Histogram-based Detection of Moving Objects for Tracker Initialization in Surveillance Video

Peter Dunne

AD-Group, Daresbury Park, Daresbury,Warrington, WA4 4HS, UK
Applied Digital Signal and Image Processing Research Centre, School of Computing Engineering and Physical Sciences, University of Central Lancashire, Preston PR1 2HE, UK
pdunne@ad-holdings.co.uk

Bogdan J. Matuszewski

Applied Digital Signal and Image Processing Research Centre, School of Computing Engineering and Physical Sciences, University of Central Lancashire, Preston PR1 2HE, UK
bmatuszewski1@uclan.ac.uk

Abstract

We present an approach to localized object detection that is not dependent upon background image construction or object modeling. It is designed to work through camera embedded software using spare processing capacity in a visual signal processor. It uses a localized temporal difference change detector and a particle filter type likelihood to detect possible trackable objects, and to find a point within a detected object at which a particle filter tracker might be initialized.

Keywords: *object modelling, particle filter, tracker initialization*

1. Introduction

Moving object detection and tracking is central to a range of security and business intelligence surveillance video applications. The information derived from the processes can be used to alert security camera monitoring staff to potential trespass or rule violation in sensitive areas. In the case of business intelligence the information can be used to gather positive statistical information relating to customer behaviour.

The approach described in this paper is one of a range of analytics components being developed for use with the Chipwrights range of visual signal processors(VSPs) [1] as camera embedded software to provide low and intermediate level data for subsequent user analysis and interpretation. The VSPs have a primary role of dealing with image capture, intensity correction, compression, camera multiplexing etc., but there is spare processing capacity available for analytics tasks. The component that we describe is a computationally economic method of object detection for counting and possible tracker initialization.

Recent developments, for example [2,3,4], integrate initialization into the overall management of the trackers. They draw upon aspects such as 3D colour representations, spatiotemporal pixel homogeneity, optical flow etc. in order to identify objects within the images. Typically tracker initialization is carried out when detected objects cannot be

accounted for by existing tracked targets yet satisfy criteria relating to size, shape, visibility, non-occlusion etc. But sophisticated approaches like those require computational commitments and feature information that are not available in our case.

We aim for grey scale image processing as artificial illumination such as fluorescent and sodium lighting can result in what is essentially monochrome footage. To accommodate both the multitasking requirements of the VSP and multiple object tracking we aim to process 5 frames per second(fps).

Some recent developments contain aspects that are closer to our requirements. For example Girisha and Murali [5] describe a method for segmenting motion objects from background that is based upon temporal frame differencing. They look at the correlation between pixel values at a given location over three sequential frames in order to determine if they represent background or belong to a foreground object. Ko et al.[6] describe a segmentation technique in which the background at each pixel location is represented by a histogram of values in the region surrounding that location. Foreground is detected using a thresholded Bhattacharyya distance between the current histogram at the pixel location and that of the temporally recursive updated background distribution.

We have avoided supervised classification approaches because of their likely demands on the processing capacity of the single VSP. In addition, our preference is for a generalized potential target detector. We avoid full frame background modeling and use temporal frame differencing to trigger our detector. As the process is not blob based we avoid difficulties associated with morphological processing.

We track objects using a grey scale histogram based particle filter and use the integral histogram approach [7] in order to extract multiple histograms efficiently. Our histogram based detector uses the same resources as the particle filters used for subsequent tracking hence keeping overall computation costs low. Rather than searching the whole image for tracker initiation at each frame we focus on expected target entry locations.

2. Method

2.1. Event trigger

The detector, which we refer to as a ‘pad’, covers an image region with dimensions half that of the bounding box of the expected target objects. Pads can be placed singly at appropriate places in an image or can be placed as arrays to form a tripwire type detection. Typical pad placement can be seen in Figure 1. The event trigger is based upon the absolute differences between pixel values in the current frame and the corresponding ones in the preceding frame. With a static camera the differences are generally small in the absence of objects and significant when an object moves in the pad region.

The first step in the process is the construction of the thresholded pad temporal difference image $\boldsymbol{D}_t$.

$$\boldsymbol{D}_t(x,y) = \left(\left|\boldsymbol{P}_t(x,y) - \boldsymbol{P}_{t-1}(x,y)\right| > \theta_D\right) \quad \forall (x,y) \in \boldsymbol{P} \tag{1}$$

where $\boldsymbol{P}_t$ is the pad image at time t and θ_D is a pixel difference threshold. A typical $\boldsymbol{D}_t$ is shown in Figure 1(c). With the low frame rate the pixel value changes from frame to frame can be significant.

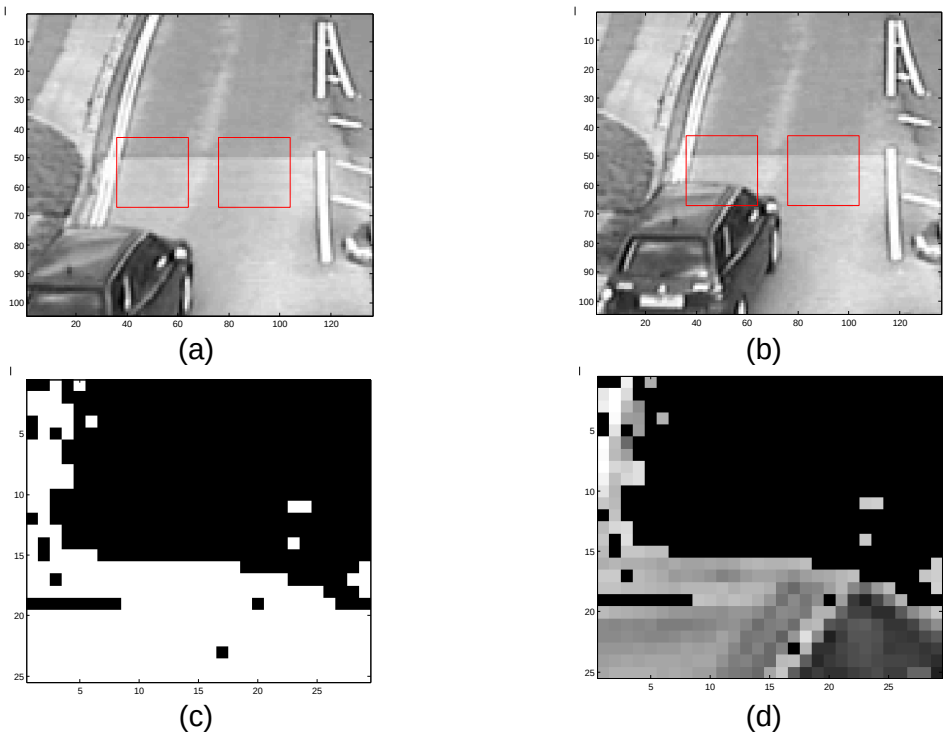


Figure 1. (a) pads 1 and 2 in frame at (t-1) (b) pads 1 and 2 in frame at (t) (c) D_t, pad(1) temporal difference mask (d) M_t, difference masked pad(1) image

The mean difference $\hat{d}_t$ across the pad is calculated:

$$\hat{d}_t = \frac{1}{N} \sum_{x,y \in \boldsymbol{P}} \boldsymbol{D}_t(x,y) \tag{2}$$

where N is the number of pixels in the pad image. A trigger occurs when $\hat{d}_t$ rises through a mean differences threshold θ_D and the mean differences are increasing:

i.e. $$\hat{d}_t > \theta_d \wedge (\hat{d}_t - \hat{d}_{t-1}) > 0 \tag{3}$$

A range of alternative trigger conditions are possible, the essence is to detect change in the pad difference image.

The video sequence shown in the figure is from the AVSS 2007 i-LIDS Challenge dataset [8]. The images were downsized by a factor of 2 and we took every 5th frame to simulate the target 5fps rate.

2.2. Extracting object information

In the absence of a trigger we make a normalized 8 bin 'probably background' grey scale histogram $\boldsymbol{q}_B^t$ of the pad image at time *t*. An integral histogram of the whole image is made at

each frame, to be used for the particle filter tracking, so this background histogram can be extracted without much extra computation. The histogram is used to recursively update a time averaged 'probably background' histogram $\tilde{\boldsymbol{q}}_B^t$:

$$\tilde{\boldsymbol{q}}_B^t = (1-\alpha)\tilde{\boldsymbol{q}}_B^{t-1} + \alpha \boldsymbol{q}_B^t \tag{4}$$

where α has a value typically of the order 0.2.

When a trigger condition is met we make a histogram $\boldsymbol{p}^t$ of the pad image. This contains information about both background and object. We use this together with $\tilde{\boldsymbol{q}}_B^t$ to make an 'indicative object' proposal histogram $\boldsymbol{q}_o^t$. The first step is to construct a difference histogram $\boldsymbol{q}_d^t$ by subtracting the averaged 'probably background' histogram from the current pad histogram:

$$\boldsymbol{q}_d^t = \boldsymbol{p}^t - \tilde{\boldsymbol{q}}_B^t \tag{5}$$

Negative bin values in the difference histogram indicate pixel value ranges that are more representative of the background rather than the foreground. Figures 2(a)-(c) show a set of typical histograms associated with this stage of the process.

Next, a difference masked pad image $\boldsymbol{M}_p$ (Figure 1(d)) is made by pixelwise multiplication of the pad thresholded temporal difference image with the pad image i.e.

$$\boldsymbol{M}_p(x,y) = \boldsymbol{D}_t(x,y) \cdot \boldsymbol{P}_t(x,y) \quad \forall (x,y) \in \boldsymbol{P} \tag{6}$$

The masked image will contain pixels associated with the object, shadows, ghosts, and zero values associated with the static region. The masked image is grey level sliced using the bin boundaries of the $n = 8$ bin histogram to produce a binary pad image $\boldsymbol{B}_p^u$ for each slice u:

$$\boldsymbol{B}_p^u(x,y) = \delta(b(\boldsymbol{M}_p(x,y)) - u) \quad \forall (x,y) \in \boldsymbol{P}, \quad u = 1,...,n \tag{7}$$

where $b(\boldsymbol{M}_p(x,y))$ is a function that compares the grey scale value $\boldsymbol{M}_p(x,y)$ against the bin boundaries and returns the bin number associated with it; δ is the Dirac delta function. The lower boundary of the first bin is set to 1 rather than 0 out of the range [0, 255]. This removes the mask static pixel contribution. The bins of the proposal histogram $\boldsymbol{q}_o^t$ are set to be the sums of the corresponding binary slice if the associated difference histogram bin $\boldsymbol{q}_d^t$ is positive, otherwise they are set to zero:

$$\boldsymbol{q}_o^t(u) = \begin{cases} \sum_{x,y \in \boldsymbol{P}} \boldsymbol{B}_p^u(x,y) & \text{if } \boldsymbol{q}_d^t(u) > 0 \\ 0 & \text{otherwise} \end{cases} \tag{8}$$

The 'indicative object' proposal histogram is then normalized:

$$\tilde{\boldsymbol{q}}_o^t : \tilde{q}_o^t(u) = \frac{q_o^t(u)}{\sum_{v=1}^{n} q_o^t(v)} \quad u = 1,...,n \tag{9}$$

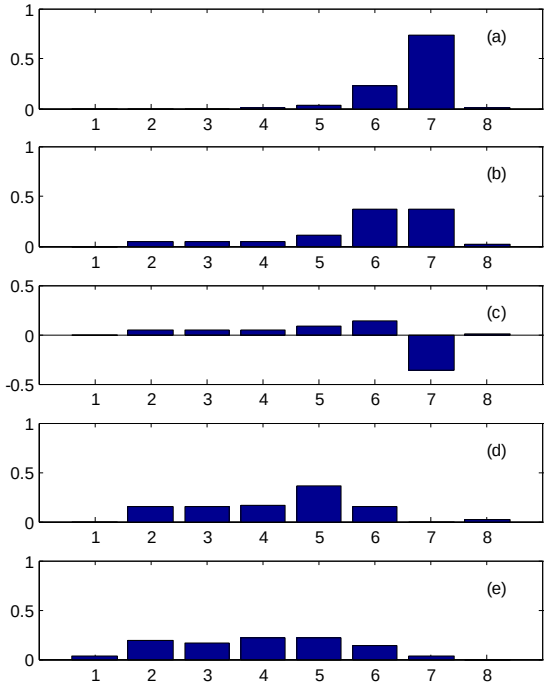


Figure 2. Histograms: (a) averaged probably background $\tilde{q}_B^t$, (b) pad image p^t, (c) differences $q_d^t = p^t - \tilde{q}_b^t$, (d) proposal $\tilde{q}_o^t$, (e) tracking histogram q_o^t extracted at the detect point

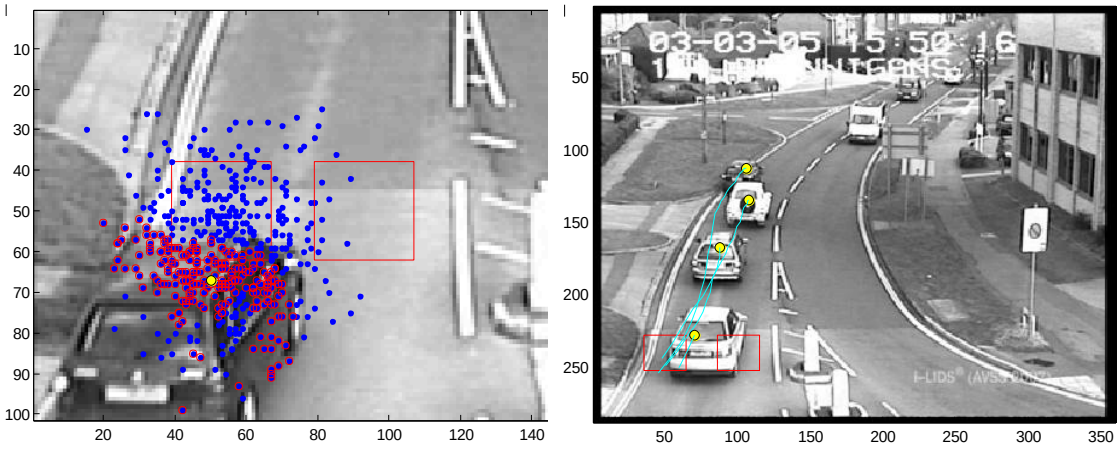
The proposal histogram, shown in Figure 2(d), indicates the grey scale values which are more characteristic of the part of the object entering the pad region than the background. We use this proposal histogram and the centre of gravity of the pad pixelwise difference image as the starting points of a search for a more representative histogram of the incoming object. Even though the proposal histogram is not a true representation of the object histogram, it is likely to be closer to it, in terms of the Bhattacharyya distance, than to histograms calculated from background in the region of the pad after the triggering event. The proposal histogram does not include bins that are dominant in the probably background histogram.

2.2. Updating object information

We use a particle filter type search of the region around the pad to get a better representation of the object. A prior is constructed by adding Gaussian random values to the centre of gravity $\boldsymbol{x}_0$ of the pad difference image $\boldsymbol{D}_t$:

$$\boldsymbol{x} = \boldsymbol{x}_0 + \nu \quad \text{where} \quad \boldsymbol{x} = (x, y) \quad \text{and} \quad \nu_i \sim \text{N}(0, \sigma) \tag{10}$$

The standard deviation σ of the distribution is taken to be half of the largest dimension of the detect pad. A typical particle prior is shown in Figure 3(a).



(a) detect (b) merged tracks

Figure 3. (a) triggered pad, showing the prior particle set (blue), the selected high weight set (red) and the object proposed detection(yellow) (b) typical merged tracks from detect points

Histograms $\boldsymbol{p}^j$ are extracted using the integral histogram at the j particle locations. The particles are given a Gaussian weight w^j in terms of their Bhattacharyya distance S from the 'probably object' histogram $\tilde{\boldsymbol{q}}_o^t$ to give a posterior distribution i.e.

$$S^j = \sqrt{1 - \sum_{u=1}^{n} \sqrt{p_u^j \tilde{q}_{o,u}^t}} \tag{11}$$

and

$$w^j = \exp\left(-\frac{S^{j2}}{2\sigma^2} \right) \tag{12}$$

The standard deviation of the weighting distribution is set at $\sigma = 0.2$ [9].

The object locations $\boldsymbol{x}_{object}$, shown in Figures 3 and 4, are obtained using a weighted mean of the distribution of particle locations:

$$\boldsymbol{x}_{object} = \sum_{j=1}^{N} \tilde{w}^j \boldsymbol{x}^j \tag{13}$$

where $\tilde{w}^j$ is the normalized weight of the j^{th} particle:

$$\tilde{w}^j = \frac{w^j}{\sum_{i=1}^{N} w^i} \tag{14}$$

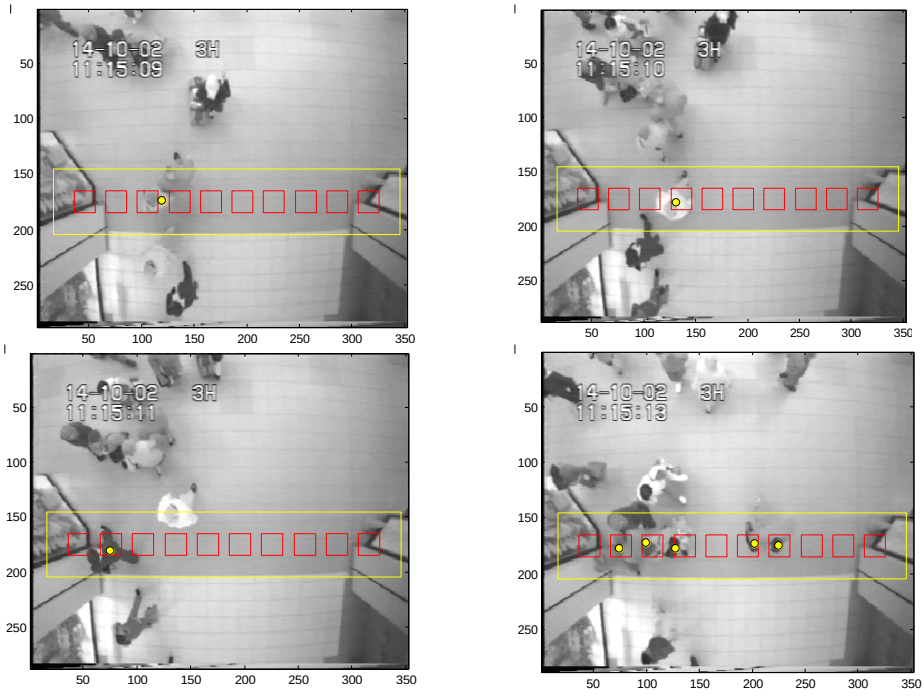


Figure 4. Examples of typical detections of pedestrians in a shopping mall, using an overhead camera

The final step of the detection process is to extract a tracking histogram at the object location and initialize a particle filter tracker at that point. An 8-bin tracking histogram $\boldsymbol{q}_o^t$ extracted at the object location in Figure 3(a) is shown in Figure 2(e). The histogram is extracted from a rectangular region with dimensions half those of the expected bounding box at the object location. In some cases, for tracking purposes, the rectangular region is split into four quadrants and a 32 bin histogram is constructed by concatenating four 8-bin histograms from those quadrants. This gives a tracking histogram with some indication of the spatial layout of the pixel values representing the foreground.

The objects are tracked using a simple SIR particle filter [10]. The particle state is described by the state vector $\boldsymbol{s}_t = [x, v_x, y, v_y]^T$. The particle filter initial velocities are taken to be zero but they pick up the velocity of the object within a couple of iterations. Typical tracks are shown in Figure 3(b).

3. Implementation

The implementation of the method is illustrated in two situations presenting different difficulties. The first one has an oblique view with slight camera movement, rigid objects and some slight illumination variation. The second with a stable camera, constant illumination, and non-rigid objects having various shapes and appearances.

Figure 3 shows a two pad array being used with the i-LIDS Challenge footage. In practice the pads can produce more than one trigger per object if passing objects are in contact with the pads for a number of frames. There are a range of heuristic strategies that can be employed to deal with trigger multiplicity. One approach is to disable the pad after the first trigger and then re-enable it once the tracker reports that the objects has cleared the area. However it is often the case that the second trigger gives a more representative histogram of

the object so it can be useful to disable after an immediate second one. In the case of target overlaps, due to closely moving traffic for example, a re-enabled pad would trigger on the following object and initiate a new tracker. In such a situation a blob based system would have to assume occlusion and fit multiple models; the pad and histogram approach is capable of identifying separate objects within a blob and initiating independent trackers. An alternative is to keep the pads active and track all triggers independently; trackers that appear to have similar positions, velocities and tracker histograms are then merged into one. A typical tracker combination outcome is shown in Figure 3(b).

Figure 4 shows the use of an array of pads with an overhead camera in a shopping mall scenario. The target objects in the mall example can challenge impositions of a pedestrian model: shopping trolleys, luggage and other baggage can present a variety of shapes to the system. The potential of the approach can be seen in this sequence: with appropriate choice of trigger threshold the shopping mall sequence detected 22 out of 23 objects, returned 3 false positives (triggering on temporal difference ‘ghosts’) and 1 false negative. This returns values of precision, sensitivity and F-score [11] of the order 0.9. The heuristics are in continuing development.

4. Conclusion

We have presented a computationally economic approach to object initial detection for counting and possible tracker initialization. It is one of a number of approaches developed to use spare processing capacity for embedded analytics in intelligent cameras. It has potential for development in contexts such as vehicle or pedestrian traffic density indication, or as a tracker initializer in situations such as tripwires or perimeter violation detection. The method does not demand the use of a target model nor does it require the development of a full background image or classifier training. It works with moderate quality monochrome footage and can be used in a range of contexts.

References

[1] Chipwrights, "Programmable Visual Signal Processors", www.chipwrights.com
[2] E. Maggio, M. Taj, and A. Cavallaro, "Efficient Multi-Target Visual Tracking using Random Finite Sets", IEEE Transactions on Systems and Circuits for Video Technology, 2008, pp1016-1027.
[3] A. Bugeau and P. Peréz, "Track and Cut: Simultaneous Tracking and Segmentation of Multiple Objects with Track Cuts", EURASIP Journal on Advances in Signal Processing, 2008, pp1-14.
[4] Z. Zhang, P. L. Venetianer, and A. J. Lipton, "A Robust Human Detection and Tracking System using a Human-Model-Based Camera Calibration", 8th International Workshop on Visual Surveillance, 2008,
[5] R. Girisha and S. Murali, "Segmentation of Motion Objects from Surveillance Video Sequences using Partial Correlation", 6th IEEE International Conference on Image Processing, 2009,
[6] T. Ko, S. Soatto, and D. Estrin, "Background Subtraction on Distributions", 10th European Conference on Computer Vision, 2008, pp276-289.
[7] F. Porikli, "Integral Histogram: A Fast Way to extract Histograms in Cartesian Spaces", IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2005, pp829-836.
[8] Home Office Scientific Development Branch, "i-LIDS Dataset for AVSS 2007 Sequence: AVSS pv medium", 2007, www.elec.qmul.ac.uk/staffinfo/andrea/spevi.html
[9] P. Dunne and B. J. Matuszewski, "Choice of Similarity Measure, Likelihood Function and Parameters for Histogram-based Particle Filter Tracking in CCTV Grey Scale Video", Image and Vision Computing, 2011, pp178-189.
[10] S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp, "A Tutorial on Particle Filters for On-Line Non-Linear/Non-Gaussian Bayesian Tracking", IEEE Transactions on Signal Processing, 2002, pp174-188.
[11] A. Baumann, M. Boltz, and J. Ebling et al., "A Review and Comparison of Measures for Automatic Video Surveillance Systems", EURASIP Journal on Image and Video Processing, 2008, Article ID 824726, 30 pages, doi:10.1155/2008/824726