Spotting Scene Change for Indexing Surveillance Video

Andrew Graves and Shaogang Gong
Department of Computer Science
Queen Mary, University of London
London, England, E1 4NS
{andrew,sgg}@dcs.qmul.ac.uk

Abstract

We address the issue of forming a pre-attentive mechanism that can be used to analyse surveillance sequences. We address the problem of spotting scene change by performing temporal segmentation on long video sequences with little colour information and observed content. This is typical in surveillance sequences. Our approach: (1) employs sustained temporal change computed for local neighbourhoods in the image frames; (2) defines a frame activity similarity metric that accounts for local spatial and temporal displacement of change; and (3) monitors the similarity over a wide period to detect changes in emphasis that are then identified as scene breaks.

1 Introduction

As the number of surveillance deployments increases there becomes an urgent need to provide rapid access to the visual content via browsing and retrieval applications. We consider that a pre-attentive mechanism with no understanding of scene content is desirable for performing in real-time a number of critical tasks, including finding the 'scene breaks' in the sequence (temporal segmentation) and for detecting the introduction of alien objects to the scene (zero-motion detection). A computationally effective but also reliable method is required for rapid deployment, as opposed to recognition based systems that require training. A pre-attentive method should require no knowledge about the expected content or structure of the scene and can provide an effective pre-processing step to both fully-automatic systems that reason about the content and semi-automatic systems that provide assistance to a specialist users (eg: as in semi-automatic visual annotation).

We address the issue of performing temporal segmentation on surveillance sequences for the purpose of decomposing the sequence into a set of small video segments each considered to contain frames of similar content. Segment (or scene) breaks are found where the holistic interpretation of the sequence changes. It is significant to point out that many existing video segmentation and indexing systems exclusively target broadcasting and media applications, assuming the accessibility of highly elaborative multi-media data (synchronised audio, text and video) or well-structured scripts/story-line or manually labelled metadata. However, this is no longer valid in surveillance when the semantics of the recordings are unknown.

We acquired a number of surveillance sequences and from those we observed that they contain little or no colour information. The colour distribution in Hue-Saturation space is heavily centred at zero indicating that only grey-level information is available. This is problematic for existing video indexing and search models as most rely upon the capture of a high-quality feature space usually dominated by colour features. This defect is due to the use of low-quality recording media in surveillance systems. The colours that do exist are not sufficient for recognition or tracking tasks in contrast to the majority of approaches (eg: [15, 9]). Due to the nature of surveillance very small activities can in fact be rather important to the outcome but can be easily missed in detection.

An image can be analysed by performing a spatial decomposition to find object level content. This has been found useful for performing retrieval of images that are required to have a similar object level content [3]. In a sequence, tracking can be achieved by evolving blob level contents [9, 5], however these approaches require good quality features, prior knowledge about the expected object-level content (faces, mammals) and considerable training. Models dependent upon spatial features are also rather vulnerable to global changes in recording conditions (eg: erratic global illumination due to changes in cloud coverage). Owing to the nature of surveillance the scene is almost completely unconstrained and the number and type of expected activities is unknown. As in surveillance we are more interested in 'what is happening' in the scene rather than 'what is present' we prefer the use of temporal features to provide an initial scene analysis.

Temporal features are computed based upon the pixel level differences that can be observed over time. In our case we are dependent entirely upon grey-level changes. In [4] a number of motion energy receptive filters are used to capture motion at different orientations and timescales for the purpose of indoor scene recognition. Unfortunately the computation is expensive and the recognition process requires training using sets of known scenes. In [2, 16] the temporal change is inexpensively estimated using the amount of change that is observed and scene recognition models are trained. In our case we do not wish to perform recognition as this implies that a limited number of scenes are known to exist, rather we aim to identify those instances in the sequence where an unknown scene change may have occurred. In non-surveillance video scene-change detection is done by performing time constrained shot-grouping [7, 12, 8, 11]. Shots are found during a preprocessing step using sharp or gradual colour feature change between frames. In our case shots do not exist and so we must determine the scene-changes at frame level.

The rest of the paper is organised as follows: In Section 2 we describe our approach to performing temporal segmentation of surveillance sequences using temporal change. This involves computing the temporal change, providing a frame similarity metric and a method for analysing the change over a period of time. In Section 3 we then describe our experimental results. We conclude in Section 4.

2 Temporal Segmentation using Local Activity

2.1 Local Activity Representation

As surveillance sequences are long and the processing is required to be quick (for retrospect or real-time analysis) the temporal features are needed to be both computationally inexpensive and reliable. Methods for computing motion such as spatio-temporal zero-crossings [6] and Gabor based motion receptive fields [4] are not considered here to be

viable due to their cost. Instead we employ Pixel Change History [16], a derivative of Motion-History Images [2], to inexpensively estimate the amount of activity occurring using temporal change:

$$PCH_{\alpha,\beta}(x,y,t) = \begin{cases} \min(PCH(x,y,t-1)+\alpha, 255) & D(x,y,t)=1 \\ \max(PCH(x,y,t-1)-\beta, 0) & otherwise \end{cases} \quad (1)$$

where α and β are the accumulation and decay factors and $D(x,y,t)$ is a thresholded frame difference function, $|I(x,y,t)-I(x,y,t-1)| > T_d$, between two successive frames each smoothed with a Gaussian filter. The filter is used to reduce the effect of noise and blur. We use $\alpha = \beta = 50$. We use a T_d of 5 to capture change but to also allow a small oscillation in pixel appearance. The *PCH* result is a value between $[0,255]$ for each pixel where a high value indicates that sustained change has taken place at that position. This is an efficient scheme that provides a measure of current pixel activity and is robust to global illumination changes.

We reduce the spatial resolution of the *PCH* space to simplify the size of the feature space and to provide a coarser scene description. A similar approach proposed in [10] used a coarse scene description to successfully identify classes of scene activities within a constrained indoor scene. The image space is sub-divided into blocks of size 16×16. A block activity measure is computed per block using the ratio-of-occupancy of pixels exhibiting sustained change. We compute the Binary Cell Activity (*BCA*) for each frame:

$$BCA(x,y,t) = \left(\sum_{i=1}^{s} \sum_{j=1}^{s} PCH(xs+i, ys+j, t) > T_p \right) > T_c \quad (2)$$

where s=16 is the spatial cell dimension, T_p is a threshold of pixel *PCH* used to determine if a pixel is considered active, and T_c is a threshold on the number of active pixels used to determine if the cell is active. We use $T_p = 50$ and $T_c = 40$. We found these thresholds ensured that the block is marked active only if had considerable evidence of pixel activity.

We additionally compute the delay based asynchrony [10] to retain the time since the last block activation:

$$ASYN(x,y,t) = \begin{cases} t - last(x,y,t) & t - last(x,y,t) < T_a \\ T_a & otherwise \end{cases} \quad (3)$$

where $last(x,y,t)$ returns the last value of t at which the specified cell was active and T_a is the largest delay permitted. We use $T_a = 255$. An illustrative frame and it's computed *PCH*, *BCA* and *ASYN* is shown in Figure 1.

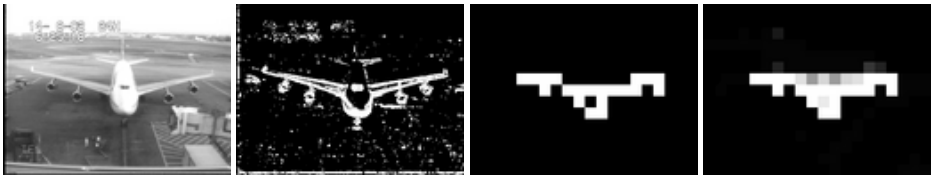

Figure 1: An illustrative frame from the aircraft docking sequence ($t = 540$) with it's computed *PCH*, *BCA* and *ASYN*.

2.2 Frame Similarity Metric

The local activity representation $BCA/ASYN$ captures action taking place at an instance in time. We now define a frame activity similarity metric as a measure of the cost of local activity change from one frame P to another Q over time. Distance metrics have been widely employed in image and video retrieval for the purpose of performing ranking [13]. In our case we wish to use the metric for monitoring the change in frame appearance over time for identifying discontinuities that can then be interpreted as scene breaks.

If we consider the BCA between two frames we can form three sets of cells: the set Ma of matching active cells; the set Mi of matching inactive cells; and the set N of non-matching cells. A frame similarity can be constructed that balances the positive evidence Ma and Mi against the negative evidence N:

$$Similarity_1(P,Q) = \left(\frac{sizeof(Ma)}{sizeof(Ma+Mi+N)} \right) \tag{4}$$

where $sizeof$ returns the size of the set. We reformulate as:

$$Similarity_2(P,Q) = exp\left(-\frac{Negative(P,Q)}{Positive(P,Q)+1} \right) \tag{5}$$

where $Negative_1(P,Q) = sizeof(N)$ and $Positive(P,Q) = sizeof(Ma)$ to ensure the result is in the range $[0,1]$ and to remove the effect of the unimportant cells Mi. A drawback is that the similarity will be low during a single continuous flowing movement in the sequence as different blocks become active at different times. This is as the similarity is brutal in that it does not consider either the locality of mismatches or the history of block activity. For example:

```
---     ---   Neg = 2        ---     ---   Neg = 1
-0- and --0   Pos = 0        -0- and -9-   Pos = 0
---     ---   Sim = 0.14     ---     ---   Sim = 0.37
```

where the numerals represent the delay since the last block activation (0 is active now) and '-' indicates a block that is not and has never been active. In the first case the activity is only displaced by 1 position and hence the similarity score should be high (a spatial displacement). In the second case the activity has recently disappeared and the similarity should also be high (a temporal displacement).

To compensate we adjust our negative evidence estimation function to account for the spatial locality and temporal delay of mis-matching cells. To estimate the amount of spatial displacement we consider the 3×3 neighbourhood of cells. To estimate the amount of temporal displacement we consider the delay since the last block activation. The negative evidence is computed as a combination of both:

$$Negative_2(P,Q) = \sum_{(x,y)\in N} \left(1 - \frac{Neigh_{(PorQ)}(x,y)}{sizeof(Neigh)} \right) \times \left(\frac{ASYN_{(PorQ)}(x,y)}{T_a} \right) \tag{6}$$

where $Neigh$ is the number of active blocks surrounding the x,y position in the frame, $sizeof(Neigh) = 8$ is the size of the neighbourhood, $ASYN$ is the delay asynchrony and T_a is the maximum delay permitted both defined in Equation 3. The choice of frame

$(P \, or \, Q)$ is made using the frame in which the particular cell x,y is not active. The result for each cell is a value between $[0,1]$ where a low value indicates that the mis-match is 'explained' by spatio temporal factors. In the previous example the similarities $(0.17, 0.34)$ are improved to $(0.17, 0.97)$ as the negative evidence is reduced from $(2,1)$ to $(1.75, 0.04)$

2.3 Temporal Segmentation

We now consider the question of performing an analysis of the sequence in order to perform temporal segmentation. Temporal segmentation requires the identification of those instances in the sequence at which the emphasis of the content is considered to have changed. In non-surveillance sequences sharp transitions are detected where the difference between two neighbouring frames exceeds a threshold. Gradual transitions are those that occur over a number of frames (eg: a 'wipe' or 'fade') [1] and are detected where the cumulative frame difference in a frame window exceeds a threshold. These methods are known to accurately detect breaks where the sequence changes from one camera position to another. This is not the case in our surveillance sequences which are taken from a single fixed camera position.

Our approach is inspired by the scene detection work of [7] and [8]. Shots are detected using the sharp/gradual method and are then grouped into scenes by computing the coherence of the past shots with the future shots. Each shot transition is so evaluated and those that have a low coherence are also identified as scene breaks. In our case we do not have shots and so we attempt to compute the coherence between past and future frames using a wide frame window:

$$Coherence(t) = \frac{\sum_{i=1}^{w/2} \text{median}\left(\forall_{j=1}^{w/2} Similarity(F_{t-i}, F_{t+j}) \right)}{w/2} \tag{7}$$

where w is the frame window size being used, F_i is the frame at time i and *Similarity* is a frame similarity function as defined in Section 2.2. The result is a value between $[0,1]$ that is high when there is similarity between the frames in the past $\{t-w/2, \ldots, t-1\}$ and future $\{t+1, \ldots, t+w/2\}$. A low result indicates a change in emphasis in the sequence. We compute the coherence at each frame in the sequence and identify the scene breaks at the minima in the result.

We automatically detect the minima at those points, t, where the coherence is the lowest in a surrounding window as suggested in [14]. In those situations where two or more points have the same lowest value the minima is retained at t that has the furthest distance to the next minima. The sequence is segmented into the desired number of breaks by choosing the minima that have the lowest score and are furthest from the neighbouring minima.

Figure 2: An illustrative frame from each of the eight manually identified scenes.

3 Experiments

A number of surveillance sequences were acquired for experiments that observe aircraft docking activities in a busy scene. One sequence was selected for manual analysis to obtain a rough understanding of the ground truth. The sequence is roughly 1.5 hours of footage sampled and digitised at 2Hz. The sequence consists of 11,000 frames each of size 320×240 represented in RGB. This produces a coarse size of 20×15. Upon inspection eight salient scenes were identified using human understanding:

frames 0-400	empty dock;
400-600	aircraft arrival;
600-2,700	passengers dis-embark and unloading;
2,700-5,700	plane re-stocked;
5,700-7,500	period of inactivity;
7,500-8,750	final loading;
8,750-9,500	engines examined; and
9,500-11,000	aircraft departure.

These manual scenes are illustrated in Figure 2. Some scenes are typified as containing a number of localised activities with a similar purpose (eg: vehicles moving to unload the aircraft), some contain small activities that occur over a long period and are semantically important (eg: engines examined), and some are used to encapsulate long periods of inactivity.

We performed automatic temporal segmentation on the first 2,000 frames using a frame window of size $w = 100$ and $w = 200$ in the computation of coherence (Equation 7). We use the frame similarity metric that explains spatial and temporal factors (Equations 5 and 6). Breaks were detected at the minimas as these are the points where the past is considered to have little correlation to the future (see Section 2.3). The coherence values and the detected breaks are shown in Figure 3. Manually identified breaks exist at positions 400 and 600. Both results detect breaks close the manual breaks due to the low level of frame coherence at this point. The difference in w makes little impact and so we prefer the smaller value owing to the reduced computational cost.

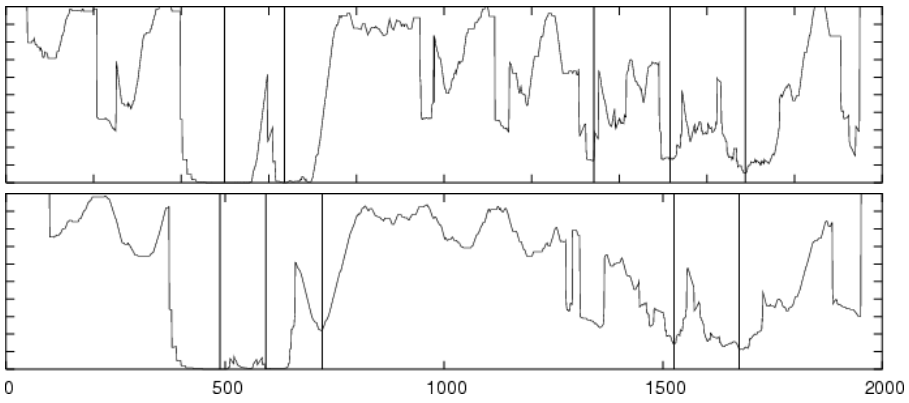

Figure 3: The frame coherence and for the first 2,000 frames using (a) $w = 100$ and (b) $w = 200$. The five automatic breaks that were detected are shown with vertical bars.

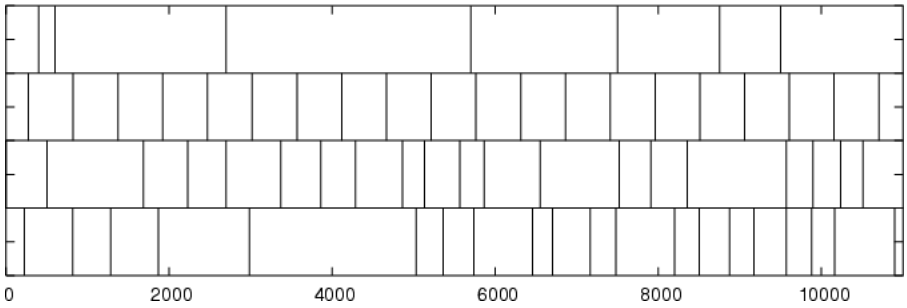

Figure 4: The positions the manual breaks in the 11,000 frame sequence: (a) the manual breaks; (b) the even temporal breaks; (c) the frame-coherence breaks; and (d) the gradual breaks.

	mean	variation
etb	154.3	69.1
fc	117.1	121.6
gb	132.4	91.4

Table 1: The mean and variance distance between the manual breaks and the automatically detected breaks for the two methods.

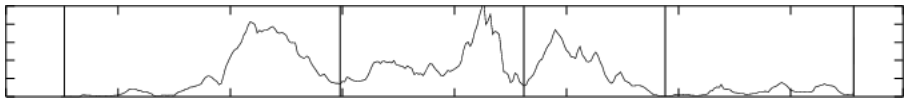

Figure 5: The coherence and automatic breaks in Sequence 2.

We performed temporal segmentation using our frame-coherence (fc) method on the entire sequence of 11,000 frames using a window size $w = 100$. To compare our method we introduced even temporal breaks (etb) at equidistant positions in the sequence. We also computed gradual breaks (gb) at the maxima of cumulative frame change, a method often used for identifying gradual transitions in traditional multimedia segmentation. In each case we used the top twenty breaks. The positions of the automatically detected breaks alongside the manual breaks are shown in Figure 4. We evaluated the correctness of the breaks using the mean distance from the manual breaks to the nearest detected break. The result shown in Table 1 indicates that the frame-coherence breaks are on average closest to the manual breaks. Both the fc and gb methods outperform etb indicating that there is some merit in performing temporal segmentation based upon frame content.

The frames and asynchrony representation (Equation 3) for the first three breaks detected using $w = 200$ are shown in Figure 6. We show a number of surrounding frames at positions $[t-100, t-50, t-25, t, t+25, t+50, t+100]$ where t is the break position $(488, 593, 722, 1525)$ to demonstrate the frame content at the detected breaks. Break 1 occurs upon the arrival of the aircraft. The position of the break is towards the middle rather than the start of the activity as this point has the lowest coherence. However, the coherence minima is very wide (see Figure 3) and so choosing the exact break position becomes difficult. Break 2 is accurately positioned where the sequence content alters from aircraft arrival to aircraft unloading. Break 3 is positioned where the unloading activities end and a number of smaller unrelated activities commence.

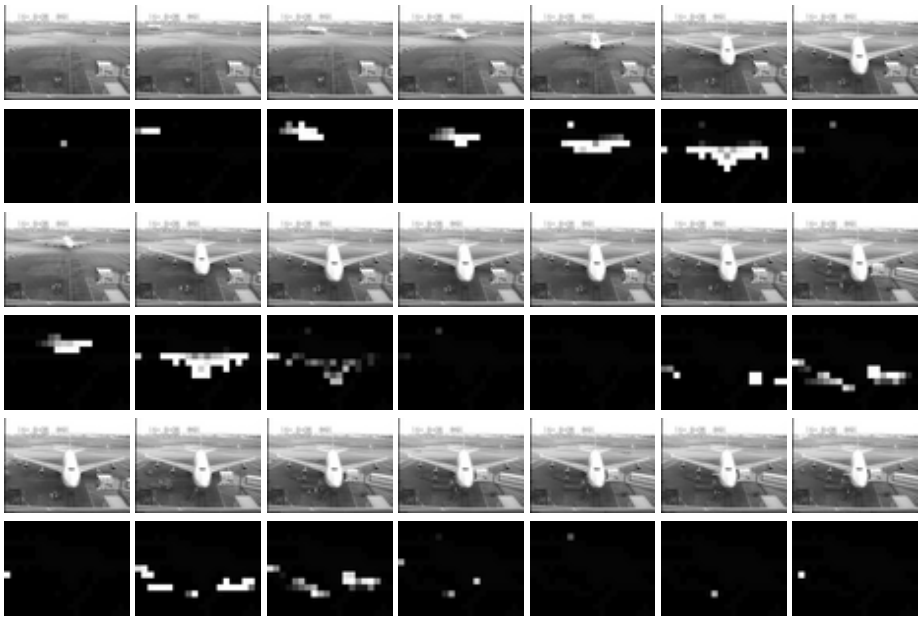

Figure 6: The frame and asynchrony representation at the first three automatically detected breaks using the frame-coherence method with $w = 200$.

As a final experiment we computed the frame coherence and automatic breaks for the first 2,000 frames on a second sequence that contained the similar aircraft docking scenario but with different global lighting characteristics. The result shown in Figure 5. Upon inspection it was found these breaks were positioned similarly to those computed on the first sequence.

4 Conclusions

We described a frame representation based upon temporal-change and a frame similarity metric. In particular, in the similarity metric we 'explain away' the inconsistancies between two frames using spatial and temporal factors. We then analyse the sequence by computing the frame-coherence between the past and future at each position. Scene breaks are identified at the minima of the coherence as they are considered to exist where the patterns of local activity change considerably over time.

The approach is computationally undemanding, operates without the use of colour information, requires no training and provides an effective scene change spotting mechanism which is essential for semantically based retrieval and browsing of surveillance video. The next steps are to consider the computation of coherence using the activities that are segmented from the sequence, to further consider the causality between representations, and to model the expected visual structure of temporal-change in order to perform scene independent zero-motion detection.

Acknowledgements

AG was funded by the EPSRC and we would like to thank BAA for providing the data under the DTI/EPSRC ICONS project. More information can be found on our website at `http://www.dcs.qmul.ac.uk/research/vision`.

References

[1] A. Del Bimbo. *Visual Information Retrieval*. Morgan Kaufmann Ed., San Francisco, USA, 1999.

[2] A.F. Bobick and J.W. Davis. The recogonition of human movement using temporal templates. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(3):257–267, March 2001.

[3] C. Carson, S. Belongie, H. Greenspan, and J. Malik. Blobworld: Image segmentation using expectation-maximization and its application to image querying. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(8):1026–1028, August 2002.

[4] O. Chomat, J. Martin, and J.L. Crowley. A probabilistic sensor for the perception and recognition of activities. In *Proceedings of the 6th European Conference on Computer Vision*, volume 1, pages 487–503, Dublin, Ireland, June 2000.

[5] M.R. Everingham and B.T. Thomas. Supervised segmentation and tracking of non-rigid objects using a 'mixture of histograms' model. In *IEEE International Conference on Image Processing*, pages 62–65, October 2001.

[6] S. Gong, S. McKenna, and A. Psarrou. *Dynamic Vision: From Images to Face Recognition*. Imperial College Press, London, England, 2000.

[7] J.R. Kender and B.L. Yeo. Video scene segmentation via continuous video coherence. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 367–373, Santa Barbara, June 1998.

[8] T. Lin, H.J. Zhang, and Q.Y. Shi. Video scene extraction by force competition. In *IEEE International Conference on Multimedia and Expo*, Tokyo, Japan, August 2001.

[9] S. McKenna, Y. Raja, and S. Gong. Tracking colour objects using adaptive mixture models. *Image and Vision Computing*, 1(17):225–231, 1999.

[10] J. Ng and S. Gong. On the binding mechanism of synchronised visual events. In *Proceedings of IEEE Workshop on Motion and Video Computing*, pages 112–117, Orlando, USA, December 2002.

[11] C.W. Ngo, T.C. Pong, and H.J. Zhang. Motion-based video representation for scene change detection. *International Journal of Computer Vision*, 50(2):127–142, November 2002.

[12] Y. Rui, S. Huang, and S. Mehrota. Exploring video structures beyond the shots. In *IEEE International Conference on Multimedia Computing and Systems*, pages 237–240, Austin, Texas, July 1998.

[13] A.W.M. Smeulders, M. Worring, S. Santini, A. Gupta, and R. Jain. Content-based image retrieval at the end of the early years. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(12):1349–1380, 2000.

[14] H. Sundaram and S.F. Chang. Video scene segmentation using video and audio features. In *IEEE International Conference on Multimedia and Expo*, New York, August 2000.

[15] M.J. Swain and D.H. Ballard. Color indexing. *International Journal of Computer Vision*, 7:11–32, 1991.

[16] T. Xiang, S. Gong, and D. Parkinson. Autonomous visual events detection and classification without explicit object-centred segmentation and tracking. In *Proceedings of British Machine Vision Conference*, volume 1, pages 233–242, Cardiff, UK, September 2002.