

ROBUST PEDESTRIAN TRACKING USING A MODEL-BASED APPROACH

Osama Masoud

Nikolaos P. Papanikolopoulos

Artificial Intelligence, Robotics, and Vision Laboratory

Department of Computer Science, University of Minnesota

4-192 EE/CS Building, 200 Union St. SE, Minneapolis, MN 55455, USA

e-mail: {masoud,npapas}@cs.umn.edu

Keywords: Pedestrian Tracking, Pedestrian Control at Intersections, Real-time Tracking.

ABSTRACT

This paper presents a real-time system for pedestrian tracking in sequences of grayscale images acquired by a stationary CCD camera. The objective is to integrate this system with a pedestrian control scheme for intersections. The system outputs the spatio-temporal coordinates of each pedestrian during the period the pedestrian is in the scene. Processing is done at three levels: raw images, blobs, and pedestrians. Our method models pedestrians as rectangular patches with a certain dynamic behavior. Kalman filtering is used to estimate pedestrian parameters. The system was implemented on a Datacube MaxVideo 20 equipped with a Datacube Max860 and was able to achieve a peak performance of over 20 frames per second. Experimental results based on indoor and outdoor scenes demonstrated the system's robustness under many difficult situations such as partial and full occlusions of pedestrians.

INTRODUCTION

There is a wealth of potential applications of pedestrian tracking. These application set certain requirements that should be present in the tracking system. For example, applications pertaining to virtual reality and performance measurement of athletes require that certain body parts be robustly tracked. Security monitoring, event recognition, pedestrian counting, traffic and pedestrian control, and traffic flow pattern identification, on the other hand, require a coarser level of tracking in which the emphasis is on tracking all individuals in the scene whose bodies can be considered as single units. Of course, a system that can perform tracking on all different levels simultaneously is highly

desirable but until now, no such system exists. A few systems that tracked body parts of one person [8,10,15] and two persons [5] have been developed. It remains to be seen how these systems generalize to track an arbitrary number of pedestrians.

The work described in this paper targets the second category of applications [6,7] (tracking the pedestrian as a single unit). Our goal is to integrate this work with a pedestrian control scheme at intersections. Several attempts have been made to track pedestrians as single units. Baumberg and Hogg [3] used deformable templates to track the silhouette of a walking pedestrian. The advantage of their system is that it is able to identify the pose of the pedestrian. Tracking results were shown for one pedestrian in the scene and the system assumed that overlap and occlusions are minimal [2]. Another use of the silhouette was made by Segen and Pingali [12]. In their case, features on the pedestrian silhouette were tracked and their paths were clustered. The system ran in real-time but was not able to deal well with temporary occlusions. Occlusions and overlaps seem to be a primary source of instability for many systems. Rossi and Bozzoli [11] avoided the problem by mounting the camera vertically in their system which aimed to mainly count passing pedestrians in a corridor. Such a camera configuration, however, may not be feasible in some cases. Our approach does not have a restriction on the camera position. More importantly, we do not make any assumptions about occlusions and overlaps. Occlusions and overlaps occur very commonly in pedestrian scenes; and hence, they cannot be ignored by a pedestrian tracking system. Therefore, robustness in arbitrary input scenes with arbitrary conditions is the pri-

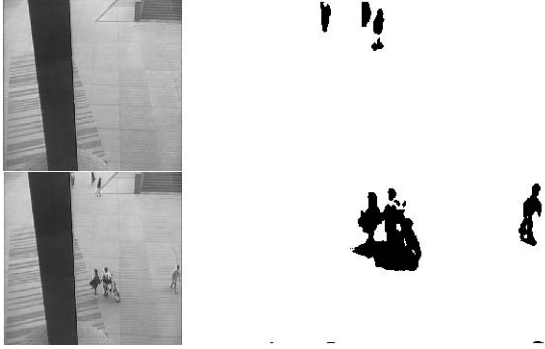


Figure 1. Top left: background image. Bottom left foreground. Right: difference image showing that a blob does not always correspond to one pedestrian.

mary motivation of this work. The use of multiple cameras can alleviate the occlusion problem. Cai and Aggarwal [4] tracked pedestrians with multiple cameras. The system, however, did not address the occlusion problem in particular but rather how to match the pedestrian across different camera views. The switching between cameras was done manually. Smith *et al.* [14] performed pedestrian detection in real-time. The system used several simplistic criteria to judge whether the detected object is a pedestrian or not but did not actually track pedestrians.

Our system uses a single fixed camera mounted in an arbitrary position. We use simple rectangular patches with a certain dynamic behavior to model pedestrians. Overlaps and occlusions are dealt with by allowing pedestrian models to overlap in the image space and by maintaining their existence in spite of the disappearance of some cues. The cues that we use are blobs obtained by thresholding the result of subtracting the image from the background. Shio and Sklansky [13] presented a method for segmenting people in motion with the use of gray scale features. This is an attractive though costly alternative. Our choice of using blobs obtained after background subtraction is motivated by the efficiency of this preprocessing step even though some information is permanently lost. In a typical scene, a blob obtained this way does not always correspond to a single pedestrian. An example is shown in Figure 1. This is the main source of weakness in many of the systems mentioned above which assume a clean one-to-one correspondence between blobs and pedestrians. In our system, we allow maximum flexibility by allowing

this relation to be many-to-many. This relation is updated iteratively depending on the observed blobs behavior and predictions of pedestrians behavior. Three levels of abstractions are used. Each level deals with a certain type of data and retains a state of the data it produces to be used in conjunction with the data received from the lower level. The lowest level deals with raw images. It receives a sequence of images and performs background subtraction producing *difference images*. In the second level, which deals with blobs, difference images are segmented to obtain blobs which are subsequently tracked. Tracked blobs are passed on to the pedestrians level where relations between pedestrians and blobs as well as information about pedestrians is inferred using previous information about pedestrians in that level.

The next section describes the processing done at the blobs level. The pedestrians level is presented next. Finally, experimental results and conclusions are presented.

BLOBS LEVEL

At the blobs level, blob extraction is performed by finding connected regions of 1's in the difference image. A number of parameters is computed for each blob. These parameters include perimeter, area, bounding box, and density (area divided by bounding box area). We then use a novel approach to track blobs regardless of what they represent. Our approach allows blobs to merge, split, appear, and vanish. Robust blob tracking was necessary since the pedestrians level relies solely on information passed from this level.

Blob tracking

When a new set of blobs is computed for frame i , an association with frame $(i-1)$'s set of blobs is sought. The relation between the two sets can be represented by an undirected bipartite graph, $G_i(V_i, E_i)$, where $V_i = B_i \cup B_{i-1}$. B_i and B_{i-1} are the sets of vertices associated with the blobs in frames i and $i-1$, respectively. We will refer to this graph as a *blob graph*. Figure 2 shows an example where blob 1 split into blobs 4 and 5, blob 2 and part of blob 1 merged to form blob 4, blob 3 disappeared, and blob 6 appeared.

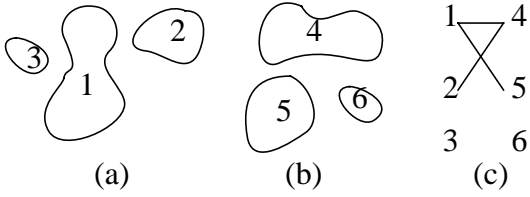


Figure 2. (a) Blobs in frame $(i - 1)$. (b) Blobs in frame i . (c) Relationship among blobs.

The process of blob tracking is equivalent to computing G_i for $i = 1, 2, \dots, n$, where n is the total number of frames. We do this by modeling the problem as a constrained graph optimization problem where we attempt to find the graph which minimizes a cost function [9].

PEDESTRIANS LEVEL

The input to this level is tracked blobs and the output is the spatio-temporal coordinates of each pedestrian. The relationship between pedestrians and blobs in the image is not necessarily one-to-one. A pedestrian wearing clothes which are close in color to the background may show up as more than one blob. Partially occluded pedestrians may also result in more than one blob or even in no blobs at all if the pedestrian is fully occluded. Two or more pedestrians walking close to each other may give rise to a single blob. For this reason, it was necessary to make the pedestrians level capable of handling all the above cases. We do this by modeling the pedestrian as a rectangular patch with a certain dynamic behavior. We found that for the purpose of tracking, this simple model adequately resembles the pedestrian shape and motion dynamics. We now present this model in more detail and then describe how tracking is performed.

Pedestrian model

Three different approaches for pedestrian modeling have been attempted. The latter two are based on the assumption that the scene has a flat ground. Small variations in ground elevation will still be tolerated especially in distant areas. This restriction can be removed if the scene topology can be determined *a priori*.

1. 2-D dynamics and 2-D shape:

The pedestrian is modeled as a fixed size rectangular patch whose dimensions are similar to

the projection of the dimensions of an average size pedestrian located somewhere near the middle of the scene. The patch is assumed to move with a constant velocity in the image coordinate system.

2. 2-D dynamics and 3-D shape:

The pedestrian is modeled as a rectangular patch whose dimensions depend on its location in the image. The dimensions are equal to the projection of the dimensions of an average size pedestrian at the corresponding location in the scene. As in the first approach, the patch is assumed to move with a constant velocity in the image coordinate system.

3. 3-D dynamics and 3-D shape:

The rectangular patch dimensions are as in the previous approach but the patch is assumed to move with constant velocity in the scene coordinate system.

In all these approaches, the patch acceleration is modeled as zero-mean, Gaussian noise to accommodate for changes in velocity. The discrete-time dynamic system for the pedestrian model can be described by the following equation:

$$\mathbf{x}_{t+1} = \mathbf{F}\mathbf{x}_t + \mathbf{v}_t, \quad (1)$$

where $\mathbf{x} = [x \ \dot{x} \ y \ \dot{y}]^T$ is the state vector consisting of the pedestrian location, (x, y) and velocity, (\dot{x}, \dot{y}) , \mathbf{F} is the transition matrix of the system, and \mathbf{v}_t is a sequence of zero-mean, white, Gaussian process noise with covariance matrix \mathbf{Q} .

Pedestrian tracking

The next five subsections describe one tracking cycle.

Relating pedestrians to blobs. We use simple rules to refine the relationship between pedestrians and blobs. If a pedestrian was related to a blob in frame $(i - 1)$ and that blob is related to another blob in the i th frame (through a split, merge, etc.), then the pedestrian is also related to the latter blob. More details can be found in [9].

Prediction. Given the system equation as in the previous section, the prediction phase of the Kalman filter is given by the following equations:

$$\begin{aligned}\hat{\mathbf{x}}_{t+1} &= \mathbf{F}\mathbf{x}_t, \\ \hat{\mathbf{P}}_{t+1} &= \mathbf{F}\mathbf{P}_t\mathbf{F}^T + \mathbf{Q}.\end{aligned}\quad (2)$$

Here, $\hat{\mathbf{x}}$ and $\hat{\mathbf{P}}$ are the predicted state vector and state error covariance matrix, respectively. \mathbf{x} and \mathbf{P} are the previously estimated state vector and state error covariance matrix.

Calculating pedestrian positions. This step provides the measurements and the associated error standard deviation for the Kalman filter. We use a heuristic in which each pedestrian patch is moved around its current location to cover as much as possible of the blobs related to this pedestrian. More details can be found in [9].

Estimation. A measurement is a location in the image coordinate system as computed in the previous subsection, \mathbf{z} . Measurements are related to the state vector by

$$\mathbf{z}_t = \mathbf{h}(\mathbf{x}_t) + \mathbf{w}_t, \quad (3)$$

where \mathbf{h} is the measurement function and \mathbf{w}_t is a sequence of zero-mean, white, Gaussian measurement noise. \mathbf{h} is a linear function in the case of 2-D dynamics modeling approaches and non-linear in the third modeling approach. In the latter case, the extended Kalman filter is used. In either case, the standard state estimation equations for the corresponding filter are applied.

Refinement. At the end of the cycle, we perform some checks to refine the pedestrian-blob relationships since pedestrians have been relocated. This step provides mechanisms for splitting pedestrians walking past each other, pedestrian initialization, re-acquiring blobs due to occlusions, and handling groups of people. More details can be found in [9].

EXPERIMENTAL RESULTS

The system was implemented on the Minnesota Vision Processing System (MVPS) which is the image processing component of the Minnesota Robotic Visual Tracker (MVRT). MVPS consists of a Motorola MVME-147 SBC running real-time operating system OS-9, a Datacube MaxVideo 20 video processor, and a Datacube Max860 vector processor.

The three modeling approaches were tested. The 3-D shape models performed noticeably better than the 2-D shape model. The 3-D dynamics model, however, only slightly outperformed the 2-D dynamics model. The system was tested on several indoor and outdoor image sequences. Several outdoor sequences in different weather conditions (sunny, cloudy, snow, etc.) have been used. In most cases, pedestrians were tracked correctly throughout the period they appeared in the scene. Scenarios included pedestrians moving at a slow or very high speeds, partial and full occlusions, bicycles, and several pedestrian interactions. Interactions between pedestrians included occlusion of one another, repeated merging and splitting of blobs corresponding to two or more pedestrians walking together, pedestrians walking past each other, and pedestrians meeting and then walking back in the direction they came from. The system has a peak performance of over 20 frames per second. In a relatively cluttered image with about 6 pedestrians, the frame processing rate dropped down to about 14 frames per second. Figure 3 shows 12 snapshots from a scene with snow falling. The snapshots span a sequence of 35 seconds. We also performed a pedestrian counting experiment for a sequence of 12 minutes in which 124 pedestrians were counted manually. The system gave a count of 130 making it successful by over 95%. Most of the failures were due to bicyclists who were double counted because the blob they generated was closer to the size of two pedestrians. There are other cases where the system failed. Those include highly crowded images. Other inevitable failures occur when a pedestrian is almost similar in color to the background. In this case, if a pedestrian box is tracking this pedestrian, it will be prone to clamp to other nearby pedestrians having bigger blobs. Also, when a pedestrian becomes totally occluded but then reappears at an unexpected location, the pedestrian box will lose track. Finally, in cases where two pedestrians walk very closely around each other, their pedestrian boxes may get interchanged erroneously.

CONCLUSIONS

We presented a real-time model-based pedestrian tracking system capable of working robustly under many difficult circumstances such as occlusions

and ambiguities. For each pedestrian in the view of the camera, the system produces location and velocity information as long as the pedestrian is visible. There are several issues that still need to be addressed. Spatial interpretation of blobs is one such issue. In the current system, the only spatial attribute of blobs taken into consideration is the blob area. The shape of the blob can give a good clue on its contents. Use of a priori known scene topology is another issue that can be considered.

ACKNOWLEDGEMENTS

This work has been supported by the Minnesota Department of Transportation through Contracts MNDOT/74708-W.O. #14, the Center for Transportation Studies through Contract #USDOT/DTRS 93-G-0017-01, the National Science Foundation through Contracts #IRI-9410003 and #IRI-9502245, and the Department of Energy (Sandia National Laboratories) through Contracts #AC-3752D and #AL-3021.

REFERENCES

- [1] Y. Bar-Shalom and T. E. Fortmann, *Tracking and Data Association*, Academic Press, 1988.
- [2] A. Baumberg and D. Hogg, "Learning flexible models from image sequences," in *Proc. of European Conference on Computer Vision*, vol. 1, pp. 229-308, May 1994.
- [3] A. Baumberg and D. Hogg, "An efficient method for contour tracking using active shape models," in *Proc. of IEEE Workshop on Motion of Non-rigid and Articulated Objects*, pp. 194-199, IEEE Computer Society Press, Nov. 1994.
- [4] Q. Cai and J. K. Aggarwal, "Tracking human motion using multiple cameras," in *Proc. of the 13th International Conference on Pattern Recognition*, pp. 68-72, 1996.
- [5] D. M. Gavrila and L. S. Davis, "3-D model-based tracking of humans in action: a multi-view approach," in *Proc. of IEEE Computer Vision and Pattern Recognition*, San Francisco, 1996.
- [6] R. Hosie, S. Venkatesh, and G. West, "Detecting deviations from known paths and speeds in a surveillance situation," in *Proc. of the Fourth International Conference on Control, Automation, Robotics and Vision*, pp. 3-6, Dec. 1996.
- [7] N. Johnson, and D. Hogg, "Learning the distribution of object trajectories for event recognition," *Image and Vision Computing*, vol. 14, no. 8, pp. 609-615, Aug. 1996.
- [8] I. A. Kakadiaris and D. Metaxas, "3D human body model acquisition from multiple views," in *Proc. of the Fifth International Conference on Computer Vision*, pp. 618-623, Boston, MA, Jun. 1995.
- [9] O. Masoud and N. P. Papanikolopoulos, "A robust real-time multi-level model-based pedestrian tracking system," in *Proc. of the ITS America Seventh Annual Meeting*, Washington, DC, Jun 1997.
- [10] K. Rohr, "Towards model-based recognition of human movements in image sequences," *CVGIP: Image Understanding*, vol. 59, pp. 94-115, Jan. 1994.
- [11] M. Rossi and A. Bozzoli, "Tracking and counting moving people," in *Proc. of Second IEEE International Conference on Image Processing*, pp. 212-216, 1994.
- [12] J. Segen and S. Pingali, "A camera-based system for tracking people in real time," in *Proc. of the 13th International Conference on Pattern Recognition*, pp. 63-67, 1996.
- [13] A. Shio and J. Sklansky, "Segmentation of people in motion," in *Proc. of IEEE Workshop on Visual Motion*, pp. 325-332, 1991.
- [14] C. Smith, C. Richards, S. A. Brandt, and N. P. Papanikolopoulos, "Visual tracking for intelligent vehicle-highway systems," *IEEE Trans. on Vehicular Technology*, vol. 45, no. 4, pp. 744-759, Nov. 1996.
- [15] C. R. Wren, A. Azarbayejani, T. Darrell, and A. Pentland, "Pfinder: real-time tracking of the human body," in *Proc. of the Second International Conference on Automatic Face and Gesture Recognition*, 1996.



frame 39



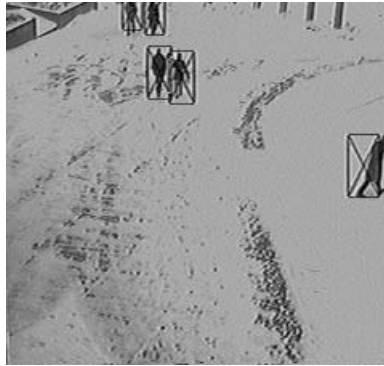
frame 78



frame 118



frame 148



frame 263



frame 364



frame 414



frame 497



frame 539



frame 587



frame 718



frame 783

Figure 3. A number of snapshots from the input sequence in a snowy afternoon overlaid with pedestrian boxes shown in black.