Tracking a Person with Pre-recorded Image Database and a Pan, Tilt, and Zoom Camera *

Yiming Ye[1], John K. Tsotsos[2], Eric Harley[2], Karen Bennet[3]
[1]IBM T.J Watson Research Center
P.O. Box 704,Yorktown Heights, N.Y. 10598
Phone: (914) 784-7460; Fax: (914) 784-7455; Email: yiming@watson.ibm.com
[2]Department of Computer Science
University of Toronto, Toronto, Canada, M5S 1A4
[3]IBM Canada Center for Advanced Studies
North York, Ontario, Canada, M3C 1H7

Abstract

This paper proposes a novel tracking strategy that can robustly track a person or other object within a fixed environment using a pan, tilt, and zoom camera with the help of a pre-recorded image database. We define a set of camera states which is sufficient to survey the environment for the target. Background images for these camera states are stored as an image database. During tracking camera movements are restricted to these states. Tracking and segmentation are simplified as each tracking image can be compared with the corresponding pre-recorded background image.

Keywords: Tracking, Sensor Planning, Image database,Image difference, Segmentation

1 Introduction

This paper approaches the task of tracking people in a fixed environment by actively controlling a pan, tilt, and zoom camera while referring to a pre-recorded image database of the environment. Visual tracking of objects moving in three-dimensions has received considerable attention in the computer vision community over the past few years [5, 6, 7, 10, 12, 13, 14, 15, 16, 19, 21, 27, 35, 36, 38]. The task is a challenging one because it not only involves the difficulties of segmenting the target from various backgrounds, but also the analysis and prediction of target motion.

The goal of most previous work in this area is to track a known object by its features projected in the image. Such features are usually points or lines which are recognized on the basis of a model of the target. For example, several vision based tracking systems [9, 24, 26] study gesture classification using finger edge features or even complete 3D models of the hand [18]. Goncalves *et al.* [11] model 3D motion of the arm using two truncated right-circular cones connected with spherical joints. Baumberg and Hogg [2] use a flexible shape model to track the silhouette of a moving human despite the changing outline.

These approaches typically involve a loop of prediction, projection, measurement, and adjustment. The prediction step generates the next three-dimensional object pose based on current

[*] Published by *Machine Vision and Applications (2000) 12: 32-43*

estimates of position and velocity. The projection step attempts to project the target onto a two-dimensional image based on the prediction. The measurement step uses information regarding the features of the projected image to direct the search for corresponding features in the real image. The adjustment step combines information from measurement and prediction steps to produce new values of the target pose and velocities, and these new values are used for the next prediction.

Recent work also examines the task of visually tracking non-rigid objects [7, 10, 12, 13, 14, 15, 16, 19, 27, 35, 36]. For example, Huttenlocher [13] constructed a system for tracking a non-rigid object moving in a complex scene. The method extracts two-dimensional models of a moving object from a sequence of images by factoring out motion and shape change. Darrell *et. al.* [7] have implemented vision routines to track people and to identify head/hand locations as they walk about in a room. Foveation cues guide an active camera to follow head or hands. The system assumes a fixed background and that the person is facing the camera. Gavrila *et. al.* [10] construct a system to track unconstrained human movement using a three-dimensional model. Image sequences acquired simultaneously from multiple views allows them to recover the three-dimensional body pose at each time instant without the use of markers. Kakadiaris *et. al.* [15] present a method to mitigate the difficulties arising due to occlusion among body parts by employing multiple calibrated cameras in a mutually orthogonal configuration. Rossi and Bozzoli [28] avoid problems of occlusion by using a vertically mounted camera to track and count people moving in and out of a scene at a specified entrance or exit.

Crowley et. al. [6] describe a system which uses multiple visual processes to detect and track faces for video compression and transmission. Visual processes for face tracking are described using blink detection, normalized color histogram matching, and cross correlation. Fusion of results into a unified estimation for tracking is made possible by estimating a covariance matrix with each observation. The result of face detection is fed into a recursive estimater, and the output from the estimator is used to control a pan/tilt/zoom camera.

In this paper, we deviate from the above-mentioned schemes in that we make no assumptions regarding the features of the object. The goal here is to investigate what can be achieved through sensor planning alone. Active control of the camera is a form of *sensor planning* advocated in [1] and analyzed in [33]. The task of sensor planning, while receiving little attention in the past, is very important during tracking because the camera's state parameters determine the quality of the resulting image and indeed whether the target will be within the image. Demonstrations of the efficacy of planned camera motion in object recognition and tracking can be found in [8] and [20, 22, 23, 30], respectively. Therefore, the work reported in this paper focuses almost exclusively on sensor planning — how to control the camera to perform the tracking task, given a target recognition algorithm. We propose the concept of a detection function to evaluate the performance of given recognition algorithms. The detection function will be used to help select the state parameters of the camera during the tracking process.

One of the difficulties faced by previous visual tracking strategies is the complexity of segmentation and recognition tasks required for the measurement and adjustment stages. To successfully track the target, the system must be able to distinguish the target from the background. This can be very difficult when the target does not have distinctive features or when the background is complex. The point or line features used by most systems can lead to instability in the recognition algorithm if the background has many similar features. We attempt to overcome this problem through the use of a pre-recorded image database.

There are four aspects of our approach: camera states selection by detection function, background image database generation, target tracking with selected camera states, and target detection with image differences. The first is to use the detection function to select a set of camera states (pan, tilt, and zoom parameters) such that wherever the target appears in the surveillance region, there exists at least one camera state for which the target is in the field of view and its image is of sufficient quality. We then take background images with these camera states and store the images in an image database. During tracking, the camera is restricted to these camera parameter settings, as they are the best for target detection and recognition. Segmentation is through simple image differencing. The target is defined to be the collection of blobs of sufficient size in the difference image. Our strategy is relatively stable in the face of increasing background complexity and can be used effectively in tracking tasks where the identity of the moving target is not an issue. For example, it can be used in visual surveillance to track an intruder moving about the environment, or in animal behavioral studies to remotely photograph animals at a water hole or den. In conjunction with more sophisticated segmentation strategies, the method of determining a minimal set of camera states and employing a pre-recorded background image database over this set of states may improve tracking in a wide variety of applications. This paper presents the tracking algorithm and a simple experiment to illustrate the concepts.

2 The Detection Function

In this section we introduce a *detection function* which specifies the ability of the recognition algorithm to detect the target, averaged over various factors and conditions that affect its performance. The detection function $\mathbf{b}(\langle w, h\rangle, \langle \theta, \delta, l\rangle)$ gives the probability of detecting the target by the given recognition algorithm when the camera's viewing angle size is $\langle w, h\rangle$ and the relative position of the target to the camera is $\langle \theta, \delta, l\rangle$; where $\theta = arctan(\frac{x}{z})$, $\delta = arctan(\frac{y}{z})$, $l = z$, and (x, y, z) are the coordinates of the target center in camera coordinate system. The value of $\mathbf{b}(\langle w, h\rangle, \langle \theta, \delta, l\rangle)$ can be obtained empirically: the target is placed at (θ, δ, l) and experiments are performed under various conditions, such as light intensity, background situation, and the relative orientation of the target with respect to the camera center. The value of $\mathbf{b}$ is given by the number of successful recognitions divided by the total number of experiments.

It is not necessary to record the detection function values of all the different camera viewing angle sizes. We only need the detection values of one camera angle size (we call it the reference angle), and those of the other camera angle sizes can be obtained approximately by transforming them into those of the known camera angle size, as follows. Suppose we know the detection function values for viewing angle size $\langle w_0, h_0\rangle$. We want to find the detection function values for viewing angle size $<w, h>$. To get the value of $\mathbf{b}(\langle w, h\rangle, \langle \theta, \delta, l\rangle)$ for a given $\langle \theta, \delta, l\rangle$, we need to find values of $\langle \theta_0,\ \delta_0,\ l_0\rangle$ for angle size $\langle w_0, h_0\rangle$ that make the following approximation true:

$$\mathbf{b}(\langle w, h\rangle, \langle \theta, \delta, l\rangle) \approx \mathbf{b}(\langle w_0, h_0\rangle, \langle \theta_0, \delta_0, l_0\rangle). \tag{1}$$

The approximation relation (Formula (1)) means that when we use the recognition algorithm to analyze the picture taken with parameters $(\langle w, h\rangle,\ \langle \theta, \delta, l\rangle)$ and the picture taken with parameters $(\langle w_0, h_0\rangle, \langle \theta_0, \delta_0, l_0\rangle)$, we should get almost the same result. To guarantee this, the images of the target object should be almost the same in both cases, i.e., they must be approximately equal in at least two geometric factors, namely the scale factor and the position factor. The scale factor refers to the size of the projection of the target object on the image plane. The position factor refers to the position on the image plane of the projection of the center of the target object. Typically, the position factor has much less influence than the scale factor.

We use the scale factor to find the value of l_0 when l is given. The sizes of the projection of the target object on the image planes for $(\langle w,h\rangle,\langle\theta,\delta,l\rangle)$ and $(\langle w_0,h_0\rangle,\langle\theta_0,\delta_0,l_0\rangle)$ are approximately determined by l and l_0, respectively. Equality of the scale factors means that for a target patch that is parallel to the image plane, the area of its projection on the image plane for $(\langle w,h\rangle,\langle\theta,\delta,l\rangle)$ should be same as the area of its projection on the image plane for $(\langle w_0,h_0\rangle,\langle\theta_0,\delta_0,l_0\rangle)$. Let W and H denote the width and height of the image plane, respectively. Since the size of the image plane remains constant for different focal lengths, W and H will be same for any focal length. (We assume here that the image plane and the focal plane of the camera are always coincident).

Let S be the area of the target patch. Let $S^{'}$ be the area of the projected target image for the desired arguments $(\langle w,h\rangle,\langle\theta,\delta,l\rangle)$, and $S_0^{'}$ be the area of the projected target image for the reference arguments $(\langle w_0,h_0\rangle,\langle\theta_0,\delta_0,l_0\rangle)$. From the similarity relation between the target patch and its projected image, it is easy to show that

$$S^{'}=\frac{f^2}{l^2}S. \tag{2}$$

Since

$$tan(\frac{w}{2})=\frac{\frac{W}{2}}{f},$$

and

$$tan(\frac{h}{2})=\frac{\frac{H}{2}}{f},$$

we have

$$S^{'}=\frac{f^2}{l^2}S=\frac{WH}{4l^2tan(\frac{w}{2})tan(\frac{h}{2})}S. \tag{3}$$

Similarly,

$$S_0^{'}=\frac{WH}{4l_0^2tan(\frac{w_0}{2})tan(\frac{h_0}{2})}S. \tag{4}$$

To guarantee $S^{'}=S_0^{'}$, we get:

$$l_0=l\sqrt{\frac{tan(\frac{w}{2})tan(\frac{h}{2})}{tan(\frac{w_0}{2})tan(\frac{h_0}{2})}}. \tag{5}$$

We use the position factor to find the values of θ_0,δ_0 when θ and δ are given. Let D denote the center of target patch with respect to $(\langle w,h\rangle,\langle\theta,\delta,l\rangle)$, and let $D^{'}(x^{'},y^{'},z^{'})$ denote the image of D on the image plane with respect to $(\langle w,h\rangle,\langle\theta,\delta,l\rangle)$. Similarly, let D_0 be the center of target patch with respect to $(\langle w_0,h_0\rangle,\langle\theta_0,\delta_0,l_0\rangle)$, and let $D_0^{'}(x_0^{'},y_0^{'},z_0^{'})$ represent the image of D_0 on the image plane with respect $(\langle w_0,h_0\rangle,\langle\theta_0,\delta_0,l_0\rangle)$, where $(x^{'},y^{'},z^{'})$ and $(x_0^{'},y_0^{'},z_0^{'})$ are in camera coordinate system. Then we have

$$x^{'}=ftan(\theta)=\frac{\frac{W}{2}}{tan(\frac{w}{2})}tan(\theta)=\frac{W}{2}\frac{tan(\theta)}{tan(\frac{w}{2})}. \tag{6}$$

Similarly,

$$y^{'}=\frac{H}{2}\frac{tan(\delta)}{tan(\frac{h}{2})}, \tag{7}$$

$$x_0^{'} = \frac{W}{2}\frac{tan(\theta_0)}{tan(\frac{w_0}{2})}, \tag{8}$$

and

$$y_0^{'} = \frac{H}{2}\frac{tan(\delta_0)}{tan(\frac{h_0}{2})}. \tag{9}$$

To guarantee $x^{'} = x_0^{'}$ and $y^{'} = y_0^{'}$, we get

$$\theta_0 = arctan[tan(\theta)\frac{tan(\frac{w_0}{2})}{tan(\frac{w}{2})}], \tag{10}$$

and

$$\delta_0 = arctan[tan(\delta)\frac{tan(\frac{h_0}{2})}{tan(\frac{h}{2})}]. \tag{11}$$

Therefore, when we want to find the detection function value for parameters $\langle\theta, \delta, l\rangle$ with respect to the camera angle size $\langle w, h\rangle$, we first find the corresponding $\langle\theta_0, \delta_0, l_0\rangle$, and then retrieve the detection function value for $\mathbf{b}(\langle w_0, h_0\rangle, \langle\theta_0, \delta_0, l_0\rangle)$ from the look up table or from the analytical formula.

In the detection function $\mathbf{b}(\langle w, h\rangle, \langle\theta, \delta, l\rangle)$ the position of the target with respect to the camera is fixed. The variations in the target orientation and the background are taken as a probabilistic factor which generate the value of the detection function. In general, the orientation of the target determines which aspect of the target is facing the camera. The appearance of the aspect has a great influence on the recognition result. Suppose the target has m different aspects, $a_1 \ldots a_m$. We define $\mathbf{b}^a(\langle w, h\rangle, \langle\theta, \delta, l\rangle)$ to be the probability of detecting the target when the camera's viewing angle size is $\langle w, h\rangle$, the relative position of the target is $\langle\theta, \delta, l\rangle$, and the aspect of the target facing the camera is a ($a \in \{a_1, \ldots, a_m\}$).

3 Minimum set of camera states

In this section we show how to choose a set a camera states such that wherever the target is in the given environment, at least one of the camera states puts the target into the field of view with good image quality. We begin by explaining the concept of effective volume with respect to a given target aspect a.

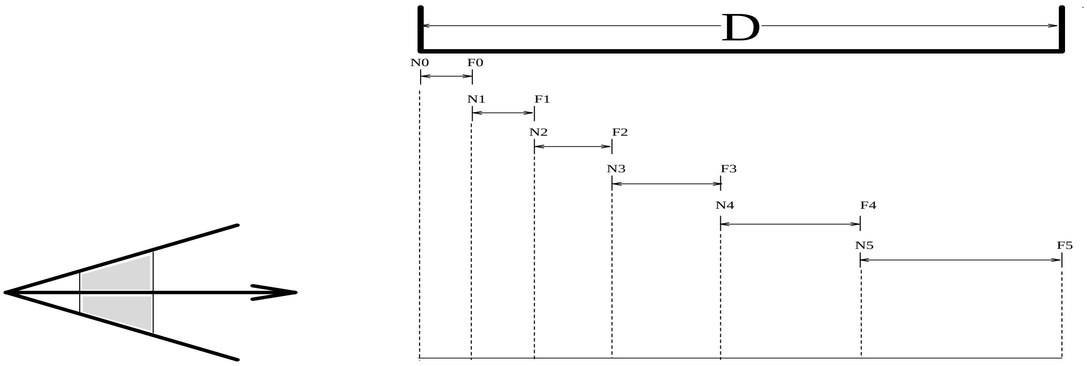
3.1 Effective Volume

For a given camera viewing angle size $\langle w, h\rangle$, the ability of the recognition algorithm and the value of the detection function $\mathbf{b}^a(\langle w, h\rangle, \langle\theta, \delta, l\rangle)$ are influenced by the parameters $\langle\theta, \delta, l\rangle$ and by the background. Since we can not predict the background, we only consider the influences of θ, δ, and l. Angles θ and δ determine the position of the projected target on the image. It is well known that the target image position has no or very little influence on the recognition results. Thus, we omit the influence of θ, δ and only consider the influence of l. Usually the recognizer can successfully recognize the target only when the image size of the target is such that the whole target can be brought into the field of view of the camera and the features can be detected with the required precision. For a given recognition algorithm, a fixed viewing angle size, and a given target aspect, the probability of successfully recognizing the target is high only when the target's distance is within a certain range. Therefore, different sizes of the viewing angle $\langle w, h\rangle$ will be associated with different **effective ranges** of distance l.

3.2 Selection of Camera Angle Size

Let D be the maximum distance from the camera center to any point in the environment. Our purpose here is to select camera angle sizes such that their effective ranges will cover the entire depth D of the environment without overlap.

Suppose that the biggest viewing angle for the camera is $\langle w_0, h_0\rangle$, and its effective range for the given aspect is $[N_0, F_0]$. We can use geometric constraints to find other required viewing angle sizes $\langle w_1, h_1\rangle$, ..., $\langle w_{n_0}, h_{n_0}\rangle$ and their corresponding effective ranges $[N_1, F_1], \ldots, [N_{n_0}, F_{n_0}]$, such that $[N_0, F_0]\bigcup\ldots\bigcup[N_{n_0}, F_{n_0}] \supseteq [N_0, D]$, and $[N_i, F_i)\bigcap[N_j, F_j) = \emptyset$ for $i \neq j$. These $n_0 + 1$ angle sizes are enough to examine the whole depth of the environment with high probability. Figure 1 illustrates the above idea in two-dimensions.



(a) (b)

Figure 1: Schematic showing selection of the camera angle size in two dimensions. (a) The effective range for a given angle size. (b) The viewing angle sizes should be selected such that their effective ranges can cover the entire depth D without overlap.

The effective range $[N_{i+1}, F_{i+1}]$ of the next viewing angle $\langle w_{i+1}, h_{i+1}\rangle$ should be adjacent to the effective range of the current viewing angle $\langle w_i, h_i\rangle$, i.e., $N_{i+1} = F_i$. To guarantee that the areas of the images of the target patch of $\langle w_i, h_i\rangle$ at N_i and F_i are equal to the areas of the images of the target patch of $\langle w_{i+1}, h_{i+1}\rangle$ at N_{i+1} and F_{i+1}, respectively, we obtain (using Equations (5)):

$$w_i = 2arctan[(\frac{N_0}{F_0})^i tan(\frac{w_0}{2})] \tag{12}$$

$$h_i = 2arctan[(\frac{N_0}{F_0})^i tan(\frac{h_0}{2})] \tag{13}$$

$$N_i = F_0(\frac{F_0}{N_0})^{i-1}; F_i = F_0(\frac{F_0}{N_0})^i \tag{14}$$

Since $N_i \leq D$, we obtain $i \leq \frac{ln(\frac{D}{F_0})}{ln(\frac{F_0}{N_0})} - 1$. Let $n_0 = \lfloor\frac{ln(\frac{D}{F_0})}{ln(\frac{F_0}{N_0})} - 1\rfloor$, then the angles that are needed to cover the whole tracking environment for the given aspect are $\langle w_0, h_0\rangle$, $\langle w_1, h_1\rangle$, ..., $\langle w_{n_0}, h_{n_0}\rangle$.

From the above discussion, we know that if we can find the first effective range with respect to the biggest camera angle size, then we can find other effective ranges. The segmentation strategy of our approach calculates the image difference of the background image and the real image and detects changing blobs. If the total sum of the areas of changing blobs is within a certain range, then a person is detected. To find the value of N_0, we can put the target at various distances d and calculate values of the detection function. There will be a segment of d with good detection function values. The biggest d with a good enough detection function value will be the value of F_0. The smallest d with a good enough detection function value will be the value of N_0.

3.3 Select Camera Viewing Direction

The effective ranges of $\langle w_0, h_0 \rangle$, $\langle w_1, h_1 \rangle$, ..., $\langle w_{n_0}, h_{n_0} \rangle$ divide the space around the camera center into a layered sphere. Each layer can be successfully examined by the corresponding effective angle. For a given effective angle size $\langle w, h \rangle$, there are a huge number of viewing directions that can be considered. Each direction $\langle p, t \rangle$ (pan, tilt) corresponds to a rectangular pyramid, which is the viewing volume determined by parameters $\langle w, h, p, t \rangle$. Within this viewing volume, only a slice of the pyramid can be examined with high detection probability by the given recognition algorithm. This slice of the pyramid is the **effective volume** for camera state $\langle w, h, p, t \rangle$. The union of the effective volumes of all the possible $\langle p, t \rangle$ given $\langle w, h \rangle$ will cover the given layer. To examine this layer, it is not necessary to try every possible $\langle p, t \rangle$ one by one — we only need to consider those directions such that the union of their effective volumes cover the whole layer with little overlap. This idea is illustrated in Figure (2).

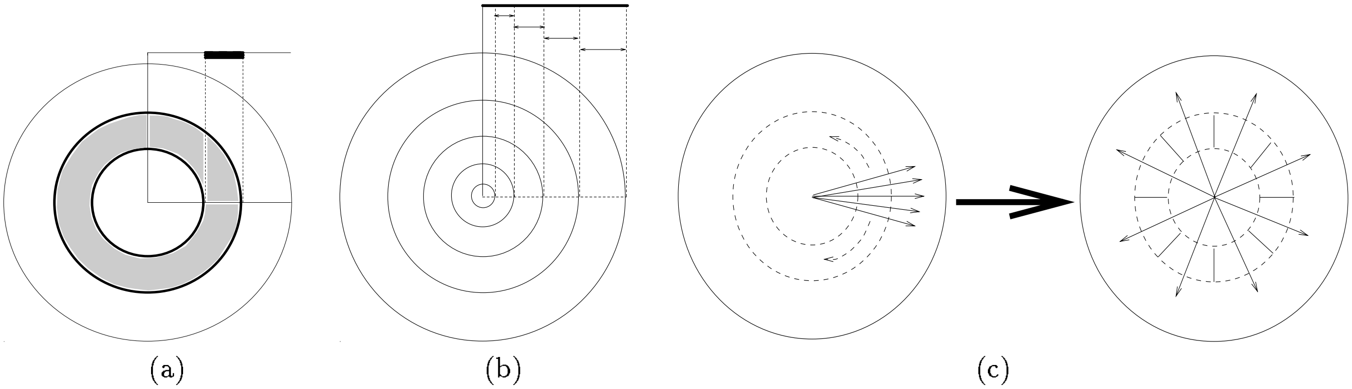


Figure 2: (a) Each viewing angle size can check a layer in the space. (b) The selected camera viewing angle sizes divide the space into a layered sphere. (c) There are a huge number of possible directions to examine a given layer, but among them, only a limited number of actions are necessary.

Let $\alpha = min\{w, h\}$, and the final viewing direction set be $S_{candidate}$. Then the following algorithm gives the necessary viewing directions to cover the whole sphere (see [39] for details).

1. $S_{candidate} = \phi$

2. $p \longleftarrow 0$, $t \longleftarrow 0$, $S_{candidate} = S_{candidate} \bigcup < p, t >$.

3. $t_e \longleftarrow \frac{\pi}{2}$

4. $t_b \longleftarrow arccos\{\frac{cos((t_e - \frac{\alpha}{2}) - \frac{\alpha}{2})}{\sqrt{1 + \frac{sin^2(\frac{\alpha}{2})\, sin^2((t_e - \frac{\alpha}{2}) - \frac{\alpha}{2})}{sin^2((t_e - \frac{\alpha}{2}) + \frac{\alpha}{2})}}}\}$

5. Cover the slice on the sphere whose tilt is within the range of $[t_b, t_e]$ and the slice on the sphere whose tilt is within the range of $[\pi - t_e, \pi - t_b]$.

   (a) Let $t \longleftarrow t_e - \frac{\alpha}{2}$.

   (b) let $\Delta_{pan} \longleftarrow 2arctan\{\frac{sin(\frac{\alpha}{2})}{sin((t - \frac{\alpha}{2}) + \frac{\alpha}{2})}\}$

   (c) Use Δ_{pan} to divide $[0, 2\pi]$ for the given slide. So, we obtain a series of $[p_b, p_e]$, *viz.*, $[0, \Delta_{pan}]$, $[\Delta_{pan}, 2\Delta_{pan}]$, ..., $[k\Delta_{pan}, 2\pi]$. Note: the length of the last interval may not be Δ_{pan}.

   (d) For each division, let $p \longleftarrow \frac{p_b + p_e}{2}$. Then perform $S_{candidate} = S_{candidate} \bigcup < p, t >$ and $S_{candidate} = S_{candidate} \bigcup < p, \pi - t >$.

6. Let $t_e \longleftarrow t_b$

7. If $t_e \leq \alpha$, stop the process. Otherwise Goto 4.

In the above algorithm, Step 2 gives the first candidate direction. Step 3 and Step 4 gives beginning value and the end value of tilt for the first slice that is going to be covered. Step 6 and Step 5 gives the beginning value and the end value of tilt for the next slice that is going to be covered. Step 5 selects viewing directions to cover the selected slice and the corresponding symmetric slice on the sphere. Step 5(d) adds the selected directions into $S_{candidate}$. The view volumes of the selected directions have a little overlap among them.

3.4 Effective Volumes for a Given Aspect

For a given aspect a_i, we can find a set of n_i effective viewing angle sizes $\langle w_{i,1}, h_{i,1} \rangle$, ..., $\langle w_{i,n_i}, h_{i,n_i} \rangle$, that can examine the whole depth of the environment. For each angle size $\langle w_{i,j}, h_{i,j} \rangle$, we can find a set of $n_{i,j}$ viewing directions $\langle p_{i,j,1}, t_{i,j,1} \rangle$, ..., $\langle p_{i,j,n_{i,j}}, t_{i,j,n_{i,j}} \rangle$. Each triple $V_{ijk} =< a_i; \langle w_{i,j}, h_{i,j} \rangle; \langle p_{i,j,k}, t_{i,j,k} \rangle >$ determines an effective volume. When the aspect of the target facing the camera is a_i, and the target to be tracked is within V_{ijk}, and when the camera state is $\langle w_{i,j}, h_{i,j} \rangle$ and $\langle p_{i,j,k}, t_{i,j,k} \rangle$, then the given recognition algorithm can be expected to detect the target. Let $V_i = \sum_{j=1}^{n_i} \sum_{k=1}^{n_{i,j}} V_{ijk}$, then V_i is the union of all the effective volumes for aspect a_i. If the target is within the environment, then there exists at least one effective volume $V_{ijk} \in V_i$ that contain the target. The corresponding camera parameters $\langle w_{i,j}, h_{i,j} \rangle$ and $\langle p_{i,j,k}, t_{i,j,k} \rangle$ determine the camera's state by which we can detect the target.

3.5 Minimum Camera Parameter Settings

Suppose the target has N aspects. For each aspect a_i there exists a set of effective volumes V_i. Let $V = \sum_{i=1}^{N} V_i$. Then V is the set of the effective volumes for the target. This means that no matter where the target might appear in the environment or what aspect the target presents to the camera, there exists at least one effective volume that contains the target. The corresponding camera parameters with respect to this effective volume will detect the target.

When discussing the effective volume, we assumed that the camera must cover the entire spherical region surrounding it. In most situations, however, the environment is only a portion of the sphere with radius equal to the depth of the environment. Let Ω denote the region occupied by the environment. Then during the tracking process, we only need to consider those effective volumes V_{ijk} that have common regions with Ω.

Thus, we define the Minimum Camera Parameter Settings ($MCPS$) as:

$$\begin{aligned} MCPS = \{\langle w_{i,j}, h_{i,j}, p_{i,j,k}, t_{i,j,k}\rangle \quad | \quad & 1 \leq i \leq N, 1 \leq j \leq n_i, 1 \leq k \leq n_{i,j}, \\ & V_{ijk} \cap \Omega \neq \emptyset\} \end{aligned} \tag{15}$$

$MCPS$ is a relatively small set of camera settings needed to track the target within the environment. The effective volumes of these states cover the entire sphere around the camera to some depth D. Wherever the target may appear in this spherical environment, there exists at least one camera state in this set appropriate for high probability of target detection.

The MCPS is particularly useful for efficient scanning of the environment in search of the target, since it defines a minimum set of camera movements that suffice for effective surveillance. Smoothness of tracking, however, can be improved by selecting additional camera states to supplement the minimum set, thus creating the Camera Parameter Settings for Tracking (CPST).

4 Segmentation

In order to detect and track a target, we must be able to segment it from the background of the image. Generally this is a very difficult task. Our strategy here is to alleviate some of the difficulties of segmentation by using the camera states of MCPS to create a database of images, IDB_{MCPS}, of the environment without the target present, and then during tracking to use these camera states and the corresponding background images for comparison when segmenting for the target. This strategy should improve the efficiency and accuracy of segmentation. We illustrate the concept using the extremely simple segmentation strategy: *calculate the difference between the tracking image and the corresponding database image, and interpret any significant difference as target.* Presumably, more discriminating segmentation routines could also benefit from sensor planning and an image database.

Details of the difference calculation in this segmentation method are described with reference to the example in Fig. 3. Image (a) is from the image database, and image (b) is taken with the same camera state, but during tracking, after the appearance of a person. Image (c) is the color difference image (b-a) calculated as follows. The color intensity (r, g, b) of a pixel at position (x, y) in (b) is compared with the intensity (r', g', b') at (x', y') in (a), where $|x - x^{'}| \leq n$ and $|y - y^{'}| \leq n$. The value of constant n is chosen to compensate for errors in camera movement and depends on camera angle size. In the Appendix we calculate an upper limit on the value of n required to compensate for an errors in the camera viewing direction and viewing angle size. The pixel intensity in the color difference image for the position (x, y) is defined to be the triple $(|r - r^{'}|, |g - g^{'}|, |b - b^{'}|)$ whose 2-norm is minimum.

Image (d) in Fig. 3 is the binary difference image obtained by converting (r, g, b) intensities first to grey intensities in the range 0 to 255, and then to black/white intensities of 0 or 255 according to a threshold (40 in this case). Some small white areas are noise, and larger white areas are target. To reduce noise, we apply standard erosion and dilation operations. Blobs are then detected as groups of connected white pixels, and blobs of size $m_i > 1000$ pixels are considered to be target. Image (e) is the same as (c), but with hash marks superimposed marking the average (x_i, y_i) pixel coordinates of target blobs. Here the algorithm found five blobs of significant size, which are

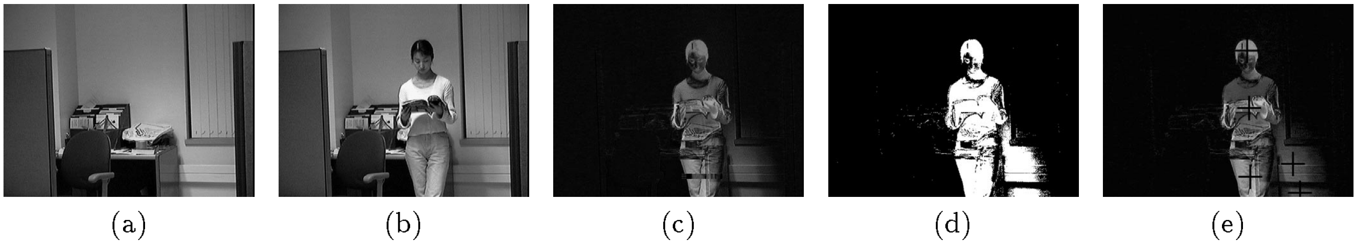


Figure 3: Image Segmentation and Recognition Algorithm

assumed to represent the human. The features of the target are represented by the total mass $M = \Sigma m_i$ and the mass-averaged position of the blobs, given by $X = \Sigma m_i x_i / M, Y = \Sigma m_i y_i / M$, where the summation is over the blobs of sufficient size.

This segmentation algorithm, although extremely simple, can successfully detect the human body, because the colors and shape of the hair, face, clothes, and other features of the human, contrast well with most backgrounds. Unfortunately the person's shadow may also be interpreted as part of the target, (cf. Fig. 3(g)), but generally this does not greatly influence the calculated mass and position of the target. In any case, a more sophisticated segmentation method can easily be substituted in this framework of tracking with an MCPS and IDB.

5 Tracking

Our tracking algorithm uses the set of camera states MCPS and the corresponding Image Database IDB_{MCPS} while continuously iterating the following four steps:

1. Choose the next camera state $\langle w, h, p, t \rangle$ based on information obtained from the previous image, such as the target position X, Y and mass M.

2. Take an image $I^*_{\langle w,h,p,t \rangle}$.

3. Attempt to segment target from background in the image $I^*_{\langle w,h,p,t \rangle}$ with reference to the corresponding image $I_{\langle w,h,p,t \rangle}$ in IDB_{MCPS}.

4. If the target is detected then calculate its position and mass.

Step 1 is performed by the **Where to Look Next** routine. When there is no information regarding the whereabouts of the target, as is the case initially or later if tracking fails, then the routine simply cycles through the states of MCPS. If the target was recently in the field of view and has now moved out, then the routine uses the last known position and orientation to guess a set of next possible positions and orientations.

Recall that the space around the camera is tessellated into layers of wedge-shaped cells, each effectively covered by a particular camera state. (This may also be done for several significantly different target aspects). We assume that images can be processed quickly enough that the target stays in any one cell long enough for the taking and processing of several images. In this case, if the target moves out of view, then it can be found in one of the adjacent cells, called the **surrounding region**. Similarly, if the target aspect changes, then the next aspect should be one adjacent to the current aspect in a graph relating the various aspects (*cf.* [17]). In this case the target's new position should be in a cell defined for the new aspect and which intersects a cell in the **surrounding region**.

The surrounding region for each cell and the neighborhood of each aspect can be determined ahead of time, so that for each target aspect and position we can plan a set of camera states called the **related camera settings, RCS** which permit relocating the target if it is last seen with this aspect and position. Further, it may be possible to dynamically order by preference choices in RCS according to the target's trajectory and rotation.

6 Example experiment

In this section we describe the tracking algorithm with reference to an experiment in a fixed office environment. The camera used in our experiment is a canon VC-C1 MKII Communication Camera (Fig. 4). The pan, tilt, and zoom of the camera are controlled by an SGI Indy machine through an RS-232 port during the tracking process. The mechanical errors are relatively small, which makes this a perfect device for our tracking strategy. The image size taken with this camera is 640×480. The rotation angle for pan is limited to Right-Left +/- 50 degrees, the rotation angle for tilt is Up-Down +/- 20 degrees. The zoom range is 8 × power zoom. To control the camera, pan can take values from 0 (leftmost) to 1300 (rightmost). Each step of pan corresponds to 0.0769 degree. The tilt can vary from 0 (lowermost) through 289 (horizontal) to 578 (uppermost). Each step of tilt corresponds to 0.0692 degree. The zoom can take values from 0 (largest camera angle) to 128 (smallest camera angle).

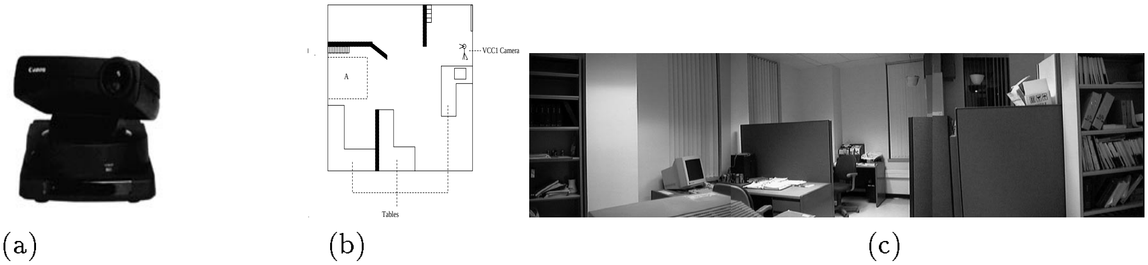


Figure 4: (a) The Canon VCC1 MKII camera used in the experiments. (b) Sketch of top view of the tracking environment. (c) Global view of the tracking environment.

The tracking environment is a normal office. Fig. 4(b) shows the top view of the environment. Region A is the most distant part of the office visible from the camera. Fig. 4(c) gives a global view of the environment, as constructed from three camera images with pan = 0, 525 and 1050, and constant tilt of 277 and zoom 0. These are states (a), (h), and (p) of Table 1.. Since these three camera settings suffice for a complete scan of the office environment, they form the Minimum Camera Parameter Settings for tracking.

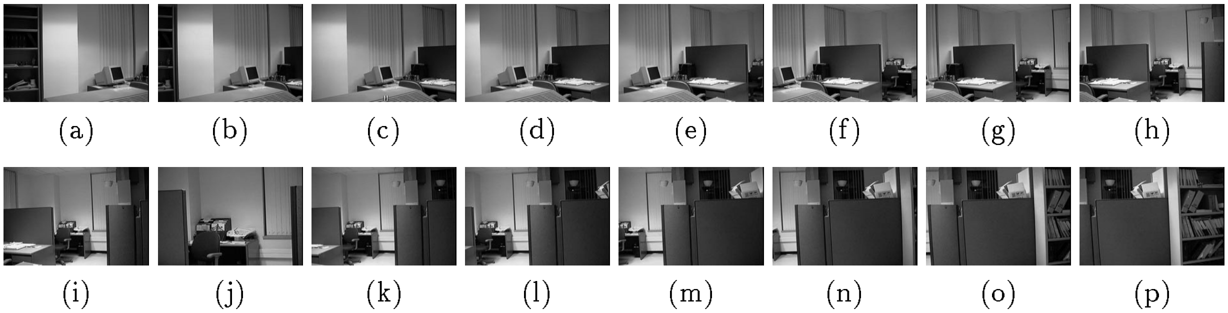
For smooth tracking, however, we increase the number of camera states to form the Camera Parameter Settings for Tracking, as listed in Table 1. The background images for these camera states are shown in Fig. 5. For this simple example, the tilt and zoom parameters remain constant except for one state (j) where they are adjusted to accommodate for the distant Region A (cf. Fig. 4). For the other states, the pan parameter is incremented in steps of 75, producing a smooth sweep of images of the environment.

The inference engine which controls the movement of the camera during tracking iterates the following steps:

1. Repeatedly scan the environment using camera states (a), (h) and (p) of Table 1 since these

State	p	t	z	n	**State**	p	t	z	n
a	0	277	0	1	i	600	277	0	1
b	75	277	0	1	j	600	199	55	5
c	150	277	0	1	k	675	277	0	1
d	225	277	0	1	l	750	277	0	1
e	300	277	0	1	m	825	277	0	1
f	375	277	0	1	n	900	277	0	1
g	450	277	0	1	o	975	277	0	1
h	525	277	0	1	p	1050	277	0	1

Table 1: Camera parameter settings for tracking: (p = pan, t = tilt, z = zoom)



(a) (b) (c) (d) (e) (f) (g) (h)

(i) (j) (k) (l) (m) (n) (o) (p)

Figure 5: The image database for the Camera Parameter Settings for Tracking

comprise the Minimum Set of Camera Parameters. If a target is detected calculate its mass M and x-coordinate X, and Goto (2).

2. If the current zoom is 0 then select the next pan, tilt, and zoom using Method (a) below, otherwise use Method (b).

   (a) **Select pan value**: Let p_1, p_2, ..., p_{15} represent the pan values 0, 75, ..., 1050. Let p_i be the current pan value, and $P = \{p_{i-3}, p_{i-2}, p_{i-1}, p_i, p_{i+1}, p_{i+2}, p_{i+3}\}$. The set P includes all of the pan values for which the viewing directions are within the current image. The x-coordinates of the intersection of these viewing directions with the image plane are: 81, 173, 233, 320, 407, 467, and 559, respectively (the calculation is omitted). Select the next pan direction p_k from P such that the corresponding x-coordinate x_k of intersection with the image plane is closest to X.
   Select tilt and zoom values: If the next pan $p_k = 600$, and $M < 10000$, then select camera state (j) ($< pan, tilt, zoom > = < 600, 199, 55 >$) as the next action for tracking. (The direction and low mass imply that the person is within Region A, which being distant from the camera requires a small angle size). Otherwise the tilt and zoom remain unchanged.

   (b) **Select pan, tilt and zoom values**: (The current zoom is 55, i.e., camera state (j).) If $M < 31,100$ then do not change the camera state. (The direction and mass suggest that the person is still in Region A.) Otherwise, select State h ($< pan, tilt, zoom > = < 525, 277, 0 >$) as the person apparently just left the region.

3. Adjust the camera to the new state and take a picture,

4. Segment as described above, using $n = 1$ for zoom 0 or $n = 5$ for zoom 55. Calculate the new mass M and x-coordinate X of the target if it is detected.

5. If the target was detected then go to Step 2, otherwise go to Step 1.

The nine actions and image sets for this experiment are shown in Fig.s 6 and 7. Each image set consists of five images: the background image, the image with the target present, the color difference image, the improved binary difference image, and the color difference image overlaid with a cross mark for each significant segmented blob. An explanation of the action at each step follows. The sequence begins with Action 1 in State p (pan = 1050, tilt = 277, zoom = 0) where the human is first detected.

1. The coordinates x, y and mass m of each of the five detected target blobs are: $(x, y, m) =$ (309, 205, 16013), (332, 68, 13006), (318, 360, 5202), (422, 180, 5714), and (416, 33, 1612), yielding a total mass of $M = 41547$ and a mass averaged x-coordinate of $X = 337$. Since the zoom is 0, Rule (2a) of the inference engine applies, and the next state selected is p again.

2. One blob is detected: $(x, y, m) = (125, 170, 29670)$. The target is calculated to be at position $X = 125$, and according to Rule (2a) the pan must be decreased three units to 825 (State m).

3. Three blobs are detected: $(289, 115, 5040)$, $(331, 212, 13111)$, $(283, 35, 2362)$. Thus, $X = 315$, implying that the person is near the center again. The state does not change.

4. Six blobs are detected: (79, 99, 4535), (50, 182, 1121), (169, 21, 5085), (109, 306, 3012), (123, 195, 1281), (175, 87, 1300). Thus, $X = 128$, implying that the person is left of center. By Rule (2a), the pan is decreased two units to 675 (State k).

5. Four blobs are detected: (279, 107, 8772), (221, 187, 1284), (291, 294, 2432), (299, 21, 3458). Thus, $X = 280$, implying that the person is near center again. Hence no state change.

6. Three blobs are detected: (210, 236, 1054), (227, 101, 4536), (260, 17, 2834). Thus, $X = 234$, suggesting a next pan value of 600. Since the calculated target size $M = 8234$ is small (less than 10,000), Rule 2(a) causes an increase in zoom to 55, i.e., State (j).

7. Five blobs are detected: (373, 221, 13438), (376, 50, 7314), (368, 364, 2307), (485, 82, 6445), (503, 10, 1346). Thus, $X = 402$ and $M = 30850$. Since the zoom is 55, Rule 2(b) is invoked. The mass is less than 31,100, thus no change in state.

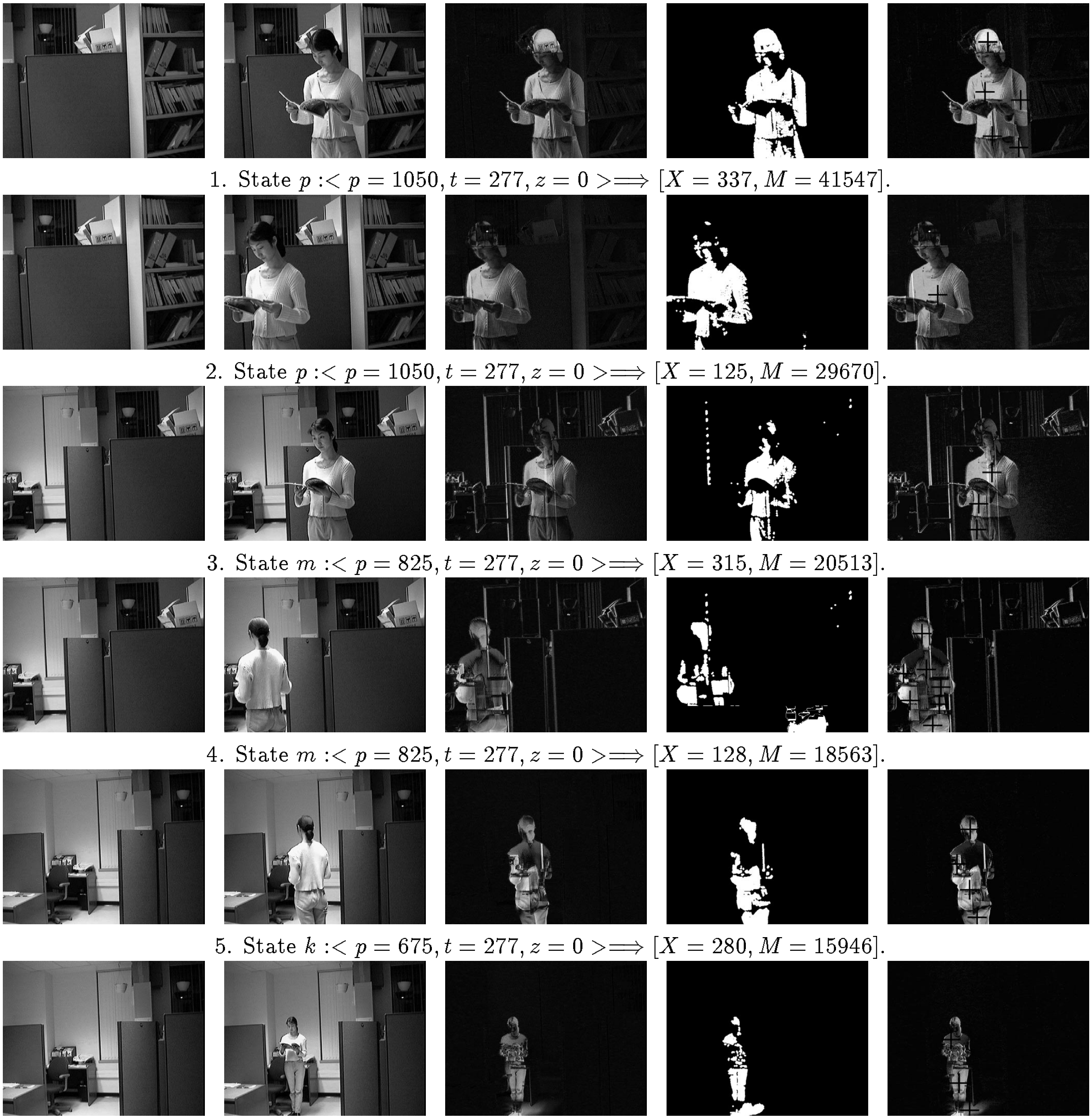
8. Four blobs are detected: (137, 204, 21174), (180, 37, 8517), (129, 387, 1262), (181, 389, 1357). Thus, $X = 149$ and $M = 32310$. The target mass is now large enough that Rule 2(b) causes a switch to State h.

9. Four blobs are detected: (258, 204, 4794), (282, 43, 2607), (322, 94, 3821), and (323, 216, 1031). At this point the experiment is terminated. Thus, the person was successfully tracked during a walk about the office.

7 Discussion

This paper proposes a novel tracking strategy that can robustly track a person, or other object within an environment by a pan, tilt, and zoom camera with the help of a pre-recorded image database. We define a concept called Minimum Camera Parameter Settings (MCPS) which gives a small but sufficient number of camera states required to detect the target anywhere within a given region. For each camera parameter setting in MCPS, we pre-record an image of the environment, and this set of camera states is used during tracking. When the target appears within an image, we segment target from the background by using the corresponding background image as a reference. This greatly simplifies segmentation, and the main part of the person's body can be detected robustly. In order to guarantee smooth tracking, we can increase the number of camera states in the above process. Our method requires to pre-store a set of environmental images. Thus, it may need more memory then other tracking algorithms. However, since the segmentation is done by simply comparing the pre-stored image and the current image, the computational cost is generally less than other methods. The set of background images taken by our method is similar to a panoramic image mosaic [31] which consists of a set of images taken around the same viewpoint. However, images within a panoramic image mosaic are taken with the same camera viewing angle size, while background images taken with our method might be associated with different viewing angle sizes. This difference in zoom is important to guarantee that a good view of the target to be tracked can always be obtained no matter where the target is within the environment.

Since the camera is actively controlled during tracking, and segmentation is based on comparison of images taken with the same camera parameters, our method requires good mechanical reproducibility. We tested our strategy with the Canon VCC1 Camera, and the tracking results are satisfactory. Complexity of the environment is not a problem in segmentation, however the



1. State $p :< p = 1050, t = 277, z = 0 > \Longrightarrow [X = 337, M = 41547]$.

2. State $p :< p = 1050, t = 277, z = 0 > \Longrightarrow [X = 125, M = 29670]$.

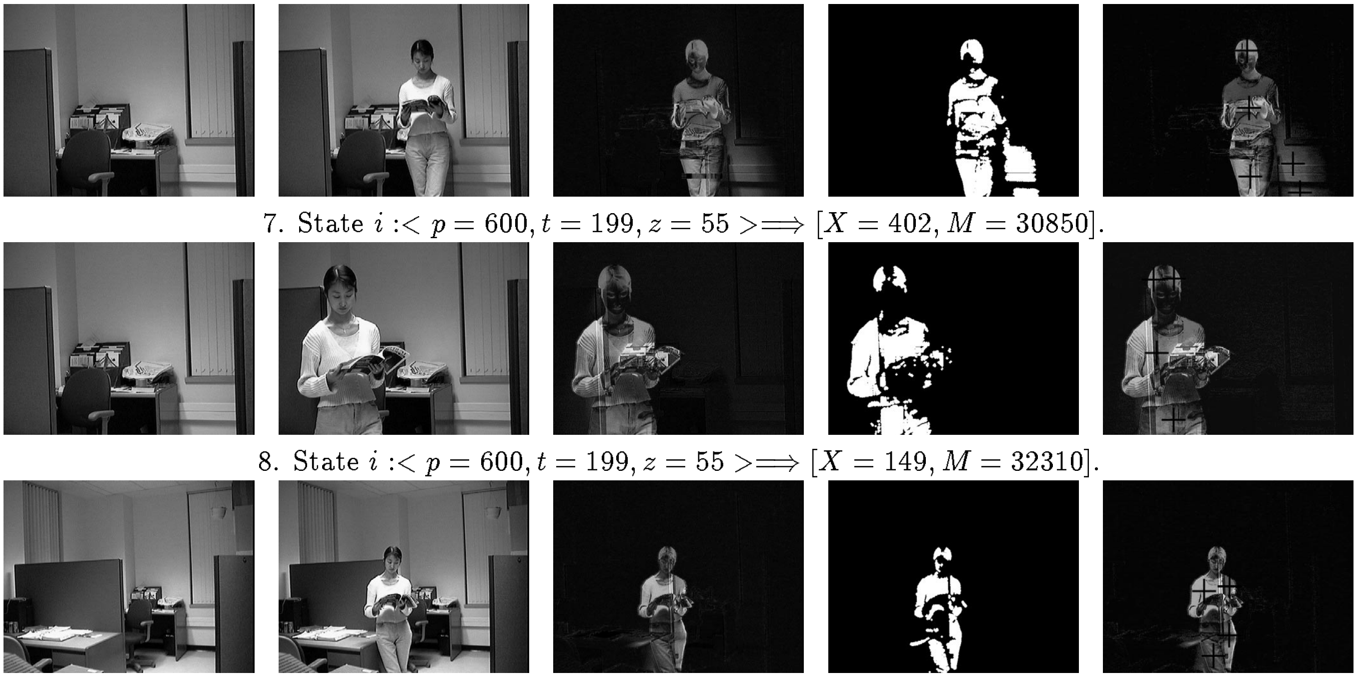
3. State $m :< p = 825, t = 277, z = 0 > \Longrightarrow [X = 315, M = 20513]$.

4. State $m :< p = 825, t = 277, z = 0 > \Longrightarrow [X = 128, M = 18563]$.

5. State $k :< p = 675, t = 277, z = 0 > \Longrightarrow [X = 280, M = 15946]$.

6. State $k :< p = 675, t = 277, z = 0 > \Longrightarrow [X = 234, M = 8424]$.

Figure 6: A tracking experiment performed in our lab.



7. State $i :< p = 600, t = 199, z = 55 > \Longrightarrow [X = 402, M = 30850]$.

8. State $i :< p = 600, t = 199, z = 55 > \Longrightarrow [X = 149, M = 32310]$.

9. State $h :< p = 525, t = 277, z = 0 > \Longrightarrow [X = 288, M = 12253]$.

Figure 7: A tracking experiment performed in our lab (continued).

simple segmentation algorithm which we use in this paper does depend on the constancy of the background. More sophisticated segmentation methods can also be incorporated in the same overall strategy. For example, the problems associated with changes of lighting or shadows can be attacked by using techniques involving color consistancy [3, 37] as illustrated in [25, 38]. Our results show that through the use of a few pre-recorded background images and active control of the camera, the task of visual tracking can be simplified. By incorporating various techniques in object recognition, this strategy may find applications in many practical situations such as human machine interaction and automated surveillance. This method may also be used to advantage in face recognition [4, 6, 7, 21, 25, 29, 32, 34, 38, 41] by setting the effective volume to regions where the facial features can be clearly captured.

Acknowledgments

We would like to thank James Maclean and Gilbert Verghese for their help. We also thank anonymous reviewers of IEEE Workshop on Visual Surveillance for their comments and suggestions. This work was funded by IBM Center for Advanced Studies, Canada and the Department of Computer Science, University of Toronto.

Appendix

The Effect of Mechanical Error on Segmentation

Our method of surveillance and tracking involves collecting a set of background images to serve as a reference during segmentation. Segmentation is based on calculating the difference between the current image and the reference image for the same camera state. Camera states, however, may not be reproduced exactly. For this reason, we introduced a parameter, n, to compensate for mechanical error (*cf.* Section entitled Segmentation). In this Appendix we relate the parameter n to the size of errors in the camera viewing direction and angle size. To do this, we study how these errors influence the projection of a point P from the environment onto the image plane.

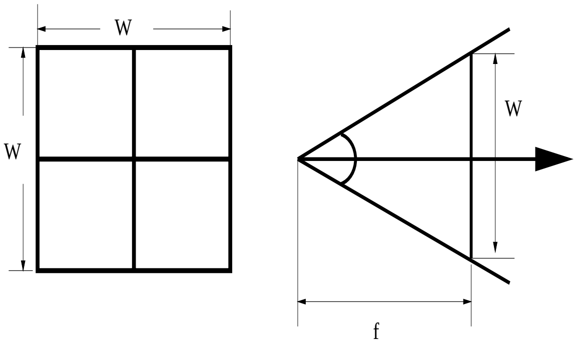


Figure 8: The viewing plane and viewing angle size.

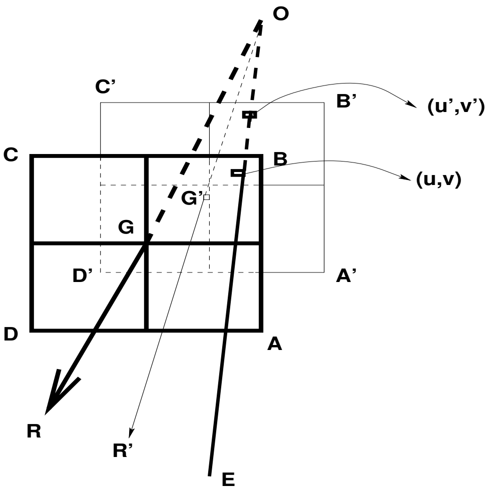


Figure 9: Change in projection of a point as camera viewing direction shifts from OR to OR'.

Without loss of generality, suppose that the viewing plane is a square with side length W, and that the viewing angle size for the reference image is $\beta \times \beta$. Let the focal length be f, as shown in Figure 8. Suppose the resolution of the camera is $m \times m$. Then a unit length is $\frac{W}{m}$.

We have the following relationship:

$$\tan(\frac{\beta}{2}) = \frac{\frac{W}{2}}{f}$$

$$f = \frac{W}{2\tan(\frac{\beta}{2})}.$$

In Figure 9 we represent the intended camera viewing direction as OR, and the erroneous viewing direction as OR'. The desired (reference) viewing angle size is $\beta \times \beta$, but mechanical error results in a size of $\alpha \times \alpha$. As a result, a point in the environment whose projection on the reference image is (u, v) projects to (u', v') in the tracking image. The head coordinate system is sketched in Figure 10 where the viewing direction of the camera is specified by pan and tilt: (p, t). The value of tilt $(0 \leq t \leq \pi)$ is the angle between the camera's viewing direction and the Z axis. The value of pan $(0 \leq p \leq 2\pi)$ is the angle between the projection of the camera's viewing direction on the XY plane and the X axis.

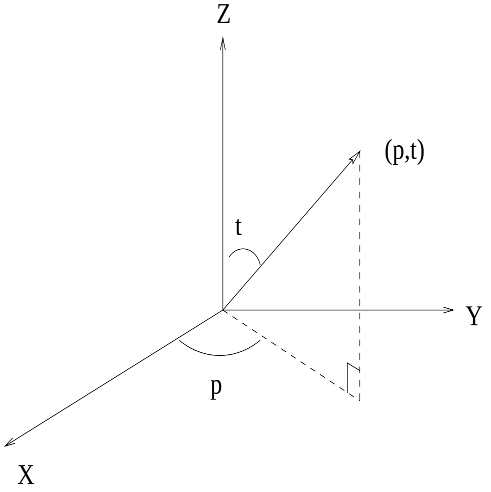


Figure 10: The head coordinate system and the viewing direction of the camera.

Let the mechanical errors in pan, tilt and viewing angle size be within ϵ_p, ϵ_t, and ϵ, respectively (usually ϵ_p and ϵ_t can be found in product specifications). Our aim is to find the maximum value of $\mid u - u' \mid$ and $\mid v - v' \mid$ given these error bounds. Here, we calculate the maximum value of $\mid u - u' \mid$; the maximum value of $\mid v - v' \mid$ can be similarly obtained.

For a given camera angle size β, the error $\mid u - u' \mid$ achieves its maximum value when $(u, v) = (\frac{W}{2}, \frac{W}{2})$ and the reference viewing direction is $(p, t) = (0, \frac{\pi}{2})$, i.e., along the X axis in the head coordinate system. The erroneous viewing direction is $(p', t') = (-\epsilon_p, \frac{\pi}{2} + \epsilon_t)$, and the erroneous viewing angle size is $\beta - \epsilon$. To calculate the value of u', we first rotate the head coordinate system around the Z axis by $-\epsilon_p$, and then around the Y axis by ϵ_t. By considering also the change in the zoom and the geometry of Figure 9 we obtain the following equations.

$$\begin{aligned} u' &= \frac{f\sin(\epsilon_p) + u\cos(\epsilon_p)}{f\cos(\epsilon_p)\cos(\epsilon_t) - u\sin(\epsilon_p)\cos(\epsilon_t) - v\sin(\epsilon_p)} \frac{W}{2\tan(\frac{\beta-\epsilon}{2})} \\ &= \frac{\frac{W}{2\tan(\frac{\beta}{2})}\sin(\epsilon_p) + \frac{W}{2}\cos(\epsilon_p)}{\frac{W}{2\tan(\frac{\beta}{2})}\cos(\epsilon_p)\cos(\epsilon_t) - \frac{W}{2}\sin(\epsilon_p)\cos(\epsilon_t) - \frac{W}{2}\sin(\epsilon_p)} \frac{W}{2\tan(\frac{\beta-\epsilon}{2})} \end{aligned}$$

$$= \frac{\frac{1}{\tan(\frac{\beta}{2})}\sin(\epsilon_p) + \cos(\epsilon_p)}{\frac{1}{\tan(\frac{\beta}{2})}\cos(\epsilon_p)\cos(\epsilon_t) - \sin(\epsilon_p)\cos(\epsilon_t) - \sin(\epsilon_p)} \frac{W}{2\tan(\frac{\beta-\epsilon}{2})} \tag{16}$$

The shift in position as measured in pixels is:

$$\begin{aligned} n &= \frac{u' - u}{\frac{W}{m}} \\ &= \frac{m}{2}\left\{\frac{\sin(\epsilon_p) + \cos(\epsilon_p)\tan(\frac{\beta}{2})}{[\cos(\epsilon_p)\cos(\epsilon_t) - \sin(\epsilon_p)\cos(\epsilon_t)\tan(\frac{\beta}{2}) - \sin(\epsilon_p)\tan(\frac{\beta}{2})]\tan(\frac{\beta-\epsilon}{2})} - 1\right\} \end{aligned}$$

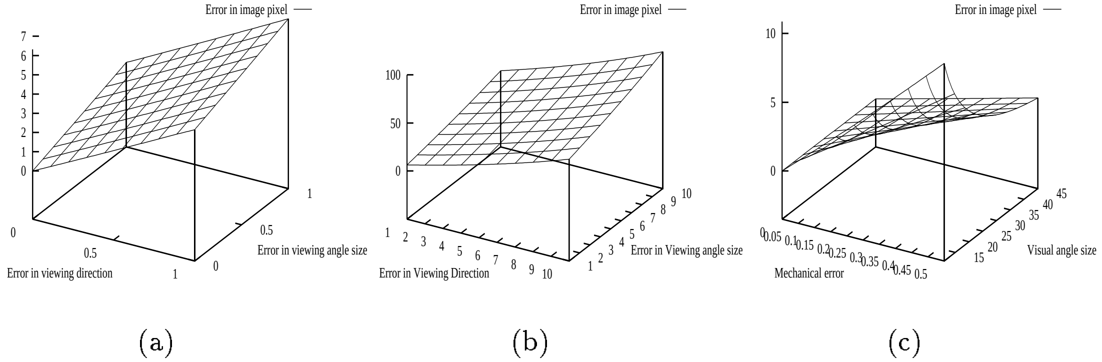


Figure 11: (a) and (b) The influence of the error in viewing direction and the error in camera viewing angle size on the image error. (c) The influence of mechanical error and the camera visual angle size on the image error. Camera resolution parameter $m = 150$.

Figure 11 (a) and (b) show the dependence of image error n (in pixels) on the errors in camera viewing direction and camera viewing angle size. Here, we set $\epsilon_p = \epsilon_t$, and take this to be the value of the error in viewing direction. The error in viewing angle size is ϵ. The errors are represented in degrees in Figure 11. Figure 11(a) shows that the image error n increases with the size of errors in viewing direction and viewing angle. The graph in Figure 11(b) shows that the segmentation algorithm requires good mechanical reproducibility, since large camera errors necessitate such large values of n as to render the segmentation is meaningless.

Figure 11(c) shows the influence of visual angle size β and mechanical error on the image error n. Here, mechanical error is defined to be $\epsilon_t = \epsilon_p = \epsilon$, The units along the X and Y axes are degrees. From this figure we can see that image error n increases with increasing mechanical error and decreasing visual angle size.

References

[1] R. Bajcsy. Active perception vs. passive perception. In *Third IEEE Workshop on Vision*, pages 55–59, Bellaire, 1985.

[2] A.M. Baumberg and D.C. Hogg. An efficient method for contour tracking using active shape models. In *IEEE Workshop on Motion of Non-rigid and Articulated Objects*, pages 194–199, 1994.

[3] D.D. Brainard, B.A. Wandell, and E.J. Chichilnisky. Color constancy: from physica to appearance. *Current Directions in Psychological Science*, 2(5):165–170, 1993.

[4] R. Brunelli and T. Poggio. Face recognition: features versus templates. In *IEEE Trans. Pattern Analysis and Machine Intelligence*, pages 1042–1052, 1993.

[5] Q. Cai, A. Mitiche, and J. Aggarwal. Tracking human motion in an indoor environment. In *IEEE International Conference on Image Processing*, pages 215–218, 1995.

[6] J.L. Crowley and F. Berard. Multi-modal tracking of faces for video communications. In *CVPR*, 1997.

[7] T. Darrell, B. Moghaddam, and A.P. Pentland. Active face tracking and pose estimation in an interactive room. In *CVPR*, pages 67–71, 1996.

[8] S.J. Dickinson, H.I. Christensen, J.K. Tsotsos, and G. Olofsson. Active object recognition integrating attention and viewpoint control. *Computer Vision and Image Understanding*, 67(3):239–260, 1997.

[9] W.T. Freeman and C. Weissman. Television control by hand gestures. Technical Report 94-24, Mitsubishi Electric Research Labs., Cambridge, MA, 1994.

[10] D.M. Gavrila and L.S. Davis. 3-d model based tracking of humans in action: a multi-view approach. In *CVPR*, pages 73–79, 1996.

[11] L. Goncalves, E.D. Bernardo, E. Ursella, and P. Perona. Monocular tracking of the human arm in 3d. In *ICCV*, 1995.

[12] H.P. Graf, E. Cosatto, D. Gibbon, M. Kocheisen, and E. Petajan. Multi-modal system for locating heads and faces. In *International Conference on Automatic Face and Gesture Recognition*, pages 88–93, Killington, Vermont, October 1996.

[13] D. Huttenlocher, J. Noh, and W. Rucklidge. Tracking non-rigid objects in complex scenes. In *ICCV93*, pages 93–101, 1993.

[14] S.X. Ju, M.J. Black, and Y. Yacoob. Cardboard people: A parameterized model of articulated image motion. In *International Conference on Automatic Face and Gesture Recognition*, pages 38–44, Killington, Vermont, October 1996.

[15] I. Kakadiaris and D. Metaxas. Model-based estimation of 3d human motion with occlusion based on active multi-viewpoint selection. In *CVPR*, pages 81–87, 1996.

[16] C. Kervrann and F. Heitz. A hierarchical statistical framework for the segmentation of deformable objects in image sequences. In *CVPR*, pages 724–728, 1994.

[17] J. Koenderink and A. van Doorn. The internal representation of solid shape with respect to vision. *Biological Cybernetics*, 32:211–216, 1979.

[18] J.J. Kuch and T.S. Huang. Vision based hand modeling and tracking for virtual teleconferencing and telecollaboration. In *ICCV95*, pages 666–671, 1995.

[19] B. Moghaddam and A. Pentland. Probabilistic visual learning for object detection. In *ICCV95*, pages 786–793, 1995.

[20] D.W. Murray, K.J. Bradshaw, P.F. McLauchlan, I.D. Reid, and P.M. Sharkey. Driving saccade to pursuit using image motion. *International Journal of Computer Vision*, 16:205–228, 1995.

[21] N. Oliver, A. Pentland, and F. Berard. Lafter: Lips and face real time tracker. In *CVPR*, 1997.

[22] T.J. Olson and D.J. Coombs. Real-time vergence control for binocular robots. *International Journal of Computer Vision*, 7(1):67–89, 1991.

[23] K. Pahlavan and J.-O. Eklundh. A head-eye system—analysis and design. *CVGIP: Image Understanding*, 56(1):41–56, 1992.

[24] V. Pavlovi, J. Kuch, and T. Huang. Hand gesture modelling analysis and synthesis. In *International workshop on automatic face and gesture recognition*, Zurich, 1995.

[25] A. Pentland, B. Moghaddam, and T. Starner. View-based and modular eigenspace for face recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 84–91, Seattle, WA, USA, 1994.

[26] J.M. Rehg and T. Kanade. Digiteyes: Vision-based hand tracking for human-computer interaction. In *Proceedings of the workshop on Motion of Non-Rigid and Articulated Bodies*, pages 16–24, 1994.

[27] K. Rohr. Towards model-based recognition of human movements in image sequences. *CVGIP: Image Understanding*, 59(1):94–115, 1994.

[28] M. Rossi and A. Bozzoli. Tracking and counting moving people. In *Second IEEE International Conference on Image Processing*, pages 212–216, 1994.

[29] H.A. Rowley, S. Baluja, and T. Kanada. Neural network-based face detection. In *IEEE Trans. Pattern Analysis and Machine Intelligence*, January 1998.

[30] P.M. Sharkey, I.D. Reid, P.F. McLauchlan, and D.W. Murray. Real-time control of an active stereo head/eye platform. In *Proceedings of the 2nd International Conference on Automation, Robotics and Computer Vision*, 1992.

[31] Harry Shum, Mei Han, and Richard Szeliski. Interactive construction of 3d models from panoramic. In *Proceedings of Computer Vision and Pattern Recognition*, Santa Barbara, USA, June 1998.

[32] K. Sung and T. Poggio. Example-based learning for view-based human face detection. In *IEEE Trans. Pattern Analysis and Machine Intelligence*, January 1998.

[33] K. Tarabanis, R.Y. Tsai, and P.K. Allen. Analytical characterization of the feature detectability constraints of resolution, focus, and field of view for vision sensor planning. *CVGIP: Image Understanding*, 59:340–358, May 1994.

[34] M.A. Turk and A. Pentland. Face recognition using eigenfaces. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 586–591, Maui, HI, USA, 1991.

[35] J. Weng, N. Ahuja, and T.S. Huang. Learning recognition and segmentation using the cresceptron. In *ICCV93*, pages 121–128, 1993.

[36] C. Wren, A. Azarbayejani, T. Darrell, and A. Pentland. Pfinder: Real-time tracking of the human body. In *International Conference on Automatic Face and Gesture Recognition*, pages 51–60, Killington, Vermont, October 1996.

[37] G. Wyszecki and W. S. Styles. *Color Science: Concepts and Methods, Quantitative Data and Formulae, Second Edition.* John Wiley & Sons, New York, 1982.

[38] J. Yang and A. Waibel. A real-time face tracker. In *WACV*, 1996.

[39] Y. Ye. Sensor planning for object search. *PhD Thesis, Comp. Sci., University of Toronto*, 1997.

[40] Yiming Ye, John K.Tsotsos, Karen Bennet, and Eric Harley. Tracking a person with prerecorded image database and a pan, tilt, zoom camera. In *Proceedings of the IEEE Workshop on Visual Servinence (an ICCV workshop)*, Bamby, India, 1994.

[41] A. Yuille, P. Hallinan, and D. Cohen. Feature extraction from faces using deformable templates. *International Journal of Computer Vision*, 8(2):99–111, 1992.