{21} & m_{22} & m_{23} & m_{24} \\ m_{31} & m_{32} & m_{33} & m_{34} \end{pmatrix} \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix}$$

It is a 3D-2D response projection (figure1), where $\lambda$ is the scale factor. Equation provides two basic projection equations describing the relationship between points defined in space (*X, Y, Z*) and their corresponding pixels (*u, v*):

$$u = \frac{m_{11}X + m_{12}Y + m_{13}Z + m_{14}}{m_{31}X + m_{32}Y + m_{33}Z + m_{34}}$$
$$v = \frac{m_{21}X + m_{22}Y + m_{23}Z + m_{24}}{m_{31}X + m_{32}Y + m_{33}Z + m_{34}}$$

where *mij* , $i = 1...3, j = 1...$ 4 are the entries of projection matrix. At least 6 points are needed to solve 12 entries of projection matrix M. A singular value decomposition (SVD) is a good method to solve the equation:

$$\begin{pmatrix} -X_1 & -Y_1 & -Z_1 & -1 & 0 & 0 & 0 & 0 & u_1X_1 & u_1Y_1 & u_1Z_1 & u_1 \\ 0 & 0 & 0 & 0 & -X_1 & -Y_1 & -Z_1 & -1 & v_1X_1 & v_1Y_1 & v_1Z_1 & v_1 \\ -X_2 & -Y_2 & -Z_2 & -1 & 0 & 0 & 0 & 0 & u_2X_2 & u_2Y_2 & u_2Z_2 & u_2 \\ 0 & 0 & 0 & 0 & -X_2 & -Y_2 & -Z_2 & -1 & v_2X_2 & v_2Y_2 & v_2Z_2 & v_2 \\ & & & & & & \vdots & & & & & \\ -X_n & -Y_n & -Z_n & -1 & 0 & 0 & 0 & 0 & u_nX_n & u_nY_n & u_nZ_n & u_n \\ 0 & 0 & 0 & 0 & -X_n & -Y_n & -Z_n & -1 & v_nX_n & v_nY_n & v_nZ_n & v_n \end{pmatrix} \begin{pmatrix} m_{11} \\ m_{12} \\ m_{13} \\ m_{14} \\ m_{21} \\ m_{22} \\ m_{23} \\ m_{24} \\ m_{31} \\ m_{32} \\ m_{33} \\ m_{34} \end{pmatrix} = 0$$

Sometimes, we need to track moving objects along a plane. In such case, points from calibration pattern are mapped to the image points on the image plane (2D-2D projection), , called homography (Figure2.2).

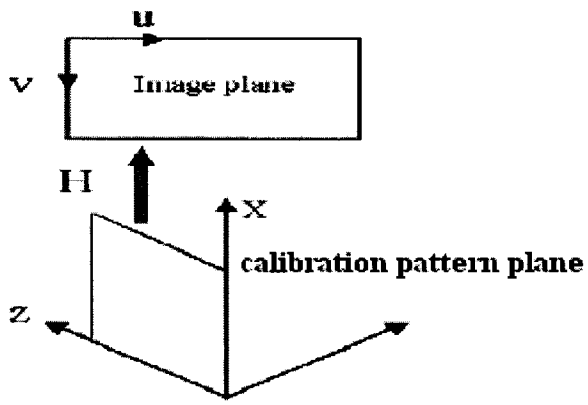



*Figure2.2. 2D-2D correspondence projection*

The projection of a space point $P = (X, Y, 1)$ onto the image point $p = (x, y, 1)$ can be expressed as:

$$\begin{pmatrix} x \\ y \\ 1 \end{pmatrix} = \lambda \begin{pmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{pmatrix} \cdot \begin{pmatrix} X \\ Y \\ 1 \end{pmatrix}$$

or,

$$p = \lambda HP$$

At least four points are needed in order to calculate homography matrix H. The calculation of the matrix H is usually very sensitive to the noise in the image. It usually requires the knowledge of more than 4 points on the motion plane and their projections on the image.

In ***[BM98],*** the calculation of H is reduced to the calculation of only 3 parameters. This minimal parametrization of the homography matrix allows a robust calculation even when the number of point correspondences is small. This is done by a change in the coordinate systems in both the image and the motion plane. Indeed, consider the image points $p_0$, $p_1$, $p_2$ and $p_3$, and their corresponding points in the scene $p_0$', $p_1$', $p_2$' and $p_3$'. The points $p_1$', $p_2$' and $p_3$' must belong to the motion plane, while the point p0'must not be on this plane. These 4 points must form a projective basis, i.e., no three of them are collinear. The image and motion plane are transformed so that:

$$\begin{array}{ll} \mathbf{p_1} = (0,0,1)^T & \mathbf{p'_1} = (0,0,1)^T \\ \mathbf{p_2} = (1,0,0)^T & \mathbf{p'_2} = (1,0,0)^T \\ \mathbf{p_3} = (0,1,0)^T & \mathbf{p'_3} = (0,1,0)^T \\ \mathbf{p_0} = (1,1,1)^T & \mathbf{p'_0} = (1,1,1)^T \end{array}$$

Under such choice of coordinate systems, the homography matrix **H**, such that $\mathbf{p'_i} \simeq \mathbf{Hp_i}(i = 1,2,3)$ is diagonal, that is $\mathbf{H} = diag(\alpha, \beta, \gamma)$. Because of the equality up to scale factor **H** now depends on only 2 parameters. The original matrix is recovered by applying the inverse transformations on the left and right hand sides of the diagonal matrix.

## 2.2 Three-dimensional Reconstruction

After having point correspondences identified, the 3D reconstruction from calibrated images is straightforward. A space point and its 2D projection point is linked by the projection matrix which has been already attained from calibration stage:

$$\begin{pmatrix} u \\ v \\ 1 \end{pmatrix} = \lambda \begin{pmatrix} m_{11} & m_{12} & m_{13} & m_{14} \\ m_{21} & m_{22} & m_{23} & m_{24} \\ m_{31} & m_{32} & m_{33} & m_{34} \end{pmatrix} \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix}$$

If we know the projection matrice and the projections of a space point in both the left and the right image, we can easily recover that space point's 3D coordinates (*X; Y; Z*) by solving the following 4 equations:

$$\begin{aligned} u &= \frac{m_{11}X + m_{12}Y + m_{13}Z + m_{14}}{m_{31}X + m_{32}Y + m_{33}Z + m_{43}} \\ v &= \frac{m_{21}X + m_{22}Y + m_{23}Z + m_{24}}{m_{31}X + m_{32}Y + m_{33}Z + m_{43}} \\ u' &= \frac{m'_{11}X + m'_{12}Y + m'_{13}Z + m'_{14}}{m'_{31}X + m'_{32}Y + m'_{33}Z + m'_{43}} \\ v' &= \frac{m'_{21}X + m'_{22}Y + m'_{23}Z + m'_{24}}{m'_{31}X + m'_{32}Y + m'_{33}Z + m'_{43}} \end{aligned}$$

We have 4 equations and 3 unknowns. The above equation can be rewritten as equation:

$$\begin{pmatrix} m_{11}-m_{31}u & m_{12}-m_{32}u & m_{13}-m_{33}u \\ m_{21}-m_{31}v & m_{22}-m_{32}v & m_{23}-m_{33}v \\ m'_{11}-m'_{31}u' & m'_{12}-m'_{32}u' & m'_{13}-m'_{33}u' \\ m'_{21}-m'_{31}v' & m'_{22}-m'_{32}v' & m'_{23}-m'_{33}v' \end{pmatrix} \begin{pmatrix} X \\ Y \\ Z \end{pmatrix} = \begin{pmatrix} m_{34}u-m_{14} \\ m_{34}v-m_{24} \\ m'_{34}u'-m'_{14} \\ m'_{34}v'-m'_{24} \end{pmatrix}$$

Equation can easily be solved by SVD method. In summary, the coordinates of a space points can be recovered by knowing its projections on at least two calibrated images.

In case of tracking a moving object along a plane, points from a calibration pattern are mapped to the image points on the image plane (2D-2D projection). A space point $P = (X, Y, 1)$ and its 2D projection point $p = (x, y, 1)$ is linked by the homography matrix:

$$\begin{pmatrix} x \\ y \\ 1 \end{pmatrix} = \lambda \begin{pmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{pmatrix} \cdot \begin{pmatrix} X \\ Y \\ 1 \end{pmatrix}$$

or,

$$p = \lambda H P$$

If we know the homography matrix and the projections of a space point in the image plane (x, y) we can easily recover that space point's coordinates $(X; Y)$.

## 2.3 Foreground segmentation methods used in object detection

Object detection first need to locate and segment a foreground object from background in a sequence of images. Image segmentation is the grouping of image pixels based on some similarly criteria, such as intensity, region or boundary. The goal of image segmentation is to partition a digital image into disjoint regions of interest. Depending on the technique, the criteria may be different. Some methods are based on pixel statistics such as intensity. Other methods try to enclose regions within a boundary. Healy in ***[HC85]*** classified segmentation techniques into global, region homogeneity and boundary finding. The global techniques are those that assign a pixel's membership based on information from the entire image. Region homogeneity groups pixels based on some similarity of criteria around pixels. Finally, boundary-finding techniques locate the region boundaries of the image. This thesis uses the

same classification, but employs more specific techniques: binarization and clustering instead of global techniques.

- **Binarization-based approach**

Binarization-based approach, same as threshold technique, is based on finding optimal threshholding to separate the foreground objects of an image from the background. Threshold techniques, which make decisions based on local pixel information, are effective when the intensity levels of the objects fall squarely outside the range of levels in the background. Frame difference and Background segmentation method can be categorized as Binarization-based approach.

- **Clustering-based approach**

Clustering-based approach is an image restoration method that classifies pixels into clusters based on some attributes. For example, in ***[HKR97],*** the algorithm estimates image motion by tracking clusters determined in the color/position feature space. Each image is divided into a given number of clusters by grouping pixels of similar color and position. Adjacent clusters with similar motion are combined to build interest objects.

- **Boundary-based approach**

Boundary-based approach depends on the information provided by the object boundaries. The boundary-based approach usually uses active contour models. The active contour model is defined as an energy-minimizing spine (also called "snake")- the snake's energy depends on its shape and location within the image. Snake is a deformable active contour used for boundary tracking which was originally introduced by ***[KWT98].*** It is basically a complex edge finding algorithm, which traces edges of objects and finds humans by their unique shape (a spherical head on a body) in human tracking. The main idea is to start with some initial boundary shape represented in the form of spline curves, and iteratively modifies it by applying various shrink/expansion operations according to some energy functions. Local minima of this energy then correspond to desired image properties. Snakes are the optimization way of finding structures. For example, suppose we are interested in the outlines of the clock faces. We might start by looking to see whether image edges will help. Maybe there *is* a simple contour round the clock face, but sometimes the contour of the clock face is broken up. In addition, bits and pieces of other structure inevitably

show up in the edge map. Clearly, using an edge detector alone, however good it is, will not separate the clock faces from other structure in the image. We also can find the clock faces by using extremely exact 2-D *models* of their images - ellipses with known dimensions and orientations. But it is unreasonable to expect that such detailed information will normally be known in advance. We might want to weaken the conditions further, and look for a shape in the image that is *smooth* and forms a *closed contour*, but which is not necessarily elliptical. Active contour models, or snakes, allow us to set up such general conditions, and find image structures that satisfy the conditions. This can be illustrated in Figure2.3, suppose we know that there is a clock face in the rectangular region of the image. We can set up a snake to start on this rectangle. Then, we can tell the snake to shrink, to try to form a smooth contour, and to avoid going onto brighter parts of the image. The final position of the snake has converged on the contour of the outside of the clock face, distorted a little by the bright flint at 1 o'clock.

*Figure 2.3. Snakes for clock face*

The snake structure has a very simple form. They consist of a set of *control points*, effectively connected by straight lines. Each control point has a position, given by (X, Y) coordinates in the image, and a snake is entirely specified by the number and coordinates of its control points. Moving the control points makes adjustments to the snakes. Snakes use an energy minimizing spline that is deformed by constraint forces. M. Kass, A. Witkin, and D. Terzopoulos in ***[KWT98]*** describe three forces:

- Internal forces modeling the bending of the contour
- Image forces representing information such as edges
- Constraint forces to provide any additional external forces

In segmentation and boundary tracking problems, these forces relate to the gradient of image intensity and the positions of image features. One advantage of the force-driven snake model is that it can easily incorporate the dynamics derived from time-varying

images. The snakes are usually parameterized and the solution space is constrained to have a predefined shape. So these methods require an accurate initialization step associated with the solution of a partial differential equation of motion. Active contour had been used in pattern location and tracking for a long time. ***[CB92], [CC96], [KWT88], [WW], [WAD97]*** and ***[MT93]*** all use active contour. It is good at attaching to an object with strong edges and irregular shapes. In ***[WW]***, the initial position of the active contour is usually the bounding box of the searching window. It searches for strong edges along the direction towards to centroid of potential region. It stops at the pixel with strong edge characteristic and close to the skin-color region. By searching along the line joining the initial position and the centroid, the pixel with strong edge information is picked as potential contour. In ***[WAD97]***, Pfinder uses a 2D contour shape analysis that attempts to identify the head, hands, and feet location. ***[CB92]*** proposes a B-spline representation of active contours, and ***[MT93]*** proposes a deformable quadric model for modeling of shape and motion of 3D non-rigid objects. They consider the case of quadric ellipsoids with tapering and bending deformations.

***Geodesic active contour models*** are not parameterized and can be used to track objects that undergo non-rigid motion. In ***[CC96]***, a three steps approach is proposed which starts by detecting the contours of the objects to be tracked. An estimation of the velocity vector field along the detected contours is then performed. At this step, very unstable measurements can be obtained. Following this, a partial differential equation is designed to move the contours to the boundary of the moving objects. These contours are then used as initial estimates of the contours in the next image and the process iterates.

- **Region-based approach**

Region-based approach groups pixels into regions based on some criteria of homogeneity among neighboring pixels and relies on information provided by the entire region such as texture and motion-based properties. A region-based method usually proceeds as follows: the image is partitioned into connected regions by grouping neighboring pixels of similar intensity levels. Adjacent regions are then merged under some criteria involving perhaps homogeneity or sharpness of region boundaries. With Region-based approach, the estimation of the target's velocity is based on the correspondence between the associated target regions at different time instants. This operation is usually time-consuming because a point-to-point

correspondence is required within the whole region. But region-based approach increases robustness due to the fact that the whole region provides information. Computing Optical flow like in [JB93] is one of the widely used methods in this category. With this method, the apparent velocity and direction of every pixel in the frame have to be computed. Although it is a little bit time consuming, this method is very useful in detecting and tracking objects in video with moving background or shot by a moving camera.

## 2.4 Image morphological operations

Sometimes when tracking system locates and detects foreground regions from background in image sequences, we cannot obtain clear foreground regions. Some small regions like noise spikes and ragged edges due to illumination changes should be eliminated first. We usually apply some basic image morphological operations to binary image to remove such noise. The morphological operations are particularly useful for the analysis of binary images and common usages including edge detection, noise removal, image enhancement and image segmentation.

### 2.4.1 Erosion and Dilation

The two most basic operations in mathematical morphology are erosion and dilation. Both of these operators take two pieces of data as input: an image to be eroded or dilated, and a structuring element. Erosion, Dilation and their combined usage are ways of adding or removing pixels from the boundaries of features in order to smooth them, to join separated portions of features or separate touching features, and to remove isolated pixel noise from the image. Dilation turns pixels "on" according to rules based on the number or arrangement of neighboring pixels, while Erosion turns pixels "off" according to similar rules. Dilation increases the size of objects, while Erosion reduces the size of objects. Morphological operations are usually performed on binary images where the pixel values are either 0 or 1. For simplicity, we will refer to pixels as 0 or 1, and will display a value of zero as black and a value of 1 as white.

#### a) Erosion

Suppose: object has white (1) pixels, background is made up of black (0) pixels
Then the erosion mask is:

| | | |
|---|---|---|
| 1 | 1 | 1 |
| 1 | 1 | 1 |
| 1 | 1 | 1 |

The logical operator for erosion is AND. The mask has the effect of removing a single pixel from the boundary of objects. This means that every pixel in the neighborhood must be 1 for the output pixel to be 1. Otherwise, the pixel will become 0. No matter what value the neighboring pixels have, if the central pixel is 0 the output pixel is 0. Just a single 0 pixel anywhere within the neighborhood will cause the output pixel to become 0. Erosion can be used to eliminate unwanted white noise pixels from an otherwise black area. The only condition in which a white pixel will remain white in the output image is if all of its neighbors are white. The effect on a binary image is to diminish, or erode, the edges of a white area of pixels (Figure2.4 cited fromn ***[GWJ]***).

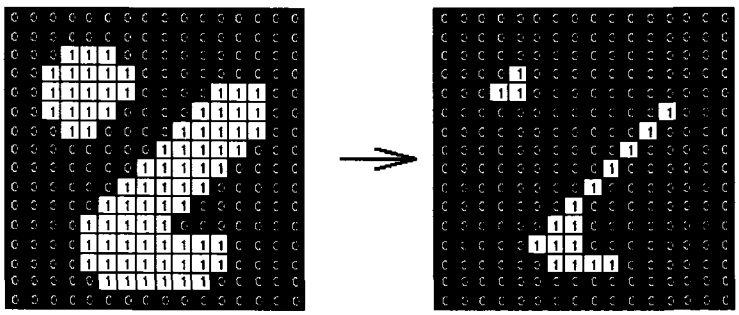


*Figure 2.4. Effect of Erosion using a 3×3 square structuring element*

## b) Dilation

Dilation is the inverse of erosion. The generalised mask is:

| | | |
|---|---|---|
| 1 | 1 | 1 |
| 1 | 1 | 1 |
| 1 | 1 | 1 |

The logical operator for dilation is OR. Dilation has the effect of adding a single pixel to the boundary of the object. This mask will make white areas grow, or dilate. Being

the opposite of erosion, dilation will allow a black pixel to remain black only if all of its neighbors are black. This operator is useful for removing isolated black pixels from an image (Figure2.5 cited from *[GW]*).

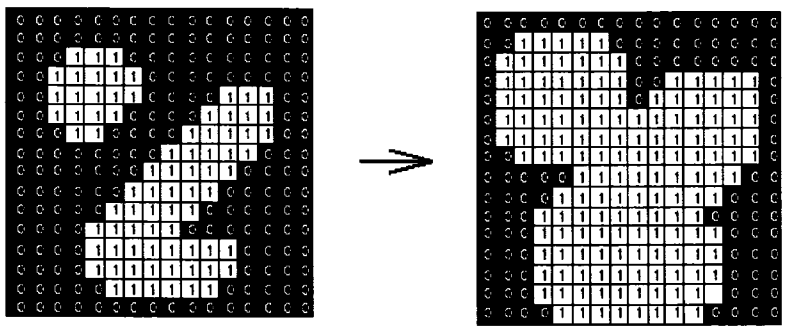


*Figure 2.5. Effect of Dilation using a 3×3 square structuring element*

## 2.4.2 Opening and Closing

Opening and closing are two important operators from mathematical morphology. They are both derived from the fundamental operations of erosion and dilation. Opening – (erosion followed by dilation) - and closing –(the reverse sequence) - attempt to restore the original area of features but with some rearrangement of the boundary pixels. Dilation increases the size of objects. Opening (Erosion-dilation) removes single pixel anomalies but maintains the original shapes and sizes of objects. We can see the result of erosion, dilation and opening to remove noise as following (Figure2.6):

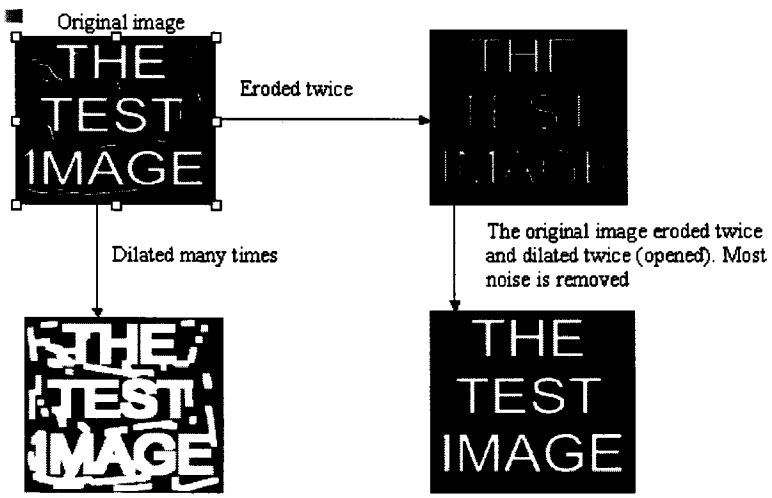



*Figure 2.6. Erosion, Dilation and Opening*

## a) Opening

The basic effect of an opening is somewhat like erosion in that it tends to remove some ragged edges of regions of foreground pixels. Although erosion can be used to eliminate small noise spikes quite effectively, it has the big disadvantage that it will affect *all* regions of foreground pixels. Opening gets around this by performing both erosion and a dilation on the image. Opening can eliminate small noise spikes without changing its orientation and size. All pixels covered by the structuring element within the foreground region will be preserved. However, all foreground pixels that cannot be reached by the structuring element will be eroded away. The effect of an opening on a binary image using a 3×3 square structuring element is illustrated in Figure 2.7 (cited from [GW]).

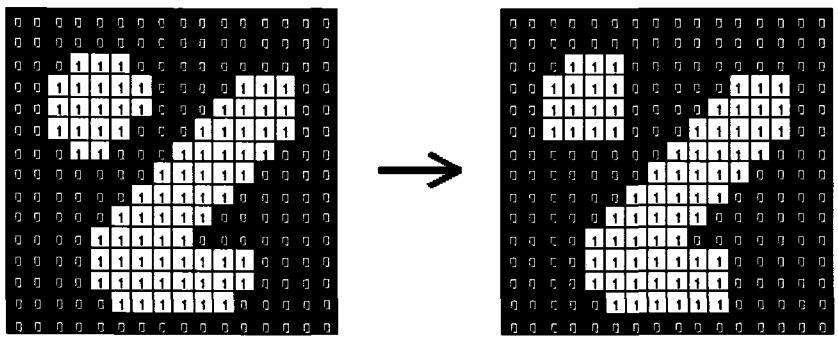


*Figure 2.7 Effect of opening using a 3×3 square structuring element*

## b) Closing

Closing is similar in some ways to dilation. Closing is opening performed reversibly. It is defined simply as a dilation followed by an erosion using the same structuring element for both operations. It tends to enlarge the boundaries of foreground regions in an image (and shrink background color holes in such regions), but it is less destructive of the original boundary shape. Although dilation can fill in small background holes in images, it will also distort *all* regions. Closing can have same effect without changing its orientation. For any background boundary point, if the structuring element can be made to touch that point, without any part of the element being inside a foreground region, then that point remains background. The effect of a closing on a binary image using a 3×3 square structuring element is illustrated in Figure 2.8(cited from [GW]).

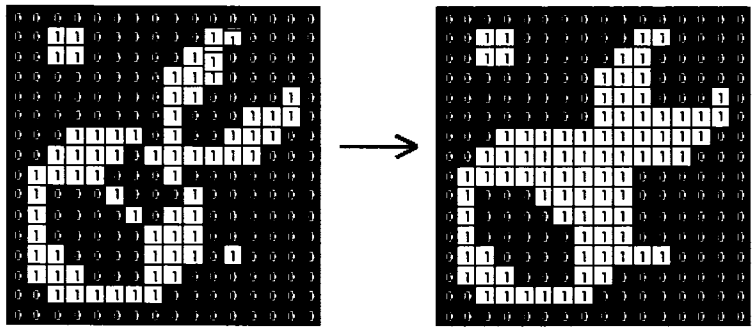


*Figure 2.8 Effect of Closing using a 3×3 square structuring element*

## 2.5 Boundary Extraction

When dealing with binary images, the principal application of morphology is extracting image components that are useful in the description of shape. In particular, we consider morphological algorithms for extracting boundaries. Here we give detail introduction about boundary extraction. The images are binary, with 1's shown shaded and 0's shown in white. The boundary of a set A is obtained by first eroding A by structuring element B and then taking the set difference of A and it's erosion. The resulting image after subtracting the eroded image from the original image has the boundary of the objects extracted. The thickness of the boundary depends on the size of the structuring element B. The boundary of a set A denoted by (where B is a suitable structuring element):

$$\beta(A) = A - (A \Theta B)$$

The effect of boundary on image A using structure B is illustrated in Figure 2.9 cited from ***[GW]***.

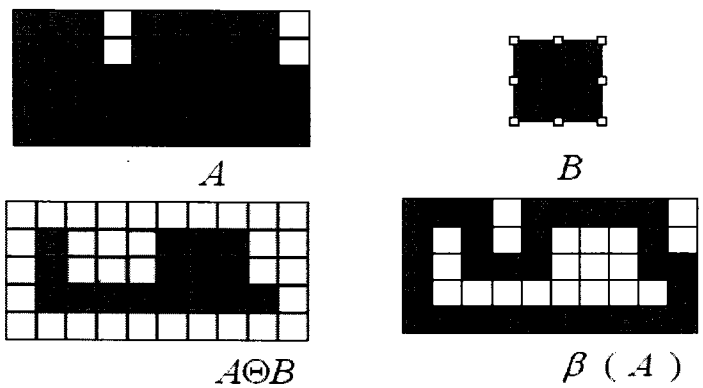



*Figure 2.9 Effect of boundary on image A using structure B*

## 2.6 Conclusion

In this chapter, we have established the necessary background for object detection and tracking. Some basic methods and techniques in computer vision, including Camera Calibration, Three-dimensional Reconstruction, Foreground segmentation, Boundary Extraction and some Image morphological operations used to eliminate image noise, were proposed and discussed.

# Chapter 3

# Tracking System Approach

The Object Tracking system recognizes and tracks interesting objects from image sequences. In general, object-tracking system usually consists of four steps (Figure3.1): pixel difference step, object detecting, object tracking and post processing. In pixel difference step, systems extract the foreground images including object images, from the static background image. Next, in object detection, they extract object blobs (which stand for interest objects, such as human, car) from the extracted foreground images based on the physical position of each blob. Pixels whose velocity and position are close to each other are grouped as a blob. Using several blob features can identify each extracted blob. Object tracking stage compacts groups of pixels, identifies the blobs, and tracks the identified blobs over time. Finally, in post processing, they output some results of the object blobs according to different requirement such as people number, object trajectory and so on. The use of

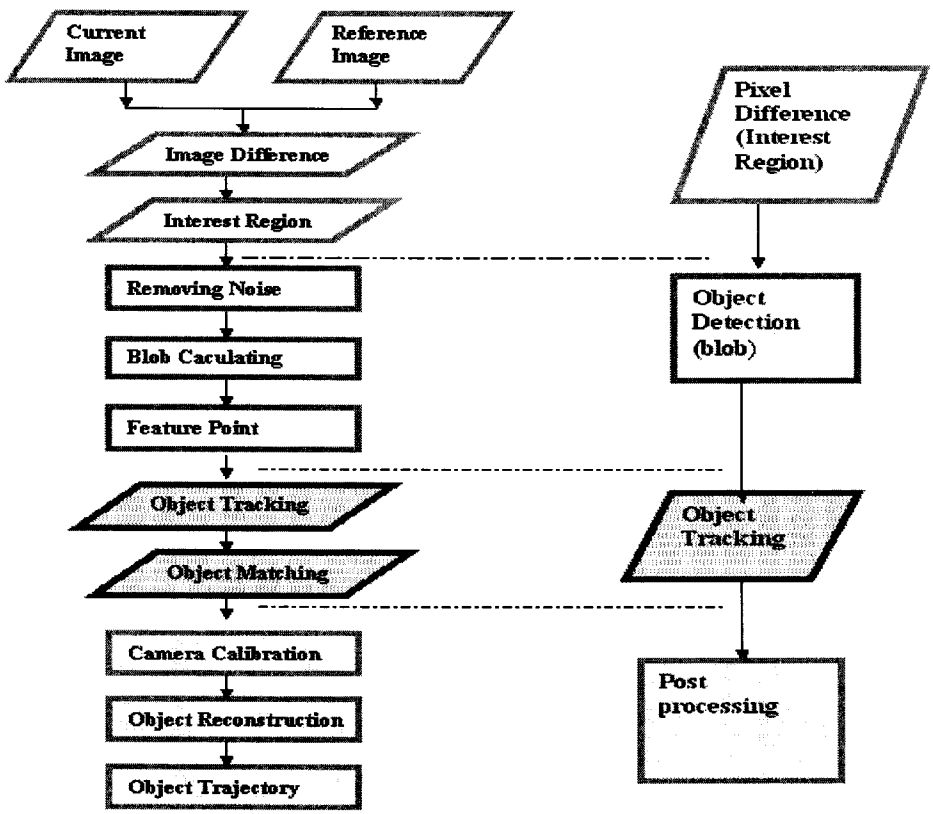



*Figure3.1. Tracking system approach*

"blobs" as a representation for image features have a long history in computer vision. The term has had many different mathematical definitions. As we use it, a blob is a compact set of pixels that share some visual properties such as intensity that are not shared by the surrounding pixels. Now we will introduce all these four steps in detail:

## 3.1 Pixel Difference Method

In many video surveillance applications, cameras are fixed and we are interested in tracking the motion of the foreground object, which could be people or cars. Pixel Difference Method detects possible moving regions. The detection will depend on motion of the target and image subtraction. Pixel Difference Method is a particularly efficient and sensitive method for detecting gray level changes between images sequences. It is widely used in motion detection, where a fixed camera is used to observe dynamic events in a scene.

**The pixel difference method usually works as follows:**

1) Select a threshold value m

The light radiation and frame grabber leads to the formation of noise in the input images, which destroys the efficiency of the tracking system. In order to efficiently detect within the viewing area of the camera; the system requires a threshold value. This threshold value can be calculated by computing the average of apparent differences between two consecutive still images.

2). Scan every pixel in the two images and compute the pixel difference between the pixel of the previous image and the same location pixel of the current image. This inter-frame difference image indicates where some movement has occurred.

3). If the difference is greater than the threshold value m, make it 0 (black). If the difference is less than m, make it 255 (white). So all pixels of the difference image that have value 0(black) include motion and *probably* belong to an moving object, whereas the pixels with value 255(white) that are not part of the moving objects can be ignored. In this way, moving region will be separated from background.

Two type of pixel difference method will be discussed as following:

### 3.1.1 Frame Difference

Many video segmentation algorithms use **Frame Difference** as their primary segmentation criterion. The position and shape of the moving object is detected from the frame difference of two consecutive frames. In our Fish-Tracker System, we used Frame Difference method. Rosin in ***[RE95]*** has pointed out the disadvantages of change detection : 1) Only the motion "wavefront" will produce any change, so that only part of the moving object is highlighted. 2) Objects that become stationary for short periods of time will "disappear". The alternative is **Background Subtraction,** which is the technique that depends on the difference of the current frame with the scene background. But extracting the background image from sequences of frames is needed.

### 3.1.2 Background Subtraction

**Background Subtraction** is the technique that depends on the difference between the current frames with the scene background. But extracting the background image from sequences of frames is needed. Obviously, frame difference only gives us a rough idea of which regions may contain moving objects, but it will not output good tracking results. Background segmentation is a technique that the foreground region can be separated from background by maintaining a background model, and classifying each pixel into either foreground or background. The assumption here is that the camera is more or less fixed such that we can maintain or train a background model. In our Human-Counter System, we used **Background Segmentation** method.

P. L. Rosin and T. Ellis in ***[RE95]*** state that **Background segmentation** Method may be sub-divided into three parts: **first**, the generation of a suitable reference or background; **secondly**, the arithmetic subtraction operation; and **thirdly**, the selection of a suitable threshold. Background images can be generated by a variety of methods. For each image pixel location, we compare the input image pixel and its background model and determine its class.

Some background models are as follows:

***1. Mean:*** a simple approach in ***[RE95] [LJH][BK]*** just uses a median filter (average pixel value of whole image) at each pixel to generate background. Y.H. Yang and M.D. Levine in ***[YL92]*** have suggested the least median of squares (LMedS)

estimate. For each pixel, a mean pixel is used to represent the background, i.e., the background at some specific time *is* $B(x; y)$. To check if an input pixel $I(x; y)$ is a foreground pixel or not, we just check

$(I(x; y) - B(x; y)) > T$ or not. Where T is a threshold value.

***2. Mean & Covariance:*** For each pixel, a Gaussian can be used to model the distribution of such pixel, i.e., the model for a background pixel is represented by a mean pixel $B(x; y)$ and its covariance C $(x; y)$. To check an input pixel $I(x; y)$, the Mahalanobis distance is used, i.e., a foreground pixel satisfies:

$[I(x; y) - B(x; y)]\, C(x; y)[I(x; y) - B(x; y)] > T$. *Where T is a threshold value.*

R´omer Rosales and Stan Sclaroff in ***[RS98]*** use Mean & Covariance as background model.

***3. Temporal Deviation:*** Since the computation of the mixture of Gaussian model is a bit intensive, the temporal deviation model has a simple representation. For each pixel, we maintain its minimum $M(x; y)$, its maximum $N(x; y)$, and the largest interframe absolute difference $D(x; y)$. A foreground input pixel $I(x; y)$ will satisfy

$(M(x; y) - I(x; y)) > D(x; y)$ or $(N(x; y) - I(x; y)) > D(x; y)$.

In ***[HHD98]***, W4 system uses Temporal Deviation as background model.

## 3.2. Removing noise and Foreground Blob computing

**Pixel Difference** provides an estimate of the similarity with the background at every pixel. Then, if a region of the image is dissimilar enough to the estimated empty region, the system marks it as an interesting region.

**Pixel Difference** alone, however, is not sufficient to obtain clear foreground regions; it results in significant level of noise due to illumination changes that should be eliminated first. In ***[HHD98]***, W4 uses region-based noise cleaning to eliminate noise regions. After background subtraction, one iteration of erosion is applied to foreground pixels to eliminate one-pixel thick noise. Then, a fast binary connected-component operator is applied to find the foreground regions, and small regions are eliminated. In ***[RS98]***, a binary map of the current frame is computed with the interest regions in it. Then basic image morphological operations are applied (erosion-dilation) in order to rule out small regions and to fill holes in probable regions of

interest. In ***[LJH]***, after binarizing the image derived from background subtraction, a filter is applied to the image to get rid of the noise. Then, a fast connected-component operator is applied to the image to locate the foreground regions. The operator assigns a clustering label to each pixel according to the labels of its upper-left and left neighbors. After this self -clustering procedure, small regions are eliminated, and big regions are considered as interesting objects.

A blob can be described as a set of connected pixels that share a common attribute. In object tracking, pixels whose velocity and position are close to each other are grouped as a blob. We identify each extracted blob by using several blob features. There are several techniques to cluster and extract blobs, such as "meandering process"***[R97]***, "Fill procedure"***[SPT]*** and "Neighborhood structure-Region growing algorithm" ***[BR02].***

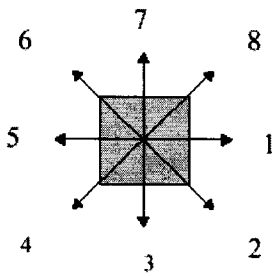



*Figure 3.2. Meandering algorithm*

In ***[R97]*** (Figure 3.2), meandering algorithm works as follows. The blob analysis algorithm begins by scanning across the image pixels from left to right, top to bottom. When a black pixel is located, it will be considered a new possible object and the algorithm will meander through all continuously connected black pixels. We will look in the eight directions to see if there is a black pixel that is connected to this one. When all the black pixels within a region have been found, the minimum bounding-box (blob) that can cover the region will be calculated. This process can extract the foreground blobs, including human blobs.

In **[BR02]**, the method is a sort of region growing algorithm. The objective of this method is to give a measure of how much a pixel belongs to a structural windowed region around it. The first step is to define the basic structure (Fig.3.3.a). And then perform a logical "AND" between the pixels pointed by the circle and each one of its three neighborhoods in each direction (i.e., horizontal, vertical, Fig.3.3.b). For

example, the kernel of Fig.7.b is applied to the 9 different sub-windows of Fig.7.c, and the value of the central pixel (the black pixel in Fig.3.3.c) is increased by the outcome of all these "AND" operations. Finally, this allows extracting the chain code representing borders and use contour lines instead of bounding boxes to highlight blob.

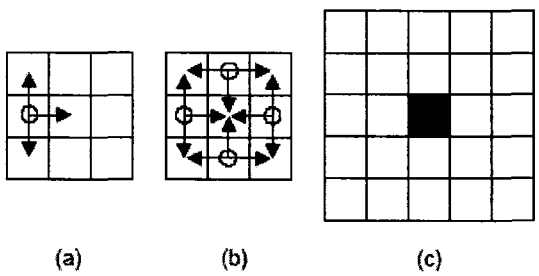



*Figure 3.3. Neighborhood structure-Region growing algorithm*

## 3.3 Object Type Classification

After segmenting a foreground object from background in image sequences, we need to classify objects into different categories such as human, vehicles using shape, color analysis. Recognition techniques based on matching are usually adopted. An unknown pattern is assigned to the class to which it is closest in terms of a predefined metric. As stated in ***[GW]***, the simplest approach is the minimum-distance classifier, which computes the (Euclidean) distance between the unknown and each of the prototype vectors. It chooses the smallest distance to make a decision. Another approach based on correlation can be formulated directly in term of images and is quite intuitive. In addition, probability considerations become important in pattern recognition because of the randomness under which pattern classes normally are generated. So an Optimum Statistical Classifiers approach can also be derived. It is optimal in the sense that, on average, its use yields the lowest probability of committing classification errors. Finally, neural networks involve architectures that consist of layers of perception computing elements and focus on decision functions of multi-class pattern recognition, independent of whether or not the classes are separable.

Collins *in* ***[CL00]*** classifies moving object blobs into general classes such as "humans" and "vehicles" using viewpoint-specific neural networks, trained for each camera. Each neural network is a standard three-layer network (Figure 3.4). They use view dependent visual properties to train a neural network classifier to recognize three

classes: single human; human group and vehicles. Learning in the network is accomplished using the back propagation algorithm. Input features to the network are a mixture of image-based and scene based object parameters: image blob dispersedness (perimeter/area (pixels)); image blob area (pixels); apparent aspect ratio of the blob bounding box; and camera zoom. There are three output classes: human, vehicle and human group. When teaching the network that an input blob is a human, all outputs are set to 0.0 except for "human", which is set to 1.0. Other classes are trained similarly. If the input does not fit any of the classes, such as a tree blowing in the wind, all outputs are set to 0.0. This neural network classification approach is fairly effective for single images; however, one of the advantages of video is its temporal component. To exploit this, classification is performed on each blob at every frame, and the results of classification are kept in a histogram. At each time step, the most likely class label for the blob is chosen.

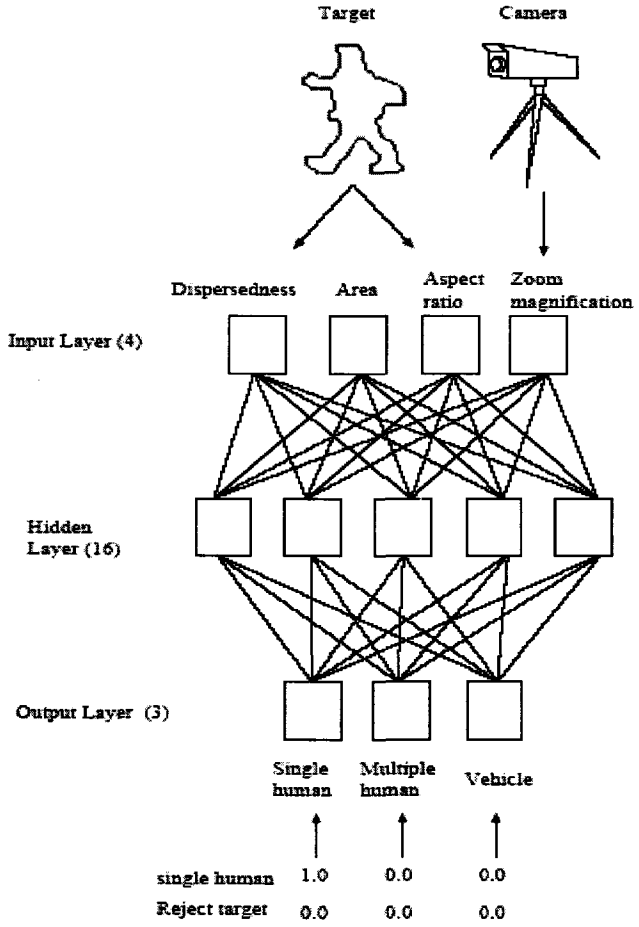



*Figure 3.4. Neural network approach to object classification.*

## 3.4 Object Tracking

The object tracking stage identifies the blobs, which until now have been just compact groups of pixels, and tracks the identified blobs over time.

The goals of the object tracking stage are to:
1) Determine when a new object enters the system's field of view, and initialize motion models for tracking that object.
2) Compute the correspondence between the foreground regions detected by the background subtraction and the objects currently being tracked.
3) Employ tracking algorithms to estimate the position of each object, and update the motion model used for tracking.

The process of tracking each moving object usually follows the following steps:
1. Initialization of the object (blob) to be tracked.
-------Determining size, shape or color features for each blob (object) when a new object enters the system's field of view, and initialize motion models for tracking that object.
2. Prediction of object future position and other property
--------Estimating the position of each object using motion model, and update
The motion model used for tracking.
3. Finding new object near predicted position and most similar to the old object
-------- Matching objects to current foreground regions by finding overlap between the estimated (via the global motion foreground regions from the current frame). For each object, all current foreground regions whose bounding boxes overlap sufficiently are candidates for matching that object.
4. Updating the object to be tracked
---------Changing object to the new object we get in step4 and continue tracking next image frame.
5. The new frame become the frame being processed and start again from step 2 and so on until all the frames have been processed.

In order to find the objects in a new frame, the "expected position" of all objects is predicted. If the distance between "predicted position" and an object in new frame is lower than a distance threshold, then the new object matches to old one. If necessary,

some properties, such as size and shapes, are used to calculate the similarity between two objects.

In ***[LJH]***, the second order Kalman filter (state including position, velocity, and acceleration) is used to model the motion of each person in the scene. The Kalman filter works in two stages: prediction and correction.

W4 in ***[HHD98]*** employs a second order motion model for each object to estimate its bounding box location in subsequent frames and then matches objects to current foreground regions by finding overlap between the estimated bounding boxes of objects and the bounding boxes of foreground regions from the current frame. For each object, all current foreground regions whose bounding boxes overlap sufficiently are candidates for matching that object. W4 uses a two stage matching strategy to update its global position estimate of an object. The initial estimate of object displacement is computed as the motion of the median coordinate of the object, which quickly narrows the search space for the motion of the object, and then they perform a binary edge correlation between the current and previous silhouette edge profiles. This correlation is computed only over a 5x3 set of displacements. Typically, the correlation is dominated by the torso and head edges, whose shape changes slowly from frame to frame.

In ***[SA]***, tracking consists of the iterative execution of three processes: (1) extracting the features of the obtained blobs in the current frame, (2) choosing the blob that seems most likely to have the same features as the tracking blob, based on Mahalanobis distance, and (3) judging the candidate blob to determine if it is similar enough to the tracking blob, based on the Bayesian probability measures.

In ***[IB97]***, four properties— average color, position, velocity, and size — are used to compute matching distance measures. The first measure is distance between the average color of an object and a blob. The second measure is the Euclidean distance between an object's position and a blob position in the new frame. At high frame rates, objects are normally close to their blob in the new frame. The third measure is distance from predicted position. Velocity is estimated using an object's current and previous positions. Then the position of the object is predicted in the new frame and the Euclidean distance computed between the predicted position and the blob position

in the new frame. Finally, the fourth measure is the size difference between an object's current blob and all blobs in the new frame, which varies slowly at high frame rates.

In ***[SPT]***, object main properties are used to calculate the similarity between two objects. That is they are similar if the weighted sum of color and size differences is larger than a threshold and the distance between the two objects is lower than a distance threshold.

## 3.5 Conclusion

There are various approaches for Object Tracking. In this chapter, a general approach is discussed in detail. As the first step, two kinds of pixel difference method were introduced. Then, methods on reducing noise and blob Calculation were discussed. Finally, different methods in object tracking and post processing were compared and discussed.

# Chapter 4

# A real-time tracking system------

# Fish-Tracker

For several decades, animals in motion have long been the subject of study for zoologists and have drawn much attention to computer vision researchers for automatic tracking and animations. Recently, automatic marker systems which can be used for tracking moving animals have become very popular in the market. Products, such as VICON, MOTION ANALYSIS and Peak Motus, all use reflective markers to capture and analyze motion patterns via videotape and optical capture methods. Windows-based Peak Motus is a software system providing the flexibility of optical marker-based motion capture as well as videotape-based motion capture. In controlled settings, optical markers are used to capture movement data. In outdoor situations where the use of markers is impossible, videotape-based motion capture can be used in creating the most dynamic capture environment. Once collected, movement data are exported into many leading animation software programs to create realistic 2D and 3D animated characters. Analog data may be easily collected within the system, and it can all be presented in full-color animated reports. The intuitive tab-card layout makes the Peak Motus motion measurement system easy to learn, use, and teach. Peak Performance Technologies is the worldwide leader in the design, development and manufacture of motion capture, motion measurement and analytical systems. Their products have been used in various industrial, medical, biological and sport science applications. However, high cost usually hits users of Peak Motus product. For example, it costs 20,000 dollars to get a full set of Peak Motus system for tracking fish movement.

## 4.1 Motivation of our Fish-Tracker System

The software system in this thesis was requested by department of Biological Science to analysis the behavior of fish being put into a shallow tank. Because the tanks are shallow, it is assumed that the fish are moving in a single plane; therefore, a 2D system is appropriate. In addition, Optical Capture is not an option because the contrast is low and reflective markers cannot be placed on the fish. So in this special case, a 2D video system with automatic tracking should be used to identify their points. Users might start by looking to see whether Peak Motus will help. Peak Motus

provides a flexible solution for video-based and optical, 2D and 3D tracking. But according to the description above, only a 2D video system can be used for analyses the behavior of a fish. As such, with far less cost and customized solution, developing a real-time software system to replace the current commercial software was justified.

## 4.2 The approach of our Fish-Tracker System

This system is developed for department of Biological Sciences to replace the current commercial software with far less cost and customized solution. The software tracks the movement of fish automatically, eliminating many time-consuming manual analytical steps. The system reads image sequences of a moving fish, obtained by a digital camera linked to a PC computer, and determines the trajectory of moving fish undergoing a planar motion in a scene observed by a still camera. The system has been developed using Microsoft Vision C++ 6.0, Matrox Imaging Library software and Qt programming. Our motion detection algorithm is based on frame pixel difference. Using motion and image subtraction, we obtain slight differences between subsequent frames, allowing us to perceive the moving fish in the image. Once the region of movement is detected, it is segmented using blobs (bounding box). Because shadows affect the magnitude of differences between images and may recognize the wrong object, we remove shadows by increasing the threshold value whenever the resultant image indicates pixels that do not belong to the object. The direction and velocity of each object are calculated between each pair of frames to predict the position of the object in the next frame. By assuming that object trajectories are continuous and that position changes at a small rate, a simple matching is performed. We calculate distance and velocity of key points in the calibration pattern plane and average these points. Our system is currently being used to determine spent time, path and velocity of moving fish. The specific objective of the our **Fish-Tracker System** was as following:

1. Calculate the fish's velocity
2. Obtain a diagram of the fish's pathway
3. Calculate how much time the fish spends in certain areas of the tank

## 4.3 Detecting Moving Fish

Using Frame Difference method we can detect moving fish. Those pixels that are not part of the fish can be ignored (Figure 4.1). The resulting change image indicates all differences due to the movement of the real objects.

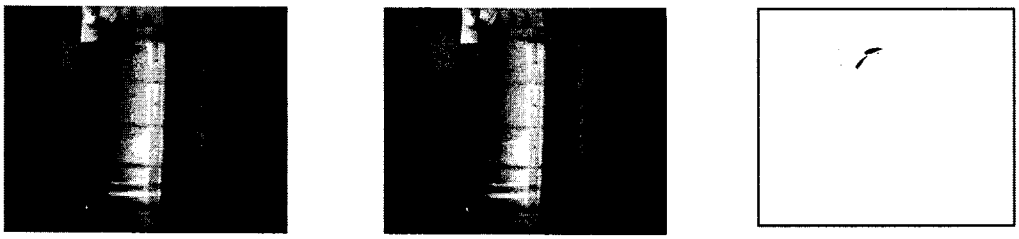


*Figure4.1. Detecting Fish – pixel difference*

Applying pixel difference method to image sequences, we can get fish moving path (Figure4.2).

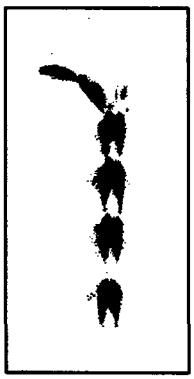


*Figure4.2. Fish moving path- image subtraction*

## 4.4 Removing shadow

Using pixel difference method, we detect not only moving fish but also some extra shadows (Figure4.3). There are many algorithms detecting shadows. The **first** algorithm in ***[BR02]*** aims to find quite large shadows. By calculating the intensity ratio for each pixel between the background and the current frame we can detect shadows. Since shadows are regions with similar intensity ratio while the objects are usually composed of significantly different gray levels. ***[BR02]*** states that the relationship between pixels when illuminated and the same pixels under shadows is roughly linear. The **second** algorithm in ***[BR02]*** is based on the edge gradient that is able to find out shadows when they are small and narrow. In addition, ***[CMC02]*** propose a simple method to reduce the shadow effects. A gradient filter filters the input images. Since in normal conditions, shadow tends to have a gradual change in luminance value. Therefore, after taking the gradient, the values in the shadow region tend to be very small while the edges have large gradient value. ***[EHD]*** has also showed how the model can use color information to suppress shadows of the targets from being detected.

Shadows can have a dramatic effect on the magnitude of difference images and recognize wrong object, despite the fact that no fish have been in that region. In our Fish-Trajectory System, we remove shadow by increasing the threshold value whenever the result image indicates that some black pixels do not belong to the fish (Figure 4.4).

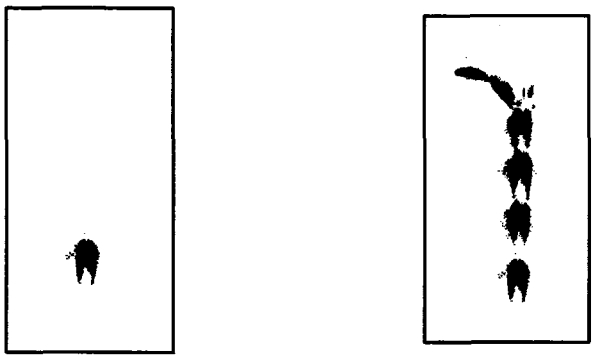


*Figure4.3. Detect Extra Shadows*

**The following are the processing steps of removing shadow:**

**1).** Initialize threshold value m=20

**2).** Remove all pixels that are less than m

**3).** Change m to the average pixel value of points remained.

**4).** Repeat step 2 and 3 until no point belongs to the shadow

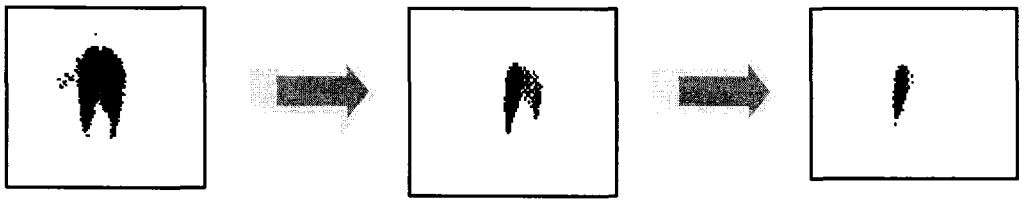


*Figure4.4. Removing Shadow*

Apply this algorithm to all sequences of images; we can get following result (Figure4.5).

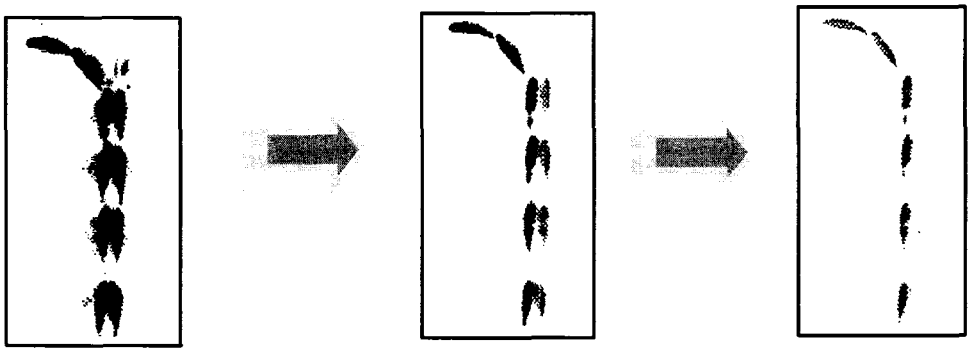


*Figure4.5. Fish path after Removing Shadow*

## 4.5 Extracting Feature Point

After fish are detected and shadows are removed, we need to get feature point that can stand for fish in image sequences (Figure 4.6).

The following are the processing steps:

1). Count the number of all pixels which are black

2). Get the minimum rectangle that can cover the black fish region.

3). Scan each black pixel within this rectangle from outside to the center

4). Make these pixels white until just left one

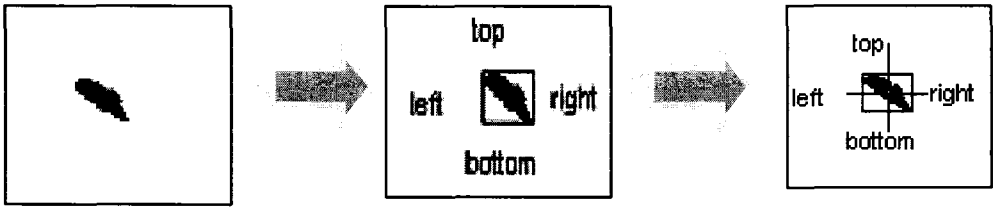



*Figure4.6. Getting Feature Point pixel stands for fish*

Applying above algorithm to image sequences, we can get sequences of points and each point stands for the fish at time instants t. And then draw line to get fish path (Figure4.7).

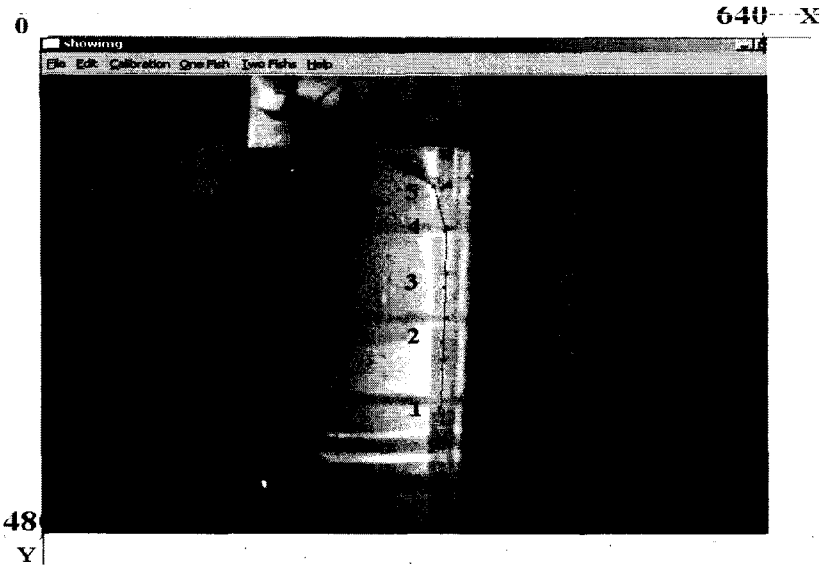



*Figure4.7. Fish path after getting feature point*

## 4.6 Camera calibration

After previous several steps, we already got 2D fish path on the image. However, our goal is to compute properties of the fish in 3-D world from these digital images

(Figure 4.8). To do so, we need to know the projection matrix: the camera parameters. Camera calibration is the process of estimating the parameters of camera.

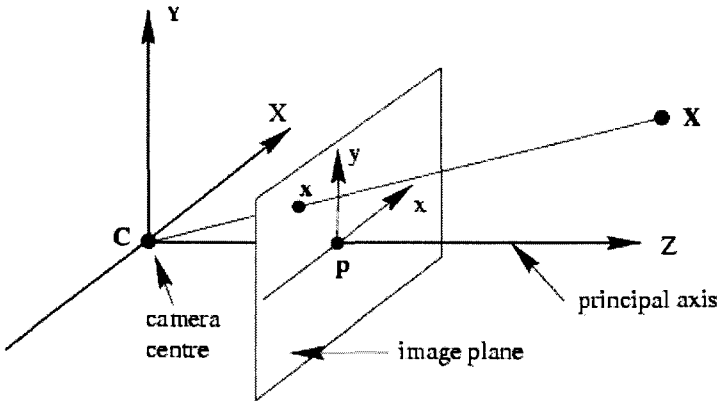



*Figure4.8. 2D ---3D correspondence*

The projection of a 3D point P=(X, Y, Z, 1) on to the pixel p= (u, v, 1) is given by

$$\begin{pmatrix} u \\ v \\ 1 \end{pmatrix} = \lambda \begin{pmatrix} m_{11} & m_{12} & m_{13} & m_{14} \\ m_{21} & m_{22} & m_{23} & m_{24} \\ m_{31} & m_{32} & m_{33} & m_{34} \end{pmatrix} \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix}$$

At least 6 points can solve 12 entries of projection matrix M (see chapter3).

In our case, Points from corner of tank on fish container bottom (Figure 4.10) are mapped to the image points on the image plane (Figure 4.9).

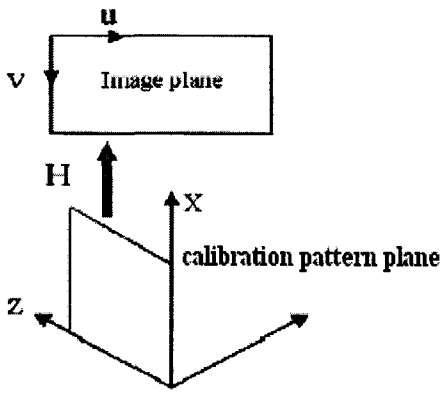



*Figure4.9. 2D ---2D correspondence*

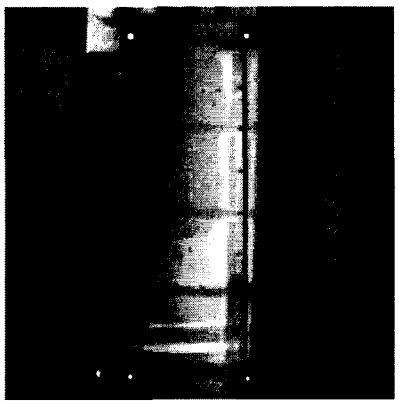


*Figure4.10. Calibration pattern*

The relationship between image pixels and points on the pattern is given by:

$$\begin{pmatrix} x \\ y \\ 1 \end{pmatrix} = \lambda \begin{pmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{pmatrix} \cdot \begin{pmatrix} X \\ Y \\ 1 \end{pmatrix}$$

or,

$$p = \lambda HP$$

At least four points are needed in order to calculate H. In our Fish-Tracker system, the calibration (Figure 4.11) is almost same as that in Peak Motus. Processing steps of calibration is as follows:

a) Input the first image

b) Draw tank region and use four corners of tank to calibrate. The first corner drawn becomes origin point and it should be closer to the position of fish in the first image.

c) Input Vertical and Horizontal Scale length.

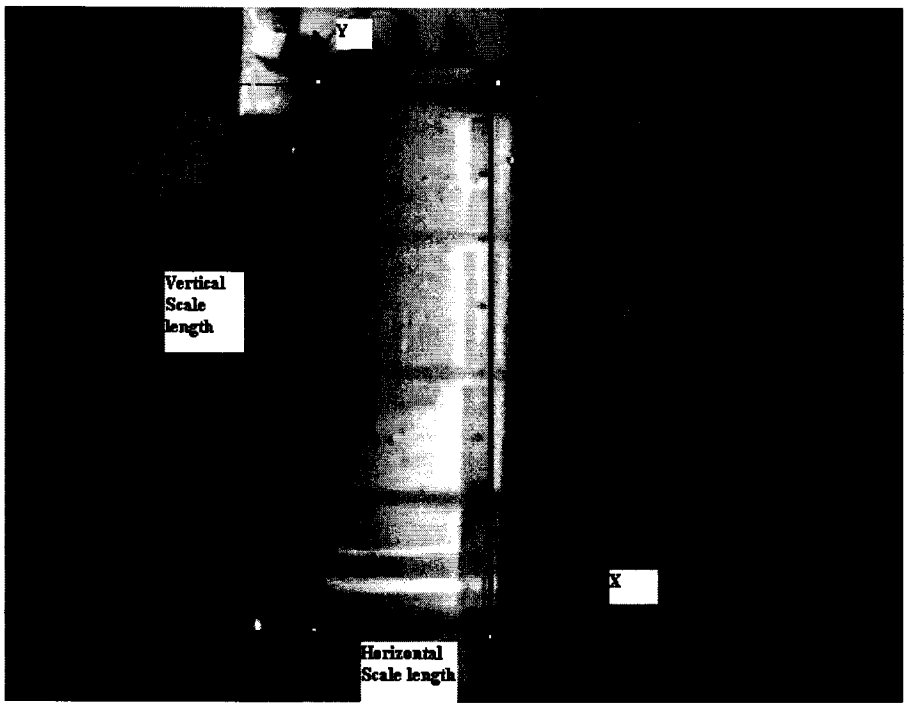



*Figure4.11. Calibration*

## 4.7 Trajectory of moving objects

If an object *i* is observed at time instants t = 1,2,...,n, then the sequence of image points $T_i = (p_{i,1}, p_{i,2},...,p_{i,t},...,p_{i,n})$ is called **the trajectory** of i. The trajectory of a moving object provides information about its velocity, direction and position.

Between any two points of the trajectory we can define their difference vector $V_{i,t} = p_{i,t+1} - p_{i,t}$ .

## 4.7.1 Computing the path of single fish

1) Calculating 2D fish path (Figure 4.12) on the image plane and X-Y 2D coordinates of key points on the path.

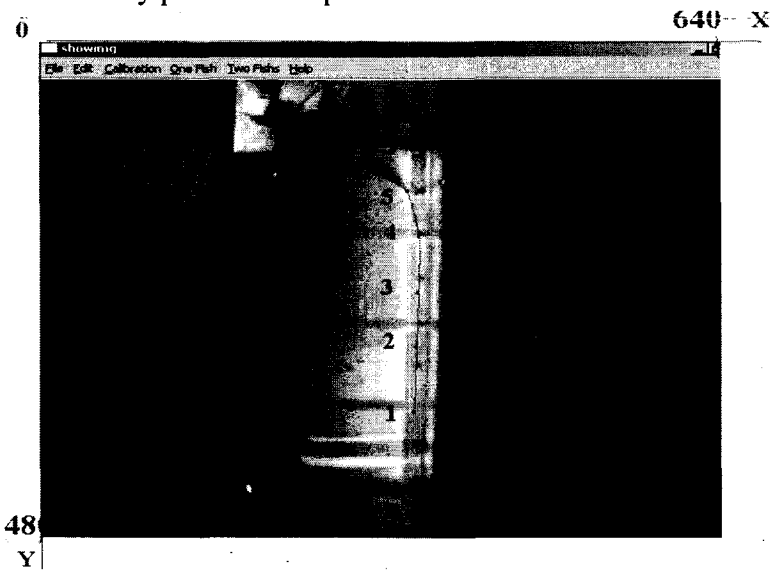



*Figure4.12. Fish path*

2) Reconstructing actual key point coordinates on the calibration pattern plane in 3-D world.

–**Pixels**: 2D coordinates X (im) and Y (im) on the image plane→ already calculated

–**H:** Project matrix h→ estimated by calibration

–**2D actual point**: coordinates X and Y on the calibration pattern plane → can be calculated

$$\begin{pmatrix} x \\ y \\ 1 \end{pmatrix} = \lambda \begin{pmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{pmatrix} \cdot \begin{pmatrix} X \\ Y \\ 1 \end{pmatrix}$$

or,

$$p = \lambda HP$$

3) Reconstructing all the actual key points on the calibration pattern plane and calculating actual fish path. (Figure 4.13)

**2D points coordinates on image plane and calibration pattern plane:**

| *Point No* | *Ximage(pixel)* | *Yimage(pixel)* | *Xpattern(cm)* | *Ypattern(cm)* |
|---|---|---|---|---|
| *1* | *339* | *351* | *33.17* | *22.92* |
| *2* | *343* | *281* | *34.08* | *41.02* |
| *3* | *342* | *222* | *33.75* | *55.31* |
| *4* | *343* | *167* | *33.91* | *67.89* |
| *5* | *333* | *118* | *31.57* | *78.55* |
| *6* | *307* | *95* | *25.76* | *83.41* |

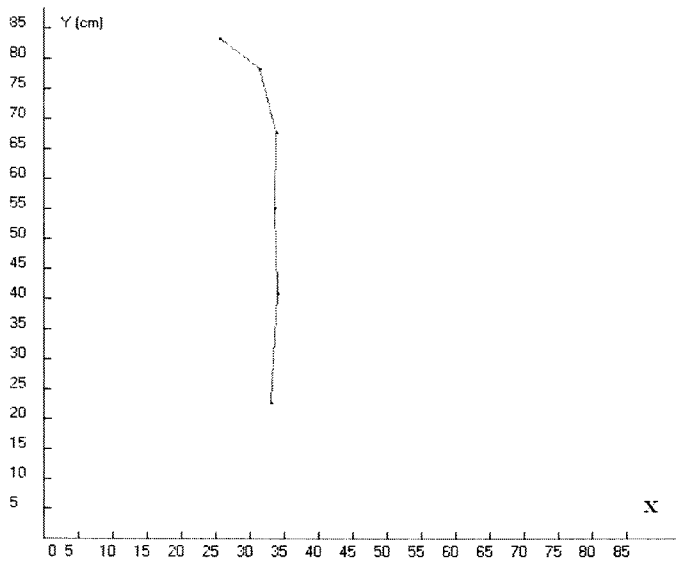



*Figure4.13. Fish path diagram*

## 4.7.2 Calculating the speed of single fish

**Fist,** calculating distance of actual key points in the calibration pattern plane.

$$\textbf{\textit{Distance}} = \sqrt{(X^2 + Y^2)}$$

**Second,** calculating velocity of actual key points in the calibration pattern plane.

$$\textbf{\textit{Velocity}} = distance \ / \ time\ interval$$

**Third,** calculating average speed of all the actual key points in the calibration pattern plane.

$$\textbf{\textit{Average speed}} = \sum(\text{velocity}) \ / \ image\ No$$

**Fourth,** calculating standard deviation and standard error of speed.

$$\textbf{\textit{Deviation}} = \sqrt{\sum(Velocity - AverageSpeed)^2 \ /(imageNo - 1)}$$

$$\textbf{\textit{Error}} = Deviation \ / \ \sqrt{imageNo}$$

**Actual Fish Velocity**

| *Second* | *Distance(cm)* | *Velocity (cm/s)* |
|---|---|---|
| *1.64* | *18.12* | *11.03* |
| *3.29* | *14.30* | *8.71* |
| *4.93* | *12.57* | *7.65* |
| *6.57* | *10.92* | *6.65* |
| *8.21* | *7.57* | *4.61* |

**Average speed of fish:**

7.73 cm/seconds

**Deviation:** 2.06

Error: **0.92**

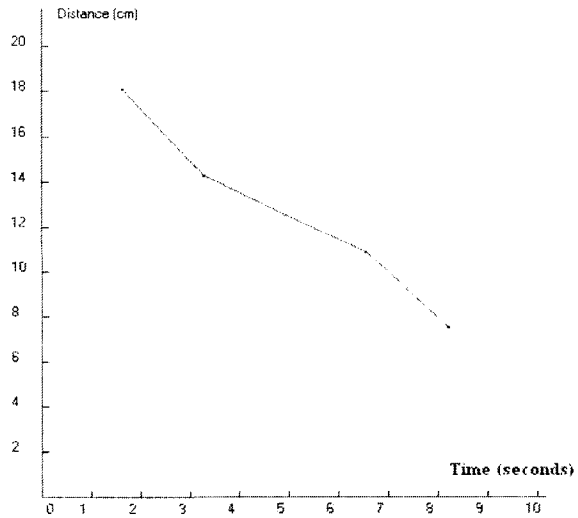



*Figure4.14. Fish speed diagram*

## 4.7.3 Computing the paths of two fishes

**1)** After fish are detected, we need to get key point that can stand for fish in image sequences. (Figure4.15)

–Use mouse to select first fish key points

–Use mouse to select second fish key points

–Get two fish key points on the image plane

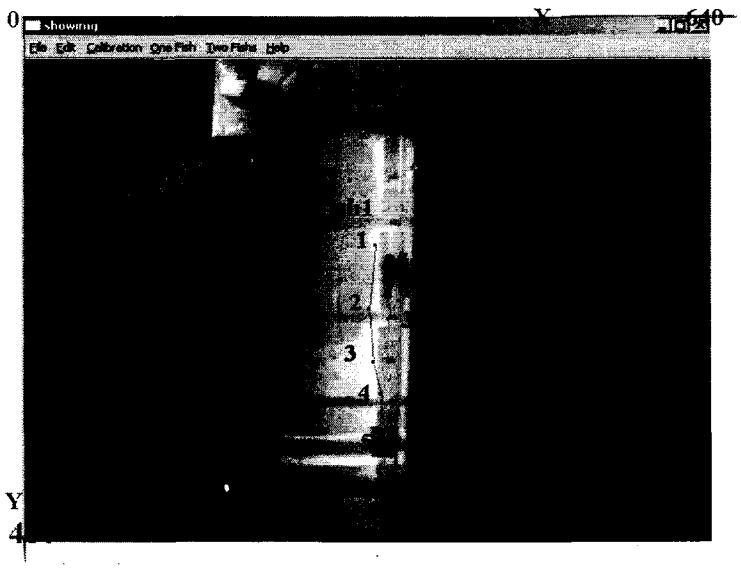



*Figure4.15. Two fish paths*

**2).** Reconstructing all the actual key points of two fishes on the calibration pattern plane and Calculating actual fishes path. (Figure4.16)

| *Point No* | *x1 (cm)* | *y1 (cm)* | *x2 (cm)* | *y2 (cm)* |
|---|---|---|---|---|
| *1* | *30.19* | *64.36* | *33.39* | *31.88* |
| *2* | *28.72* | *49.74* | *32.85* | *40.30* |
| *3* | *29.65* | *36.63* | *33.07* | *47.21* |
| *4* | *31.38* | *28.30* | *33.53* | *52.48* |
| *5* | *31.14* | *15.10* | *33.49* | *59.51* |

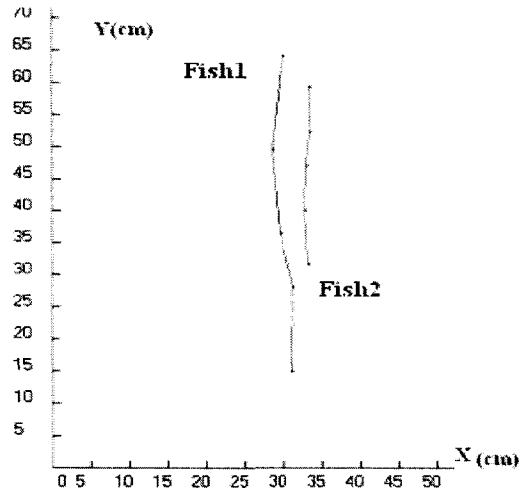



*Figure4.16 Two fish path diagram*

## 4.7.4 Calculating the speed of two fishes

**Velocity of two fishes on the calibration pattern plane**

| *Seconds* | *Distance1* | *Distance2* | *Velocity1* | *Velocity2* |
|---|---|---|---|---|
| *0.71* | *14.70* | *8.44* | *20.58* | *11.81* |
| *1.43* | *13.14* | *6.90* | *18.40* | *9.67* |
| *2.14* | *8.50* | *5.30* | *11.91* | *7.42* |
| *2.86* | *13.21* | *7.03* | *18.49* | *9.84* |

**Average speed of fish on the calibration pattern plane:**

*Fish 1* *17.34 cm/seconds*

*Fish 2* *9.68 cm/seconds*

**Deviation:** *Fish1: 2.56* *Fish2: 2.10*

**Error:** *Fish1: 1.28* *Fish2: 1.05*

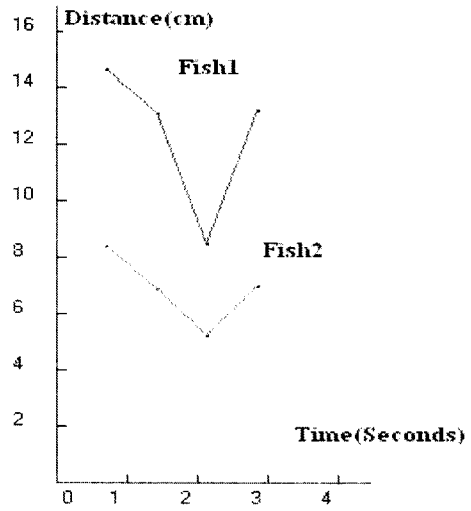



*Figure4.17 Two fish speed diagram*

## 4.8 Time Analyses

We need to be able to define specific areas of the tank in order to determine how much time a fish spends there in a given time period. For example, the two kinds of tank regions are the following (Figure4.18):

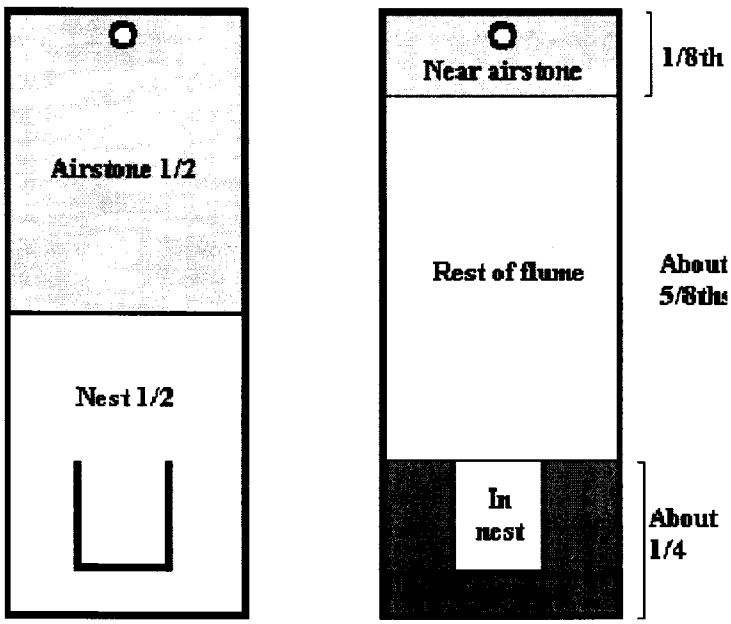



*Figure4.18 Defined Flume Regions*

After getting the path of fish, Fish-Tracker allows you to input total region number first, then draw regions one by one, finally save time data in files.

## 4.9 Conclusion

In this chapter, we proposed a fish tracking system that has been developed as part of the work of this thesis. In particular, to address the fish tracking related problems and challenges, methods in computer vision being used in this system are discussed in detail. The system read in image sequences of multiple moving fish and determines the trajectory of moving fish that passed through the scene. Our motion detection algorithm is based on frame pixel difference. With far less cost, our system provided a custom solution for fish tracking system with unique functionalities such as fish velocity calculation, obtaining a diagram of fish' pathway, etc. Technically, we remove shadows by increasing the threshold value whenever the resultant image indicates pixels that do not belong to the object. This approach is proven as an effective solution.

# Chapter 5

# Another real-time tracking system ------ Human-Counter

Human tracking is gaining increasing interests recently, especially in areas where people identification and activity recognition are important. Real-time human tracking is very useful in many areas, such as security applications, pedestrian traffic management, tourist estimation and so on. Object tracking is to tracking the location of an object. But Human tracking is more complicated and is far more than just location tracking. In human tracking, we need to know the size, orientation, shape and 3D pose of the target, sometimes even deformation and articulation. For example, in some cases, it is necessary to track some feature points or some geometric primitives such as lines and curves. In other cases, we need to track a shape or contour, e.g., face tracking. In addition, when estimating 3D information, we need to infer 3D poses of the object. And, when determining human motion, we need to know the articulation and deformation of human.

## 5.1 Human identification and activity recognition

Human identification and activity recognition can be classified into three categories. **(1) Human presence detection,** which essentially classify moving objects as human or non-human . **(2) Human motion classification,** which recognize different types of human locomotion, such as walking, running, limping, etc. **(3) Gait recognition,** which identify people from their gait (i.e. the 'way they walk'). Some representative research works in these three categories are briefly surveyed as follows.

### 5.1.1 Human presence detection

The objective of Human presence detection is to detect the presence, and to identify the movement and interaction of people through "blob". The system in ***[CL00]*** tracked the human body as a whole blob. They use a hybrid algorithm by combining adaptive background subtraction with a three-frame differencing technique to detect moving objects, and use the Kalman filter to track the moving objects over time. Detected objects are classified into semantic categories such as human, human group, car, and truck using shape and color analysis, and these labels are used to improve tracking

with temporal consistency constraints. They have developed a method for classifying vehicle types and people using linear discriminant analysis (LDA). LDA is often called supervised clustering. The method has two sub-modules: one for classifying objects "shape", and the other for determining "color". Each sub-module computes an independent discriminant classification space, and calculates the most likely class in that space using a weighted *k*-class nearest-neighbor (*k*-NN) method. Further classification of human activity, such as walking and running, has also been achieved. This system is very successful at tracking humans and cars. But it did not put much emphasis on activity recognition.

### 5.1.2 Human motion classification

Here, the goal is to recognize generic activities such as the movement of arms and legs, instead of trying to tie the action to a particular person. Using these methods, it is possible to refine the "blob" representation of a person through hierarchical, articulated models. This allows main body parts, such as head, arms, torso, and legs, to be individually identified to specify the activities more precisely. For example, J. Ben-Arie, Z. Wang, P. Pandit in ***[AWP02]*** uses a 2-D stick model to represent torso, arms, and legs. The recognition algorithm consists of two phases. The first phase computes the angles between connected body parts such as the upper arms and the lower arms. The algorithm matches the angular trajectories with trained data and accumulates the best matches into a hash table. The second phase then computes the vote for a whole motion sequence and identifies the activity as the one in the database receiving the largest number of votes. The above algorithm uses 2D information. It is also possible to employ explicit 3D models. Some current research is focused on tracking human body parts using generic 3-D models, e.g., head and hands. Q. Delamarre and O. Faugeras in ***[DF01]*** build a 3D model from multiple views. They initialize the 3D model at the first frame, then, estimate the model state by the Kalman filter and physical forces which pull the template model to confirm to the real object pose observed in image. The information extracted from image includes depth information from stereo-correlation and the silhouettes. The distance between extracted features and those predicted by 3D model is computed. And the distance of deviation between image features and the 3D model determine the strength of physical force. The above process is computed iteratively until it converges. Finally, E. Ong, and S. Gong in ***[OG99]*** make a compromise between 2D and 3D analysis. To build and track a 3D-model is difficult. So E. Ong, and S. Gong employ a hybrid 2D-

3D model. The 2D information in ***[OG99]*** includes a set of feature points, e.g., head, right and left hands, and the contour of the human body. The 3D model used is a skeleton model. It describes the bone structure of a person represented by joints and vertex points, such as hands and head. Given 2D information, 3D structure can be inferred through inverse kinematics. But it is often ambiguous to infer 3D structure just from a single viewpoint. Thus this method employs multiple views.

### 5.1.3 Gait recognition methods

The term gait recognition is typically used to signify the identification of individuals in image sequences 'by the way they walk'. Tracking many gait recognition techniques fall in this category, which aim to identify individuals against a pre-established gait database. Approaches to gait recognition can be classified roughly as either model-based, which recover a structural model of the human body and use this structure for motion recognition or model-free, which directly model, extract and recognize the motion patterns generated by any particular body movement. One example of the model-based method is [LG02], in which seven ellipses are used to represent different parts of the silhouette of a person. For each ellipse, the centroid, aspect ratio of the major and minor axes, and the orientation of the major axis are extracted. For each image frame, combining these parameters of the seven ellipses forms a "region feature". Given a gait sequence, two kinds of features are computed over time. One feature is the mean and standard deviation of the region features, combined with one additional parameter: the height of the centroid of the whole silhouette. Together, they provide a "gait average appearance feature." The other feature is computed based on the magnitude and phase of the Fourier transform of the region features in the sequence, which gives a "gait spectral component feature." Since both features could be affected significantly in the presence of noise in the silhouette, so to combine these two features will be a possible improvement. One example of the model-free methods is ***[CKC03]***. In ***[CKC03]***, various features are extracted from the gait sequence, such as the swing of the hands/legs, the sway of the upper body and static features like height. For different kinds of features, different methods such as DTW (dynamic time warping) and HMM (hidden Markov models) are used for classification. For example, the features for the swing of the hands/legs are projection vectors, and are matched by DTW. And the HMM is used to represent the leg dynamics. The results of these classifications are combined by Sum, Product

and Minimum rules to achieve a decision fusion. This approach improves the overall recognition performance. *[CKC03]* is not view invariant and need camera calibration information. *[KCC03]* proposes a view invariant method to synthesize the side view from any other arbitrary view using a single camera if the person is far enough from the camera. By using the perspective projection model and the optical flow based structure from motion equations, the azimuth angle of the original view is estimated, and a video sequence at the new side view is synthesized. This approach can be combined with other gait analysis techniques for efficient and invariant recognition.

## 5.2 Human Extraction

The human extraction process extracts human blobs from the foreground blobs. In general, there are two main approaches for extracting and tracking human body, namely Static feature extraction (e.g. points and contours) approach and the dynamic feature extraction approach *[WTN]*. For static feature extraction, whole object is tracked based on its shape and appearance. In *[WAD97]* and *[WW]*, the human has to be recognized and located in the image using activity-specific static body parameters without directly analyzing gait dynamics, feature like corners and contour are extracted and tracked from image to image, and it is necessary to solve the correspondence of the points in different images. Pattern matching techniques are usually adopted. For dynamic feature extraction approach, the human has to be recognized using action of gait like in *[HHD98]*. A structural model of the human body can usually be recovered and used for motion recognition in this approach. Although comparison between the image and the projection of the models provide flexibility and stability in tracking, the time complexity is usually high, especially for object with a deformable shape. To handle this kind of pattern matching and object recognition problem, window searching may help. By restricting the search region, the searching time can be reduced. In order to have a small search region, the prediction of the location of the object in next time frame has to be accurate. Based on the idea that human body includes both the appearance of human body and the dynamics of gait motion measured during walking, *[WTN]* attempt to combine the two different sources of information to recognize and track people. The proposed method is shown in Figure 5.1. For each image sequence, background subtraction is used to extract moving silhouettes of the walker. Static pose changes of these silhouettes over time are represented as an associated sequence of complex vector configurations in a common coordinate, and are then analyzed using the Procrustes

shape analysis method to obtain an eigen-shape for reflecting the body appearance, i.e., static information. Also, a model-based approach under a Condensation framework together with human body model, motion model and constraints is presented to track the walker in image sequences. From the tracking results, they can calculate joint-angle trajectories of main lower limbs, i.e., dynamics of gait. Both static and dynamic information may be independently used for recognition using the nearest exemplar pattern classifier. They are also combined on decision level to improve the final performance.

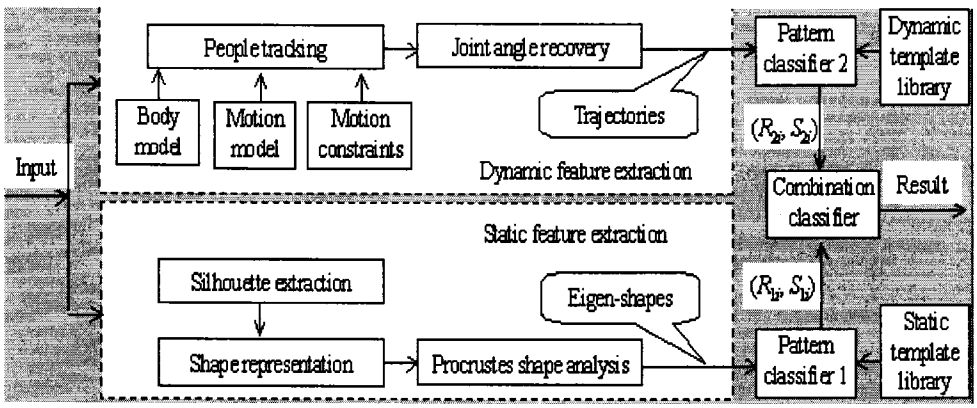



*Figure 5.1. Static feature extraction and dynamic feature extraction*

## 5.3 Human body models

There are lots of human body models used in detecting and tracking people. In ***[HHD98]***, A second-order motion model including the velocity and acceleration terms is used to track the overall body motion and the motions of body parts. A cardboard human model (Figure5.2) of a person in a standard upright pose is used to model the human body and to predict the location of human body parts. Cardboard human model locates 6 body parts: Head, Torso, Hands, Feet and takes height of the bounding box of an object as the height of the cardboard model. Finally, it uses fixed vertical scales to find the location of body parts. The lengths of the initial bounding boxes of the head, torso, and legs are calculated as 1/5, 1/2 and 1/2 of the length of bounding box of the object, respectively. The locations of these parts are verified and refined using dynamic template matching methods.

*Figure5.2. Cardboard human model*

In ***[SA]***, Koichi Sato and J. K. Aggarwal use correlation with binary person templates to detect the people present in the foreground layers. ***[AWP02]*** uses a 3-D stick model (Figure5.3) to represent torso, arms, and legs and recognizes people actions using two phases. The human body is represented by nine cylinders for the torso, upper arms, lower arms (forearms +hands), legs and feet.

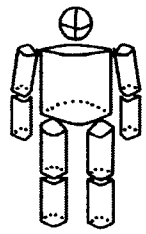


*Figure 5.3. 3-D stick model*

The Ellipses Human model has been applied in ***[LG02]***(Figure5.4). A foreground walking person is divided into 7 regions. Seven ellipses are used to represent different parts of a person. For each ellipse, the centroid, aspect ratio of the major and minor axes, and the orientation of the major axis are extracted. For each image frame, combining these parameters of the seven ellipses forms a "region feature".

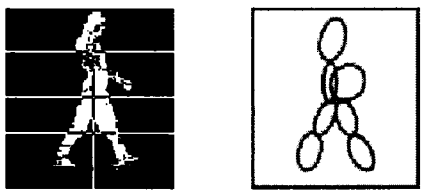


*Figure5.4. Ellipses Human model*

The Truncated Cones human model used in ***[WTN]*** (Figure5.5) is composed of 14 rigid body parts, each of which is represented by a truncated cone except for the head represented by a sphere. Truncated Cones human model locates the global position and track each limb separately. Then, it can be formulated as a tree-like structure and represented by a hierarchical estimation.

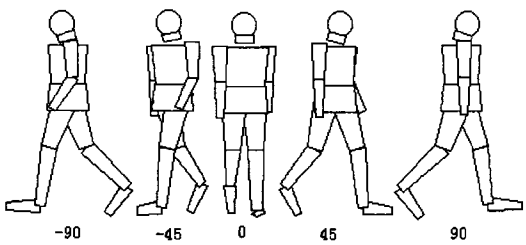



*Figure5.5. Truncated Cones human Model*

## 5.4 Real-time Human Tracking System

There are lots of real-time systems developed for human tracking. In order to give a basic idea about the approach of human tracking, we will introduce two important real-time systems in human tracking here.

Pfinder ("Person finder") introduced in ***[WAD97]*** is a real-time system for tracking single person and interpreting their behavior. Pfinder runs at 10Hz on a standard SGI computer, works with stationary, YUV color-based video sources and has performed reliably on thousands of people in many different physical locations. Pfinder tracks the human body using a set of "blobs" which correspond to the person's hands, head, feet, shirt and pants, uses 2D model of color and shape to represent these body parts. Pfinder has some limitation. First, Pfinder expects the scene to be significantly less dynamic than the user. Although Pfinder has the ability to compensate for small, or gradual changes in the scene or the lighting, it cannot compensate for large, sudden changes in the scene. If such changes occur, they are likely to be mistakenly considered part of the foreground region, and an attempt will be made to explain them in the user model. Another limitation, related to the dynamic scene problem, is that the system expects only one user to be in the space. Multiple users don't cause problems in the low-level segmentation or blob tracking algorithms, but do cause significant difficulties with the gesture recognition system that attempts to explain the blob model as a single human figure. The processing steps of Pfinder system is as following:

**1.Build the background model**

A model of the scene is built by observing the scene when no person is present. For each pixel, the mean color value and the covariance of the associated distribution are determined.

**2. Detect moving object by measuring color deviations from background model**

Pfinder first detects a differently colored region as change in scene and then uses 2D contour shape analysis to identify a set of "blobs" which correspond to the person's hands, head, feet, shirt and pants locations. Each blob has a spatial (x, y) and color (Y, U, V) component, and also have a detailed representation of its shape and appearance (Figure 5.6).

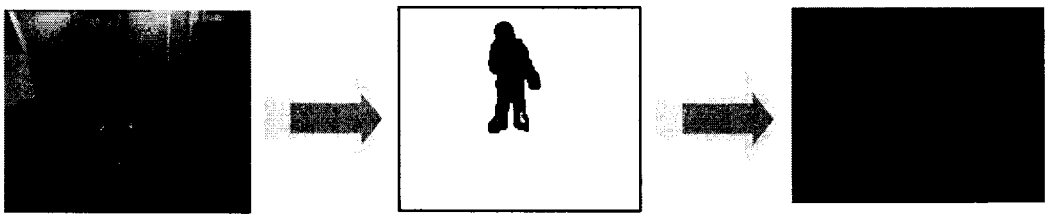


*Figure 5.6 Pfinder System--build blob model*

**3. Tracking single person**

Pfinder first predicts the appearance of blob in the new image using the current state of person model and calculates the likelihood that the pixel is a member of the blob for each image pixel and for each blob model. Secondly, Pfinder resolves this pixel-by-pixel likelihood into a support map and indicate for each pixel whether it is part of one of the blobs or of the scene. Finally it updates the statistical models of all blob models.

Another important real-time tracking system is W4 system. The W4 system introduced in ***[HHD98]*** is designed for outdoor surveillance tasks and particularly for low light level situations. It operates on monocular grayscale video imagery and constructs dynamic models of people's movements to answer questions about what they are doing, and where and when they act. A second-order motion model including the velocity and acceleration terms is used to track the overall body motion and the motions of body parts. A cardboard human model of a person in a standard upright pose is used to model the human body and to predict the location of human body parts. The locations of these parts are verified and refined using dynamic template matching methods. Region splitting and merging are handled and individual body parts such as head, hands, torso and legs are tracked in order to understand actions. Unlike many of systems for tracking people, W4 makes no use of color cues. Instead, W4 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. W4 is capable of simultaneously tracking multiple people even with occlusion. There is also limitation for W4. The W4 system represents a good first step to the problem of recognizing and analyzing human activities, but the cardboard model used in W4 to predict body

pose and position is restricted to upright people. The processing steps of W4 system is as following:

**1. Foreground Object Detection.**

Foreground regions are detected by a combination of background analysis and simple low level processing of the resulting binary image. Background scene is modeled by three values for each pixel, m (x): minimum intensity value, n (x): maximum intensity value and d (x): maximum intensity difference between consecutive frames. Foreground region is classified using background model

$$\left\{\begin{array}{ll} 0 \text{ background} & \left\{\begin{array}{l}(I^t(x)-m(x))<kd_\mu \\ \vee I^t(x)-n(x))<kd_\mu\end{array}\right. \\ 1 \text{ foreground} & \text{otherwise.}\end{array}\right.$$ fference

Region-based noise cleaning is applied to eliminate noise regions after thresholding. Binary connected component analysis is applied to each foreground object

**2. Constructing appearance model using global shape features for object**

Global shape features includes Centroid (median coordinate of foreground region), Major axis and Shape of 2D silhouettes that is represented by horizontal and vertical projection histogram

**3. Distinguishing single person, people in group and others**

Analyze vertical projection histogram to determine if a region contains multiple people (Figure 5.7) and distinguish a single person using a set of average normalized templates from database.

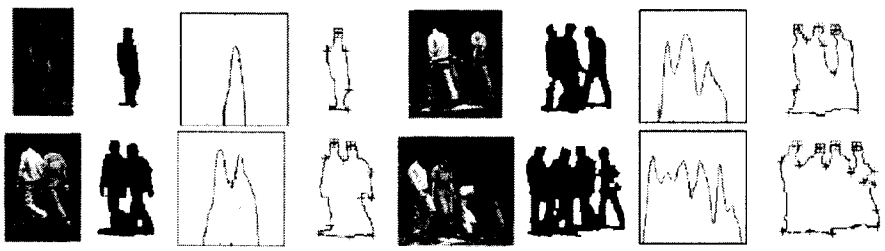


*Figure 5.7.Distinguishing single person and people group in W4*

**4. Tracking single person**

First the system model motion for each person to estimate its location in subsequent frames and compare estimated bounding boxes with actual ones. And then determine current position to update motion model using displacement of median coordinate of the person and binary edge correlation between 2 silhouette edges (Figure 5.8).

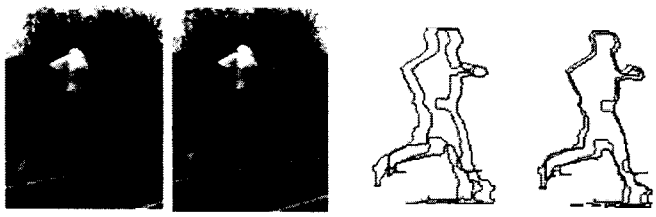


*Figure 5.8. Tracking single person*

5. **Tracking people in groups**W4 counts number of people in the group by identifying their heads using Vertical projection histogram (significant peak) or Geometric shape cues (convex hull-corner vertices) (Figure 5.9).

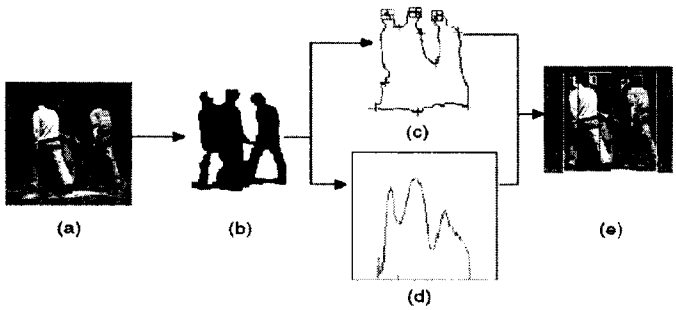



*Figure 5.9. Counting number of people*

## 5.5 The Approach of Our Human-Counter System

To track and count moving people is important for the office security or the marketing research. Many of such measurements are still carried out on manual works of persons. Therefore it is necessary to develop the automatic method of counting the passing people. Several attempts have been made to track people. **[Janne]** is developing person-counting software. This system uses motion detection algorithm for a digitized image sequence for extracting regions of activity from the static background. They infer the number of the persons in each of these regions by applying snake-technique for finding the boundaries of the objects in the region. The number of persons can be obtained by following the shape of the snake-curve. The image in Figure 5.10 is one frame from a video sequence that shows how the algorithm has identified two moving objects; it should be clear how the peaks in the red outlines represent the two humans.

*Figure 5.10 Snake-curve representing the number of people*

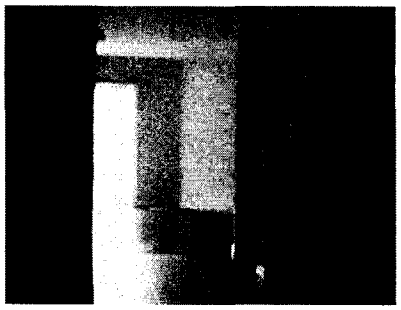
The W4 system *[HHD98]* counts number of people in the group by identifying their heads using –Vertical projection histogram (significant peak). They Generate shape and appearance features for each detected foreground object using horizontal and vertical projection histogram and then analyze vertical projection histogram to determine if a region contains multiple people. Finally they count the number of people according significant peak (Figure 5.9).

In this paper, we propose a **Human-Counter System** with a single camera for security inside the building. The camera is setup at the high area of the room so that the image data of the passing people can be recorded. The implemented system recognizes people movement and track people even when their images are partially overlapped. The main goal of **Human-Counter System** is to count the number of people that pass in front of the camera (in any direction) during a given time period. The development of the **Human-Counter System** was taken in the following stages:

1) Find moving pixels (pixels which do not match the background image) using background subtraction and display as a binary (moving or not moving) image
2) Remove unnecessary noise from this image using opening and closing
3) Connect up moving pixel regions that are very close to one another
4) Create object groups when two tracked objects collide (one object may occlude another)
5) Count the number of object groups.
6) Analyze each object group to find the total number of people.

### 5.5.1 Background Subtraction:

Background subtraction is a method typically used to detect unusual motion in the scene by comparing each new frame to a model of the scene background. Given a background image of a scene, moving pixels can be located simply by obtaining the absolute values of the difference between the corresponding pixels in current frame and the background and large values in the difference map then indicate locations of change. The difference map is usually binarised by thresholding it at some pre-determined value to obtain a change/nochange classification. The result image after background subtraction is shown as following (Figure 5.11):



*The background image for this scene*

*The current frame of a video with three people visible*

*The binarised moving pixels image resulting from the background subtraction operation*

*Figure5.11. – Example of background subtraction*

## 5.5.2 Noise Reduction (Opening-Closing)

After background subtraction has occurred, there will be pixels in the resulting image that appear having moved, but are not a valid moving object. This is referred to noise. Although the number of these noise pixels can be significantly reduced by accurate threshold, there will always be some noise generated. The binary morphological operations that will be used in this Human-Counter system to remove noise are Opening , Closing and with a structuring element. The result of opening an image is to remove small noisy image elements and the result of closing on a binary image is to connect objects that are close together, fill up small holes that may appear in objects and smooth object outlines. These operations will be used after the background subtraction stages to:

1. Remove objects, which are too small to be valid traceable regions
2. Connect moving objects that have been split by a noisy background
3. Fill in the holes within moving objects that may appear due to the background subtraction technique

The Result image after noise Reduction is shown as following (Figure 5.12):

*Image resulting from the background subtraction*

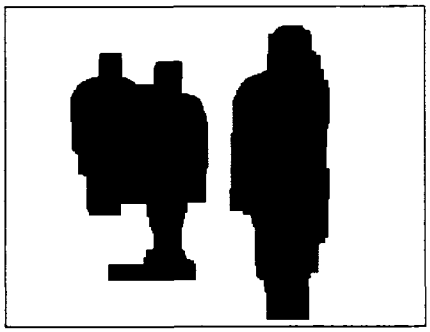


*Image resulting from the Noise Reduction*

*Figure5.12 – Example of Noise Reduction*

## 5.5.3 Object group connection

This stage will find homogenous regions in an image – in this case a binary image. White pixels are considered to be background pixels and so any black pixels indicate foreground pixels (the objects the program is interested in). The essential idea behind object group connection is to connect all the pixels that touch other black (foreground pixels) together in order to form a region. The fundamental method is relatively simple. The image, in this case the background subtraction image of the current frame with noise reduction completed, is searched through pixel by pixel, from top left corner to bottom right and any black pixel is labeled. The label is either a new label if none of the surrounding pixels are labeled or one of the surrounding pixel's labels. If there is more than one distinct label around the pixel,"equivalence" is noted between them. After all the pixels have been thus labeled, all the equivalent labels are relabeled so that connected pixels have the same label and so become the same connected region.

***Algorithm:***

***For each pixel in the image***

***If pixel is a moving pixel (i.e. possibly part an object)***

***If pixel has no label***

***Get a new label and Assign Label to pixel***

***Check the neighbouring pixels***

***For all unlabelled neighbours which are moving pixels***

***Label with current label***

***For all differently labeled neighbours***

***Note the equivalence***

***If pixel has label***

***Check neighbouring pixels*** *(as above: check if this pixel is a moving pixel and if it has label.)*

The Result image after Group Connections are shown as following (Figure 5.13):

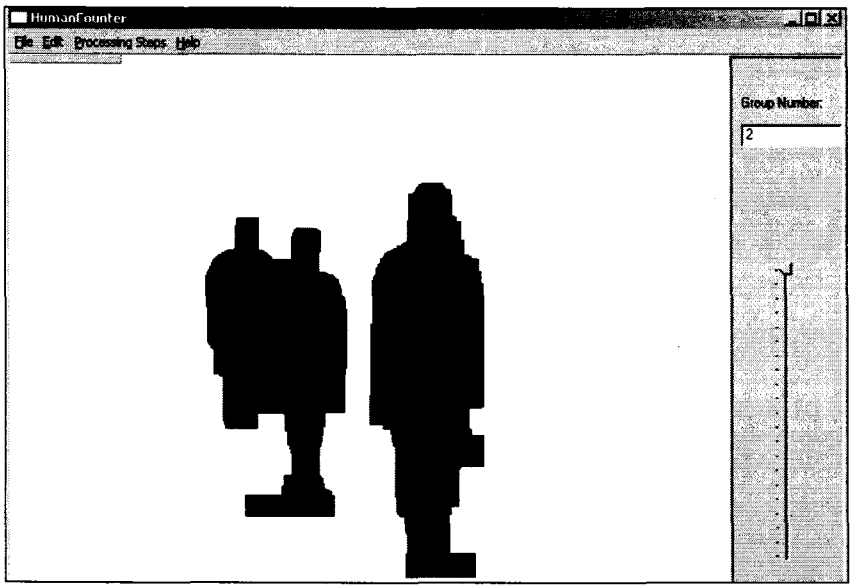



***Image resulting from the Group Connection (two groups with different color)***

*Figure 5.13 – Example of Group Connection*

## 5.5.4. Analyze each object group

After Region Connecting we can get the number of connected group. However sometimes a group consists of one or more people walking together and whose object regions overlap so that they become one connected component. So we need to analyze each connected group to get actual people number. We take each region being tracked and analyses certain characteristics and makes an estimate of the number of people within the group. First, we get boundary of each connected region and draw a bounding box of each region. Secondly, we draw a line at 1/8 of the height of each region. Finally, we count the number of intersection points between the boundary and this line. If there are two points number, the result is one person in this connected region; If there are four points number, the result are two people in this connected region and so on (Figure 5.14). Besides, we also compare the width of each region with the width of single person sample to estimate the number of people within this region.

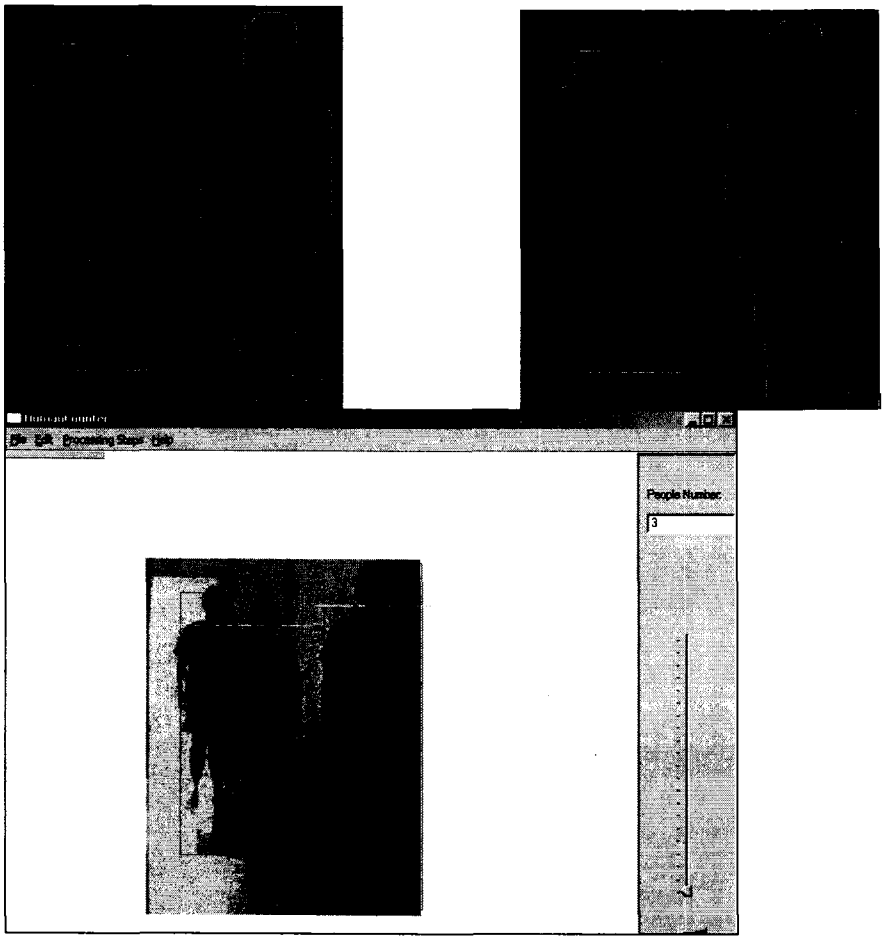



*Figure 5.14 – Example of Counting People*

## 5.6 Conclusion

In this Chapter, we first introduced three categories of human identification and activity recognition and some human body models used in real-time human tracking systems, and then we proposed a **Human-Counter System** in detail. In this system, some basic image morphological operations such as Opening and Closing are employed to reduce noise. And a connected components algorithm is used to get the number of connected regions for each image. Finally, we count total number of people by analyzing each connected region.

# Chapter 6

# Conclusion and Future Work

Tracking systems use different algorithms and procedures for different applications. In this thesis, we first presented a work that we have done on the motion detection in Fish-Tracker system. The system reads in image sequences of multiple moving fish and determines the trajectory of moving fish that passed through the scene. Our motion detection algorithm is based on frame pixel difference. The direction and velocity of each object are calculated between each pair of frames and are used to predict the position of the object in the next frame. The calibration procedure and the problems due to the presence of noise and shadows in the images are also addressed and the adopted strategies are detailed. Our system has successfully been tested on a fish tracking application and is currently being used to study the behavior of the fish in response to changes in environmental conditions. In addition, Human-Counter system is another tracking system presented in this thesis. People region are detected by finding the difference between the previously obtained background and current image. Some basic image morphological operations such as Opening and closing are employed to reduce noise and obtain clear foreground regions. A connected components algorithm is used to get the number of connected regions for each image. Finally, we count total number of people by analyzing each connected region.

## Future Work

Our system can be improved and extended in following areas:

- One area is the techniques for removing shadows. In this thesis, the method for removing shadows used in fish tracking system is efficient but it does have some limitations. For instance, in the situation of shadows are darker than the moving objects, it will not work very well. Methods of removing shadows are very important topic in computer vision and deserve a more extensive discussion and research.

- Another area is to identify main human body parts and recognize different types of human activities using human model and pattern matching. For example, in the shopping market, customer behaviour and shopping patterns need to be detected and analysed for security and marketing purpose.

# Bibliography


**1. [AC97]** J. K. Aggarwal and Q. Cai, "Human Motion Analysis: a Review," in Proc. Of IEEE Computer Society Workshop on Motion of Non-Rigid and Articulated Objects, 1997.

**2. [AP96]** A. Azarbayejani and A. Pentland. Real-time 3d tracking of the human body. In *Image Com*, 1996.

**3. [AWP02]** J. Ben-Arie, Z. Wang, P. Pandit, and S. Rajaram, "Human Activity Recognition Using Multidimensional Indexing", PAMI, Vol. 24, No. 8, August 2002.

**4. [B94]** M. Bichsel. Segmenting simply connected moving objects in a static scene. *PAMI*, 16 (11):1138-1142, 1994.

**5. [B95]** A.M. Baumberg, "Learning Deformable Models for Tracking Human Motion," PhD thesis, The University of Leeds, School of Computer Studies, UK, October 1995.

**6. [B97]** C. Bregler "Learning and Recognizing Human Dynamics in Video Sequences" In Proc. CVPR 97, June 1997.
http://mrl.nyu.edu/~bregler/bregler_humandyn.pdf

**7. [B01]** C. BenAbdelkader, "Gait as a Biometric for Person Identification in Video Sequences", Dissertation, University of Maryland, 2001.
http://citeseer.ist.psu.edu/cache/papers/cs/25419/http:zSzzSzwww.cs.umd.eduzSzLibraryzSzTRszSzCS-TR-4289zSzCS-TR-4289.pdf/benabdelkader01gait.pdf/

**8. [BB94]** Barron J. L, Fleet D. J, and Beauchemin S. S. Performance of optical flow techniques. *International Journal of Computer Vision*, Vol. 12, pp 43–77, 1994.
http://citeseer.ist.psu.edu/cache/papers/cs/1847/http:zSzzSzwww.qucis.queensu.cazSzhomezSzfleetzSzresearchzSzPaperszSzijcv-94.pdf/barron92performance.pdf/

**9. [BD01]** A. F. Bobick, and J. W. Davis, "The Recognition of Human Movement Using Temporal Templates", PAMI, Vol. 23, No. 3, 2001.

**10. [BH93]** Baumberg, A. M. and Hogg, D. C. Learning flexible models from image sequences. Technical Report 93.36, University of Leeds School of Computer Studies, 1993.

**11. [BJ96]** M. J. Black, A. D. Jepson, "EigenTracking: Robust Matching and Tracking of Articulated Objects Using a View-Based Representation", ECCV, 1996 http://www.cs.jhu.edu/~hager/Public/teaching/CS600.641/black96eigentracking.pdf

**12. [BK]** David Beymer, Kurt Konolige. "Person Counting Using Stereo", http://www.iser02.unisa.it/papers/6.pdf

**13. [BK99]** Beymer, D. and K. Konolige. *Real-Time Tracking of Multiple People Using Stereo.* in *IEEE Frame Rate Workshop.* 1999. Corfu, Greece. http://www.ai.sri.com/~beymer/vsam/vsam-slideshow/

**14. [BM98]** B.Boufama and R. Mohr. "A stable and accurate algorithm for computing epipolar geometry". In *International Journal of Pattern Recognition and Artificial Intelligence,* (12) 6 (1998), pages 817--840.

**15. [BR93]** J. Ben-Arie and K.R. Rao, "A Novel Approach for Template Matching by Non-Orthogonal Image Expansion," IEEE Trans. Circuits and Systems for Video Technology, vol. 3, no. 1, pp. 71-84, Feb. 1993.

**16. [BR02]** A. Bevilacqua M. Roffilli, "Robust denoising and moving shadows detection in traffic scenes" 2002

**17. [BY96]** S. Ju, M. Black, and Y. Yacoob. Cardboard people: A parameterized model of articulated image motion. *Proc. Gesture Recognition,* 1996. http://www.cs.brown.edu/~black/Papers/fg96.pdf

**18. [CB92]** R. Curwen and A. Blake, Dynamic Contours: Real-Time Active Splines. *Active Visio*n, A. Blake and A. Yuille, eds., pp. 39-58. MIT Press, 1992.

**19. [CC96]** V. Caselles and B. Coll, Snakes in Movement. SIAM J. Numerical Analysis, vol. 33, pp. 2,445-2,456, 1996.

**20. [CD00]** Ross Cutler and Larry Davis. Robust real-time periodic motion detection, analysis, and applications. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8):781 –796, 2000.

**21. [CK77]** J. Cutting and L. Kozlowski, "Recognizing friends by their walk:gait perception without familiarity cues," *Bulletin of the Psychonomic Society*, vol. 9, pp. 353–356, 1977.

**22. [CKC03]** N. Cuntoor, A. Kale and R. Chellappa, "Combining Multiple Evidences for Gait Recognition", ICASSP 2003.
http://www.metaverselab.org/pub/paper2/icsp.pdf

**23. [CL00]** R. Collins, A. Lipton, *et al.* A System for Video Surveillance and Monitoring. CMU-RI-TR-00-12, Robotics Institute, CMU, May, 2000.

**24. [CMC02]** Shao-Yi Chien, Shyh-Yih Ma, and Liang-Gee Chen, *Fellow, IEEE* , "Efficient Moving Object Segmentation Algorithm Using Background Subtraction", IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY, VOL. 12, NO. 7, JULY 2002

**25. [CMJ95]** Dmitry Chetverikov , Zoltán Megyesi, Zsolt Jankó
Geometric Modelling and Computer Vision Laboratory, "Image and Pattern Analysis research group ", Geometric Modelling and Computer Vision Laboratory, Computer and Automation Research Institute, Hungarian Academy of Sciences, H-1111 Budapest Kende u.13-17, HUNGARY.

**26. [CNN95]** D. Cunado, J.M. Nash, M.S. Nixon, and J. N. Carter, "Gait extraction and description by evidence- gathering," Proc. of the International Conference on Audio and Video Based Biometric Person Authentication, pp. 43–48, 1995.
http://eprints.ecs.soton.ac.uk/archive/00001948/01/cunadonash.pdf

**27. [CS95]** C_edras, C., and Shah, M. Motion-based recognition: A survey. Image and Vision Computing,13(2):129-155, March 1995.

**28. [DB97]** J. Davis and A. F. Bobick. The representation and recognition of human movement using temporal templates. In *CVPR*, 1997.

**29. [DF01]** Q. Delamarre, O. Faugeras, "3D Articulated Models and Multi-View Tracking with Physical Forces", CVIU Vol. 81, pp328-357, 2001.

**30. [DGH98]** T. Darrell., G. Gordon, and M. Harville. *Integrated person tracking using stereo, color, and pattern detection.* in *CVPR* 1998. Santa Barbara, California. http://www.ai.mit.edu/people/trevor/papers/1998-021/TR-1998-021.pdf

**31. [DN90]** N. Diehl, Object-Oriented Motion Estimation and Segmentation in Image Sequences. IEEE Trans. Image Processing, vol. 3, pp. 1,901-1,904, Feb. 1990.

**32. [EHD]** Ahmad Elgammal, David Harwood, Larry Davis "Non-parametric Model for Background Subtraction", Computer Science Library, University of Maryland, College Park, MD20742

**33. [FL98]** H. Fujiyoshi and A. Lipton. Real-time human motion analysis by image skeletonization. In *Proceedings of the 1998 Workshop on Applications of Computer Vision*, 1998. Robotics Institute, CMU – 66 – VSAM Final Report

**34. [FM94]** Fesharaki, Mehdi N. A robust real-time edge detection algorithm. In *OFOGH The Journal of Computer Science and Engineering.* Vol.1No2.1994.Web, http://www.eleceng.adelaide.edu.au/Personal/habib/OFOGH/v1-n2.html#Vol1-No2-P1.

**35. [FW96]** Bobick, A. F. and Davis, J. W. Real-time recognition of activity using temporal templates. In *Workshop on Applications of Computer Vision,1996.*

**36. [G99]** D.M. Gavrila, "The Visual Analysis of Human Movement: A Survey", Computer Vision and Image Understanding, Vol.73, No.1, pp.82-98, 1999. http://www.cs.jhu.edu/~hager/Public/teaching/CS600.641/human_motion_survey.pdf

**37. [GD95]** D. M. Gavrila and L. Davis, "Towards 3-D Model-based Tracking and Recognition of Human Movement: a Multi-View Approach," in *IEEE International Conference on Automatic Face and Gesture Recognition*, (Zurich, Switzerland), 1995.

**38. [GD96]** D. Gavrila and L.S. Davis, "3D Model-Based Tracking of Humans in Action: A Multi-View Approach," To appear in *CVPR*, 1996, 73–80.

**39. [GW]** Rafael C. Gonzalez, Richard E. Woods, "Digital Image Processing", second version, chapter 12: object recognition p693-732

**40. [HC85]** Glenn Healy and Jorge L. C. Sanz. Contam: An edge-based approach to segmenting images with irregular objects. In CVPR, pages 486-489, 1985.

**41. [HD99]** I. Haritaoglu and L. Davis, "Hydra: Multiple people detection and tracking using silhouettes",*IEEE Workshop on Visual Surveillance*, pp.6-13,1999. http://www.umiacs.umd.edu/users/hismail/Hydra_Outline.htm

**42. [HHD98]** Ismail Haritaoglu, David Harwood and Larry S. Davis, W4: Who? When? Where? What? A real System for detecting and tracking people. 3. International Conference on Face and Gesture Recognition, April 14-16, 1998, Nara , Japan
http://www.umiacs.umd.edu/users/hismail/Publications/fg98W4.pdf
http://www.umiacs.umd.edu/users/hismail/W4_outline.htm

**43. [HKR97]** B. Heisele, U. Kreßel W. Ritter, " Tracking Non-Rigid, Moving Objects Based on Color Cluster Flow " , Daimler-Benz AG, Research and Technology, Pattern Understanding, F3M, P.O. Box 2360, 89013 Ulm, Germany.Proc. IEEE Conf. Computer Vision Pattern Recognition (CVPR '97), pp. 257-260, 1997.

**45. [IB96]** M. Isard and A. Blake. Contour tracking by stochastic propagation of conditional density. In *Proceedings of the 1996 European Conference on Computer Vision*, pages 343–356, 1996.

**46. [IB97]** S. Intille and A. F. Bobick. "Real-time closed-world tracking". *CVPR*, 1997.

**47.** [Janne] Dr. Janne Heikkilä, Department of Electrical Engineering, University of Oulu, Finland. (http://www.ee.oulu.fi/~jth/monitor/Monitor.html)

**48. [IB98]** M. Isard and A. Blake, "Condensation – conditional density propagation for visual tracking," *International Journal of Computer Vision*, vol. 29, no. 1, pp. 5–28, 1998.

**49. [IDB97]** S. Intille, J. Davis, A. Bobick "Real-Time Closed-Word Tracking", In Proc. of CVPR, June 1997

**50. [JB93]** A. Jepson and M.J. Black. Mixture models for optical flow computation. In *Proc. IEEE Conf. Computer VIsion Pattern Recognition*, pages 760–761, New York, 1993.

**51. [JZL96]** A.K. Jain, Y. Zhong, and S. Lakshmanan, Object Matching Using Deformable Templates. IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 18, no. 3, pp. 267–278, Mar. 1996.

**52. [K60]** R. E. Kalman, "A new approach to linear filtering and prediction problems," *Transactions of the ASME–Journal of Basic Engineering*, vol. 82, no. Series D, pp. 35–45, 1960.

**53. [KCC03]** A. Kale, A. K. Roy Chowdhury and R. Chellappa, "Towards a View Invariant Gait Recognition Algorithm", AVSS, 2003.

**54. [KCY03]** A. Kale, N. Cuntoor, B Yegnanarayana, A.N Rajagopalan, R. Chellappa, "Gait analysis for human identification", Proceedings of the 3rd International conference on Audio and Video Based Person Authentication, 2003.
http://www.cfar.umd.edu/~kale/avbpa.pdf

**55. [KPT77]** Kauth, R.J, A.P. Pentland, G.S Tomas, "Blob: an unsupervised clustering approach to spatial preprocessing of MSS imagery", XI Int. Symp. of RS of the Environment, Ann Arbor, MI, 1977

**56. [KW94]** D. Koller, J.Weber, and J. Malik. Robust multiple car tracking with occlusion reason. Technicalreport, U.C.Berkeley, 1994. http://sunsite.berkeley.edu/Dienst/Repository/2.0/Body/ncstrl.ucb/CSD-93-780/pdf

**57. [KWT88]** M. Kass, A. Witkin, and D. Terzopoulos, Snakes: Active Contour Models. Int'l J. Computer Vision, vol. 1, pp. 321-332, 1988.

**58. [L]** Jure Leskovec. Detection of Human Bodies using Computer Analysis of a Sequence of Stereo Image English article submitted for "11th European Union Contest for Young Scientists" http://ai.ijs.si/jure/mat/vision99.html

**59. [LB98]** James Little and Jeffrey Boyd. Recognizing people by their gait: The shape of motion. *Videre*, 1(2), 1998. **http://www-mitpress.mit.edu/e-journals/Videre/001/articles/v1n2001.pdf**

**60. [LG02]** L. Lee and W.E.L. Grimson, "Gait analysis for recognition and classification," Proceedings of the IEEE Conference on Face and Gesture Recognition, pp. 155–161, 2002.

**61. [LJH]**Wei Niu, Long Jiao, Dan Han, and Yuan-Fang Wang, "Real-Time Multi-person Tracking in Video Surveillance" . Department of Computer Science, University of California, Santa Barbara, CA 93106 http://www.cs.ucsb.edu/~yfwang/papers/PCM03.pdf

**62. [MKS89]** L. Matthies, T. Kanade, and R. Szeliski. Kalman filter based algorithms for estimating depth from image sequences. *IJCV*, 3(3):209-236, 1989.

**63. [MT93]** D. Metaxas and D. Terzopoulos, Shape and Nonrigid Motion Estimation Through Physics-Based Synthesis. IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 15, no. 6, pp. 580-591, 1993.

**64. [NSK94]** H.H. Nagel, G. Socher, H. Kollnig, and M. Otte, Motion Boundary Detection in Image Sequences by Local Stochastic Tests. Proc. European Conf. Computer Vision, vol. II, pp. 305-315, 1994.

**65. [OG99]** E. Ong, and S. Gong, "A Dynamic Human Model Using Hybrid 2D-3D Representations in Hierarchical PCA Space", BMVC99. http://www.bmva.ac.uk/bmvc/1999/papers/04.pdf

**66. [PD00]** N. Paragios, R. Deriche. "Geodesic Active Contours and Level Sets for the Detection and Tracking of Moving Objects", PAMI, Vol. 22, No. 3, March 2000.

**67. [PN94]** R. Polona, R. Nelson "Low Level Recognition of Human Motion", In Proc. Non Rigid Motion Workshop, November 1994.

**68. [PT99]** N. Paragios and G. Tziritas, Adaptive Detection and Localization of Moving Objects in Image Sequences. Signal Processing: Image Comm., vol. 14, pp. 277-296, 1999

**69. [QPK98]** Georges M. Quénot, Jaroslaw Pakleza, Tomasz A. Kowalewski ,"Particle image velocimetry using optical flow for image analysis", 8TH INTERNATIONAL SYMPOSIUM ON FLOWVISUALISATION (1998)

**70. [R94]** K. Rohr. Towards model-based recognition of human movements in image sequences. *CVGIP:IU*, 59(1):94-115, 1994.

**71. [R97]** LEE ROSSEY, "Design and Implementation of Opinion-Based Behaviors for an Autonomous Mobile Robot with Vision", Master Thesis, 1997 UNIVERSITY OF FLORIDA. http://walden.mvp.net/~mminnis/thesis.pdf

**72. [RE95]** P. L. Rosin, T. Ellis, "Image difference threshold strategies and shadow detection," *Proceedings of the 6th British Machine Vision Conference, Birmingham, UK, September*, pp.347-356, 1995.

**73. [RS98]** R´omer Rosales and Stan Sclaroff, http://www.cs.bu.edu/techreports/pdf/1998-007-tracking-multiple-humans.pdf Improved Tracking of Multiple Humans with Trajectory prediction and Occlusion Modeling. To appear in Proceedings **IEEE Conf. on Computer Vision and Pattern Recognition. Workshop on the Interpretation of Visual Motion**, Santa Barbara, CA, 1998.

**74. [RS99]** R´omer Rosales and Stan Sclaroff, Computer Science Department, Boston University. 3D trajectory Recovery for tracking multiple objects and trajectory guided recognition of actions http://www.cs.bu.edu/techreports/pdf/1998-019-3D-trajectory-guided-action-recog.pdf. BU CS TR98-019 rev. 2. To appear in **Proc. IEEE Conf. on Computer Vision and Pattern Recognition**, June 1999_

**75. [RW89]** Yoav Rosenberg , Michael Werman, "Object tracking with a moving camera". In Proc. IEEE Workshop on Visual Motion, pages 2-12, 1989.

**76. [SA]** Koichi Sato and J. K. Aggarwal, Department of Electrical and Computer Engineering, The University of Texas at Austin, Austin, TX 78712, USA http://www.ece.utexas.edu/projects/cvrc/koichisato.pdf Recognizing two person interaction

**77. [SFP00]** Y. Song, X. Feng, and P. Perona, "Towards Detection of Human Motion," in *Proceedings of the Computer Vision and Pattern Recognition*, 2000.

**78. [SG99]** C. Stauffer, W. E. L. Grimson, "Adaptive background mixture models for realtime tracking," *Proc. IEEE Computer Vision and Pattern Recognition*, (Fort Collins, Colorado), June 23-25, 1999. http://www.ai.mit.edu/projects/vsam/Publications/stauffer_cvpr98_track.pdf

**79. [SPT]** L. A. Sechidis, P. Patias, V. Tsioukas, " LOW-LEVEL TRACKING OF MULTIPLE OBJECTS" The Aristotle University of Thessaloniki, Department of Cadastre Photogrammetry and Cartography, Univ. Box 473, GR-54006, Thessaloniki, Greece

**80. [ST94]** J. Shi and C. Tomasi, "Good Features to Track ", 1994 IEEE Conference on Computer Vision and Pattern Recognition (CVPR'94), 1994, pp. 593 – 600

**81.[T87]** R. Y. Tsai. A versatile camera calibration technique for high accuracy 3D machine. vision metrology using o®-the-shelf TV cameras and lenses. *IEEE Transaction. on Robotics and Automat*, 3(4):323C344, 1987.

**82. [VSB02]** Rene Visser, Nicu Sebe, and Erwin Bakker
http://carol.science.uva.nl/~nicu/publications/CIVR02_visser.pdf
Object Recognition for Video Retrieva

**83. [w]** Ying Wu , Visual Tracking
http://www.ece.northwestern.edu/~yingwu/teaching/ECE510/Notes/tracking.pdf

**84. [WAD97]**Christopher Wren, Ali.A, Trevor. D
pfinder: real-time tracking of the human body. Published in IEEE Transaction on Pattern Analysis and machine Intelligence July 1997,vol 19 pp760-765

**85. [WLB91]** J.Wang, G. Lorette, and P. Bouthemy. Analysis of human motion: A model-based approach. In *7th Scandinavian Conf. Image Analysis*, Aalborg, Denmark, 1991.

**86. [WPR02]** J. Ben-Arie, Z. Wang, P. Pandit, and S. Rajaram, "Human Activity Recognition Using Multidimensional Indexing", PAMI, Vol. 24, No. 8,August 2002.

**87. [WTN]** Liang Wang, Tieniu Tan, Huangzhong Ning, and Weiming Hu, Fusion of Static and Dynamic Body Biometrics for Gait Recognition, *IEEE Transactions on Circuits and Systems for Video Technology Special Issue on Image- and Video-Based Biometrics*.

**88. [WW]** Shu-Fai Wong and Kwan-Yee Kenneth Wong, "Real time human body tracking using Wavenet"

**89. [YL92]**Y.H. Yang and M.D. Levine. The background primal sketch: An approach for tracking moving objects. *Machine Vision Applic.*, 5:17–34, 1992.

**90. [YXC97]** J. Yang, Y. Xu, and C. S. Chen, "Human action learning via hidden Markov model", *IEEE Transactions on Systems, Man and Cybernetics*, vol. A, pp.34-44, 1997.

**91. [Z00]** Zhengyou Zhang. A flexible new technique for camera calibration. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(11):1330–1334, 2000.

# Vita Auctoris

NAME: Bo Shen

EDUCATION: University of Windsor, Windsor, ON, Canada
2002-2005 M .Sc.

University of Windsor, Windsor, ON, Canada
2000-2002 B .Sc.

DoHua University, ShangHai, China
1986-1990 B. Eng.