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

Yiming Ye
IBM T.J Watson Research Center
P.O. Box 704,Yorktown Heights, N.Y. 10598
yiming@watson.ibm.com

John K. Tsotsos
Department of Computer Science
University of Toronto
Toronto, Canada, M5S 1A4
tsotsos@vis.toronto.edu

Karen Bennet
IBM Canada Center for Advanced Studies
North York, Ontario
bennet@vnet.ibm.com

Eric Harley
Department of Computer Science
University of Toronto
eharley@db.toronto.edu

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 called the Minimum Camera Parameter Settings (MCPS) which contains just enough camera states as required to survey the environment for the target. This set of states is used to facilitate tracking and segmentation. The idea is to store a background image of the environment for every camera state in MCPS, thus creating an image database. During tracking camera movements are restricted to states in MCPS (or a version of this set that is augmented to improve smoothness of tracking). Scanning for the target and segmentation of the target from the background are simplified as each current image can be compared with the corresponding pre-recorded background image.

1. Introduction

The task of visually tracking objects moving in three-dimensions has received considerable attention in the computer vision community over the past few years [3, 4, 6, 7, 8, 9, 10, 11, 13, 14, 16, 19, 22, 23, 24]. The task is a challenging one because it not only involves the difficulties of segmenting the target from various backgrounds, but also analysis and prediction of the target's motion. Approaches to this problem include the use of multiple cameras [6, 10], two- and three-dimensional models of the target [6, 8] and attempts to follow specific features of the moving target, such as head or hands through the use of an active camera [4]. The stability of these tracking methods is adversely affected by the complexity of the environment. In this paper we present a novel tracking strategy which 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. The method is based on active control of a pan, tilt, and zoom camera and the use of a pre-recorded image database of the environment.

Active control of the camera is a form of *sensor planning* advocated in [2] and [1] and analyzed in [21]. 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 [5] and [15, 17, 18, 20], respectively.

Whereas segmentation is generally difficult and unstable, we show that some problems can be alleviated through the use of a pre-recorded image database and intelligent control of the camera. We first select a set of camera states (i.e., pan, tilt, and zoom settings) such that wherever the target may appear in the given environment, there exists at least one camera state appropriate for target recognition. The background images for these camera states are stored in an image database. These same camera states are used during tracking, so that the background images form references to facilitate segmentation. This paper presents the tracking algorithm and a simple experiment to illustrate the concepts.

2. Minimum set of camera states

We first would like 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. For a given recognition algorithm and fixed camera viewing angle size $\langle w, h \rangle$, the probability of successfully recognizing a target appearing in an image is high only when the distance l from the target to the camera is within a certain range. This *effective range* is such that the whole target is within the camera's field of view and the target features are represented with sufficient clarity. A set of viewing angles $\langle w_0, h_0 \rangle$,$\langle w_1, h_1 \rangle$, ..., $\langle w_{n_0}, h_{n_0} \rangle$ can be selected such that their effective ranges divide the space around the camera center into a layered sphere, covering the depth D of the environment:

$$w_i = 2arctan[(\frac{N_0}{F_0})^i tan(\frac{w_0}{2})], \quad (1)$$

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

where the biggest viewing angle for the camera is $\langle w_0, h_0 \rangle$, and its effective range for the given aspect is $[N_0, F_0]$. The number of such layers or angles is $n_0 = \lfloor \frac{ln(\frac{D}{F_0})}{ln(\frac{F_0}{N_0})} - 1 \rfloor$. These equations are derived using geometric constraints and the requirement that the area of the target patch in the image remain constant from one layer to the next (see [25] for details).

Each layer of the layered sphere can be successfully scanned for the target using the corresponding angle size $\langle w, h \rangle$ by sweeping the pan and tilt parameters $\langle p, t \rangle$ of the camera. A single camera direction $\langle p, t \rangle$ produces a viewing volume which is a rectangular pyramid, the intersection of which with the spherical layer produces an effective viewing volume for camera state $\langle w, h, p, t \rangle$. A target appearing in the **effective volume** will be detected with high probability by the given recognition algorithm when the camera is in the corresponding state. To examine the entire layer for the target we need a set of camera directions, $\langle p, t \rangle$, such that the union of their effective volumes cover the whole layer with little overlap. The following algorithm generates a set S of viewing directions required for covering the whole sphere. See [25] for details of derivation.

1. $S = \langle 0, 0 \rangle$; $t_e \longleftarrow \frac{\pi}{2}$

2. while $(t_e > \alpha)$ **do**

Cover the two slices of spherical surface with tilt in the range $[t_b, t_e]$ and $[\pi - t_e, \pi - t_b]$.

1. $t_b \longleftarrow arccos\{\frac{cos((t_e - \frac{\alpha}{2}) - \frac{\alpha}{2})}{\sqrt{1 + \frac{sin^2(\frac{\alpha}{2}) sin^2((t_e - \frac{\alpha}{2}) - \frac{\alpha}{2})}{sin^2((t_e - \frac{\alpha}{2}) + \frac{\alpha}{2})}}}\}$
2. Let $t \longleftarrow t_e - \frac{\alpha}{2}$.
3. Let $\Delta_{pan} \longleftarrow 2arctan\{\frac{sin(\frac{\alpha}{2})}{sin((t - \frac{\alpha}{2}) + \frac{\alpha}{2})}\}$
4. Use Δ_{pan} to divide the range $[0, 2\pi]$ for the given slice into a series of intervals $[p_b, p_e]$, as follows: $[0, \Delta_{pan}]$, $[\Delta_{pan}, 2\Delta_{pan}]$, ..., $[k\Delta_{pan}, 2\pi]$. Note: the length of the last interval may not be Δ_{pan}.
5. For each interval, let $p \longleftarrow \frac{p_b + p_e}{2}$ and $S = S \bigcup < p, t > \bigcup < p, \pi - t >$.
6. $t_e \longleftarrow t_b$

The result is a set of camera states whose effective volumes 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. If the accuracy of the recognition algorithm is very sensitive to the orientation (aspect) of the target relative to the camera, then we may define a set of camera states for each of several target aspects. In most situations, the target cannot move about the entire sphere of radius equal to the depth of the environment, but rather is physically restricted to a subregion Ω. The camera states of importance during tracking then are the subset whose effective volumes intersect Ω. This set becomes our Minimum Camera Parameter Settings ($MCPS$), i.e., the minimum set of camera states needed to track the target within the environment.

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).

3. 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 the 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*

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

Figure 1. Image Segmentation and Recognition Algorithm

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. 1. 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 (typically less than 6) is chosen to compensate for errors in camera movement and depends on camera 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. 1 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 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 $X = \Sigma m_i x_i / \Sigma x_i$, $Y = \Sigma m_i y_i / \Sigma y_i$, 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. 1(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.

4. 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.

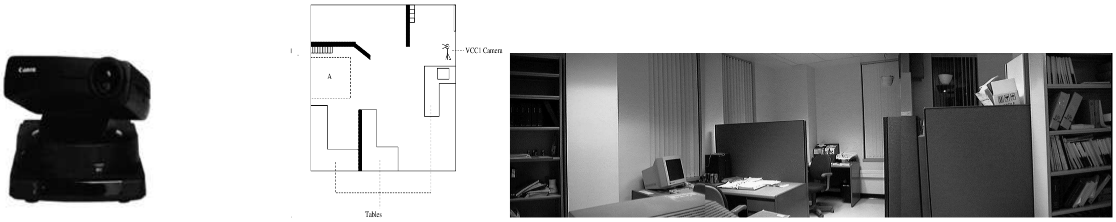
2. Take an image $I^*_{\langle w,h,p,t \rangle}$.

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.* [12]). 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**.



(a) (b) (c)

Figure 2. (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 surrounding region for each cell and the neighbourhood 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.

5. 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. 2). 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 $8\ \times$ 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).

The tracking environment is a normal office. Fig. 2(b) shows the top view of the environment. Region A is the most distant part of the office visible from the camera. Fig. 2(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

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

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

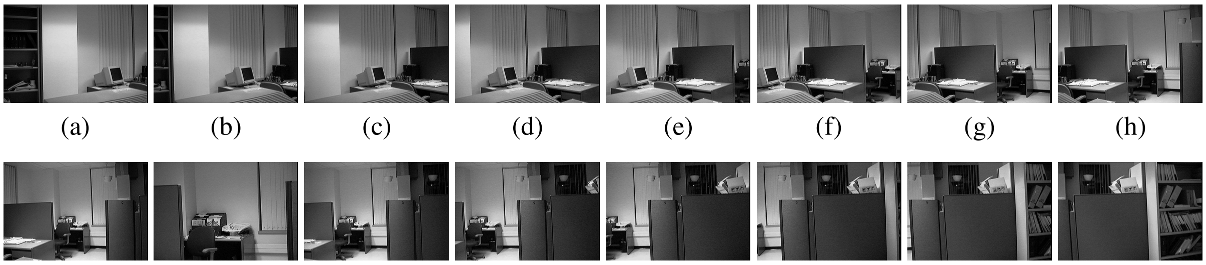
these camera states are shown in Fig. 3. For this simple example, the tilt and zoom parameters remain constant except for one state (j) where they are adjusted to accommodate for the distant Region A (cf. Fig. 2). For the other states, the pan parameter is incremented in steps of 75, producing a smooth 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 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, $\ldots$, p_{15} represent the pan values 0, 75, $\ldots$, 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



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

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

Figure 3. The image database for the Camera Parameter Settings for Tracking

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 4 and 5. 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.

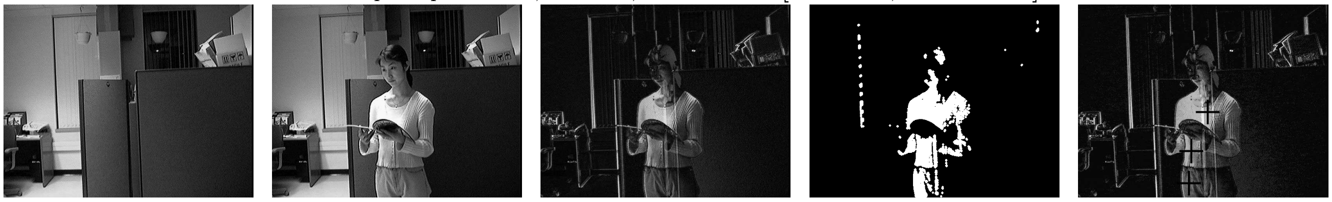
6. Three blobs are detected: (210, 236, 1054), (227, 101, 4536), (260, 17, 2834). Thus, $X = 234$, suggesting a next pan value of 600. Since the calculated target size $M = 8234$ is small (less than 10,000), Rule 2(a) causes an increase in zoom to 55, i.e., State (j).

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,

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

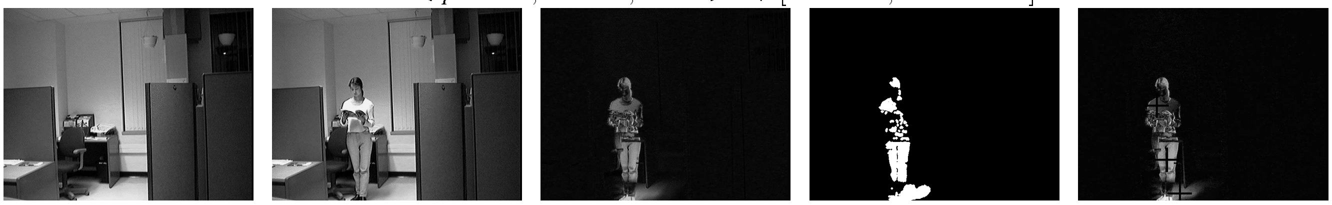
2. State $p :< p = 1050, t = 277, z = 0 > \Longrightarrow [X = 125, M = 29670]$.



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 4. 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 5. A tracking experiment performed in our lab (continued).

$X = 149$ 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.

6. Conclusion

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 the minimum 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 while using the corresponding background image as a reference. This can greatly simplify 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.

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 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. 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. This strategy may find applications in many practical situations such as human machine interaction and automated surveillance.

Acknowledgements

We would like to thank James Maclean and Gilbert Verghese for their help. This work was funded by IBM Center for Advanced Studies, Canada and the Department of Computer Science, University of Toronto.

References

[1] J. Aloimonos, I. Weiss, and A. Bandyopadhyay. Active vision. *International Journal of Computer Vision*, 1(4):333–356, 1988.

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

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

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

[5] 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.

[6] 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.

[7] 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.

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

[9] 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.

[10] 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.

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

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

[13] J.J. Kuch and T.S. Huang. Vision based hand modeling and tracking. In *Proceedings of International Conference on Computer VIsion*, pages 81–87, 1996.

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

[15] 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.

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

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

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

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

[20] 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.

[21] 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.

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

[23] 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.

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

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