People Counting and Tracking for Surveillance

Anup Doshi
CSE 252C Project Report
Draft - Nov. 29, 2005

Abstract

The computer vision community has expended a great amount of effort in recent years towards the goal of tracking people in videos. Much more recently, algorithms have been developed to track multiple people in videos robustly and in real-time. The goal of this project is to implement a system based on one of those algorithms, in order to count and track the people in a database of surveillance footage. Due to several constraints and performance issues, however, a more straightforward algorithm based on background subtraction is implemented and shows acceptable performance levels. Further improvements are considered to improve the performance, including implementations of algorithms such as BraMBLe [3].

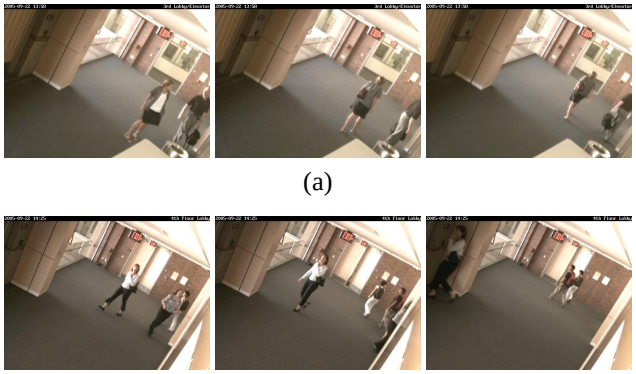
1. Introduction

Tracking people using surveillance equipment has increasingly become a vital tool for many purposes. Among these are the improvement of security and making smarter decisions about logistics and operations of businesses. Automating this process is an ongoing thrust of research in the computer vision community.

With this in mind, a department at UC-Irvine recently conducted a fire drill and recorded the entire drill into a database of footage. With many different camera locations, they are very interested in finding out how many people exited, and which routes they used to exit the building. Their ultimate goal is to uniquely identify the people who exited, however that is beyond the scope of this paper.

Thus the aim of this work is to automatically count the number of people to use each exit in a particular video from the UC-Irvine database. To do so, it will be necessary, to first detect the people in the video, then to track the movements of each person, and finally decide if they exit. The researchers at Irvine would like to see a program whose input is essentially the name of the video, and an output of the count of people entering and exiting.

The contribution of this project so far is a fully functional tracker based on background subtraction, which can count individual people with a great deal of accuracy. Several heuristics are attempted to deal with cases of multiple people, however, none works with an acceptable level. Finally an attempt at implementing a more advanced algorithm is not yet functional, and thus work is continuing forward.



(a)

(b)

Figure 1: A sample sequence from the surveillance footage. (a) shows an example of shadows and different views of the same person. (b) is an example of a tough sequence involving several people crossing in front of each other.

1.1. Previous Work

While tracking one person in a stationary background may be relatively simple, the problem becomes very complicated with multiple people. The may be crossing in front of each other, behind occlusions, through different lighting, with shadows, and in groups. Indeed many of these issues show up in the UCI database (see Figure 1.1). So one goal of this work is to come up with a tracker that can perform robustly under all of those conditions.

Blob tracking may be simple and quick, but it does not work generally, especially with people moving in groups. Several candidate algorithms claim to be able to distinguish people in groups. Among them are Siebel and Maybank [1], who fuse several algorithms including a head-tracker to help distinguish people. Lipton [2] uses classification to segment the image, while Isard and Maccormack [3] use a particle-filtering algorithm they term “BraMBLe”. This algorithm is

described in more detail below(Section 1.2). Rittscher et.al [4] are successfully able to distinguish groups of people into individuals using combinations of model and feature-based segmentation. Finally, a promising algorithm looks to be Zhao and Nevatia [5] who use human shape models in 3D. They claim good results under many of the conditions listed above. There are likely many more algorithms which could be considered, however these seem to be state-of-the-art.

Several issues involved with any of these trackers include tradeoffs in performance and runtime. Also, a major issue in the algorithms that use classification or statistical shape models, is the necessity to train the algorithm. As will be made evident in Section 4.1, the lack of labeled data proves to be a hurdle in implementing any of those algorithms.

1.2. BraMBLe[3]

The BraMBLe algorithm presents several innovations into the people-tracking arena. Primarily, it learns a separate statistical model for the foreground and (static) background, and uses this to generate a likelihood function for an input image. This observation likelihood is then fed as an input into a particle filtering algorithm. The particle filter is able to track an unknown and varying number of objects in an image.

A version of this algorithm was implemented by Kristin Bransom, who was kind enough to provide her code for use in this project. However several changes needed to be made to suit this application. Among them were changing the appearance model to a generalized cylinder, and training the statistical models of foreground and background. Work is ongoing to achieve these goals. However, several problems with this algorithm are apparent which may hinder the performance after all.

One issue involves, as discussed in the BraMBLe paper, the inability to maintain track identities when two people cross in front of each other. The algorithm may get confused and switch labels, and so it may be necessary to use more complex models or other information. The second issue is the lack of labeled foreground and background images, which should be used to train the Gaussian mixtures. However this may be overcome by using the results of the background subtraction method that follows, which does relatively well at separating foreground images from background images. A final issue is seemingly that different background models must be learnt for different videos. This may prove to be too much overhead computation, but that remains to be seen as the algorithm is implemented.

Because of these problems, focus was shifted to improving the performance of the background subtraction algorithm (Section 2). While a system has been developed based on the background subtraction algorithm, further work is necessary to implement BraMBLe so as to improve performance on multiple-person tracking.

2. Background Subtraction Algorithm

2.1. Preprocessing

The original videos came in asf format, and were each on the order of 30-40minutes long. In order to handle this large amount of data, the video was broken up into segments of length ˜37seconds (250 frames) long. This was done by reading the movie into MATLAB (which does its own frame-rate adjustments), cropping the data and saving it to MATLAB .dat files. The data is accessed much quicker in this format than in the original. Splitting the data up also relieves the memory constraints of the computer. However doing so leads to problems when the tracks of people are broken up between to segments, and so in an ideal case the entire video should be analyzed at once. In the development case it was found that a length of 250 frames was sufficient to gauge performance of the algorithm while satisfying the memory constraints of the computer.

After loading the sequence, it turns out that most frames are redundant and not necessary. In other words, frames n and n+1 look extremely similar because there can be very little movement between frames. Thus for the purposes of this algorithm the frame rate was downsampled by 5, and so every 5th frame was analyzed.

2.2. People Detection

In the data sets involved, the environment is extremely constrained and so detection of people becomes relatively simple. Most of the security cameras are situated in hallways or lobbies, with fixed lighting conditions and stationary cameras. Most of the time the background is stationary as well. Occasionally there are doors opening or closing, however during the fire drill they remain shut. There are very few opportunities for occlusions in the foreground as the cameras are located in a position to specifically prevent that. The only major issue faced turns out to be detecting multiple people when they are in groups, occluding each other (see Section 2.2.1).

The main idea of the basic detection algorithm is to use background subtraction. After converting from RGB to grayscale, a background image is first found by taking the mean image of the video segment. This averages out any foreground movements so that only background pixels are left over. Each frame in the sequence is then subtracted from the background image, and the resulting non-zero pixels are taken to be foreground pixels. There are some random background pixels which remain as the result of noise, however. Most of these are discarded by zeroing out those pixels which are within an epsilon of zero, i.e., those pixels were relatively close to the background.

At this point each image is converted to a binary 0-1 image for further processing. In order to fill in any gaps between body parts, a morphological closing operation is ap-

plied to this binary image. The filter that is used is a disk with diameter 5; other masks may be more suitable but this one works well. The closing operation also removes a majority of the rest of the noise in the image. After this series of operations, what is left is a sequence of images containing blobs which represent the people or other moving objects in the image.

During the development process a quick overview of the data proved that there were very few moving objects in the video sequences outside of people. The majority of these uninteresting objects were fairly small, as an example, a small flower moving as a result of a person walking by. Thus in the interest of speed, these smaller objects are removed and the rest are assumed to be people. A smaller number of large non-human objects are also detected but only appear in one frame, such as the shadow of a door that was just opened. These will be taken care of in the tracking portion by discarding those objects whose track only persists for a short period of time (see Section 2.3).

2.2.1 Dealing with Multiple people

Several heuristics were implemented to help deal with multiple people. However none of them worked well for various reasons, thus they are not included in the current version of this algorithm.

One simple measure involves measuring the width of the blob of a person, and if it exceeds some threshold classifying it as several people. However due mostly to the effect of camera angles, single people appear wider in certain locations and orientations, thus this does not work well.

Another attempt involves keeping track of some color information of each person, so that if one crosses in front of another their identities could be maintained. However because people change color as they turn (see Figure 1.1(a)), this does not work well either.

A smarter way to deal with this issue would be to use a head-based or full human appearance model (See Section 1.1). These provide a better ability to distinguish people and even identify them. Work is ongoing to implement the BraMBLe algorithm (Section 1.2).

2.3. Tracking

Once the people are located in each individual image, it is necessary to track them across images. This is achieved using a particle tracker developed by Crocker et.al, available online at http://www.deas.harvard.edu/projects/weitzlab/matlab/.

For each image in the blob sequence obtained above, the centroids of the blobs are located. These points and their frame number are then fed into the particle tracking algorithm. The algorithm is based upon the IDL tracking algorithm developed by Crocker et.al [6]. To assign a track label to a blob, it essentially finds the closest particle in the previous image and assigns that track to the current case. Constraints are set in the maximum translation a particle can move. Also, the algorithm allows a particle to disappear for several frames and still be tracked. This accounts for quick occlusions when people pass in front of each other. Furthermore if a particle appears for less than 2 frames, it is not tracked and assumed to be noise. This accounts for quick changes in lighting and shadows.

See the results of the tracking algorithm in Figure 2.

The algorithm works fairly well in putting together tracks for individual particles. It does get confused when people pass over each other slowly, and in some occasions when people move very quickly such that the displacement is large. However, for the most part, so long as a track is found it is sufficient to make a decision without maintaining identity.

2.4. Decision Making

Finally a decision needs to be made on whether the person is exiting or entering through the exit door of interest, or neither.

The first problem arises in the specification of the exit door. For the purposes of this algorithm, the user is asked to pick four points which define the four corners of the door of interest. This is because there are potentially multiple doors in each situation, and it would be extremely difficult to automatically find the door of interest.

Once the door is specified, it is fairly straightforward to determine whether a track ends or begins within that region. This is the quite simple basis for deciding if a person enters or exits.

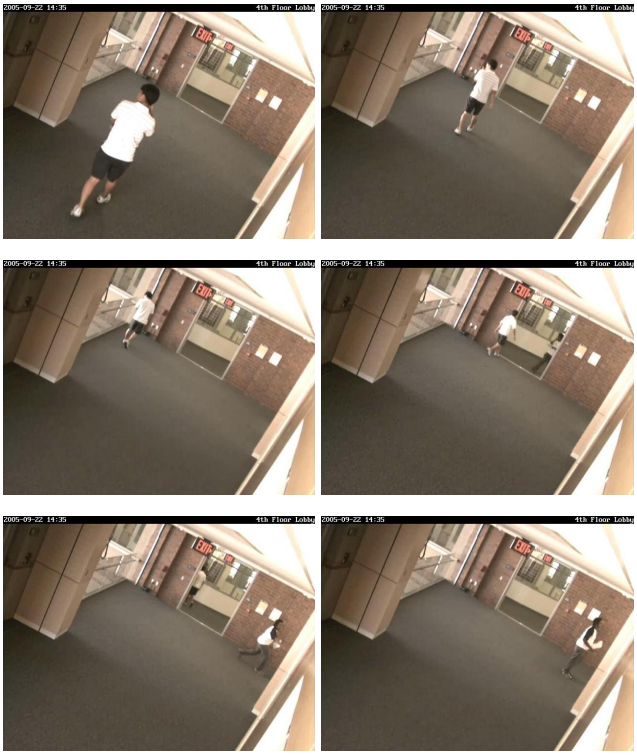
There is much room for extension in this decision making step. One step could determine that if a track splits in two eventually, then two people must have entered. Similar steps could be implemented, however their necessity would be precluded by involving a multiple-person tracker such as BraMBLe.

3. Performance

The preprocessing step of importing the videos turns out to take the most time, however this step is dependent upon platform and input data types. The step takes approximately 4-5 hours for a 45 minute video.

The rest of the algorithm is implemented in MATLAB and as quite unoptimized code. Overall it performs at approximately 2 frames/sec on a 1.6GHz Pentium M. By optimizing the code and porting it to C, a realtime implementation could easily be developed.

Since this algorithm is run offline anyway, performance is not an important criteria. However in the final implemen-



(a)

(b)

Figure 2: (a) A sample sequence from the surveillance footage IrSep22_1409. A person walks in from the bottom at time 22:03 and leaves through the door, while another enters through the door at time 22:13 and stops to the right. (b) The tracks of those two.The detector successfully decides -1 at 22:03 and +1 at 22:13 (see Section 4).

tation it should run as fast as possible to reduce computational time.

4. Results

The following table presents the output of the detector on the video, IrSep22_1409. Sample frames can be seen in Figures 1.1(b) and 2.

Time	$TrueEvents$	$DetectedEvents$
13.08	-1	-1
13.11	-1	-1
14.24	+1	+1
15.13	-1	-1
15:32	-4	$miss$
17.07	-2	-1
17.32	-1	-1
18.00	-6	$miss$
18.51	-1	-1
18.57	-1	-1
19.26	-3	-1
19.52	-3	$miss$
19.53	+1	+1
19.58	+1	+1
20.40	-1	-1
21.12	-1	-1
22.03	-1	-1
23.13	+1	+1

The detector works very well on frames where only one person is involved, or where the people are well separated. However in cases of multiple people, it fails to detect the people moving out the door. Unfortunately it is difficult to place the blame for this on any particular part of the algorithm, as there are many causes for the failures.

As an example, at 18:00 when 6 people leave in quick succession, they move in a large blob such that the center of the blob is never inside the door until the very last few frames. At that point it suddenly jumps from one side of the image to the center, and so the tracking algorithm fails to register that as a single track.

Similar results were found to be the case on two other videos, and more tests are being run. Most of the errors seem to occur on frames where multiple people move around. This is the main reason for moving to a more complex algorithm.

The reader will note the lack of a quantified error rate. The reason for this is explored in the following section.

4.1. Ground Truth

A major issue in quantifying the error is determining the ground truth. In other words, it is necessary to have some baseline with which the output of the algorithm can be compared. On the forefront, it would seem quite simple to have

a person sit and label the data. While this in itself is a daunting task for hours upon hours of video, it is not the real problem. Actually, even determining the nature of what the ground truth is becomes extremely difficult. Unless the person knows what precisely to label, no ground truth can be found.

Thus determining *what* to label is the problem. Take as an example the data given in the results section. The ground truth was clearly massaged to fit a format that would appear nicely alongside the output of the algorithm. For example, a human may decide to clump together two people exiting at 13:08 and 13:11 as one single '-2'. However the algorithm may (and does) split these up into two separate events. The ground truth can thereby easily become ambiguous.

Even if the time of the event is be settled upon (currently it is the first appearance of the person of interest) - a human can easily make mistakes in labeling the time. This would lead to more confusion, especially when multiple events are occurring simultaneously.

Further, at 19:26 in reality three people leave in a clump, where the algorithm detects only one 'person'. This is clearly not a complete failure, since the one 'person' is actually a blob consisting of three real people. However this would be considered a failure in any quantified analysis.

At the end one may say, just compare the total number of people that have left. However this statistic could be inaccurate as the algorithm may make two mistakes and have them cancel out in the end.

Thus it is clear that quantifying the performance of the overall detector is be quite difficult. One might consider looking at the performance of each individual step of the algorithm. However even this is somewhat difficult. For example, in the people-detection step, it is hard to determine when to count a person after they begin moving behind an occlusion (Some have said only when 80% of a person is visible, do they count as a positive). Further, in the tracking step, it is easy for a person to say that, for example, the tracker was wrong to split one track into two. That sentiment is, again, somewhat difficult to quantify.

In summary, all these problems may be overcome in one way or another, but to do so would require extreme care in labeling. This labeling process would be very time consuming and have large room for error. At this point, therefore, a qualitative analysis seems sufficient to gauge the performance of the algorithm.

5. Further Work

As has been enumerated by previous sections, further work is ongoing to expand this algorithm into the domain of multiple-person tracking. The primary thrust of this work is focusing on incorporating the BraMBLe code and adjusting it to suit this data set. It may be possible now to use the output of the background subtraction algorithm as training data for the BraMBLe shape models.

Finally, once a polished algorithm has been produced, it will be necessary to package it in a format palpable to the interested users. This includes potentially porting the algorithm to C and creating a user interface.

6. Conclusions

In summary, an algorithm to track and count the number of people exiting a door in a given surveillance video has been presented. This algorithm has been qualitatively shown to work well on sparsely populated videos, but fail when multiple people and events overlap, as expected. Further work is continuing to implement a more complicated algorithm to deal with these failures.

While the ultimate goal of this project has not been achieved quite yet, it certainly seems to be within reach. More problems were encountered than initially envisioned, but these can, for the most part, be overcome. Hopefully within the next few weeks a polished algorithm will be available for more general use.

References

[1] Siebel, N.; Maybank, S., "Fusion of Multiple Tracking Algorithms for Robust People Tracking," *ECCV 2002*, pp.373–387, 2002

[2] Lipton, A.; Fujiyoshi, H.; Patil, R., "Moving target classification and tracking from real-time video," *Proc. of the Workshop on Application of Computer Vision, IEEE*, pp. 8–14, October, 1998.

[3] Isard, M.; MacCormick, J., "BraMBLe: a Bayesian multiple-blob tracker," *ICCV 2001*, Vol. 2, pp. 34–41, 2001.

[4] Rittscher, J.; Tu, P.; Krahnstoever, N., "Simultaneous Estimation of Segmentation and Shape," *CVPR 2005 2:486-493*, 2005

[5] Zhao, T.; Nevatia, R., "Tracking Multiple Humans in Complex Situations," *PAMI 2004*, pp1208–1221, 2004

[6] Crocker, J.C.; Grier, D.G., "Methods of Digital Video Microscopy for Colloidal Studies", J. Colloid Interface Sci. 179, 298 (1996). http://www.physics.emory.edu/~weeks/idl/ http://www.deas.harvard.edu/projects/weitzlab/matlab/