Human Activity Tracking for Wide-Area Surveillance

Patrick D. O'Malley
pomalley@mil.ufl.edu

Michael C. Nechyba
nechyba@mil.ufl.edu

A. Antonio Arroyo
arroyo@mil.ufl.edu

Machine Intelligence Laboratory
Department of Electrical and Computer Engineering
University of Florida, Gainesville, FL 32611-6200

Abstract

We present a method for tracking and identifying moving persons from video images taken by a fixed field-of-view camera. Specifically, we have developed a system to first track moving persons in a given scene and generate color-based models of those persons to accomplish identification. The tracking is non-invasive meaning that it does not require persons to wear any particular electronics or clothing to be able to track them. Tracking is accomplished using a position-based data association algorithm while the color modeling is accomplished using a mixture-of-Gaussians statistical model. The expectation-maximization algorithm is used to generate the color models over a sequence of frames of data in order to make the entire process near-realtime. Once a color model is developed for a given person, that model will be placed in a database where it will be used for future identification. Any similar identification scheme can be used in place of color modeling for greater accuracy.

1. Introduction

Tracking people using only the data from a video camera is a difficult computer vision problem. However, research in this area is spurred on by the promise of many enhancements to current technology that a robust person tracking system would provide. Already we are seeing the use of computer generated overlays to add information to sports broadcasts. These overlays, however, are usually dependent on special sensors and emitters placed on the tracked object. The RACEf/x(c) [1] system, for example, uses a GPS system to accomplish tracking of NASCAR (c) racecars in order to superimpose car statistics such as the name of the driver, velocity and position onto the broadcast. The people tracking problem, on the other hand, cannot usually be solved by using similar transmitters. In surveillance applications wherein people are tracked in order to detect unauthorized entry or suspicious behavior for example, there is no possibility of using transmitters. A non-invasive method using only the information obtained by a passive sensor such as a video camera is necessary.

The specific problem which we wish to address is surveillance over a wide-area. A typical coverage zone would be an airport, the downtown of a large city or any large public area. Surveillance over these wide areas would be utilized to search for suspicious behavior - persons loitering, for example - or unauthorized access - a person attempting to enter a restricted zone, for example. Currently, surveillance of this type is accomplished by a human operator who does all of the detection of unauthorized activity over many cameras. Given the challenge of watching multiple monitors, it is easy to see how some suspicious activity would go unnoticed by a human operator. A computer vision system, however, can monitor both immediate unauthorized behavior and long-term suspicious behavior. The system would then alert a human operator for a closer look. In this way, one operator could monitor more locations or monitor the same number of locations more closely.

The most common method of tracking any moving object, including persons, is to use some form of image subtraction to find motion regions. These motion regions are then classified as either noise or a person. Researchers such as [2] use the difference between consecutive video images to find the regions of motion. Others such as [3] and [4] use a background known to be empty of any moving objects (a "reference image") which is then subtracted from current or foreground images to find the motion regions. It should be noted that this use of a reference image is limited to fixed field-of-view cameras whereas the consecutive image subtraction method can be used with movable cameras.

The more difficult step occurs after finding the motion regions. This step is to determine which motion regions should be tracked and which are noise. A simple tree swaying in the breeze, for example, could cause severe noise problems leading to false positives (noise tracked as a person) or false negatives (a person not tracked because of nearby noise.) [5] assumes that an unknown object is to be tracked (and thus a person) if it can be tracked over several frames using a second-order motion model. Other tracking methods using color alone [3] have a difficult time tracking new objects - noise or a person - because they do not have color information available initially.

Identification of a given individual is difficult under any circumstances. If the video imagery is taken from a distance such that the face is obscured, it is questionable whether a good likelihood of positive identification is pos-

sible. If the video can be taken close-up, a face recognition algorithm such as that by [6] or [7] could give good identification accuracy. A full-body color modeling scheme as used by [8] would be useful assuming the statistical model chosen can robustly compare two slightly dissimilar persons. It can be seen that a simple color model cannot be used over great time periods for identification but should be limited to short spans and localized regions - a city block, for example.

In order to track a person over a wider area and to be able to connect tracks found across multiple cameras, a single frame of reference will be needed to transform coordinates from individual cameras into a global coordinate system. One method (for two cameras) is given by [9]. Extended to many more cameras, we can see how a human tracking system can be used for wide-area surveillance.

2. Overview

The method for tracking people presented here is fundamentally based on utilizing two classification systems: position and color. As with [5], we will use a first-order motion model to track the individual people as they move throughout the scene. The second classification system is a mixture-of-Gaussians model of the color of each person. Initially, these two classifiers act separately - that is, they do not "share" information between them to support more robust classification. However, further work in the area (see Section 7) will make the tracking more robust by fusing the two systems.

By separating the tracking system from the color modeling system, we can solve the bootstrapping problem faced by systems such as [3] which use only color modeling as the tracking system. We can maintain tracking using the position-based tracker while at the same time generating a color model of the tracked person. Once that color model is developed, we can use it for identification or to resolve occlusion events as described in Section 7.

Before any tracking work can be done, the video frames (images) need to be reduced to centroids of motion regions using image processing techniques.

3. Image Processing

In order to determine where a potential person is in a given scene, we reduce the scene to the centroids of motion regions. Those centroids are what finally gets tracked by the position-based tracker. The original frame is retained, however, to extract color information later.

Like [3] we use a background subtraction scheme based on making a normal distribution model for each pixel. Each frame is first converted from RGB (red, green, blue) colorspace to HSV (hue, saturation, value.) We chose HSV because it tends to reduce the effects of shadows on the subtraction process.

The normal distribution for each pixel is found by taking the first n frames of the video sequence (which we assume to not have any foreground or moving objects in them.) A three dimensional vector, x_{ij}^k given in (1) where k is the frame number and (i,j) is the pixel coordinate, is then used to model the mean and variance as given in (2) and (3).

$$x_{ij}^k = \begin{bmatrix} h \\ s \\ v \end{bmatrix} \tag{1}$$

$$\mu_{ij} = \frac{1}{n} \sum_{k=1}^{n} x_{ij}^k \tag{2}$$

$$\Sigma_{ij} = \frac{1}{n-1} \sum_{k=1}^{n} (x_{ij}^k - \mu_{ij})(x_{ij}^k - \mu_{ij})^T \tag{3}$$

The Mahalanobis distance is used to classify a pixel from a frame as either foreground or background. However, to speed the calculation, we only use the diagonals of the covariance found in (3). We experimentally choose a distance threshold above which a pixel is classified as foreground.

Once motion regions are found, we get masks such as that in Figure 1. Using these masks, we apply a standard connected-region segmentation algorithm to determine the following properties of each region: centroid coordinates (x,y), extents (xmax, ymax, xmin, ymin) and size in pixels. Size thresholds are applied to eliminate any motion regions too small to be considered a person. The remaining centroids are sent to the position tracking system to be classified as a trackable object (a person) or noise.

4. Position Tracking System

The position tracking system is based on similar systems used in RADAR tracking situations where there is a high noise density. The first stage of tracking is known here

Fig. 1: Motion region mask

as the Track Initiation Processor (TIP). After the TIP has acquired a target (assumed here to be a person) it passes that target to the Data Association Filter (DAF) which is designed to keep a lock on the target. The final stage is the track-drop ruleset which is a heuristic approach to determining when a target is no longer trackable such as when it has gone behind an object or has left the scene.

4.1 Track Initiation Processor

The job of the Track Initiation Processor (TIP) is to extract from a series of video frames, all of the objects which could potentially be people. At this stage we assume that the image processing has reduced each video frame to a series of centroids of motion regions that are as yet unclassified as targets. Therefore, the TIP will classify them as either trackable objects (people) or noise. If noise, those regions aren't eliminated outright because they could be as-yet untracked objects so the TIP algorithm takes them into consideration.

The TIP is a faster implementation of a common algorithm used in RADAR known as "retrospective processing" [10] wherein three assumptions made about moving objects are evaluated on all potential targets. By evaluating all potential targets - that is, all unclassified motion regions whether noise or people - over many frames, we will eventually find only those targets which can be people. That is considered a good track.

The three assumptions we wish to follow are very simple: First, that our noise is randomly distributed through the scene. Second, that a person moving through the scene moves with linear velocity over short distances. Thirdly, that a person moving through the scene moves with a limited maximum velocity. These three assumptions are not unreasonable given that people do not tend to move with a quickness that is going to violate either of the last two assumptions.

We find objects which follow these three assumptions by using *n* frames of data. Three frames is the minimum. The first frame of centroid data is assumed to be all potential targets (potential people.) We will prune this assumption over the next two frames.

The situation when the first frame of data is present is shown in Figure 4a. In this simulation (of randomly generated 2-d data), we see simulated centroids of objects. The first image (a) shows the objects as blue dots with green circles indicating the maximum distance that this object (if it is in fact a person) could travel from this frame to the next. The second image (b) shows the frame #2 objects as red dots. Only those red objects inside the green circles can be people because the others are moving too fast (or are noise.) Isolating only those pairs of blue and red objects (frame #1 and #2 objects, respectively) we predict where those objects would be in the third frame if they follow the constant velocity assumption. The third image (c) shows these predictions as magenta circles. Finally, when the third frame of data arrives (shown in Figure 4d) we can see that only one of those objects satisfies the constant linear velocity assumption. Therefore, we can eliminate all the other objects as noise. The three-point track we have found can now be used to converge the Kalman filter described in the next section.

The algorithm described here can be extended from three frames of data to any number. The more frames used to find a valid track, the more likely it is to be a valid track and not simply well-placed noise. In this way, our image processing can be poor in that it leave a lot of noise yet tracking would not suffer seriously.

4.2 Data Association Filter

To continue the tracking process, a data association filter is used. We use a first-order Kalman filter to generate predictions of the positions of each object in the next frame. When that frame arrives, we associate the predictions with the incoming objects using a nearest-neighbor algorithm. We determine "nearness" based on the Mahalanobis distance from the incoming centroid to the prediction with its associated uncertainty covariance given by the Kalman filter. This method tends to give more robust results than using a simple Euclidean distance measure.

4.3 Track Drop Rules

In order to determine when to drop a track, a heuristic approach is used. Currently, we allow the DAF to "miss" - that is, fail to associate with an incoming object - twice before that object is considered gone and the track is dropped. A "miss" is defined to be when the DAF cannot make a nearest-neighbor association because there are no objects within a certain distance to associate with. The distance threshold is determined experimentally.

5. Color Modeling

Color modeling is used for local identification. That is, because it is assumed that a person would change color because they change clothing from one day to the next or even from one hour to the next, using color models for identification does not work over long time spans. Therefore, we use color models in localized temporal and spatial regions. The downtown of a city for one day, for example. After that day, the color models would be expired. But for that day, the identification is assumed to work well enough to allow systems such as this to follow a given person even if they disappear from then reappear in the camera's field of view. In this way, the person's piecemeal path can be recorded.

5.1 Mixture-of-Gaussians Modeling

We chose to use a mixture-of-Gaussians model of the person's color data in RGB (red-green-blue) colorspace. Mixture modeling like this is typically a good fit to the data

given that most persons tend to appear as only a few regions of color. Blue jeans and a t-shirt, for example, are usually only a few colors. Skin tones are well modeled as a region of color.

To simplify the mixture model, we fix the number of Gaussians to three. Three was chosen based on the Expectation-Maximization (EM) algorithm we use to fit the Gaussians to the data. Equations (4), (5) and (6) give the update equations for the iterative EM solution to the mixture-of-Gaussians problem for $i \in \{1, 2, ..., k\}$ where k is the number of Gaussians in the model and n is the number of points in the data set. As can be seen, reducing the number of Gaussians and the amount of data to be modeled can significantly increase the computational speed of this algorithm. Therefore we used only three Gaussians and limited the data to 300 points in RGB colorspace which were found by subsampling all of the person's color data. Note that each of these equations needs to be iterated many times until the Gaussians converge.

$$\overline{P(\omega_i)} = \frac{1}{n} \sum_{j=1}^{n} P(\omega_i | x_j, \Theta) \tag{4}$$

$$\overline{\mu}_i = \frac{\sum_{j=1}^{n} P(\omega_i | \boldsymbol{x}_j, \Theta) \boldsymbol{x}_j}{\sum_{j=1}^{n} P(\omega_i | \boldsymbol{x}_j, \Theta)} \tag{5}$$

$$\overline{\Sigma}_i = \frac{\sum_{j=1}^{n} P(\omega_i | x_j, \Theta)(\boldsymbol{x}_j - \overline{\mu}_i)(\boldsymbol{x}_j - \overline{\mu}_i)^T}{\sum_{j=1}^{n} P(\omega_i | \boldsymbol{x}_j, \Theta)} \tag{6}$$

It was found that further speed enhancements were required to improve the algorithm. The choice was between finding ways to do the above computations more efficiently or trying to hide the computation time entirely. We chose the latter by instead of modeling the color all in one frame, we spread the iterations out over multiple frames. We also allowed the color data which the model is trained on to be changed with each frame. In this way we were able to assure that we would be modeling more than one view of the person.

We use the incoming color data (data that has not yet been included in the color model) as "unknown" validation data by taking the sum log-likelihood of the data given that person's color model. Once the log-likelihood converges (plateaus) we consider it well-formed and do not continue to include any more data in the model.

6. Results

Figure 3 shows a typical sequence using the tracking and identification algorithms together. The images are non-sequential from the top-left to the bottom-right.

The first frame shows the person being tracked initially as he enters the scene. (A green box indicates a tracked object. A blue box indicates an unknown object, probably noise.) Two identifiers are given to the person: a name - in this case "ann" - and a target identifier - in this case '0'. The name is associated with the color model and is therefore unique to a given person. The target identifier is not unique to a person as it is merely the position-based tracker's identification information.

The name ("ann") is assigned to this person initially because it did not have a pre-recorded color model to be associated with. As this person is tracked, a color model is developed and will from then all be associated with the name "ann."

The second image shows that the position-based tracker loses the person because of an image processing problem. (This problem was later fixed but re-introduced into the code to cause the code to lose tracking for demonstration purposes.) The third image shows that the person is still not tracked.

In the fourth image, the person is again tracked. The code compares this person's color to the color models in the database and matches it to the name "ann." (In this demonstration there was only one model in the database from when this person was tracked previously - as shown in frame one. However, it could have determined that this person didn't exist in the database at all and assigned a new name. Because it did not, it indicates that it successfully matched this person to the previously developed color model.)

The last two images show how it continues to track the subject. Not shown is a failure to track caused when the subject is occluded by the sign.

7. Future Work

Two major areas of work could be done to improve this tracking system. They are related, however, and could be accomplished at the same time. The first problem is to improve the robustness of the tracking by fusing the position--based tracking system with the color modeling system. In this way, if a situation happened as in Figure 3, frame 2 wherein the tracker loses the person because of an image processing problem, we could instead decide if the two halves of the previously connected person are really supposed to be one using the color information.

Similarly, we can resolve occlusion such as that shown in Figure 2 where the motion region is seen as only one ob-

ject (bounded by the blue box) but which is really two persons who were previously unconnected. It can be seen how, once we detect that occlusion of this sort has occurred (using a heuristic technique, for example) we can resolve the occlusion and separate the people by using the previously determined color models. We do this by classifying each pixel of the big motion region as either belonging to one or the other color model. By those classifications, we can make a fairly good guess as to the relative locations of the persons.

8. Conclusion

We have shown how a combination of two methods - position-based tracking and color modeling - can be used to robustly track people non-invasively using only standard stationary cameras. We have also shown how this method can be expanded to recover from difficult problems such as occlusion by allowing the position-based tracker to essentially bootstrap the tracking until a color model can be developed which will then make the whole system more robust. We believe this system can be useful in a wide variety of surveillance situations.

9. References

[1] Sportvision website [online]. URL: http://www.sportvision.com. Retrieved on April 9, 2002.

[2] A. J. Lipton, H. Fujiyoshi, and R. S. Patil, "Moving target classification and tracking from real-time video," *Proc. IEEE Image Understanding Workshop*, pp. 129-136, 1998.

[3] S. Khan and M. Shah, "Tracking People in Presence of Occlusion," *Asian Conference on Computer Vision*, Taipei, Taiwan, January 2000.

[4] Q. Cai, A. Mitiche, and J. K. Aggarwal. "Tracking human motion in an indoor environment", *2nd Intl. Conf. on Image Processing,* pages 215-218, Washington D.C., October 1995.

[5] I. Haritaoglu, D.Harwood, and L.Davis, "Who, when, where, what: A real time system for detecting and tracking people", *Proceedings of the Third Face and Gesture Recognition Conference*, pp. 222-227, 1998.

[6] C. Beumier, M.P. Acheroy, Automatic Face Authentication from 3D Surface, *British Machine Vision Conference BMVC 98*, University of Southampton UK, 14-17 Sep, 1998, pp 449-458.

[7] R. Brunelli and D. Falavigna. "Person identification using multiple cues", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 17, No. 10, pp. 955-966, October (1995).

[8] I. Zapata, "Detecting Humans in Video Sequences Using Statistical Color and Shape Models," Master's Thesis, Department of Electrical and Computer Engineering, University of Florida, 2001.

[9] S. Nichols, "Improvement of the Camera Calibration Through the Use of Machine Learning Techniques," Master's Thesis, Department of Electrical and Computer Engineering, University of Florida, 2001.

[10] E. Brookner, "Tracking and Kalman Filtering Made Easy," Wiley, New York, 1998, pp. 111-116.

Fig. 2: Example of two-person occlusion

Fig. 3: Example of tracking and identification video sequence

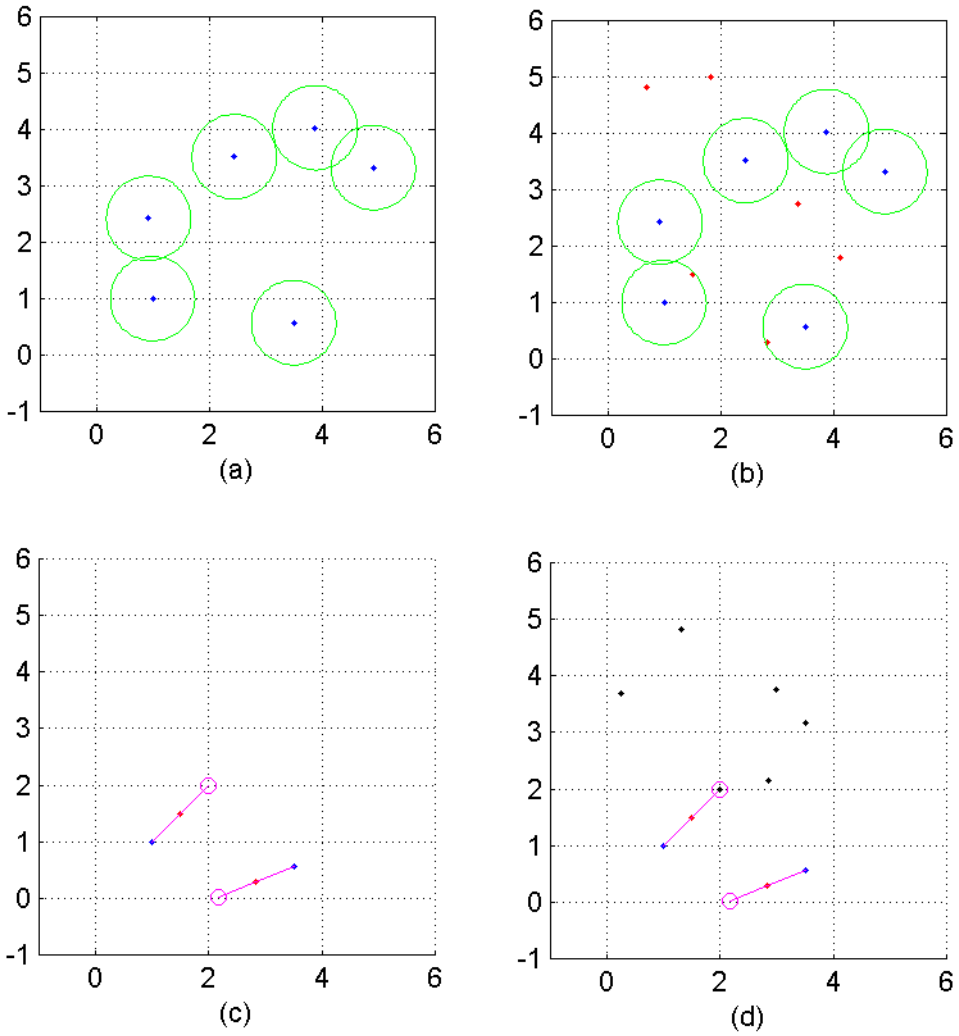

Fig. 4: Track Initiation Processor simulation