Detection and Tracking in the IBM PeopleVision System*

J. Connell, A.W. Senior, A. Hampapur, Y.-L. Tian, L. Brown, S. Pankanti†
IBM T. J. Watson Research Center,
PO Box 704, Yorktown Heights, NY 10598

Abstract

The detection and tracking of people lie at the heart of many current and near-future applications of computer vision. In this paper we describe a background subtraction system designed to detect moving objects in a wide variety of conditions, and a second system to detect objects moving in front of moving backgrounds. Detected foreground regions are tracked with a tracking system which can initiate real-time alarms and generates a Smart Surveillance Index which can be searched to find interesting events in stored video.

1. Introduction

The PeopleVision project uses vision to track and understand human motion. The project spans several application from surveillance to human computer interaction. At the heart of these applications lie detection and tracking algorithms.

In the surveillance domain we apply them in the construction of a Smart Surveillance Index that describes the activity in a scene and can be used to derive real time alerts or to search for events in many hours of recorded video. The system is designed to automate much of the task of watching banks of video monitors, calling the attention of a human operator to "interesting" occurrences that occur rarely in streams of mesmerizing, uneventful video. In some scenarios detection of moving objects is sufficient for the raising of an alarm, but in busier areas where there is constant benign motion, tracking is required to follow the actions of each individual. These tracks can then be used to detect a greater variety of "interesting" behaviour.

Sections 2 and 3 describe object detection using background subtraction and salient motion detection. Section 4 describes how detected objects are tracked through the scene and through occlusions. Sections 5 and 6 describe how the tracking information is used to generate real time alarms and is stored in a Smart Surveillance Index which can be used for searching databases of surveillance video.

2. Background Subtraction

The background subtraction (BGS) system compares the current image with a stored reference background model as in previous BGS methods [4, 5]. Aside from the basic differencing operation, there are many other engineering enhancements as shown in Figure 1. These generally help the system maintain a usable background model and adapt to changing real-world conditions.

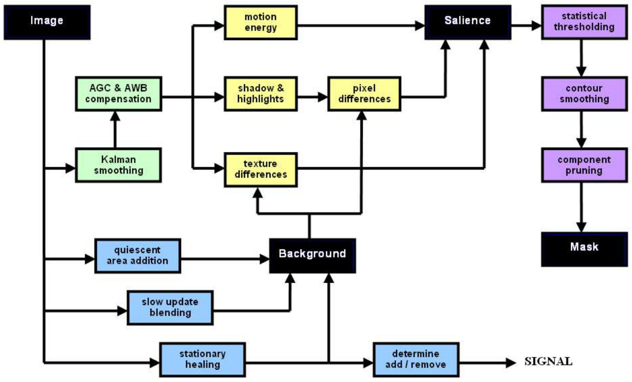

Figure 1: The background subtraction (BGS) system compares the current image with the background image to generate a salience map which is analyzed to create a foreground mask. The yellow boxes comprise the basic saliency computation engine. All the other boxes are various pre-processing, post-processing, and adaptation mechanisms added for robustness under real-world conditions. The BGS system also detects the simple addition and removal of objects.

The basic comparison functionality is comprised of three parts. The pre-processing stages (green boxes) attempt to compensate for camera and channel effects. A spatially-variant temporal smoothing is first applied to the incoming video stream, which improves color stability and reduces sparkle artifacts from compression. The system then estimates and corrects for AGC (automatic gain control) and AWB (auto white balance) shifts induced by the camera as the ambient lighting and scene composition changes. The core engine (yel-

[*] See http://www.research.ibm.com/peoplevision
[†] Contact aws @ us.ibm .com

low boxes) does the basic comparison and combines evidence from differences in color, texture, and motion weighted by overall channel noise estimates. The use of multiple modalities improves the detection of objects in cluttered environments and mitigates the sometimes harmful effects of over-aggressive shadow removal. Finally, the resulting saliency map is subjected to post-processing (purple boxes) to generate a cleaner foreground mask. The salience map is thresholded, smoothed using morphology-like operators, and then small holes and blobs are eliminated.

The remainder of the modules (blue boxes) are used to build and update the background model. A challenge for real-world systems is to acquire a background model even if there are moving objects in the scene. Sometimes it is not practical or feasible to quickly acquire an "empty" reference image. Our BGS system solves this by keeping track of where there has recently been motion or a detected foreground object. For regions which are sufficiently "quiescent", the corresponding portion of the input image is incrementally added to the background model until a complete reference image is obtained. Over the longer term, these same stable non-foreground regions control where it is acceptable to blend each new input image in with the background image. This mechanism allows the system to track slow overall changes, such as the sun passing behind a cloud in outdoor scenes.

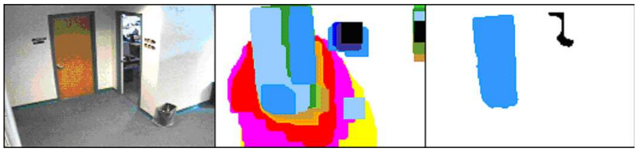

Figure 2: The BGS system can automatically "heal" regions, such as this recently closed door (left), by forcing them into the background. The system maintains a map of the number of frames since motion was observed (middle) and finds the minimum of this map for each current foreground component (right). When there has been no motion associated with an object for a specified amount of time, it is considered a candidate for healing.

A novel aspect of the BGS system is its ability to "heal" the background image by selectively adding uninteresting, stationary foreground objects to it. This process, shown in Figure 2, is advantageous because it makes detection of objects passing in front of such areas easier. A foreground object is deemed uninteresting if it has not moved, or has had no internal motion, for a specified amount of time. Even when people stop walking slight motions prevent them from being inadvertently pushed into the background model. The BGS system also gives the top level program the option of "vetoing" any of its proposed healing actions if, for instance, it wishes to continue tracking an object.

When healing, the BGS system can also discriminate between removed versus abandoned objects. It does this by analyzing the change in the amount of edge energy associated with the boundaries of the foreground region, as shown in Figure 3. Barring extremely cluttered environments, if there are significantly more edges then an object has been added. Conversely, less associated edge energy suggests that an object has been removed. If the edge measure is similar before and after, it typically means that there has been a state-change (e.g. a door closing). However, if the total edge energy is low in both cases (i.e. indistinct boundaries) the region is more likely to be something like a persistent soft shadow that the system did not fully compensate for. The category determined for each proposed region is important because it can serve as a filter to allow automatic healing of certain types. It also forms the basis for useful security alerts, such as abandoned object detection (Section 5).

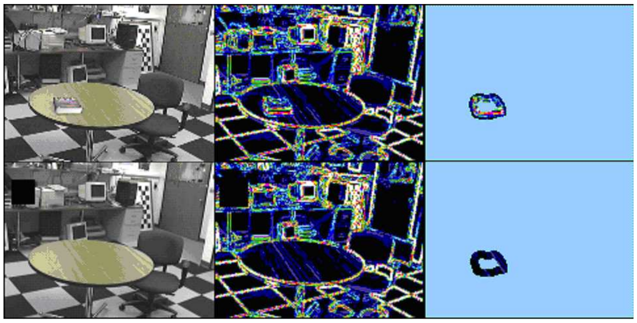

Figure 3: Healing category determination. The deposited book in the current frame (top left) is about to be "healed" and made part of the background model (lower left). The system computes the edge strengths for the current frame (top middle) and the original background (bottom middle) and compares the average edge energy near the boundary of the region in the current frame (top right) and background model (bottom right).

3. An Alternative Object Detection Approach

Background subtraction identifies regions that are different from some background model. In some scenes, however, there is constant motion or appearance change, which make it difficult to build a model for the appearance of a pixel. Water surfaces, wind-blown vegetation and video displays are all classified as foreground by conventional background subtraction algorithms, as are rapid scene-wide lighting changes, though previous authors have devised systems to cope with some of these problems [3, 5]. We are not interested in tracking these regions, but do wish to track objects that move in front of them. Consequently, we have developed a salient motion detection system that detects objects by a method complementary to that of background subtraction. Here we approach the problem from a motion filtering perspective. Consider Figure 4. The image on the left shows a scene where a person is walking in front of a bush which is waving in the wind. The next image shows the output of a traditional background subtraction algorithm, that detects the bush and the person as moving regions.

Salient motion detection uses optical flow to detect objects moving in front of a constantly changing background. First, a simple image difference detects regions where motion is occurring and within these optic flow is calculated. Optic flow vectors are chained together over a temporal window of n (typically 10–15) frames. Pixels for which the optic flow vectors are consistent in both x and y directions are labeled as foreground candidates. A chain of vectors is deemed consistent (in x or y) if the sign of that component is the same for 2/3 of the frames. The candidate regions from the motion filtering are subjected to a morphological region growing process to obtain the final detection mask, shown in the right of Figure 4. Background subtraction and salient motion detection are complementary approaches. Background subtraction is more suited for indoor environments where lighting is fairly stable and distracting motions are limited, whereas salient motion detection is well suited to detection in outdoor situations.

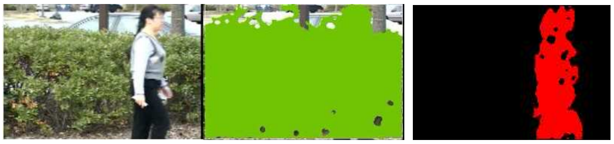

Figure 4: A person moving in front of wind-blown foliage. Centre: the result of a conventional background subtraction algorithm. Right: The object detected using salient motion.

4. Tracking

Tracking can be seen as a problem of assigning consistent identities to visible objects. Over time we obtain a number of observations of objects (detections by the background subtraction algorithm) but need to label these so that all observations of a given person are given the same label. When one object passes in front of another, partial or total occlusion takes place, with background subtraction detecting a single moving region. By occlusion handling, we hope to be able to segment this region, labelling each part appropriately, and correctly labelling the detected objects when they separate. In more complex scenes, occlusions between many objects must be dealt with.

When objects are widely separated as simple bounding box tracker is sufficient to associate a track identity with each foreground region. Bounding box tracking works by measuring the distance between each foreground region in the current frame and each object that was tracked in the previous frame, a match being declared if the object overlaps the region or lies very close.

If the foreground regions and tracks form a one-to-one mapping, then the tracking is complete and tracks are extended to include the regions in the new frame using this association. If a foreground region is not matched by any track, then a new track is created, and if a track matches no foreground region, it continues at a constant velocity, but is considered to have left the scene if it fails to match any region for a few frames.

Occasionally, a single track will be associated with two regions. For a few frames this is assumed to be a failure of background subtraction and both regions are associated with the track, but if there are consistently two or more foreground regions, then the track is split into two, to model such cases as when a group of people separate, a person leaves a vehicle, or an object is deposited by a person.

4.1. Appearance models

More complex interactions where more than one track is associated to one or more foreground regions are handled by a mechanism that uses an appearance model of each tracked object.

An appearance model consists of an image of the object — a two dimensional array of colour values with a mask indicating which pixels belong to the object. An appearance model is initialized by copying the foreground pixels of a new track. The appearance model can be correllated with detected foreground regions to track the motion of the centroid of an object being tracked by bounding box tracking. At each frame the appearance is updated by copying the current foreground pixels. During an occlusion, the foreground models of all the tracks in the occlusion are used to explain the pixels labelled as foreground by the background subtraction mechanism. We assume a depth

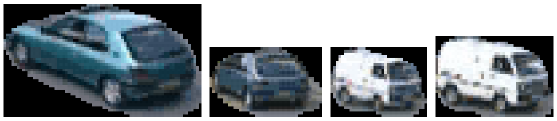

Figure 5: Appearance models from a PETS 2001 [2] video sequence, showing the appearance of model pixels, as one model recedes (left) and another approaches (right). Pixels not in the model appear black.

ordering among the tracks and try to fit the models front-to-back, building up evidence in an explanation map. The position of each object is predicted with a velocity motion model, then the front-most is localized through correlation. Pixels that fall within the foreground mask of the object are entered into the explanation map as potentially being explained by the track. Subsequent objects are correlated with only those pixels in the foreground region which have no explanation so far, and are entered into the explanation map in their turn.

The explanation map is now used to update the appearance models of objects associated with each of the existing tracks. The depth ordering is recalculated by examining those pixels where two objects overlap. Models which account for these disputed pixels better are considered to lie in front of models which match the colour of the foreground less well. The initial depth ordering at the start of an occlusion is considered to be arbitrary since such occlusions generally occlude only a small fraction of the objects. Each model is only updated in those pixels where the model was the front-most object. Regions of foreground pixels that are not explained by existing tracks are candidates for new tracks.

5. Real-time alarms

One of the principal aims of an automatic video surveillance system is to generate alarms as important events occur. This serves to focus the attention of human operators on interesting events, that might otherwise be missed due to the inevitable lack of attention after even a short period of watching surveillance video.

Many kinds of alarms can be generated using the information derived from the tracking system. Our system implements the following alarms, each of which can be limited to a particular area of the imaged scene, or be applied only to objects of a particular type. The object type (e.g. vehicle vs person) is determined by a classification module [1].

- Motion: A moving object in the region;
- Motion characteristic: Motion with particular speeds or directions;
- Abandoned object: A static object left by a person
- Removed object: A scene object that is moved
- Object count: Too many or too few objects of a particular type in the region

6. The Smart Surveillance Index

Information about tracks, including their position, size, appearance and type, is written into a Smart Surveillance Index. The index is either stored locally or, in a system with multiple cameras, each smart surveillance engine (of which several may be running on a single machine) transmits the tracking information across a network to a server which handles the storage and redistribution of the data. From the index, a very low bandwidth reconstruction of the video can be rendered, with track paths and labels superimposed. A summary bar indicates, for a selected time period, all occasions when moving objects were being tracked. Further search terms, such as motion with a particular direction or speed, or the class of an object, can be specified to filter the displayed activity. In this way sections of video where particular activity occurred are easy to identify and replay. More details of the Smart Surveillance Index and the distributed architecture of the system can be found in Hampapur et al. [1].

7. Conclusions

We have described a sophisticated background subtraction algorithm incorporating the following features: a camera model for pixel noise and AGC/AWB; colour, motion and texture difference detection; automatic background acquisition and updating. These features enable object detection in a wide variety of scenarios (indoor/outdoor); variable lighting and weather; objects of radically different scales. The model runs faster than 30fps on current hardware. In addition we have described a motion filtering approach to object detection that distinguishes salient motion from distracting motion.

On top of these detection algorithms we have written a tracking system that tracks moving objects through occlusions and generates real time alarms and a searchable video index. The tracking system has successfully run to gather continuous indeces for weeks of live video.

References

[1] A. Hampapur et al. Multi-scale tracking for smart video surveillance. Signal Processing, 2004. to appear.

[2] A. Senior et al. Appearance models for occlusion handling. In International Workshop on Performance Evaluation of Tracking and Surveillance, 2001.

[3] A. Monnet, A. Mittal, N. Paragios, and V. Ramesh. Background modeling and subtraction of dynamic scenes. In International Conference on Computer Vision, pages 1305–1312, 2003.

[4] C. Stauffer and W. Grimson. Adaptive background mixture models for real-time tracking. In Computer Vision and Pattern Recognition, pages 246–252, 1999.

[5] K. Toyama, J. Krumm, B. Brumitt, and B. Meyers. Wallflower: Principles and practice of background maintenance. In Proc. IEEE International Conference on Computer Vision, volume 1, pages 255–261, 1999.