Using Colour Templates for Target Identification and Tracking

Simon Brock-Gunn*
Machine Vision Group,
Computer Science Department, City University, London
e-mail: simon@cs.city.ac.uk

Tim Ellis
Machine Vision Group,
Centre for Information Engineering, City University, London
e-mail: t.j.ellis@city.ac.uk

Abstract

A surveillance system is presented which uses colour cues to track people moving in sparse crowd scenes. The erratic motion of the targets, together with their changeable outline, means that they are conventionally difficult to model. However, by examining the colours present in an object in a given frame of a sequence, and looking for these in a later frame, identification and tracking are achieved. The colours are transformed into a template space in which it is easy to match objects to those held in a database, so that even when a person is occluded or disappears totally from view they may be re-located as soon as they can be clearly seen again. A hierarchical approach to template storage and searching reduces the effort required to search through the database and ensures that the system is efficient even with a database containing the templates of hundreds of people. Since work is carried out on a moving sequence, the problems of changing object shape and maintaining colour constancy are minimised by allowing for small changes in these parameters between frames and by continually updating each object's template.

1 Introduction

When examining the problem of following people in sparse crowd scenes, such as in a shopping centre or entering and leaving an aircraft, a number of possible solutions arise. At first, it might seem sensible to model the shape and motion of a person, in order to be able to distinguish them from the rest of the scene [4]. Unfortunately, this task is made difficult by the changing shape of a person, as the camera perceives it, when they move around or change direction and because their movements are influenced by many external and personal factors which are unavailable to a system. Even if we suppose that it is possible to track without modelling the shape [1], any system that follows the target from frame to frame will fail as soon as the targets become occluded when they pass in front of each other or behind stationary objects, such as an advertising

*Supported by a SERC award.

BMVC 1992 doi:10.5244/C.6.22

hoarding or the aircraft, if it is unable to relocate them when they re-appear. One further consideration which can be overriding, even in current systems, is the storage space and processing power required to maintain large numbers of object representations and to match them between frames in a sequence.

It is clear that if a more general approach is taken to analysing the problem, any practical implementation that arises will be more feasible, but the question remains as to which properties of the application should be examined.

Colour – in machine vision in general and object analysis in particular – is a property which would seem obvious, yet is apparently under-exploited since almost all current machine vision systems involve the use of only monochrome data. When even black and white images take up so much storage space and necessitate so much processing themselves, it cannot seem an attractive option to increase this by at least three-fold in order to utilise the extra information provided by colour data. However, with advances over recent years in computer technology, this is no longer such a concern, and now one of the more salient reasons may be the fact that colour is not an absolute property; rather it depends as much on the frequency content of the illumination as on the surface reflectance. The perceived colours of a target outdoors will change as a cloud passes in front of the sun, or as the target moves into the shadow of a building. The human visual system overcomes some of these problems by using a poorly-understood mechanism referred to as colour constancy [6], whereby the perception of constant colours is maintained under varying illumination levels. However, as yet no widely-accepted algorithms exist in machine vision which will reliably mimic colour constancy and, as a result, colour is seen as an unreliable property.

In this research we are analysing dynamic scenes and so we are able to minimise this problem by allowing the small changes in illumination, and thus the reflected colours, which will occur from frame to frame to fall within a small tolerance level and by continuously updating the object templates to take these changes into account. The way we use colour properties enables us to claim that our method is not sensitive to the small changes in object shape, viewing angle and scale which may also occur between frames.

The use of template definitions naturally avoids the need for large amounts of storage, since a template takes up a fraction of the amount of space that a colour representation of the object requires, and quick object matching within the database is achieved by the use of a hierarchical system of data representation. This ensures that time is not wasted in matching pairs of templates which clearly represent different objects (as will be the case on the majority of occasions) and instead such matches may be dismissed quickly.

Colour templates allow the representation of objects without the need for any type of model. This quality means they can be used in identifying objects of unknown shape and size from a list of possible choices [8], but the principle can be extended so that objects are "learned" by a system, based on the quality of match within frame sequences.

2 Target Identification and Tracking

The underlying technique of the system presented is to look at the objects seen in each frame of a sequence, and learn the combination of colours present

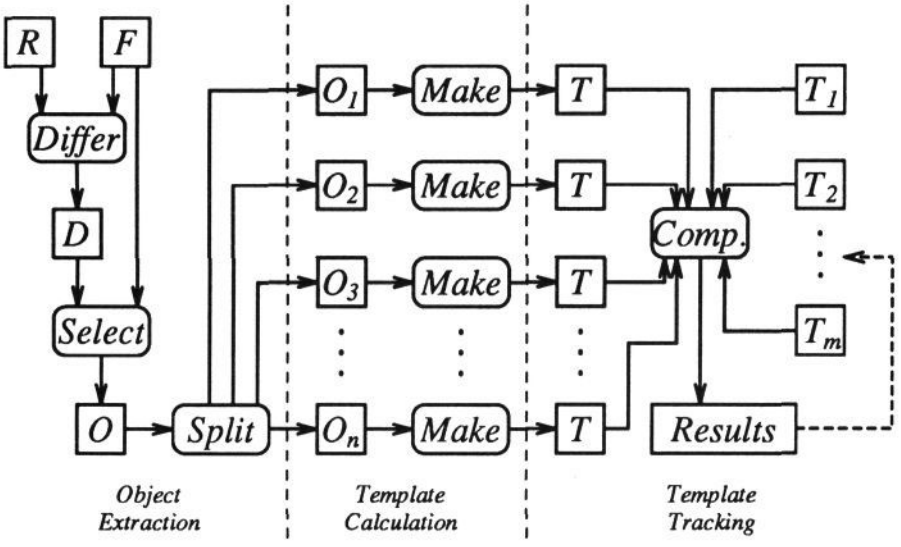

Figure 1: *The Processing Cycle, explained in sections 2.1 - 2.3*

in each object. This combination then becomes associated with the object and is remembered during the rest of the sequence, enabling identification and tracking of targets.

The processing cycle, show in Figure 1, is carried out on each frame and consists of three parts which are detailed below. Firstly, Object Extraction is carried out to locate any moving objects within the field of view and extract them from the image frame. This part of the process is not central to the ideas being presented here and, in our system, a simple and crude technique is used. When the objects have been found, the Template Calculation process examines the colours present within each of them and quantifies these colours by means of a template which can easily be compared to all those previously encountered in the Template Tracking process. The object will be located in a previous frame because of its similar colour "signature", providing a link between the object in two different frames and, therefore, determining its motion.

2.1 Object Extraction

The application uses a static camera, which allows us to assume that there is a one-to-one correspondence between movement in the scene and pixel changes in a frame sequence. This ensures that a simple operation is all that is required in order to extract the objects from a given frame image.

Before the processing begins, a colour reference image, R, is captured and stored. This image represents the background scene, which is composed only of static objects which will remain stationary during processing. Each time around the cycle, this reference image is compared to the incoming colour image frame, F. A resultant binary difference image, D, is then calculated by taking the difference between the intensity images of these two frames and

subjecting it to a threshold, t_d, to reduce noise errors:

$$D_{x,y} = \begin{cases} 1 & \text{if } F_{x,y} - R_{x,y} > t_d \\ 0 & \text{otherwise} \end{cases} .$$

$F_{x,y}$ and $R_{x,y}$ may be defined in a number of ways, the simplest being the sum of the red, green and blue point values at position x, y in each respective frame.

Then the object frame, O, is determined by taking an empty image and then copying in all the areas from the frame image where there is an entry in the difference image:

$$O_{x,y} = \begin{cases} F_{x,y} & \text{if } D_{x,y} = 1 \\ 0 & \text{otherwise} \end{cases} .$$

This object frame now contains all the objects which have moved between the acquisitions of R and F, presented on a blank background. These objects are then separated using a simple exploration algorithm which takes all the objects from O over a threshold size, t_o (again, to reduce the effects of noise errors), and relocates them in single object sub-images, $O_1, \ldots, O_n$.

2.2 Template Calculation

There are many different ways of choosing a template, T, such that it represents in some way the colours present in one of these objects, O. Given that the raw data of an object sub-image is generally composed of planes of the three primary colours, red, green and blue, the most obvious and straightforward to calculate is a three-dimensional colour frequency histogram, T_H, which may be created of suitable size $n \times n \times n$ (so n will be the number of bins on each axis), where each $T_{H_{r,g,b}}$ represents the amount of the object which has the colour (r, g, b).

However, we can employ a model that more closely resembles human physiological processes [5] by examining opponent colours. This still gives us three linearly-independent axes, but whereas (r, g, b) represent dimensions of black to red, green and blue respectively, the opponent colour axes of (rg, by, wb) represent dimensions of red to green, blue to yellow and white to black respectively. The transformation [2] from the input data is therefore

$$T_{O_{rg,by,wb}} = \left(r - g, \frac{2b - r - g}{2}, \frac{r + g + b}{3} \right) .$$

This type of template has been used to provide object detection cues [8] for the recognition of static objects, and Figure 2 shows how a template in the rg and by dimensions might look for a person wearing a red shirt and blue trousers. For this template, $n = 16$, so there are sixteen bins on each axis, and regions of intensity (represented by darker areas) are more prevalent in the red and blue parts of the template.

The use of these axes ensures that the template is invariant to rotation, translation and reflection of the object in the image plane since it is only the amount of each colour which is being measured, not its location.

Further analysis of the application shows that it is clear that the objects under study are subject neither to two-dimensional rotation in the image plane

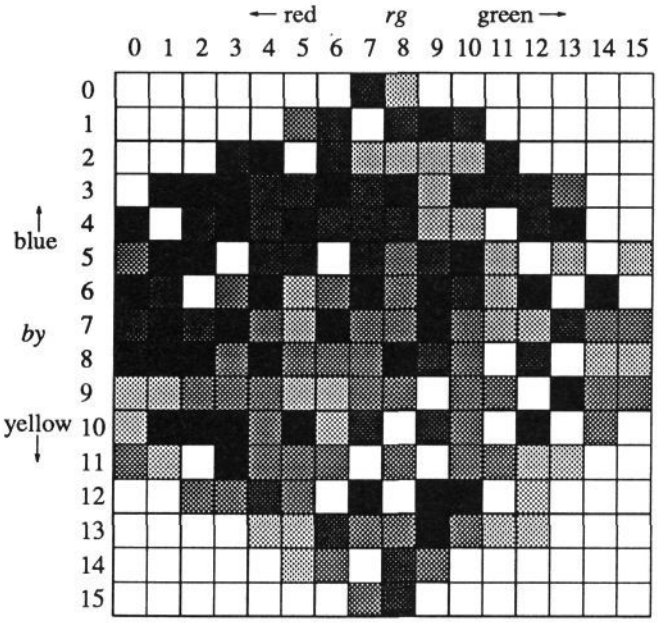

Figure 2: *rg and by plane of a sample template*

nor to reflection. This means that the discriminatory power of the colour may be made greater by weighting the information with respect to its position within the object. When we look at this in the context of our application, it can be seen that the person wearing a red shirt and blue trousers will cause there to be highlights in the appropriate areas of the template. However, someone else wearing a blue shirt and red trousers might be associated with a very similar template, yet it is clear to the observer that these are very different people. By making the spatial distribution of the colours an important factor, it is possible to ensure that such confusion does not arise, without compromising the generality of the system.

To this end, we add two further axes to the template which implement this spatial weighting: r and θ. These are analogous to the polar co-ordinates of a given point, o, in an object, O, such that, as shown in Figure 3, if c is the "centre of gravity" of the object then

$$\theta = \angle vco;$$
$$r = l/r,$$

where v is the point of intersection between the circle, centre c, radius R, which circumscribes the object and a vertical line through c, and l is the distance between o and c.

Finally, it can be recognised that wb is simply a measure of monochrome point intensity, a property which is more susceptible to shadows and changes in lighting than the others being measured. Furthermore, since there are already many methods which make use of monochrome techniques to track objects, and it is our aim to develop the use of a colour technique which may be comple-

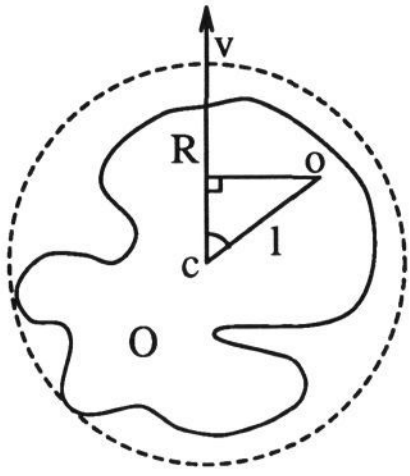

Figure 3: *calculation of the r and θ values*

mentary to these methods, this axis is discarded.

Thus we are left with a template of four dimensions: rg, by, r and θ.

2.3 Template Tracking

The template, T, is subject to a comparison technique together in turn with each of those already stored in the database, $T_1, \ldots, T_m$, where m is the number of templates stored. These T_i are the templates representing all the objects so far encountered. The match coefficient, v, is calculated as

$$v = \sqrt{\sum_{rg=0}^{n-1}\sum_{by=0}^{n-1}\sum_{r=0}^{R}\sum_{\theta=0}^{2\pi}\left(\frac{T_{rg,by,r,\theta}}{S_T} - \frac{Ti_{rg,by,r,\theta}}{S_{Ti}}\right)^2},$$

where

$$S_T = \sum_{rg=0}^{n-1}\sum_{by=0}^{n-1}\sum_{r=0}^{R}\sum_{\theta=0}^{2\pi} T_{rg,by,r,\theta},$$

such that $0 \leq v \leq 1$, where $v = 0$ indicates a perfect match (templates identical) and $v = 1$ indicates a perfect mismatch (templates completely differ). The scaling down by the sum of values in the template, which is the same as the number of image points in the corresponding object, ensures that the coefficient is invariant to scale.

This coefficient is then subject to previously determined minimum and maximum thresholds, t_{min} and t_{max} respectively:

- If $v < t_{min}$ then the two templates are similar enough to say that they represent the same object. In this case, T_i is updated by replacing it with the template definition of T and the object represented by these templates is said to have moved from the position associated with T_i to the position associated with T.

- If $v > t_{max}$ then the two templates are different enought to say that they represent different objects. If $v > t_{max}$ $\forall$ T_i then T is defined as representing an object not previously encountered. A new template, T_{m+1}, is created in the database, consisting of the template definition of T and the object is said to be at the position associated with T.

- If $t_{min} \leq v \leq t_{max}$ the templates are neither sufficiently similar nor sufficiently dissimilar to make a firm decision and the match is said to be inconclusive. This may occur if, for example, T represents a partially-occluded object or two objects occluding each other — situations in which we do not wish to make any judgements about the object associated with T. In this case, T is discarded and there is therefore no information with which to make any decision about the position of any object.

  The values of t_{min} and t_{max} are determined by experiment, and will be implementation-independent. Ideally, $t_{min} \rightarrow 0$ and $t_{max} \rightarrow 1$ and it is only the frame-to-frame change in parameters as well as the usual noise errors which mean that more realistic values need to be chosen. These values will depend on the distribution of unique colours across targets, so that if the objects are all similarly coloured t_{min} and t_{max} will need to be close together in value, whereas when the objects are all distinctly coloured they may be further apart.

3 A Hierarchical Approach

The use of a four-dimensional template means that matching can take a considerable time. A modestly-sized template with sixteen bins on each axis necessitates the comparison of over 65000 pairs of values. In a small application, with 30 different people in the database, millions of calculations would have to be performed in the Template Matching stage for each single frame. Even if each calculation took only one millisecond, Template Matching would take several seconds per frame — clearly unacceptable when the requirement is for a real-time system. The fact that so much time needs to be spent in these comparisons seems particularly absurd since the most cursory inspection by an observer reveals that almost all of the template pairs are clearly different.

The approach taken to alleviate this problem is to manufacture a "pyramid" of templates for each object, instead of just one [3]. This pyramid consists of progressively coarser-resolution templates so, for example, a sixteen-bin template would be accompanied by eight-bin, four-bin and two-bin templates. Although this provides a lot of useful new information, the overheads of this extra work are very low. If using four-dimensional, sixteen-bin templates, the highest-resolution template in a pyramid would hold 65536 values (16^4), whereas the next-highest would only hold 4096 (8^4), the next 256 (4^4) and finally the lowest-resolution which would contain just 16 (2^4) values — a combined total of less than ten percent extra. Furthermore, since each successively lower-resolution template is simply interpolated from the next higher-definition template by averaging the appropriate surrounding values, minimal further processing time is required.

In this hierarchical system, each comparison begins with the lowest definition resolution pair of templates. Of course, in almost all cases, the templates

	Single Template System		*Hierarchical System*	
Database size	*Values (millions)*	*Calculations (millions)*	*Values (millions)*	*Calculations (millions)*
30	2.0	2.0	2.1	0.1
100	6.6	6.6	7.0	0.1
1000	65.5	65.5	69.9	0.5

Table 1: *Storage requirements for a template database and processing requirements for comparing a new template with those currently in the database, exemplifying the advantage of a hierarchical system over a single-template system*

will actually be representing different objects, and in the majority of these cases, the difference between the templates is sufficient at this low level to determine that they are not representing the same object. Only if $v \leq T_{max}$, such that the objects are not definitely different, need the process continue with the next higher resolution pair, and so on until the highest resolution pair confirm that the templates are representing the same object (or otherwise). In practice, when trying to match an object's template with 30 others stored in a database, on about 25 occasions the lowest-resolution, two-bin template will provide sufficient information to dismiss the match, and only five will require examination of some of the higher-definition templates, with only one (the correct object match) necessitating a look at the highest-definition, sixteen-bin templates.

As Table 1 shows, the advantages of the pyramid system become abundantly clear as the size of the database increases.

4 Experimental Results

A simple sequence of frames was chosen in order to demonstrate the system in practice. This sequence depicted two people walking towards each other, passing (occluding) and walking away and was augmented with several still frames of other people, in order to provide a database of seven people.

A hierarchical arrangement of four dimensional-templates was used, with sixteen bins on each axis of the highest-resolution template. The system successfully tracked the two moving people and suspended tracking for the period of occlusion, as required. Within the hierarchical implementation, all of the matches into the database were determined to be false at the two-bin or four-bin template level (except, of course, the correct match). This was to be expected since the subjects chosen for the database were all clearly different to the observer, and the choice of more similar-looking subjects would undoubtedly require comparison of the higher-resolution templates.

Figure 4 shows six sample frames from the sequence. Alongside these is shown one of the targets which has been selected to be tracked. During the third frame, this target is lost, as it is occluded. However, the target is matched and found again for the fourth and subsequent frames.

Figure 4: *Six frames from the test sequence and a target which has been tracked*

5 Conclusion

Since the system proposed is very general in nature, it clearly has many potential applications in addition to our own. However, because of its particular advantages, there are three areas of tracking to which it is particularly suited:

- *Where the target objects are of similar shape and/or size.* There is no need to model any object-based parameters such as shape or size so, for example, it would be able to follow people in a street or cars on a road, whereas a system based on distinction by shape and size alone would fail.
- *Where the target objects have irregular motion.* Since there is no predictive element involved, targets may be tracked no matter how they move through the scene since at any given time they may be expected to appear in any part of the image or, indeed, not at all.
- *Where object occlusion occurs.* If an object disappears from view or is occluded for a while it will be re-located when it can be sufficiently seen again as long as the change in colour reflectance over the intervening period falls within the allowable small tolerance level, so there is no need to attempt to track it through every single frame.
- *Where a large database of objects is necessary.* Many database systems suffer because the time taken to search through the database for an object match is proportional to the number of objects held (or even proportional to some higher power of the number of objects held). By using a hierarchical storage and processing system, the time taken to search through the database increases only slowly as the database grows.

It is not suggested that the system presented here is capable of performing in all situations, rather that combination with existing techniques may be the best way of providing a solution to any given application problem.

6 Further Work

The Object Extraction process does not form an explicit part of this work and the simple method chosen has proved to be somewhat unreliable under certain circumstances where areas of the background sometimes show up as "moving" due to a change in lighting conditions between acquisition of the reference image, R, and the current image, F. These are currently removed by a filter based on allowable object sizes but this is not a satisfactory solution since, although generally successful when using the system indoors, the fluctuations sometime cause processing of outside scenes to break down. As one of the aims of this work is to reduce problems associated with colour constancy, a different method of generating R needs to be employed which will ensure the background is constantly updated [7, 1]. Furthermore, there are potential applications where the camera will not be stationary and so a more sophisticated method of Object Extraction must be used which will also take into account ego-motion.

The usefulness of this system depends on the ability to carry out processing in real time and, although the use of hierarchical storage and processing helps enormously in the Template Tracking stage (especially with a large database), the time involved in producing a template from an input frame is still of the order of several seconds (on a Sun SPARCstation). Although the ability to process at frame-rate is not a requirement for real-time processing, some rationalisation of program and storage structures must be undertaken to reduce the program cycle time towards the sub-one second scale. Specialist hardware would probably be required to achieve this but the repetitive nature of the tasks involved make this an ideal candidate for implementation in a parallel environment.

References

[1] A. T. Ali and E. L. Dagless. Computer vision for security surveillance and movement control. In *IEE Colloquium on Electronic Images and Image Processing in Security and Forensic Science*, pages 6/1–6/7. IEE, May 1990.

[2] Dana H. Ballard and Christopher M. Brown. *Computer Vision*. Prentice-Hall, Inc., Eaglewood Hills, NJ, 1982.

[3] Peter J. Burt. Smart sensing within a pyramid vision machine. In *Proceedings of the IEEE*, volume 76, pages 1006–1015. IEEE, August 1988.

[4] David Hogg. Model-based vision: a program to see a walking person. *Image and Vision Computing*, 1(1):5–20, February 1983.

[5] Leo M. Hurvich and Dorothea Jameson. An opponent-process theory of color vision. *Psychological Review*, 64(6):384–404, 1957.

[6] Peter Lennie and Michael D'Zmura. Mechanisms of color vision. *CRC Critical Reviews in Neurobiology*, 3:333–400, 1988.

[7] Simon W. Lu. A multiple target tracking system. In *Proceedings of the SPIE*, volume 1388, pages 299–305, 1991.

[8] Michael J. Swain. Color indexing. Ph.D. thesis, University of Rochester, New York, November 1990.