Colour Based Object Tracking

T.D.Grove and K.D.Baker
Department of Computer Science, University of Reading, Reading RG6 6AY, England.
Email: T.D.Grove@reading.ac.uk, K.D.Baker@reading.ac.uk

T. N. TAN
National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences, Beijing, China.
Email: TNT@prlsun2.ia.ac.cn

Abstract

A method of detecting and tracking objects using colour is presented. We track objects using a simple colour histogram based technique, which is fast and more robust to occlusion and camera motion than simple "blob" based techniques.

Keywords: colour vision, object tracking, motion segmentation, visual surveillance

1. Introduction

Unlike many other image features (e.g. shape) colour is relatively constant under viewpoint changes. Using colour also provides, as will be discussed later, some robustness with respect to occlusion. Another property of object colour that is particularly attractive is the ease with which it can be acquired. In the intended application (visual surveillance) of our work, this is attractive since systems that must continuously monitor vehicle or pedestrian activity should operate in real-time (which we define to be video rate, i.e. 25 frames/second) in order to be of maximum usefulness. We note that colour is not always appropriate as the sole means of tracking objects but the low computational cost of the algorithm we present here makes colour a desirable feature to exploit when appropriate.

2. Background and related work

A system that tracks objects typically has two components: an event detection component that locates potential objects and a tracking component that identifies and follows these objects as they move through the image. Typical approaches to event detection are background subtraction, image differencing and feature grouping.

Once objects have been detected they are tracked. The simplest tracking systems simply find similar moving regions (e.g. "blobs" obtained asconnected components in the difference image) in adjacent frames. Simple blob tracking is fast and can be implemented in real-time but performs very badly in the presence of occlusion. Other systems use more complex 2D shape descriptions [2][5] or 3D geometric models [3], both of which are less sensitive to occlusion. Unfortunately, as systems become more complex, the cost of tracking becomes greater. McKenna et al. [1] describe a face tracking system based on colour that is able to track a number of people in real-time. The technique we describe in this paper is similar to that developed by McKenna et al, but is novel in that it is intended to track arbitrary objects rather than just faces.

3. Outline of the approach

In remainder of this paper we describe our approach to colour based object tracking and discuss its performance.

3.1 Background modelling and event detection

We take the following approach to event detection: given a set of images representative of the background against which objects will appear, we create a Gaussian mixture model (GMM) to describe the distribution of colours within these images. Colours are represented as two element hue and saturation vectors. The GMM takes the form:

$$P(x) = \sum_{i}^{n} \frac{w_i}{2\pi\left|\Sigma_i\right|^{-1}} \mathbf{e}^{\frac{(x-\bar{u}_i)\Sigma_i^{-1}(x-\bar{u}_i)^T}{2}}$$

Where $\bar{u}_i, \Sigma_i, w_i$ are the mean colours, covariances and mixture weights of the n Gaussians. These parameters are estimated using the standard expectation maximisation (EM) technique [4]. Figure 1 illustrates a typical distribution obtained after fitting a GMM to the colours of pixels ina car park image. Four Gaussians were used in this case, with individual Gaussians covering those parts of the colour space which roughly correspond to the colour of the

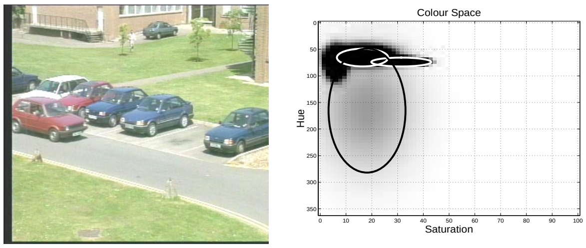
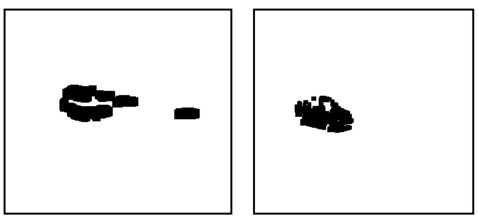


Figure 1: Car park image and colour distribution

grass, road, buildings and vehicles. We can use this colour model to segment the image into background and not-background using a low threshold. Morphological processing is then performed to remove noise and merge larger regions. Connected components are then found and some simple features (bounding box, centroid, area, eccentricity, etc.) are calculated. Unlike background subtraction, this technique is reasonably insensitive to modest camera movements (see section 4). However, its applicability is limited to situations where objects are distinctly coloured with respect to the background. This is often the case in the scenarios we are considering, such as road traffic surveillance, but is by no means the case in general. In the case shown in Figure 2 the segmentation obtained from the colour based technique is poorer than that obtained from background subtraction, since only those parts of the object that have a distinct colour are obtained. Furthermore, stationary vehicles which form part of the background are also detected, perhaps undesirably. However, these results are still adequate for event detection.



Colour based segmentation Motion based segmentation

Figure 2: Segmentation comparison

3.2 Tracking

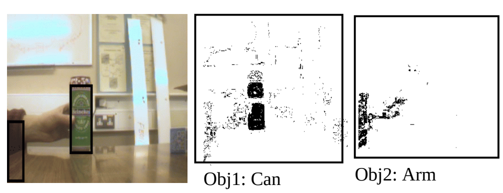
Given a number of regions obtained from the event detector described above, we extract those that are likely to be objects base on an area threshold. To track objects we adopt a similar algorithm to that presented in [1]. Given a region of interest (ROI), initially obtained from the event detector, at a time *t*, we estimate its position at time *t+1* by calculating the centroid and spatial extent of all those pixels within it that have non-zero probabilities of being non-background according to the GMM described above. The probability of a pixel (x,y) with colour *C* is given is by $P(C(x,y)) = 1 - P_{background}(C(x,y))$. The new centroid is found by

$$C_x = \frac{1}{N}\sum_y\sum_x xP(C(x,y)) \qquad C_y = \frac{1}{N}\sum\sum_x yP(C(x,y))$$

Where N is the sum of all probabilities (i.e. $N = \sum\sum P(C(x,y))$). Once the new centroid has been found, the ROI is moved so that its centre is at the new location. The new width and height of the ROI are found by computing the variances along the two image axes:

$$\sigma_x = \frac{1}{N}\sum_y\sum_x (C_x - x)^2 P(C(x,y)) \qquad \sigma_y = \frac{1}{N}\sum_y\sum_x (C_y - y)^2 P(C(x,y))$$

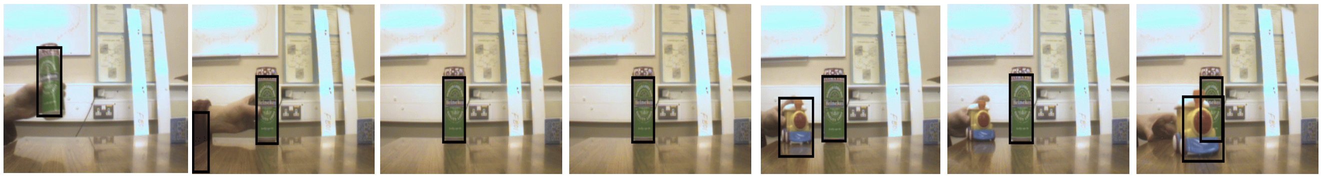
While a blob is being tracked using this algorithm, colours of non-background pixels are used to determine a colour histogram approximating the distribution of non-background colours within the object. Tracking an object using the non-background model continues until the colour histogram is thought to represent the object well enough that tracking should continue using the more specific description. The convergence and adequacy of the colour histogram is tested by means of the normalised cross-correlation between the recent observation and the histogram to date. If they are strongly correlated, the histogram is considered adequate and replaces the background model. Otherwise, the object continues to be tracked using the non-background model and samples are collected to refine the histogram. Whilst an object is being tracked some percentage of each observation is added into the histogram to allow for changes in the observed colour distribution. This helps to reduce the effects of changes in the appearance of objects due to rotation or changes in the lighting conditions. The fact that we know whether pixels belong to the background is exploited to avoid adding background pixels into the histogram. In frame 220 of Figure 4 there is considerable overlap between the two objects being tracked. In a simple "blob" based tracker we would expect these regions to merge and one or both objects to be lost. However, when each object is being tracked with its own colour histogram the other object is effectively invisible and does not interfere. This is illustrated in Figure 3 where the segmentations obtained from different colour models are compared.



Obj1: Can Obj2: Arm

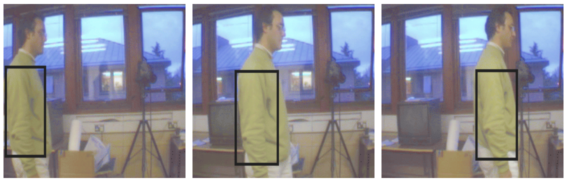
Figure 3: Segmentation under different colour models

Occlusion is also dealt with through the use of occlusion buffers. Each object has an occlusion buffer registered on the camera image. This is used to hold the last unoccluded



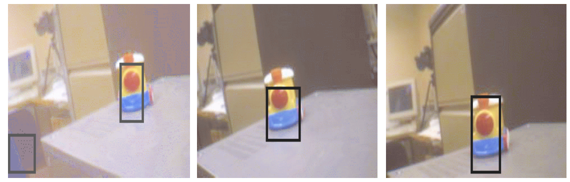
Frame 45 Frame 60 Frame 75 Frame 160 Frame 180 Frame 200 Frame 220

Figure 4: Colour Tracking of multiple objects with occlusion



Frame 70 Frame 80 Frame 90

Figure 5: Simple Object Tracking



Frame 30 Frame 60 Frame 90

Figure 6: Tracking with a moving camera

value observed at that pixel. When part of an object is occluded, the system reads from this buffer rather than from the camera image. The occlusion status of a point is determined by sorting objects according to their lowest point (in the image) and then making the valid assumption that objects on a ground plane with lower points are nearer to the camera than those with higher, and consequently occlude them if their ROIs overlap.

4. Results

In this section we report some typical results. Figure 4 shows seven frames from a sequence in which two objects are tracked against a reasonably cluttered background. The objects are correctly detected and tracked throughout the duration of the short sequence, despite the severe occlusion at frame 200. Figure 5 shows another sequence with a person walking against a reasonably cluttered background. Only the person's green clothing is tracked, since skin colour appears to be too close in terms of hue to some of the objects in the background (boxes, etc.). Again the object is correctly detected and tracked throughout the sequence. Figure 6 illustrates the case where the camera is moving and the object to be detected remains static. Note that this is a situation where background subtraction would fail as a means of locating objects. This problem occurs when detecting objects from a camera mounted on a moving vehicle. It can be seen from frame 30 that the colour based system fails to completely eliminate the background when camera movement is large. Parts of the background (in this case the side of the computer) that were not taken into account during the background model construction are detected and tracked. Tracking takes place at approx. 10 frames/sec (for two objects) on an SGI O2 and there remains considerable room for optimisation.

5. Conclusions and future work

In this paper we have described a system for tracking objects based on their colour. The main attractions of the approach are its potential speed and greater robustness to occlusion when compared to simple blob based trackers. The principal problem with the method is the necessity that objects are reasonably distinctly coloured with respect to their backgrounds. This prevents it from being useful in all circumstances. Future work focuses on improving the algorithm's performance through the use of additional image features (e.g. shape and texture).

6. References

[1] S. McKenna, S. Gong and Y. Raja. Face Recognition in Dynamic Scenes. In Proc. of British Machine Vision Conference, 1997.

[2] A. Baumberg and D. Hogg. Generating spatiotemporal models from training examples. In Proc. of British Machine Vision Conference, 1995.

[3] Sullivan, G. D. Visual interpretation of known objects in constrained scenes, Philosophical Transactions of the Royal Society, 1992.

[4] A. Dempster, N. Laird and D. Rubin. Maximum Likelihood from incomplete data via the em algorithm. Journal of the Royal Statistical Society, 39-B, 1977.

[5] D. Koller, J.Weber and J.Malik. Robust Multiple Car Tracking with Occlusion Reasoning. in Proc. Third European Conference on Computer Vision, 1994.