Real-Time Imaging **5**, 295–304 (1999)
Article No. rtim.1998.0174, available online at http://www.idealibrary.com on IDEAL®

Multi-Object Tracking in Video

This paper reports on tracking of multiple objects using color histogram backprojection and motion cues. Four tasks which facilitate this are discussed. The first is an adaptive color histogram backprojection (which builds upon the works of Swain and Ballard) and its application to tracking of multiple objects in video sequences. The second task is designing efficient fast blob detectors for selecting regions of interest in video sequences. The third is motion detection based on color histogram backprojection. Achieving these tasks led to multi-objects tracking. Various video sequences were used to demonstrate effective tracking of multiple objects. Notably, we created an interactive multiple objects tracker (CLICK-IT) which in its present form is set at three objects but can be extended easily. CLICK-IT (CSIRO Laboratory for Imaging by Content and Knowledge—Interactive Television) is a PC-based system which provides the user with an intelligent highlighter pen for sports action replay. It is intended as a truly interactive improvement on the drawing pad technology currently used for video annotation in sports broadcasting. The system uses computer vision techniques to focus attention and track particular objects (player(s), ball, horse(s), . . .) and semi-automatically annotate the dynamic scene. This paper describes the system including the user interface, the tracking technology based on color and motion information, and system performance evaluation in applications to surveillance-like sequences, running, rugby league football, basketball and soccer. Finally, video scene detection based on color histogram is discussed.

Johnson I Agbinya and David Rees

CSIRO Telecommunication and Industrial Physics, Image & Signal Processing Group, PO Box 76, Epping, NSW 2121, Australia
E-mail: {johnson.agbinya, david.rees}@tip.csiro.au

Introduction

To facilitate real-time object tracking in video, efficient image segmentation algorithms need to be used. This is especially important for software based tracking systems running on PCs. The complexity of the problem is increased if multiple moving objects need to be tracked, as is the case in many applications including surveillance, sports reporting, video annotation, supervisory management of shopping outlets, production lines and traffic control. Color-based algorithms have advantages for real-time software applications. For example, probabilistic color distribution models have been used for real-time face tracking in videotelephony [1] and visual control of computer games [2]; see [2] for a discussion of the relative merits of various tracking methods. Color histogram backprojection [3] is a low complexity, active vision algorithm for finding objects in complex scenes. Kahn *et al.* [4] combined color histogram backprojection with edge, motion and pixel disparity for real-time gesture recognition in the so-called Perseus system. Color histogram backprojection is a tracking method used for a mobile robot vision system known as Gargoyle [5].

In this paper we describe tools which enhance the suitability of color histogram backprojection for multiple object tracking. Instead of the fixed histogram

1077-2014/99/100295 + 10 $30.00/0

method used in [3] and [6] we introduce an adaptive algorithm, updating the histogram of the tracked object in order to account for changes in color, illumination, pose and scale. Tracked objects appear as "blobs" in the backprojection image. We have developed a fast blob detector using row and column intensity projections in the backprojection image. Empirically determined thresholds, blob area estimates and simple motion prediction are combined to resolve occlusion ambiguities, including "color occlusion" where objects have similar color. In their Kalman tracking system, Seo *et al.* [7] used color histogram backprojection for occlusion reasoning, but only to distinguish objects of different color.

One of the key problems in tracking is target acquisition. In automatic systems this is usually solved by some form of motion detection. Surveillance video cameras are often set up with specially defined regions, e.g. a doorway, areas containing valuable items such as a computer or exhibits in a museum, or items in a supermarket aisle. An alarm is triggered when motion is detected in one of these "hot spots". Such an alarm system could be used to initialise the tracker. However, we avoid these complications by initializing the tracker *interactively* using the mouse.

Using this interactive multi-object tracker we have built a system called CLICK-IT (CSIRO Laboratory for Imaging by Content and Knowledge — Interactive Television), which is a 32-bit object-oriented C++ software application on a Pentium PC running Windows 95 or NT. The motivation for CLICK-IT is as follows. Interactive television (IT) is an emerging technology which will dramatically change the way we access video information. An exciting prospect is immersive video with control at the receiver. Here a person watching, say, a sporting event or a theater production might choose a virtual "best seat in the house", perhaps even within the action itself, sharing the point of view of one of the players or actors. Examples of such sophisticated systems are already under development [8,9]. A simpler IT technology is action replay with control at the transmitter. ORAD's Digital Replay [10] occupies the high end of the market in this technology, but most sports broadcasters still annotate action replay segments with very simple tools, "interacting" with the video using a drawing pad to superimpose arrows and other scribbles highlighting aspects of the play.

CLICK-IT is designed to be a truly interactive improvement on this simple method of sports annotation. Our system operates like an intelligent highlighter pen, using computer vision techniques to focus attention on particular objects (player(s), ball) and semi-automatically annotate the dynamic scene. Papers relating to tracking given sports like American football [11], soccer [7, 12] and volleyball [13], require significant domain knowledge. By contrast, CLICK-IT is not domain specific and requires little image understanding. Although the emphasis here is on sports, one can imagine other applications where interactive tracking would be helpful. For example, security personnel might wish to highlight and track several hooligans in a crowd at a football match in real-time or in playback mode.

The rest of the paper is organised as follows. "Tracking Using Adaptive Color Histogram Backprojection" is a discussion of the various components of the adaptive multi-object tracking algorithm, and "CLICK-IT System Overview" is a description of CLICK-IT. "CLICK-IT System Performance" is a report on the performance of CLICK-IT applied to a surveillance-like video obtained in our laboratory, and to a variety of sports video clips of running, rugby league football, basketball and soccer. In "Conclusion" we outline work in progress on camera handshake protocols for an intelligent multi-camera tracking system using our adaptive multi-object tracking algorithm, and on scene detection using color histograms.

Tracking Using Adaptive Color Histogram Backprojection

Most of the algorithms available for segmenting objects in complex video scenes require significant processing and fail to solve the focus of attention problem in a scene where many objects are to be segmented and tracked at the same time. The focus of attention problem, that is, finding the region and object of interest in a complex scene was addressed in [3] and extended in [6] using color histogram backprojection algorithms. In both cases a fixed model of the object of interest was used to search for the object. In a video sequence, our task is to segment and track multiple objects while at the same time paying attention to variations such as color, illumination, pose and scale. We describe here an adaptive backprojection algorithm which accounts for these variations, and which also handles problems of occlusion, enabling unique identification of objects in complex video scenes. Figure 1 shows the system diagram for this algorithm. Details of each step are given in the rest of this section.

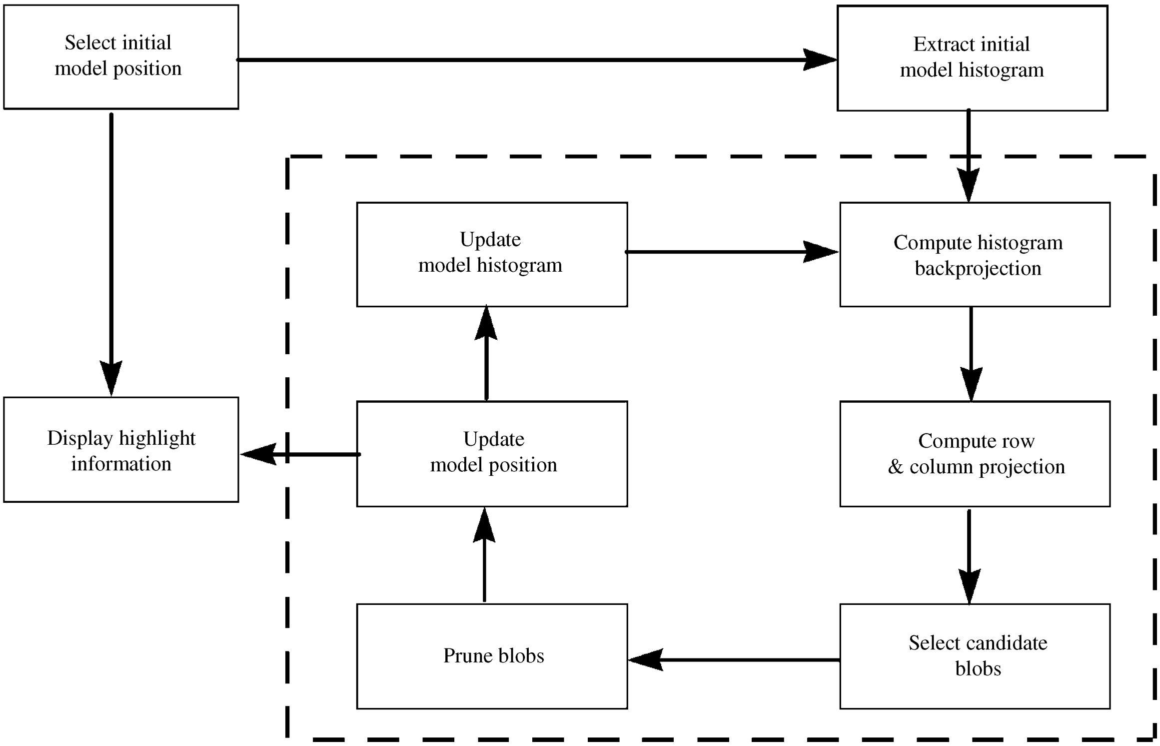

Figure 1. Tracking algorithm system diagram. The dashed box encloses the main logical loop after initialization in the first frame.

Initial training model position and histogram

The system is interactive so we avoid the usual difficulties of object detection and tracker initialization encountered in automatic systems. Off-line and real-time versions were implemented. In the off-line version, the video sequence is held in a buffer and to initialize the tracking, we use a mouse to define "rubber banded" rectangular bounding boxes around the objects to be tracked from the first frame. The dimensions of these regions are kept for future use. Each box defines an object "training model" which is characterized by its color histogram on a quantised color space; 24bit RGB sample (8 bits in each red, green, blue) images and 256 color histogram bins. Once tracking is initialized, further capturing of new training model histograms is done on-line by software based on position and motion cues. The real-time version supports a video camera and a frame grabber linked to the tracking software so that live video can be used for tracking. Here objects being tracked have to be cooperative, i.e. hold still, while the box contents are captured in the first frame.

Accurate segmentation is not required to track a person; typically the box surrounding the person's shirt, and perhaps head, is adequate. More care is required with small objects like a ball, where background pixels in the box could dominate.

Color histogram backprojection

Color histogram backprojection works as follows [3]. Let M be the histogram of the object of interest, or training "model", and I be the histogram of the image frame. Compute the ratio $min\ (M_i/I_i,\ 1)$ where i denotes the histogram bin. This is a confidence measure of how much the color i is characteristic of the model. Then a gray-scale backprojection image is constructed from the frame by replacing each pixel with color i by its scaled confidence measure, 255 $min\ (M_i/I_i,\ 1)$. In the backprojection image, the model appears as a bright "blob". Figure 2 shows an example where the object is the orange shirt worn by one of the authors (JIA) in a frame from an outdoor sequence.

When tracking several (N say) objects we compute the histogram of the frame I as usual, and the histogram of each model M_j for $j=1,\ldots N$. Then we construct N backprojection images, one for each model. Bearing in mind that the sizes of the models are much smaller than

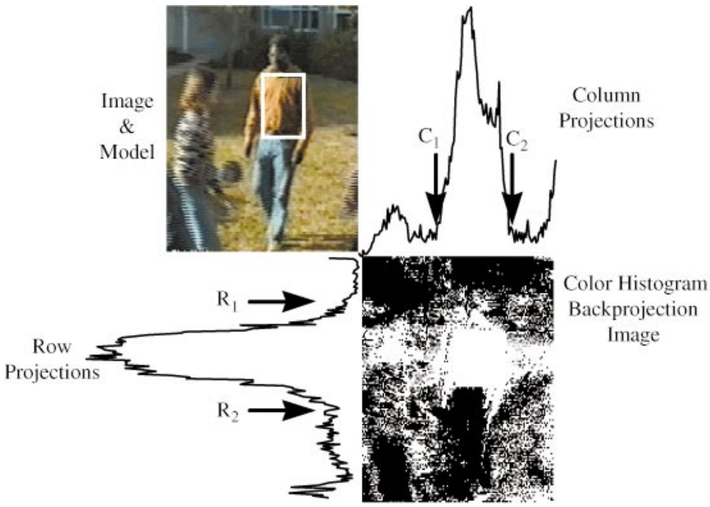

Figure 2. Example of color histogram backprojection and row and column projections. The model histogram is extracted from the region inside the white box.

the frame sizes, the additional overhead is inconsequential.

Row and column projections and blob selection

In [3] a convolution mask, typically circular, is used to estimate the position of the object on the backprojection image. We have developed more efficient algorithms [14, 15] based on row and column projections, as shown in Figure 2. In this example the location of the object of interest is unambiguous, and to a good approximation the coordinates of the centroid correspond to the peaks in these projections. In general however there may be other blobs due to similarly coloured objects; see Figure 3. Our problem then is to identify the correct blob automatically.

We experimented with several statistical methods for initial blob selection. These include the cumulative frequencies of row and column projections and their direct means. We opted for the following technique which is both simple and efficient. First we smooth the row and column projections with a five dimensional averaging filter. This filter eliminates any very narrow peaks in the column and row projections, but retains information from blobs that are significant, i.e. the blob of interest plus any others due to large objects (people) with similar colors.

The smoothed projections are then thresholded. In the current system we use the means of the row and column projections as thresholds; thus the thresholds are adaptive. The row and column projections of a tracked object are roughly Gaussian in shape (see Figure 2). We have tested ways to improve the choice of thresholds using estimates of the variances of these distributions, but have found that using the means as thresholds is adequate.

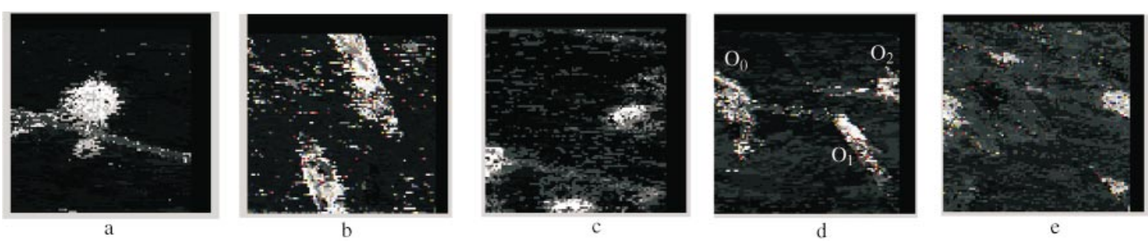

Figure 3. Object blobs resulting from color histogram backprojection for players wearing similar jerseys (a) a single object (an athlete), (b) is the case for two players, (c) and (d) for three players, and (e) four players. The difficulties associated with isolating each blob are apparent. In all cases, however, our blob detector locates the tracked object correctly. See discussion in "Multi-object tracking with a pre-defined model" regarding the blobs labeled in (d).

Candidate blobs are then identified as those regions with at least five connected points above the thresholds in both the smoothed row and column projections. An example of a candidate blob selected by this method is shown in Figure 2. It is defined by the rectangular region with corners (R_1, C_1) , (R_1, C_2), (R_2, C_1), and (R_2, C_2). Here, this method of segmentation produces a larger or smaller than expected rectangular blob (about 2 to 7 pixels in each dimension). This is not a problem as exact edge detection is not the goal.

A major advantage of this method is that the segmentation adapts to the new size of the tracked region, thereby accounting for changes in scale.

Blob pruning

At this point there may be one, several or no candidate blobs. In the last frame the tracked object may have left the field of view, or could be occluded by an object of a different color. Whenever object occlusion occurs, we repeat the algorithm using the previous model on the next frame, and re-use the model until the object is once again visible. This effectively implements a criterion to "wait for the object" to emerge from occlusion. It is efficient as long as occlusion does not persist. In the case of color occlusion, similarly colored objects will appear as candidate blobs, and sometimes these blobs will merge when one or more objects collide or shield each other. This occurs in contact sports such as rugby league where players wear the same colored jerseys.

Distance measure. Let d_i be the distance between the centroid of the ith candidate blob and the centroid of the blob in the previous backprojection, A_i its area, and A_p the blob area in the previous backprojection. We then list the blobs in order of increasing (combined) distance:

$$S_t = d_i|A_p - A_i| \tag{1}$$

In most cases the blob with the minimum combined distance corresponds to the tracked object, but to resolve any ambiguities we apply an extra criterion, similar to local block matching. For each candidate blob in the jth frame we check that the direction of motion that this blob has moved relative to the previous frame is consistent with motion prediction based on the velocities;

$$\begin{aligned} v_x &= xpos(j-1) - xpos(j-2) \\ v_y &= ypos(j-1) - ypos(j-2) \end{aligned} \tag{2}$$

estimated from the positions ($xpos(j-1)$, $ypos(j-1)$) and ($xpos(j-2)$, $ypos(j-2)$) of the centroids of the blobs in the previous two backprojections. There are of course special cases where this will not work, e.g. when two objects with the same size and same colors collide and recoil. In this case it is not possible to distinguish between these after collision.

This method of blob pruning has been successfully applied to a large number of video sequences. Examples of the complexity of the backprojection images it can handle are shown in Figure 3.

Updating model position and histogram

Once the blob corresponding to the tracked object is identified, we prune the interfering candidate blobs by placing a window around the selection, recomputing the row and column projections locally and deriving a new rectangular box which defines the updated model region. The centroid of the new rectangle is passed to the image display software to update the highlight information on screen. A new model histogram is extracted from the region in the image frame bounded by the new rectangle. Thus the algorithm adapts not only to the size of the object, as noted above, but also color, illumination and pose changes. The updated model position and histogram are used to find the object in the next frame.

Multi-object tracking with a pre-defined model

The process described above is repeated for all N backprojection images. It works for tracking objects with different, or similar colors. In the latter case one can take advantage of the color similarity to reduce the computational load, especially if one wishes to track all objects of similar color simultaneously. This would be of interest, e.g. in tracking all the players on the same team. Here we might have a pre-defined model histogram of the team jersey already stored, or have acquired a histogram from one of the players jerseys.

For the sake of illustration, we label the three blobs in Figure 3(d), as O_0, O_1, and O_2 respectively. To identify for example the object from blob O_0 properly, we use two descriptors, the object position in the previous frame (O_{00}) and the location of the blob in the present frame. This means, it has moved from O_{00} to O_0.

Therefore, O_1 and O_2 are interferences. Because of their positions, they affect the row and column means from the backprojection, leading to a detection for a false position for the target object. Observe however that the distance travelled by the target object between two frames, d_0 is much smaller than the distances to the interfering objects d_1 and d_2 respectively. The algorithm to correct for interfering objects uses the information from Eqns (1) and (2) to isolate the target.

CLICK-IT System Overview

This section describes CLICK-IT [16], a system which demonstrates the adaptive multi-object tracker technology. As indicated in Figure 4(a). CLICK-IT is a 32-bit object-oriented software application based on Visual C++ 4.0 on a Pentium PC running Windows 95 or NT. A tracking rate of 15 frames/s is achieved in 24bit RGB format using a 4Mbyte video card and a 200 MHz PC. The video archive on hard disk contains files that are either downloaded from the Internet or captured using a video camera or VCR, and a frame grabber. There is a switch to allow real-time operation. A planned system upgrade (indicated by the dashed lines) is an option to retrieve and display information from a player biography and game statistics archive during replay. A more ambitious development would be to allow content-based queries such as “Find and display similar play sequences” from previous games stored in a video database.

The control panel on the user interface shown in Figure 4(b) permits a number of operations on a video

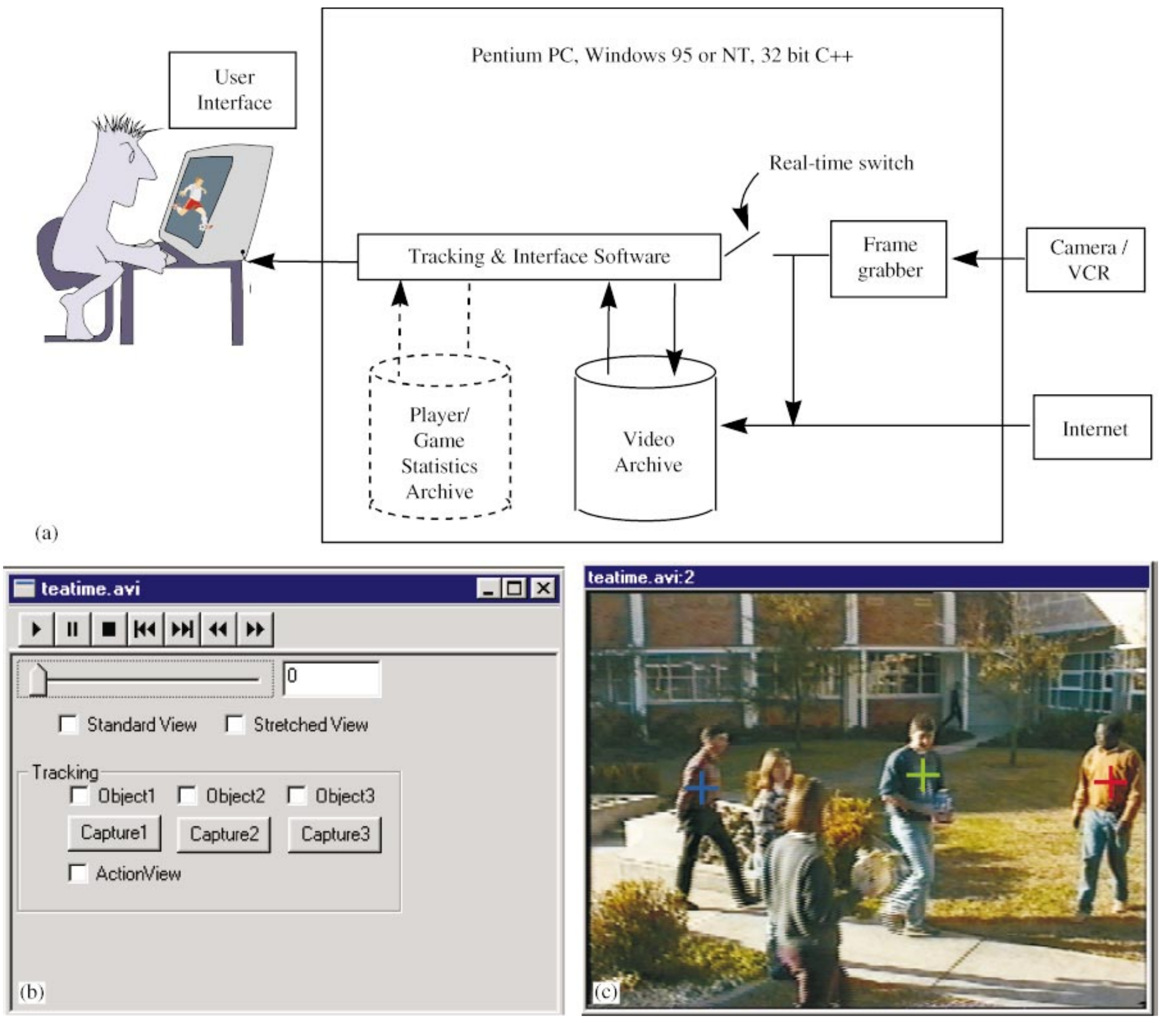

Figure 4. (a) CLICK-IT system diagram, (b) control panel, (c) typical standard view with three people highlighted by different colored crosses.

sequence loaded into memory. The buttons on the top of the panel are respectively start, pause, stop, fast rewind, fast forward, previous frame, next frame. The slider allows for quick browsing through a sequence. There are three viewing options: *standard*, for display of the original QCIF frame; *stretched*, for full screen display; and, in the case of single object tracking, *action* for QCIF-size display focusing on a region centred on the tracked object.

The control panel allows up to three objects to be selected for tracking, though more than three could easily be accommodated. Tracked objects are highlighted by different coloured crosses that automatically follow the objects throughout the action replay, as shown in Figure 4(c). Other highlighting tools can be added. For example we have implemented an object coloriser which automatically changes the color of the tracked object. Using this one could impart a "glow" to focus attention on a favorite player or a small object like a ball that is difficult to see, somewhat like an interactive software version of the FoxTrax ice hockey puck system [17]. One could also automatically annotate players with their names, or other information from the player/game archive.

CLICK-IT System Performance

We present below results of CLICK-IT's performance on surveillance-like video and to a variety of sports video clips. Signal trackers like coding algorithms need a measure of performance to enable one system to be compared with respect to another. No established standard is available for comparing the tracking performance of video object trackers. To establish performance, the choice of sequence used for tests should be wide enough to include many possible situations encountered in practice. This has been done in our tests. The sequences were selected in house, based on availability and on the measure of difficulty for tracking. The performance measure of tracking and coding systems is usually determined statistically. Therefore, to achieve acceptable measure, it is necessary to use a statistically significant amount of data. It is also necessary to track more than one object per sequence. We considered at least three objects per frame in the sequences as necessary. We limited the sequence sizes to between 20 and 100 frames. The selected sequences were ranked in terms of the degrees of tracking difficulty which include shielding, color occlusions, faded captures, low object resolution (where a few pixels are used to represent an object in a frame), blurred colors, color similarity of background with the object being tracked, mixed-shots and multi-object environments.

In any given frame a tracked object is deemed to be a "success" if the highlighting cross touches the object. The tracking success rate for a particular object in a sequence is given as a percentage of the total number of frames which contain the object. The statistics shown in Table 1 are averages over several replays of each sequence to allow for variations in the size and shape, and hence color histogram, of the initial model of the selected object. The letters "a", "b" and "c" in the table represent different objects in the same sequence.

Surveillance and sports applications

The surveillance-like video sequences were obtained in our laboratory: "brawl" involves three persons in a secluded environment and is akin to scenes in public disturbances, i.e., street fighting, while "teatime" represents a case where a suspect tries to mill with a crowd to avoid detection. A 100% success rate was achieved with the two sequences. Figure 4(c) is a typical frame from "teatime" showing three people being tracked. Figure 5 contains four frames from "teatime" illustrating the ability of the tracker to cope with occlusion.

The sports video clips, downloaded from the Internet or recorded from TV broadcasts are running (100 m, 400 m and marathon), rugby league football, basketball and soccer. The 400 m clip shows a prominent Australian

Table 1. Tracker performance statistics

Video clip	Object tracked	No. of frames	Success rate %
Brawl	a	80	100
	b	80	100
Teatime	a	60	100
	b	80	100
100 m	a	80	88
	b	80	100
	c	80	100
400 m	a	100	98
	b	28	96
	c	100	98
Marathon	a	36	100
	b	40	100
Rugby league	a	43	100
	b	25	88
Basketball	a	15	100
	b	80	88
Soccer	a	80	88

middle-distance runner, Cathy Freeman, winning the 1997 World Athletics Championships in Athens. As shown in Table 1, serious tracking failures of 12% occur only in four experiments. The rest show greater than 96% successes.

The two sequences "100 m" and "400 m" reflect sports application where more than color occlusion is important. In both cases object motion is rapid and background effects are significant. In the "100 m" case, the closeness of the runners and preponderance of dark skin tones from the runners make object separation difficult. In the "400 m" case, motion, object closeness, color occlusion and shielding are relaxed slightly but are still significant enough to make object separation difficult. In both cases however, tracking is fairly successful. Observe from the "marathon" sequence that tracking is relatively better than the previous two, leading to 100% success rate. Contact, object proximity, shielding and color occlusion are prominent in rugby. Motion is also of importance. Despite these, we were able to track multiple players with similar and different jerseys. To illustrate the possibility of tracking both basket and soccer balls, the last two sequences in the Table were used. Tracking of a ball to some extent answers the question of "how small may the object become before tracking fails?". The size of the balls used for tracking are less than 20×12 and 26×14 pixels respectively for the soccer and basketball sequences. These are the smallest objects we have used for testing so far. In both cases, ball tracking did not fail.

Figure 6 illustrates the problem of tracking failure due to incorrect initialization of the model histograms. Players 1, 2 and 3 are tracked in a sequence of frames from a rugby league football replay. The size of the players in the first frame (Figure 6(a)) makes it difficult to initialise tracking of player 3; the initial model histogram might include too many background pixels. Figure 6(b) shows an example where this has happened and the tracker has lost player 3. With careful initialization the tracker follows player 3 until he tackles player 1.

Conclusion

We have presented a set of algorithms based on color histogram backprojection which simplify the tracking of multiple moving objects within video sequences. The adaptive color histogram backprojection algorithm uses both adaptive object models and adaptive blob detector thresholds. In addition, object position is tracked to enhance the performance of the algorithm. To achieve this we derive motion cues from existing backprojection and thereby avoid complex computation of object velocities. An interactive tracking system CLICK-IT was implemented to demonstrate multi-object tracking for various surveillance-like and sports video sequences. Tests on these sequences so far indicate that the system is quite robust, on average a tracking success rate of greater than 93% was achieved on sequences of lengths about 10 seconds. Robustness might be improved using a combination of color histograms and Kalman filters, at the expense of increased computational complexity.

A possible development of the system would be to "piggyback" onto a camera system with pan-tilt-zoom functions controlled remotely by a joystick. The tracker could be linked by a feedback loop to help keep objects of interest centered and highlighted. An obvious next step would be automated camera control.

Work in progress

Intelligent cameras are being used more and more for manufacturing in bottling plants, surveillance, traffic monitoring, and high-speed production end inspections. The cameras have the capability to grab, pre-process and store video frames, pushing aside the need for integrated host PC, frame grabber and separate camera. Three models can be identified for camera networking. The cameras may be multiplexed in time, daisy-chained using the Universal Serial Bus or analog/digital video cameras with or without frame grabbers. In each case, routines to address the cameras and for collecting the video frames need to be written. The framework for networking cameras requires efficient handshake protocol. A camera handshake protocol is an algorithm which enables tracking of objects across video frames from several cameras multiplexed together or networked in such a manner that when the object being tracked starts to appear in the next camera, it is handed over so that tracking can continue. The objective is to first create a platter of concatenated images or frames obtained from several cameras and to track objects of interest in them. This problem is a direct derivative of tracking in a single camera. The key tasks inherent in implementing the system are to identify the correct object from camera to camera and to pass the subject from one camera to the other. We call these tasks "multi-camera handshake" protocol. Techniques for camera addressing and handshake between arrays of multiplexed cameras will be discussed in another communication.

Figure 5. Successful tracking of one of the author (JIA - indicated by arrow in (a)) in ‘‘teatime’’ sequence. (a) Before occlusion; (b) after occlusion by a group; (c) and (d) after two other occlusions by individual people.

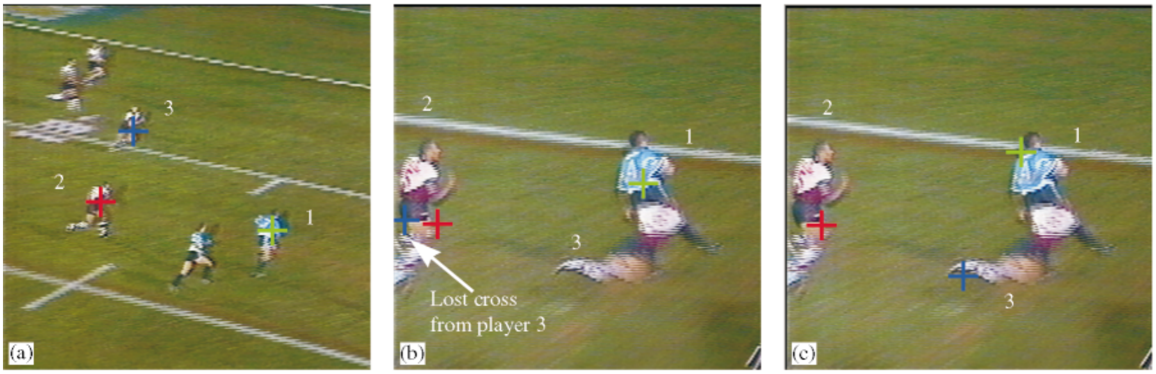

Figure 6. (a) First frame of rugby league replay with three players highlighted by crosses. Player 1 has the ball. (b) Last frame of replay. The tracker has lost player 3 due to inadequate initialisation. (c) Last frame with all three players tracked successfully.

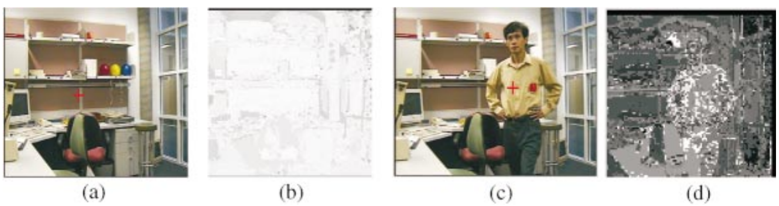

Figure 7. Scene change detection using color histogram.

Another related area of current work in our laboratory is scene change detection using color histograms. Scene change detection is an essential component in active news reporting, annotation and retrieval. Color histogram backprojection is essentially a scene change detector. To understand why this is so, we consider for example a scene of related video frames depicting a typical background. To detect scene change, we analyse the frames with color histogram back-projection and look for blob changes, in this case, significant patches of white on black. In Figure 7, the scene is a set of 50 frames of an unoccupied room enclosure. Matching the histogram of frame k+1 to that of frame k of this scene produces basically white blobs (Figure 7(b), i.e. the scene remains the same) only. When the scene changes, e.g. a person walks into the room, the correlation between the color histograms of frame k+1 and frame k produces white-on-black blobs showing where the change has occurred. By performing blob detection, we arrive at a decision of scene change. The means of rows or columns of Figure 7(b) are relatively equal, and if plotted on a Cartesian axes appear relatively ‘‘flat’’, i.e. no outstanding peaks are found. However, the means of the rows and columns of Figure 7(d) show distinct peaks when a person enters the room.

This algorithm is suitable for identifying sections of stored news video when the news reader moves to either an advert or inserted video commentary of a news item.

Acknowledgements

We thank Peter Machen at Channel 9 TV Sports, Sydney, Australia, for helpful discussions and for permission to use the rugby league football images. We also express our gratitudes to Nick Stone, Serath Seneviratne, Mark de Burgh and Fu Chen for coding various sections of CLICK-IT.

References

1. Hunke, M. & Waibel, A. (1994) Face locating and tracking for human computer interaction. *Proceedings of the 28th Asimolar Conference on Signals, Systems Computing*, pp. 1277–1281.
2. Bradski, G.R. (1998) Computer vision face tracking for use in a perceptual user interface. *Intel Technical Journal* http://developer.intel.com/technology/itj/q21998/articles/art 2a.htm.
3. Swain, M.J. & Ballard, D.H. (1991) Colour indexing. *International Journal of Computer Vision* **7**(1): 11–32.
4. Kahn, R.E., Swain, M.J., Prokopowicz, P.N. & Kirby, M.J. (1996) Real-time gesture recognition with the Perseus system. *University of Chicago Technical Report* TR-96-04
5. Flachsbart, J. (1998) *GARGOYLE:Active Vision Systems* http://www.cs.uchicago.edu/~josh/gargoyle/.
6. Ennesser, F. & Medioni, G. (1995) Finding Waldo, or focus of attention using local color information. *IEE Trans. PAMI*, **17**(8): 805–809.
7. Seo, Y., Choi, S., Kim, H. & Hong, K.-S. (1997) Where are the ball and players? Soccer game analysis with color-based tracking and image mosaic, *Proc. ICIAP'97*, http://cafe.postech.ac.kr/~dragon/soccer.html.
8. Katkere, A., Moezzi, S., Kuramura, D.Y., Kelly, P. & Jain, R. (1997) Towards video-based environments. *Multimedia Systems* **5**(2):69–85.
9. Moezzi, S. (Guest Ed.). (1997) Special issue on immersive telepresence. *IEEE Multimedia* **4**: Jan–Mar.
10. ORAD (1998) Digital Replay. http://www.orad.co.il/sport/index.htm.
11. Intille, S.S. & Bobick, A.F. (1994) Tracking using a local closed-world assumption: Tracking in the football domain. *MIT Media Lab*. **TR No. 296.**
12. Yow, D., Yeo, B.-L., Yeung, M. & Liu, B. (1995) Analysis and presentation of soccer highlight from digital video. *Proc. ACCV'95* pp. 499–503.
13. Chang, C.-W. & Lee, S.-Y. (1997) A video information system for sport motion analysis. *Journal of Visual Languages and Computing* **8**: 265–287.
14. Agbinya, J.I. (1997) Zone selector for backprojection images. *CSIRO Telecomm. & Ind. Phys. MIDAS Project Report* **RFN 465.**
15. Agbinya, J.I. & Stone, N. Blob detector for backprojected video sequences. *CSIRO Telecomm. & Ind. Phys. MIDAS Project Report* **RFN 473.**
16. Rees, D., Agbinya, J.I., Stone, N., Chen, F., Seneviratne, S., de Burgh, M., & Burch, A. (1998) CLICK-IT: Interactive television for sports action replay, *Proc. 14th International Conference On Pattern Recognition. ICPR'98*, August 17-20, Brisbane, Australia, pp. 1484–1487.
17. Cavallaro, R. (1997) "The Foxtrax hockey puck tracking system", *IEEE Computer Graphics Applications* **17**: 6–12.