AUTOMATED DETECTION OF LOCALISED VISUAL EVENTS OVER VARYING TEMPORAL SCALES

Jamie Sherrah
Department of Computer Science
Queen Mary, University of London, London E1 4NS, UK
jamie@dcs.qmw.ac.uk

Shaogang Gong
Department of Computer Science
Queen Mary, University of London, London E1 4NS, UK
sgg@dcs.qmw.ac.uk

Abstract

The characterisation and detection of events in visual scenes is difficult to define in a general framework. What constitutes an event is generally defined by contextual semantics. Therefore past approaches at event detection in visual surveillance have typically involved low-level segmentation and/or tracking, thus restricting their generality. In this work, a general semantics-free method is proposed for the extraction of visual events. The technique is based on the definition that an event is any sort of visual change. It is therefore necessary to determine visual changes that occur at different rates. Wavelet analysis is employed to detect different rates of pixel-wise change in the image, while a Gaussian mixture background model is used to determine absolute temporal change in pixel values. Clustering is then performed in a feature space for these pixel-wise change events. The result is a grouping of low-level events into high-level events. We show results on an artificial shopping scenario. Preliminary results are also presented on the extraction of high-level causal rules connecting events.

1. Introduction

In general terms, scene understanding for visual surveillance from a fixed camera is so ambiguous as to be unattainable with current vision technology. Identification of people, vehicles, objects, interactions between people, manipulation of objects and so on is made difficult primarily by poor problem definition, lack of training examples for statistical learning, and the general difficulty of automatically quantifying semantics, or meaning, in a situation. Furthermore, segmentation in the image is difficult for surveillance applications due to the arbitrary and relatively small size of objects in the image. Generally, significant high-level knowledge would be required for successful segmentation.

It therefore comes as no surprise that most previous approaches have been based on low-level segmentation or tracking of objects in the scene (Stauffer and Grimson, 2000; Morris and Hogg, 2000; McKenna et al., 2000; Haritaoglu et al., 2000). Immediately, the definition of a visual event is constrained to moving objects of a certain size or shape. For example, in many cases the definition of an event only concerns people in the scene (Morris and Hogg, 2000; McKenna et al., 2000). The tracked trajectories of the objects are then generally used in some high-level knowledge acquisition. Naturally some form of pixel grouping must be performed to obtain high-level knowledge from the scene, but our point is that it should be performed robustly in stages, rather than initially at a low level where insufficient information is available.

Of late, several attempts have circumvented the problem of segmentation and tracking by learning localised or pixel-wise variations (Stauffer and Grimson, 1999; Chomat et al., 2000; Ng

and Gong, 2000). The local models are then used to detect local visual events. Provided only detection is required, this approach overcomes the problems of segmentation and definition of the meaning of behaviour in the scene, which inevitably is problem-specific. However, pixel-wise modelling is computationally expensive, and if high-level knowledge is required then pixel grouping still must be performed.

Given that a pixel-based approach is preferable, a further issue is whether learning should be supervised or unsupervised. Supervised learning is generally less difficult since the user can supply examples of the background without any objects (*ie:* clean frames), and examples of typical behaviours. In this way, semantics are introduced. However there are two difficulties with supervised learning. First, training data or clean frames may be unavailable. Second, it may be infeasible to manually label sufficient training data. That labelling also involves a subjective element, especially when it comes to human behaviour. Due to these difficulties, and in pursuit of the ultimate point-and-shoot surveillance system, an unsupervised learning approach is preferable. This is true in particular for the application of behaviour or scene profiling, which means accumulating information about the scene, rather than imposing prior models.

An assumption that most approaches do not explicitly discuss concerns temporal scale. The spatial bounds of the image domain are pre-determined by the camera and grabbed image size, however time extends forever. Thus the question arises: over what temporal scale should behaviour or events be defined?

We propose a method for unsupervised scene profiling that can learn, classify and detect high-level visual events in an unsupervised manner. The algorithm does not require prior specification of scene semantics. Based on the most general definition of an "event" as visual change in the scene, the approach detects pixel-wise temporal change occurring at varying temporal scales. These low-level detections are then characterised by a set of local features, and unsupervised clustering is performed on the set of observed events. The result is a categorisation of events into classes that can be labelled by a human as having some high-level meaning. The classification enables events to be detected and classified on-line, information that can be used in high-level reasoning.

In section 2, different types of scene change are discussed and a definition of an event is offered. Section 3 describes the wavelet histogram method for detecting change at various temporal scales. A method is described for detecting, classifying and interpreting events in section 4. The experimental results on a test sequence are given in section 5, and the conclusion is found in section 6.

2. Defining an "Event"

The problem at hand is *behaviour* or *scene profiling*, defined here as follows:

> *Given a fixed camera and long-term video acquisition, learn to detect abnormal behaviour or events in the scene.*

Before moving on, let us expand on this definition. Long-term exposure may constitute years, captured at frame rate. Therefore the system may not wantonly accumulate data or models. Iterative optimisation algorithms should be avoided for timeliness constraints. The system must learn to detect. The traditional method is to learn models of normal behaviour and events, then test observations for deviations from these models. We assume the definition of "normal" to be "regularly occurring". Therefore the frequency of occurrence of an event may be low so long as it is regular.

People generally have a conceptual understanding of the term "visual event" as relating to some kind of visual change. Indeed, many past surveillance research approaches are based on motion or change detection (Stauffer and Grimson, 2000; Morris and Hogg, 2000; McKenna et al., 2000; Haritaoglu et al., 2000). However, these approaches immediately impose some semantics, either explicitly or implicitly, by grouping pixels into high-level events. When performed at a low level, the grouping implicitly requires definition of information such as spatial scale. More importantly, the change detected by these approaches is generally defined as the absolute difference from a reference frame. However, many dynamic scenes contain a rich variety of change occurring at

different rates. In (Toyama et al., 1999), high-level visual change is classified into ten categories. Consider the list of different events and the manner of visual change they cause, shown in Table 1. It is clear that if the definition of visual change were restricted to absolute difference, then many events would become indistinguishable or undetectable. For example, a person passing by and a person stopping for a few minutes would not be differentiated at the pixel level by the dynamic background modelling technique of (Stauffer and Grimson, 2000). It should be noted that the events from the table can all occur at different times in the same region of the image. For example, long-term lighting changes due to the weather are temporally superimposed with short- and medium-term changes due to passers by at the same pixel.

Event or Behaviour	**Pixel Change**
person walks past	short-term perturbation with frequency corresponding to speed of moving edges and texture
sun goes down	long-term smooth change
bag dropped	instantaneous step change
item removed from background	instantaneous step change
light switched on	instantaneous step change everywhere
television switched on	instantaneous step change then arbitrary change thereafter
person stops at vending machine	medium-term perturbation due to moving edges and texture
trees moving in wind	continual change of certain frequency

Table 1. Examples of some common scene events and a summary of their resultant pixel change.

In order to encompass the broad range of visual events experienced in everyday scenes, we define an event as *any kind of significant visual pixel-wise change.* Although this definition allows an automated system to detect all manner of events, it presents the challenge of finding a single technique that can cope with these different types of change. The work of (Ng and Gong, 2000) overcomes the restriction of absolute change models by augmenting the dynamic background model with localised temporal models of pixel change. At each pixel, change models are matched to novel observations over time to determine whether the observed change is normal or abnormal. However, the method requires supervised learning and is computationally expensive, and works at a fixed temporal scale.

We propose a temporal multi-resolution approach to change detection. Although pixel grouping is inevitably required for high-level definition of events, in the first instance there is insufficient information to perform robust segmentation. Therefore each pixel is analysed in isolation over time to distinguish change occurring at different rates. Local features can then be attached to occurrences of significant change, and unsupervised learning can determine classes of event in the scene.

3. Temporal Multi-resolution Analysis of Visual Change

General low-level event detection requires a method to detect change occurring at different temporal scales. The method must represent different super-imposed frequencies localised in time. It must also be computationally efficient enough so be applied to the huge space-time volume of image data. Clearly a time-frequency analysis algorithm is required. The Fourier transform would be inappropriate since it gives no localisation in time. The Short-Term Fourier Transform (STFT) would be more appropriate, but it is highly redundant which would contravene the requirement for computational efficiency. The STFT also requires the selection of a time window, again restricting temporal scale. Wavelets were designed to overcome these problems (Akansu and Haddad, 1992; Wickerhauser, 1994), and are used in our approach. Wavelets are essentially time-local band-pass filters that yield a multi-resolution time-frequency representation of a signal. The lowest band is a low-pass filter which provides the absolute pixel values over some time frame. The wavelet representation can then be used to form models of pixel change.

The approach taken is to form a Discrete Wavelet Transform (DWT) at each pixel over time. Consider the intensity values of a single pixel over time. The input time series $x(t)$ can be decomposed into a set of basis functions called *wavelets* (**?**):

$$x(t) = \int\int \gamma(s,\tau)\psi_{s,\tau}(t)d\tau ds \tag{1}$$

The basis functions $\psi_{s,\tau}(t)$ are scaled and translated versions of the *mother wavelet* $\psi(t)$:

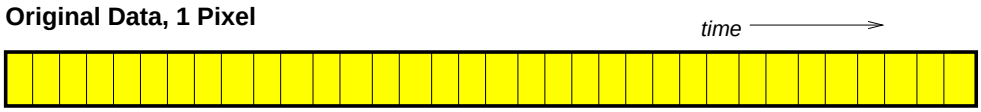
$$\psi_{s,\tau}(t) = \frac{1}{\sqrt{s}}\psi\left(\frac{t-\tau}{s}\right) \tag{2}$$

Note that the scale s is inversely proportional to frequency, so that small scale means high frequency. $\gamma(s,\tau)$ is the projection of $x(t)$ onto $\psi_{s,\tau}(t)$:

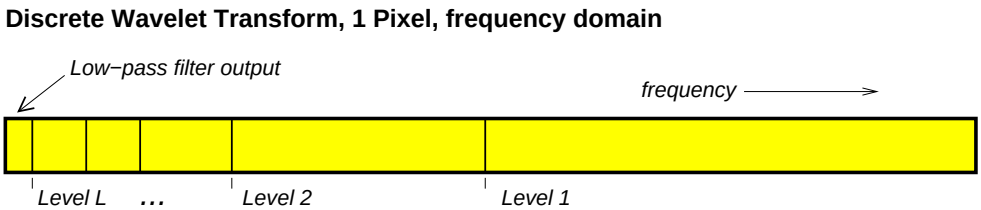
$$\gamma(s,\tau) = \int x(t)\psi^*_{s,\tau}(t)dt \tag{3}$$

The wavelets themselves are localised in time and frequency. The integral of equation 1 is infinite, so in practical applications s and τ are sampled on a dyadic grid, resulting in discrete wavelets. If $x(t)$ itself is discrete, then the discrete wavelet transform is obtained. As the scale increases, the wavelet projections $\gamma(s,\tau)$ have decreasing spatial resolution and highlight features of decreasing frequency. The result is a set of band-bass filters with logarithmic frequency coverage. Since a finite number of scales $s = [1,\ldots,L]$ must be used and the frequency coverage is iteratively halved with scale increase, a low-pass filter called a *scaling function* is used to cover the remaining low-frequency region.

To further illustrate, the input and output of a DWT is schematically shown in Figure 1. The input sequence $x(t)$ is shown in Figure 1(a). The output $y(t)$, shown in Figure 1(b), contains the $\gamma(s,\tau)$ packed into the same array, with small scales (levels) having a higher sampling density, a consequence of the fixed time-frequency bandwidth product for filters imposed by the Heisenberg uncertainty principle. As a result, the output requirement of a DWT is the same as the input (*ie:* T samples go in, T samples come out), and the computation is $\mathcal{O}(T)$ (Wickerhauser, 1994). The DWT is satisfactory for our large volumes of data, because of the linear computation and storage requirements.



(a) Input Data



(b) Frequency sub-division performed by DWT

Figure 1. Input/Output relationship of the Discrete Wavelet Transform.

After the DWT is performed, the resulting frequency information can be stored in a time-frequency histogram to characterise events occurring at different instants. A schematic example of such a histogram is shown in Figure 2. For an image containing N pixels, the storage requirement of the histogram is $N.T.L$, which quickly becomes unmanageable. A more feasible approach is to accumulate frequency information at each pixel over time, which requires bins for each pixel and

scale level only. This information can subsequently be used to determine which frequencies of change occur over time in different regions of the image. Note that the DWT method described here is a block processing method. For example, if $T = 1$ hour, then 1 hour's worth of video data would have to accumulate before the DWT can be performed. For an on-line system, a rolling DWT would have to be used, which can compute the DWT on-line as image frames arrive.

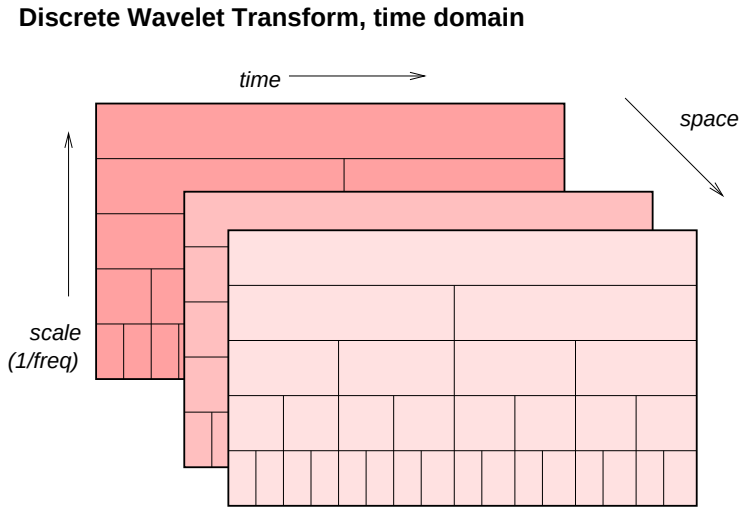


Figure 2. Multi-resolution scene profiling histogram.

4. Extraction of Events from Visual Data

We are ultimately interested in events that can be assigned high-level descriptions, such as the arrival of an automobile or a change in the weather. These events will naturally consist of many pixel-wise change events. For an unsupervised scene profiling algorithm, the distribution of events must be learned over time in some feature space. Here we describe a method for learning classes of events in the scene. First, events are detected locally in space and time. These local events are accumulated over time in an unsupervised learning algorithm, resulting in a set of event classes. The class information is subsequently used to detect and label local events, and for establishing high-level assocations between events.

4.1. Local Event Characterisation

Two methods are described for detecting pixel-wise change events, and representing localised events in a feature space. First a Gaussian mixture background model is used to detect absolute pixel change, and the pixels are clustered spatially to form local events. Second the DWT is used to detect differential pixel change at different rates, and local events remain at the pixel level. These two methods are compared in section 5.

The Gaussian mixture background model of (Ng and Gong, 2000) is used to detect pixel events as absolute colour change in RGB space that does not fit the model. Pixel events are then spatially grouped to form higher-level event boxes. We used a connected components algorithm for spatial grouping. Those foreground box events that remain in predominantly the same position for a non-trivial period of time are subsequently defined as local events. These events v_i are then characterised by the feature vector:

$$v_i = \{t, x, y, d, w, h\} \tag{4}$$

where t is the start time of v_i, (x, y) is the central position of the event box in the image, d is the duration, and (w, h) are the average event box dimensions.

To detect pixel-wise change events at different time scales, the Discrete Wavelet Transform is used in a two-pass fashion. First a wavelet histogram is formed, then on the second pass the

actual events are extracted. The purpose of the histogram is to identify regions of the image where change occurs regularly so that spurious detections can be avoided. We use $L = 7$ levels, which means the DWT is performed in blocks of $2^7 = 128$ time samples (frames). Frames are accumulated until 256 frames have been collected, then the DWT is performed over this time for each pixel. A histogram of spectral power is accumulated for each scale or level as follows. For each scale, the output array is visited and those absolute values that are over a noise threshold contribute to the histogram at that level. The final histogram is thresholded on a per-level basis to remove noise. This step is currently manual, but will be automated in the future. The 18-sample Daubechies wavelet was used (Wickerhauser, 1994). The low-pass result of the DWT is discarded; in the future it will be replaced with the Gaussian mixture background model.

The second-pass of the algorithm involves going over the sequence again and computing the DWT. For each pixel, level and time frame, form an event if the DWT output value is greater than the threshold. Only pixels with histogram values above threshold are included. These pixel-wise detections are our local events, and are characterised by:

$$v_i = \{t, s, x, y, h, g\} \tag{5}$$

where s is the temporal scale, h is the DWT output value for that scale, and g is the grey level at the pixel (x, y).

4.2. Grouping to Form High-Level Events

Given the set $v_1, \ldots, v_N$ of local events, the final step in training is to perform clustering in feature space to determine the distinct classes of event in the scene. We use k-means clustering with k a manually-determined parameter. The EM algorithm would also be a candidate for learning the clusters. Given that we now have K classes of events, the v_i can be classified on-line.

4.3. Correlations Between High-Level Events

Given that classes of events have been identified, the ultimate aim for artificial intelligence is to determine high-level information about the events and the objects that cause them. An example is establishing causal connections between events in space and time. Here we present a naive attempt at establishing causal rules connecting classes of events.

Suppose we are at the beginning of an event v_a of class A. We want to know if v_a causes events of class B. What we are essentially saying is that v_b (nearly) always occurs in the not-too-distant future of v_a, but not necessarily vice versa. For example, if a shopper takes an item from the shelf, he will pay within the next few minutes, but afterwards the next shelf item may not be taken for another half an hour if it is not a busy day. Using a one-sided Gaussian weighting function centred on the starting time of event v_a, the Gaussian-weighted time difference between the beginnings of events v_a and v_b can be determined. The purpose of the weighting function is to give higher weight to events that are temporally proximate. The weighted differences are accumulated in a covariance matrix for all combinations of events:

$$C(A, B) = \sum_{i:v_i \in A} \sum_{j:v_j \in B, t(v_j) > t(v_i)} G(t(v_j); t(v_i), \sigma) \tag{6}$$

where $t(v_i)$ is the time at which event v_i occurred. The element $C(A, B)$ is the accumulated weighting of events of class B following events of class A. The matrix is upper-triangluar because a one-sided Gaussian was used so that only connections forward in time are sought. By thresholding these correlations, rules can be established such as "if A happens then expect B" to happen. These rules are then triggered by detection of the causing event (A), and in response an alarm shows the expectation of the occurrence of the caused event (B). This is a very simple algorithm, and there is much scope for exciting research to further develop these ideas.

5. Experimental Results

In this section, the results of event detection are presented using the two approaches. First, the Gaussian background model is used to detect local events. Second, the wavelet transform is

used to detect pixel-wise events. After the listing of low-level local events, the same clustering method is applied. In the following sub-sections, we first present the experimental data, then describe the two low-level event extraction methods. Finally the detection of high-level events in the video sequence is shown, and the extraction of high-level rules linking events is described.

5.1. Shop Data

We have collected a 20 minute test sequence called the *shop* sequence, involving an artificial shopping scenario. The scene is shown in Figure 3. A shop keeper sits behind a desk on the right side of the view. An assortment of drink cans is laid out on a display table. Customers enter from the left and either browse the wares without purchasing, or take a can and pay for it. Abnormal behaviour would be to take a can and leave without paying. We will show how the system can (a) detect the events of browsing, taking a can, and paying, and (b) learn the temporal association between these events. The data were sampled at about 8 frames per second.

Figure 3. The shop scenario.

5.2. Detection of Absolute Change

Absolute change event boxes were extracted from the sequence, an example is shown in Figure 4. Using this method, $N = 213$ local events were extracted from the *shop* sequence.

These local events were then clustered using k-means with k=4. The absolute change local events v_i and their respective classes are shown in Figure 5. Although clustering is performed in a high-dimensional space, only the x-y co-ordinates of the events are shown here. The four different colours show the four clusters found. These have been manually labelled as being caused by the shop-keeper, browsing, paying and cans taken. Note that the results are quite noisy due to mis-detections and poor segmentation. In particular, the payment events are on the ground, and segregated from the change caused by the upper-body of the customer. This is due to errors made by the spatial clustering algorithm when forming the local events.

Figure 6 shows some examples of video footage annotated with events v_i that were detected on-line using the absolute change local event method.

5.3. Detection of Multi-Scale Temporal Change

In contrast, the DWT was then used to extract events from the same sequence. Since the frame rate is about 8 fps, the longest event we can characterise with 7 levels is 8-16 seconds in duration. An example of the wavelet histogram is shown in Figure 7. At each scale level and pixel, the thresholded histogram value is true (white) if significant change occured there. It can be seen that different occurrences are highlighted at different scales with continuous variation over scales. For example, paying the shopkeeper is quite prominent at level ??. The histogram was then used to extract pixel-wise events at different levels from the *shop* scenario, the resulting event list contained $N = 250,297$ events.

(a) Original Scene (b) Pixel Change Events (c) Grouped Pixel Events

Figure 4. Example of local event characterisation using Gaussian mixture background model.

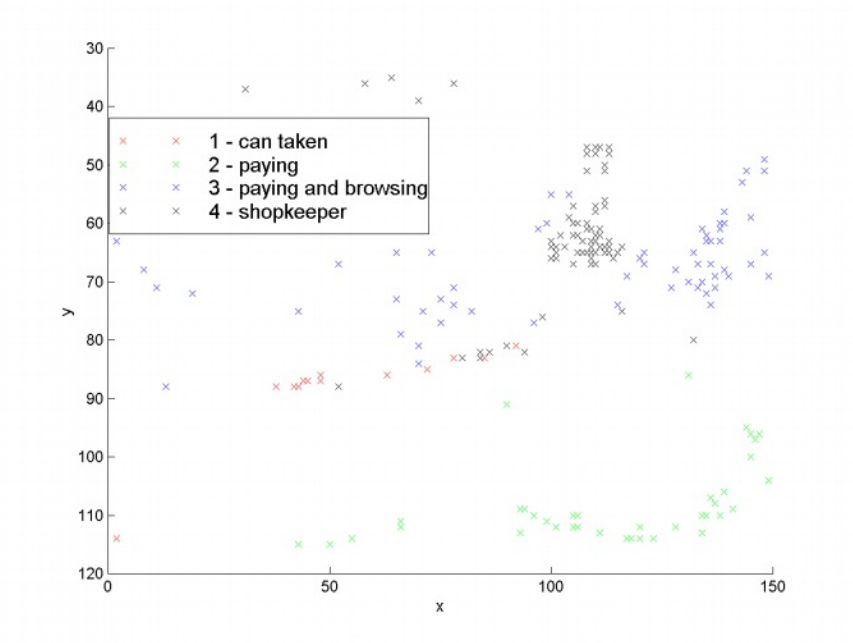


Figure 5. Clustered events using the Gaussian mixture background model to detect absolute change local events. Only the x-y positions of the events are shown.

Figure 6. Example event detections. Blue box indicates browsing. Red box indicates can taken. White box indicates shop-keeper moving. Green box indicates paying.

After clustering the multi-scale temporal change local events, the v_i and respective classes are shown in Figure 8. Again the 4 different classes, though found using unsupervised learning, have been manually labelled, this time as shop-keeper, paying, browsing-left and browsing-right. In

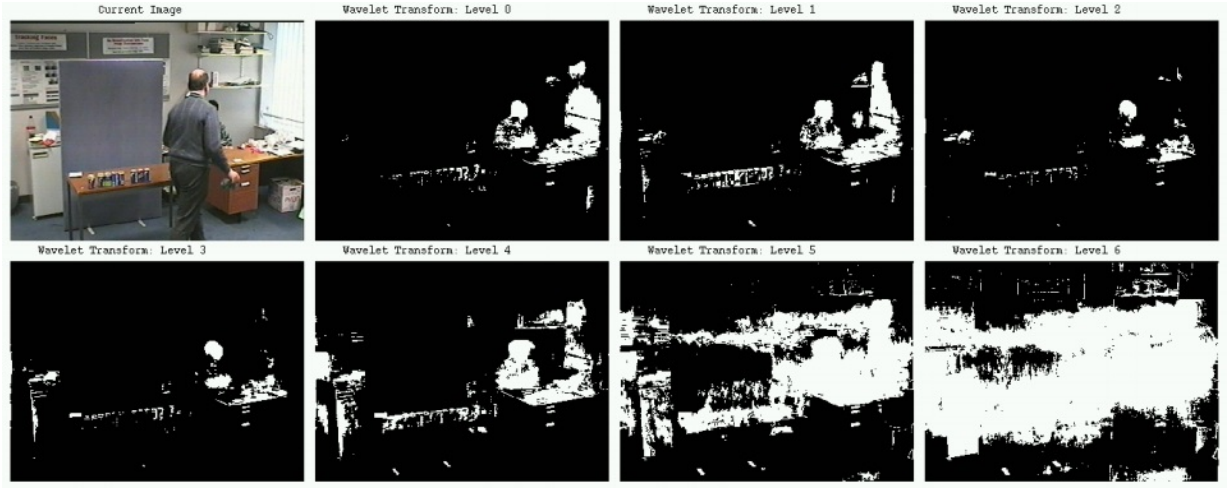


Figure 7. 7 level wavelet histogram of the shop scenario. From left to right, top to bottom, the figures are: original scene, then thresholded histograms in binary form from level 7 to level 1.

this case, the events are very clearly distinct from each other, with very dense detection rates. The main difficulty is that the cans being taken have not fallen into a class of their own. In fact they are not very distinct in the histogram of Figure 7. The reason is that the can is taken suddenly, with no subsequent change. In terms of temporal change, this would be hard to distinguish from noise. Since the wavelet method is based only on temporal difference and not absolute difference, it cannot detect such events. Rather it needs to be combined with the absolute change method.

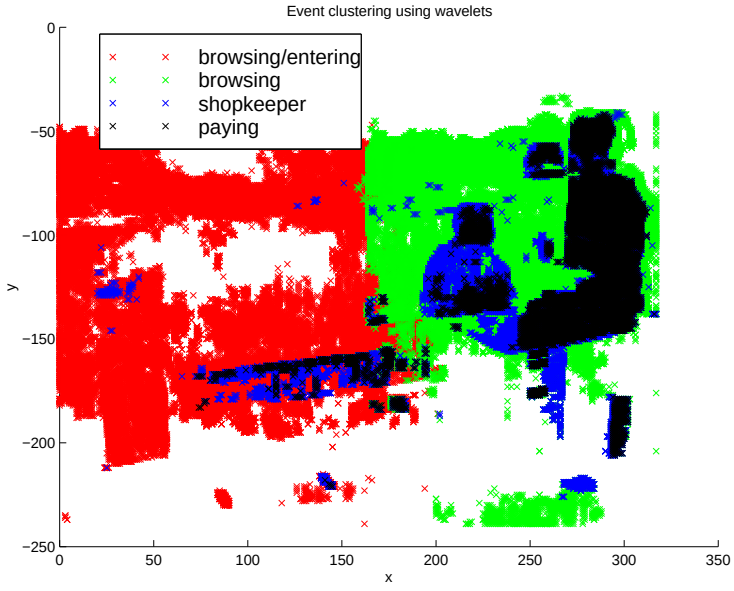


Figure 8. Clustered events using the wavelet model to detect multi-resolution change local events. Only the x-y positions of the events are shown.

5.4. Comparison

In comparing the two methods, we can see that the wavelet-based approach gives a much better clustering. One advantage of the Gaussian mixture model method is that one ends up with fewer events, making the algorithm computationally less expensive. The trade-off is that some grouping must be performed at a level that is too low to be robust. Note that the absolute change method

	can	paying	shopkeeper	browsing
can	0.0	**3.3698**	0.5837	0.6546
paying	0.0	0.0	1.3320	1.3819
shopkeeper	0.0	0.0	0.0	0.5691
browsing	0.0	0.0	0.0	0.0

Table 2. Causal correlation matrix showing causal connections between events.

involves a much more restrictive definition of an event at the low level. Semantics have been imposed through the interpretation of the background modelling results, spatial clustering of pixels into local events, and temporal grouping of instantaneous events at the same position. The multi-resolution method imposes no such semantics, but again suffers from a large number of resulting events. Both of the local event detection methods need to be combined in one final solution.

5.5. Causally Linking Events

Experimentally, the causal correlation matrix C shown in Table 2 was established. The standard deviation of the time-weighting Gaussian was $\sigma = 100$ frames. The element $C(A, B)$ shows the accumulated weighting of event B following event A. It can be seen in the table that there is a very strong causal link between taking a can and paying, highlighted in bold. There is also a relatively strong connection between paying and the shopkeeper moving, which makes sense since he is roused from his book when receiving the money.

In our case, the system developed the rule "if can is taken, then expect payment". An example is shown in Figure 9. A can been stolen (left box) which triggers display of the centre of the payment event cluster (right box). Until a payment event transpires, the box will continue to flash. Although this algorithm is quite simple and naive, the results are still quite powerful.

Figure 9. Example of the causal rule linking cans taken and payment. A can been stolen (left box) which triggers display of the centre of the payment event cluster (right box).

6. Conclusion

A methodology for detecting general visual events in a scene has been presented. Two algorithms for detecting local events have been presented and compared, one based on absolute change and the other on different rates of differential change. The methods were tested on real data.

Future work will involve further investigation of the wavelet approach, including the use of an adapted wavelet analysis to remove noise and reduce the number of detections. The two detection methods need to be merged to detect both absolute and relative change. An on-line implementation of the algorithm would enable experiments conducted over many days in real

situations. Finally, there is great scope for more work on using these visual events for high-level reasoning, such as finding causal connections.

References

Akansu, A. and Haddad, R. (1992). *Multiresolution Signal Decomposition: Transforms, Subbands, and Wavelets.* Academic Press.

Chomat, O., Martin, J., and Crowley, J. (2000). A probabilistic sensor for the perception and the recognition of activities. In *Proceedings of the Sixth European Conference on Computer Vision*, volume 1 of *Springer-Verlag Lecture Notes in Computer Science*, pages 487–503, Dublin, Ireland.

Haritaoglu, I., Harwood, D., and Davis, L. (2000). W^4: Real-time surveillance of people and their activities. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8):809–830.

McKenna, S., Jabri, S., Duric, Z., and Wechsler, H. (2000). Tracking interacting people. In *IEEE International Conference on Face & Gesture Recognition*, pages 348–353, Grenoble, France. IEEE Computer Society.

Morris, R. J. and Hogg, D. C. (2000). Statistical models of object interaction. *IJCV*, 37(2):209–215.

Ng, J. and Gong, S. (2000). Exploiting pixel-wise change for inferring global abnormal events in dynamic scenes. Technical Report QMW RR-00-07, Queen Mary, University of London.

Stauffer, C. and Grimson, W. (1999). Adaptive background mixture models for real-time tracking. In *IEEE Conference on Computer Vision and Pattern Recognition*, Colorado, USA.

Stauffer, C. and Grimson, W. (2000). Learning patterns of activity using real-time tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8):747–758.

Toyama, K., Krumm, J., brumitt, B., and Meyers, B. (1999). Wallflower: Principles and practice of background maintenance. In *IEEE International Conference on Computer Vision*, volume 1, pages 255–261, Corfu, Greece.

Wickerhauser, M. (1994). *Adapted Wavelet Analysis: from Theory to Software.* IEEE Press.