Automatically Locating an Area of Interest and Maintaining a Reference Image to Aid the Real-Time Tracking of Objects

Matthew T. Cornish and Jonathan P. Wakefield

Division of Computing and Control Systems
School of Engineering
The University of Huddersfield
Queensgate, Huddersfield. HD1 3DH ENGLAND
Email : m.t.cornish@hud.ac.uk

Abstract

Real-time tracking systems often make use of a reference image and, where processing power is limited, an 'area of interest'. To obtain such information commonly requires user interaction. Typically, a reference image is obtained by the user capturing an image when the objects being tracked are not present in the scene and an 'area of interest' may also be defined by the user; both being specific to a particular scene. An alternative approach that would automatically generate and maintain an up-to-date reference image is investigated in this paper. This consists of, essentially, 'cutting and pasting' areas of image from a sequence of frames to obtain an image containing no moving objects. Furthermore, a method for automatically generating an 'area of interest' is described. This method identifies areas of movement in a sequence of frames in order to build the 'area of interest'. These techniques have been successfully developed and proven using video sequences of more than one traffic roundabout.

1. Background

This paper describes the early progress achieved in the development of a real-time tracking system. The work carried out provides foundations useful to many real-time tracking systems and is not specific to any particular application.

In real-time tracking, to speed up the locating of objects and reduce the processing power necessary, it is often desirable to identify the 'area of interest' (AOI). It is, also, common to have a reference image, to show the appearance of the scene without the presence of the object(s) being tracked.

The example application considered in this paper is a vehicle tracking system for use on complex road junctions. So far, we have, specifically, focused on roundabouts (see figures 5.1 & 5.5). Here, the AOI is the road and the reference image

BMVC 1996 doi:10.5244/C.10.11

would show the road with no vehicles on it. It is envisaged that this knowledge of the road scene will enable a more efficient search for vehicles.

The hardware used for this work is an IBM PC compatible Pentium™, capturing video or camera footage through a PCI frame grabber. Given this relatively limited processing power, there is a need to reduce the necessary processing to a minimum.

2. Review of Traffic Tracking

Knowledge of the area of interest in a scene is often used to predict likely locations for the object(s) to be tracked and to reduce the processing power necessary. This area is typically pre-defined by the user (e.g. using a mouse). For example, Thomson, et. al., [1] define two points in the scene that can be used to count vehicles and measure their speed. Campbell, et. al., [2] use neural networks to automatically recognise features in a scene, such as road, grass, etc. This approach could be used to identify the area of interest in a scene, such as the road in a traffic tracking application, though this was not the purpose of their system. Marchant [3] describes a system for creating masks of individual farmyard animals from a sequence of frames. This is similar to the generation of an AOI described in this paper, however, we require the full paths of all the moving objects in the image.

Commonly, a reference image is taken from footage of the scene when the object to be tracked is not present [1][4][5][6][7]. However, this is not always convenient, as the scene may be in constant use. These reference images must also be adjusted to take account of ambient lighting conditions, shadows and camera movement, etc., though shadows, reflections and camera movement are particularly difficult to compensate for. Twilight was cited as a particular problem, as this is when the ambient light, shadows and reflections change very rapidly and also tends to occur at the same time as 'rush hour' [4].

Cruz, et. al., [8] describe a system that can create a reference image from a scene incorporating the object to be tracked, using a technique they call 'temporal smoothing'. This increments or decrements grey-levels in the reference image according to whether the corresponding grey-level in the current frame is higher or lower. However, this technique does not respond to rapid changes in light intensity, such as is caused by clouds, and tends to leave 'car trails', where cars have been in the past. Also it requires dedicated hardware to achieve sufficient speed.

3. Generating a Map Image

3.1 Introduction

A 'map' of the road (or, more generally, AOI), where moving vehicles have, previously, been detected would allow processing to be restricted to just the road area, within the image. It would also mean that, in terms of tracking, if an object leaves the mapped area, it can be said to be no longer on the roundabout.

3.2 A Basic Algorithm

In its simplest form, a map could be built up by repeatedly **adding** the result of **differencing** consecutive frames into an, initially blank, map image (Figure 3.1). The idea being that **differencing** would detect moving vehicles along their paths. By repeating the process, vehicle movement would be detected all the way around the roundabout.

$$M = \sum_{n=1}^{k} \left| F_n - F_{n-1} \right|$$

where M = *map image,* F = *frame,* n = *frame no.,* k = *total no. iterations.*

Figure 3.1 : Simple Map Building Algorithm

However, because of the algorithm's cumulative nature, the map also contains all of the noise (falsely interpreted as movement) detected throughout the process. It was found that there is quite a lot of noise associated with the images taken from a video source. This posed a problem when trying to set a threshold level that would extract the vehicle paths from the background noise in the map image, and was made more difficult by the level of noise varying. This variance depends many factors; such as ambient lighting conditions, the number of iterations used to build the image and the amount of traffic flow. So, rather than attempting to remove the noise from the map image, each **difference** image was **thresholded** *before* being added into the map image.

As the range of grey-levels can vary substantially between different scenes, a threshold level was re-calculated for each scene at one third of the grey-level range.

3.3 Opening

In order to remove any noise in the map image, an opening filter (an erode followed by a dilate) was used. **Opening** was chosen in preference to a simple anomalous pixel filter, as noise pixels often occurred in clusters. One of the major causes of these 'clusters' was 'camera shake' (there are few sturdy locations to place a video camera at road junctions).

As well as its effects on the background, noise also affected the quality of the vehicle images in the map. Noise, shadows, variations in grey level, windscreens (which may be the same colour as the background), etc., combined in the difference image to yield 'broken' images of vehicles. Therefore, a balance had to be found, in designing the noise filter, such that it could discriminate between these 'broken' vehicle images and actual noise.

3.4 Smoothing

A technique that proved far more successful than the above process for removing noise from the map image was to **smooth** the frames prior to all other

processing. This also had the effect of producing more 'solid' vehicle images. Followed by **thresholding**, this technique produced a continuous map of the road. Any remaining noise was then removed with a single **opening**.

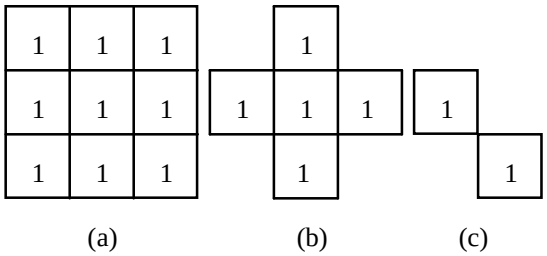

Figure 3.2 : Filter Configurations Tested for Smoothing Filter

These smoothing filter configurations were tested with various different weightings in order to reach a balance between speed and effectiveness. The mask shown as figure 3.2a best met this criterion.

3.5 The Complete Map Building Algorithm

Figure 3.3 shows the processes described above placed in order of execution. Only one new frame is grabbed each cycle, the timing derived from the processing time for each frame. A delay between each grab is desirable, as this allows vehicles to move to yield the maximum area (not overlapping) when the two frames are **differenced**. By updating two frame buffers alternately, only one frame has to be **smoothed** (the longest process) each cycle and maximum use is made of the host machine's resources.

- Grab New Frame
- *Smooth* New Frame
- REPEAT
  - Make New Frame be Old Frame
  - Grab New Frame
  - *Smooth* New Frame
  - *Difference* Old and New Frame
  - *Threshold* Difference Image (Threshold set to 1/3 Grey Level Range)
  - OR Thresholded Difference Image into Map Image
  - Count New Pixels Added to Map Image
  - *Open* Map Image
  - *Dilate* Map Image
- UNTIL No New Pixels are Added to Map Image
- Map Complete

Figure 3.3 : The Final Map Generating Algorithm

3.6 Processing Each Frame in Bands

Section 3.5 described how a time delay between the two frames being differenced is desirable to obtain maximum yield of area for each vehicle. It is also the case that frames (and, therefore, vehicles) should not be too far apart, as more efficiency can be gained by logging the full path of a vehicle, rather than occasional positions.

In order to grab frames alternately and close enough together, in time, the image had to be divided into several bands. Each time a new frame was captured, only one of these bands would be processed, the next band being processed for the next pair of frames. The width of each band, and thus, the processing time for each pair of frames, was adjusted such that vehicles in the next pair of frames had moved to give the maximum yield for each **differencing**.

Bands were processed going both down and then up the image (figure 3.4), as vehicles could be captured moving down and up the image through consecutive bands. This gave an equal result for both sides of roads vertically oriented in the map image.

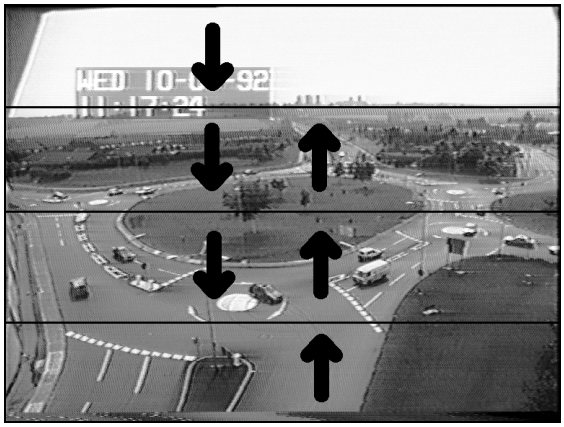

Figure 3.4 : A Frame Divided into Bands Showing the Order of Processing

3.7 Automation

An important aspect of this work was to make the system fully automated. To this end the map building process had to be designed to stop when all the road had been found (i.e. when no more, significant, movement could be detected).

The algorithm, figure 3.3, shows how the number of new pixels (pixels added to areas already mapped are discounted) being added into the map each iteration, are counted. When the number of new pixels remains steady for some time the process stops. The number of new pixels being added is filtered, using a running average, to prevent the process from stopping if several frames go by with no pixels (vehicles) being added into the image. Also, from the difference between the average number of pixels and the number of new pixels, the level of background noise is calculated and accounted for (which is significant, since the level of background noise can vary considerably between scenes).

New pixels were, actually, counted before the **opening** to remove noise. Otherwise, a buffer stage would have to be added, where noise pixels could be removed, counted and then this number subtracted from the number of new pixels added into the map. This would be an additional computational expense on, already, limited resources.

3.8 Coding the Map Image

Because the map image was to be used, both, to reduce the search area for vehicles and to minimise necessary image processing, it was necessary to code the image in two ways. Each method of coding was appropriate to the different uses of the map image, although both represent identical areas.

The search area was described by the map image as a binary image, white representing areas to be processed and black representing areas not to be processed. This was effective with our simple tracking algorithm, which would frequently predict the next location of a vehicle to be outside the road area.

Image processing was limited by run-length encoding the map image. Each entry in the RLC (run-length code) holds the start and end point of a given row in the image, each row being simplified to only one run. Thus, rather than check before processing each individual pixel, *all* the pixels between the start point and end point are processed. This leads to a simple and efficient implementation.

4 Generating a Reference Image

4.1 Introduction

Many traffic tracking systems use some kind of reference image[4][5][6][7][8]. The reference image shows the appearance of the object scene without the presence of the object being tracked. A simple **differencing** of the reference image from the object scene yields just the object.

4.2 Generating a Reference Image

The approach adopted for creating a reference frame was, essentially, one of 'cut and paste'. Here, the reference image was built up by copying into it areas from a sequence of frames in which there was no movement. Holes where vehicles were present, eventually being 'pasted over' by image content from other frames. This was, actually, implemented on a 'pixel by pixel' basis, comparing corresponding pixels in two frames, temporally far enough apart to allow vehicles to have completely moved between them, and, if they were the same, averaging them into a new frame (the reference frame).

Figure 4.1 shows the algorithm used to perform this process. Each frame was **smoothed** beforehand, to reduce noise, and the threshold value was set to 1 grey level. This threshold was intentionally chosen to be severe, rejecting all but definitely desirable scene information. Although this meant that some 'satisfactory' scene information was rejected, this could be obtained from subsequent frames. The

algorithm took approximately 10 iterations before the reference image was virtually fully generated.

$$if \quad \left| P_{F_n}(x,y) - P_{F_{n-1}}(x,y) \right| < Threshold$$

$$then \quad P_{R_{n+1}}(x,y) = \left(P_{F_n}(x,y) + P_{R_n}(x,y) \right) / 2$$

$$where\ P_F = pixel\ value\ in\ frame,\ P_R = pixel\ value\ in\ reference,\ n = frame\ no.,$$
$$x = row,\ y = column$$

Figure 4.1 : Reference Generating Algorithm

The equal weighting for P_{Fn} and P_{Rn} was chosen, as a result of experimentation, to allow a reasonably quick change in the reference for lighting changes, while, not allowing momentarily stationary cars waiting to enter the roundabout to appear in the reference image.

It is intended that this process run continually, as a background process to tracking, to provide an up-to-date reference image and may, typically, be updated every two seconds, taking around one fifth of a second to execute. In this way, the reference frame would be, temporally, just behind the current frame.

5 Experimental Results

Figures 5.1 to 5.8 show experimental results for the techniques described in this paper. Two typical roundabout scenes which were used to test these techniques are shown in figures 5.1 & 5.5. These scenes both show many cars that are clearly visible, but also some cars that are virtually impossible to detect, even by the human visual system.

Figures 5.2 & 5.6 show the mapped area in each scene, the remainder of the image being blacked out. As can be seen, the road area has clearly been correctly identified. In figure 5.6, parts of the car park have been identified as road. This is due to the movement of vehicles in this area of the scene. Although this is not desirable, it is consistent with the correct operation of the system, and will not cause any problems, apart from an increase in the area requiring processing.

From figure 5.6, it may seem that some areas of the image containing no moving vehicles (e.g. the buildings at the back of the image) have been falsely identified as road. This has been caused by the presence of some particularly large vehicles (e.g. articulated lorries) in previous frames. This is not a problem since, although this area does not precisely correspond to the road, it is an appropriate area to search for vehicles.

The reference generator gave exceptionally good results for the two scenes. As can be seen in figures 5.3 & 5.7, no cars are present on the road. Figure 5.7 also shows how parked cars are interpreted as background as they are not moving.

Finally, figures 5.4 & 5.8 show the results of a simple centroid tracking system, implemented to test the usefulness of the map and reference images. In each, a car has been successfully tracked across a portion of the roundabout, shown by the highlighted trail behind each vehicle. Thus, demonstrating the validity of this approach.

Figure 5.1 : A Typical Roundabout Scene

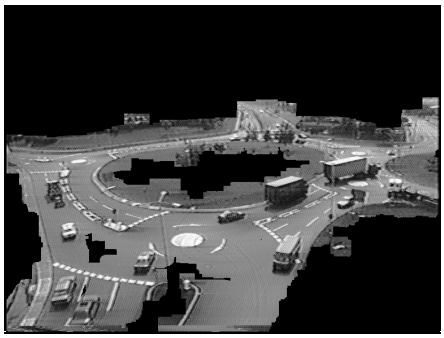

Figure 5.2 : The Mapped Road Area

Figure 5.3 : The Reference Image (Showing all Moving Objects to have been Removed)

Figure 5.4 : Highlighted Path of a Vehicle

Figure 5.5 : A Typical Roundabout Scene

Figure 5.6 : The Mapped Road Area

Figure 5.7 : The Reference Image (Showing all Moving Objects to have been Removed)

Figure 5.8 : Highlighted Path of a Vehicle

6 Conclusions

This paper has described work carried out that provides useful foundations for real-time tracking applications; namely, a method for automatically generating a map image and a method for automatically generating a reference image. These have been successfully proven to work independently of user interaction and used, in combination, as part of a vehicle tracking system, to track vehicles in more than one roundabout scene.

7 Further Work

Arising from this work, additional scene information could be automatically generated and that would be of use. This information includes: vehicle entry/exit points, an estimation of the 'ground plane' and an estimation of velocity vectors for vehicles to help deal with occlusion.

Additionally, it is now our intention to develop the tracking algorithm to its conclusion.

8 References

[1] Thomson, M. S., Wan, C. L., Binnie T. D. (1993) 'Development of a Real-Time Image Analysis System for Traffic Monitoring Applications.' *Knowledge-Based Systems for Civil and Structural Engineering pp.189-195*

[2] Campbell, N. W., Mackeown, P. J., Thomas, B. T., Troscianko, T. (1995) 'Automatic Interpretation of Outdoor Scenes.' *British Machine Vision Conference **1** pp.297-306*

[3] Marchant, J. A., (1992) 'Accurate Boundary Location from Motion' *Britsh Machine Vision Conference pp 89-92*

[4] Chou, Y. J., Sethi, V. (1993) 'Machine Vision Based Traffic-Adjusted Intersection Signal Control.' *Proc. Conf. Digital Image Tech. Appl. Civ. Eng. pp.150-157*

[5] Nicchiotti, G., Ottaviani, E. (1994) 'Automatic Vehicle Counting from Image Sequences.' *Proceedings, 4th Int. Workshop, Time Varying Image Proc. Moving Obj. Recognition **3** pp.410-417*

[6] Sethi, I. K., Brillhart, W. L. (1991) 'Traffic Analysis Including Non-Conforming Behaviour via Image Processing.' *Proceedings - Society of Automotive Engineers **p-253** pt:1 pp.193-201*

[7] Dagless, E. L., Ali, A. T., Cruz, J. B. (1993) 'Visual Road Traffic Monitoring and Data Collection.' *Proceedings of the IEEE-IEE Vehicle Navigation and Information Systems Conf. pp.146-149*

[8] Cruz, J. B., Ali, A. T., Dagless, E. L. (1993) 'A Temporal Smoothing Technique for Real-Time Motion Detection.' *5th Int. Conf. Computer Analysis of Images and Patterns*