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

Yiming Ye[1], John K. Tsotsos[2], Karen Bennet[3] and Eric Harley[2]

[1] IBM T.J Watson Research Center, P.O. Box 704,Yorktown Heights, N.Y. 10598[†]
[2] Department of Computer Science, University of Toronto
[3] IBM Canada Center for Advanced Studies, North York, Ontario

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. 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.

Keywords: Tracking, Image database, Segmentation,

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. 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 [1, 4], two- and three-dimensional models of the target [1, 7] and attempts to follow specific features of the moving target, such as head or hands through the use of an active camera [6]. The stability of these tracking methods is adversely affected by the complexity of the environment.

In this paper we show that some of these problems can be alleviated through the use of a pre-recorded image database and intelligent control of the camera (sensor planning). 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.

[†] yiming@vis.toronto.edu, tsotsos@vis.toronto.edu, bennet@vnet.ibm.com, eharley@db.toronto.edu

These same camera states are used during tracking, so that the background images form references to facilitate segmentation. We illustrate these ideas with an experiment basesd on a simple segmentation method and tracking algorithm.

2 The 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. These angles can be obtained empirically or derived from geometric constraints and the requirement that the size of the target in the image remain constant from one layer to the next (see [12] 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. An algorithm that generates this set given $\langle w, h \rangle$ is presented in [12]. Thus, we can produce a set of camera states $\langle w, h, p, t \rangle$ whose effective volumes cover the entire sphere around the camera to some depth D. This set becomes the Minimum Camera Parameter Settings ($MCPS$) required to track the target within the environment.

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 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.

Fig. 1. Image Segmentation and Recognition Algorithm

This segmentation algorithm, although extremely simple, can successfuliy 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, using the set of camera states MCPS and the corresponding Image Database IDB_{MCPS}, continuously iterates the following four steps:

1. **If** the target is not detected then camera state is assigned by the **Where to Look Next** routine; **otherwise** the camera state is left unchanged.
2. Take an image $I^*_{\langle w,h,p,t\rangle}$ with camera state $\langle w,h,p,t\rangle$ set in Step 1.
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. From the results of Step 3, decide if the target is detected or not.

The **Where to Look Next** routine performs the task of selecting the next camera state $\langle w^*,h^*,p^*,t^*\rangle \in MCPS$ in an attempt to bring the target into the field of view of the camera for recognition. 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.

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. The pan, tilt, and zoom of the camera are actively controlled by an SGI Indy machine through an RS-232 port 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, the pan value can take values from 0 (leftmost) to 1300 (rightmost). Each step of pan corresponds to 0.0769 degree. The tilt value 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. Figure 2(a) is a sketch of the top view of the environment. Region A is the most distant part of the office visible from the camera. Figure 2(b) 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 three camera settings suffice for a complete scan of the office environment, and thus comprise the Minimum Camera Parameter Settings for our tracking task. To improve smoothness of tracking, however, we allow the pan to increment in steps of 75, from 0 to 1050, with tilt constant at 277 and zoom of 0.

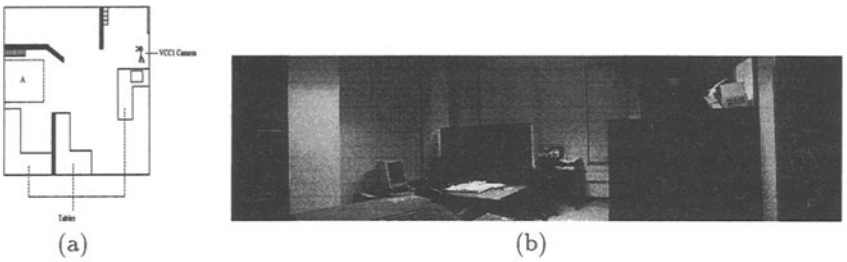

Fig. 2. (a) Top view of the tracking environment. (b) Global view of the tracking environment.

We identify these states in what follows by their pan value. One additional state, called 600′, with pan 600, tilt 199 and zoom 55 is included to capture the more distant area A (cf. Fig. 2). We next describe the inference engine which controls the movement of the camera during tracking in the environment of Fig. 2(b):

1. Repeatedly scan the environment using camera states (pan) 0, 525, and 1050, since these comprise the Minimum Set of Camera Parameters. If a target is detected calculate its total mass M and average x-coordinate X, and Goto (2).
2. If the current zoom is 0 then select the next $< pan, tilt, zoom >$ using Method (a) below, otherwise use Method (b).
   (a) **Select pan value**: Let $p_i = i * 75$ be the current pan value, and P be the set of pan values including p_i and the next three lower and three higher pan values allowed. The set P includes all of the pan values with viewing directions that fall in 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, from lowest to highest pan values in P. 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 600′ ($< pan, tilt, zoom >=< 600, 199, 55 >$) 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). Else the tilt and zoom remain unchanged.
   (b) **Select pan, tilt and zoom values**: (The current zoom is 55, i.e., camera state 600′). If $M < 31,100$ then do not change the camera state. (The direction and mass suggest that the person is still in Region A.) Else select camera state 525, ($< pan, tilt, zoom >=< 525, 277, 0 >$), as the next action. (Apparently the person has just left region A).
3. Adjust the camera to the new state, take a picture and calculate the new mass M and x-coordinate X of the target.
4. Goto Step 2 to select the next camera parameters for tracking.

The nine actions and image sets for this experiment are shown in Fig. 3.

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. The sequence in Fig. 3 begins with Action 1 in state 1050 where the human is first detected. The coordinates and mass 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 1050 again. In Action 2, (blobs: $(125, 170, 29670)$), the target is calculated to be at position $X = 125$, and according to Rule (2a) the pan must be decreased three units to 825. Action 3 (blobs: (289,115,5040),(331,212,13111),(283,35,2362)) finds the person near the center again, so the state does not change. In Action 4 (blobs: (79, 99, 4535), (50, 182, 1121), (169, 21, 5085), (109, 306, 3012), (123, 195, 1281), (175, 87, 1300)) the person is left of center, and the pan is appropriately changed for tracking to that shown in Action 5. Here the target (blobs: (279,107,8772),(221,187,1284),(291,294,2432),(299,21,3458)) is near center again, hence no camera change for Action 6. In Action 6 the target is left of center, (blobs: (210,236,1054),(227,101,4536),(260,17,2834)) suggesting a next pan value of 600, which invokes Rule (2b). This rule checks the size of the target, which being small causes an increase in zoom to that shown in Action 7. Action 7 (blobs: ((373 , 221 , 13438), (376 , 50 , 7314), (368 , 364 , 2307), (485 , 82 , 6445), (503 , 10 , 1346)) produces no change in state for the next action. In Action 8, (blobs: (137 , 204, 21174), (180 , 37, 8517), (129 , 387, 1262), (181 , 389, 1357)) the target mass increases sufficiently to reset the zoom, as shown in Action 9. Blobs found in Action 9 are: (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 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

Fig. 3. A tracking experiment performed in our Lab.

method requires excellent 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. 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.
2. 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.
3. 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.
4. 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.
5. 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.
6. B. Moghaddam, T. Darrell and A. P. Pentland. Active face tracking and pose estimation in an interactive room. In *CVPR*, pages 67–71, 1996.
7. J. Noh, D. Huttenlocher and W. Rucklidge. Tracking non-rigid objects in complex scenes. In *ICCV93*, pages 93–101, 1993.
8. N. Oliver, A. Pentland and A. Lafter. Lips and Face Real Time Tracker. In *CVPR*, 1997.
9. K. Rohr. Towards model-based recognition of human movements in image sequences. *CVGIP: Image Understanding*, 59(1):94–115, 1986.
10. 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.
11. J. Yang and A. Waibel. A Real-Time Face Tracker. In *WACV*, 1996.
12. Y. Ye. Sensor Planning for Object Search. PhD Thesis, Department of Computer Science, University of Toronto, January 17, 1997.