Moving Region Detection in Compressed Video

B. Ugur Töreyin[1], A. Enis Cetin[1], Anil Aksay[1], and M. Bilgay Akhan[2]

[1] Department of Electrical and Electronics Engineering
Bilkent University 06800 Bilkent
Ankara, Turkey
{ugur,cetin,anil}@ee.bilkent.edu.tr
[2] Visioprime 30 St. Johns Rd., St. Johns, Woking, Surrey
GU21 7SA, UK
bilgay.akhan@visioprime.com

Abstract. In this paper, an algorithm for moving region detection in compressed video is developed. It is assumed that the video can be compressed either using the Discrete Cosine Transform (DCT) or the Wavelet Transform (WT). The method estimates the WT of the background scene from the WTs of the past image frames of the video. The WT of the current image is compared with the WT of the background and the moving objects are determined from the difference. The algorithm does not perform inverse WT to obtain the actual pixels of the current image nor the estimated background. In the case of DCT compressed video, the DC values of 8 by 8 image blocks of Y, U and V channels are used for estimating the background scene. This leads to a computationally efficient method and a system compared to the existing motion detection methods.

1 Introduction

Video based surveillance systems are widely used in security applications. A typical system may be required to handle many cameras recording various locations. Some digital cameras have built-in data compression systems and provide only compressed video. In order to realize a computationally efficient automatic video processing system, it is required to process video in the compressed domain.

In this paper, it is assumed that the video is compressed either using the Discrete Cosine Transform (DCT) or the Wavelet Transform (WT). In the case of wavelet compressed video, the proposed moving object detection algorithm compares the WT of the current image with the WTs of the past image frames to detect motion and moving regions in the current image without performing an inverse wavelet transform operation. Moving regions and objects can be detected by comparing the wavelet transforms of the current image with the wavelet transform of the background scene which can be estimated from the wavelet transforms of the past image frames. If there is a significant difference between the two wavelet transforms then this means that there is motion in the video.

If there is no motion then the wavelet transforms of the current image and the background image ideally should be equal to each other or very close to each other due to quantization process during compression. Stationary wavelet coefficients belong to the wavelet transform of the background. This is because the background of the scene is temporally stationary [1]-[5]. If the viewing range of the camera is observed for some time, then the wavelet transform of the entire background can be estimated as moving regions and objects occupy only some parts of the scene in a typical image of a video and they disappear over time. On the other hand, pixels of foreground objects and their wavelet coefficients change in time. Non-stationary wavelet coefficients over time correspond to the foreground of the scene and they contain motion information. A simple approach to estimate the wavelet transform of the background is to average the observed wavelet transforms of the image frames. Since moving objects and regions occupy only a part of the image they can conceal a part of the background scene and their effect in the wavelet domain is canceled over time by averaging.

A similar argument is also valid for DCT compressed video. DCT of the background scene can be estimated from the DCTs of the past image frames [3]. Both AC and DC coefficients are used in [3]. In this paper only the DC values of 8 by 8 DCT blocks are used for motion detection. In [3], only the luminance information is used whereas in this paper both luminance and chrominance channels are used for motion detection. A significant change in the DC values of 8 by 8 image blocks of Y, U and V channels of the estimated background image and the DCT of the current image indicates a motion in video. Since only the DC values are used, a computationally efficient system is achieved.

Any one of the space domain approaches [2]-[8] for background estimation can be implemented in compressed domain providing real-time performance. For example, the background estimation method in [2] can be implemented by simply computing the wavelet or discrete cosine transforms of both sides of their background estimation equations.

2 Hybrid Algorithm for Moving Object Detection

Background subtraction is commonly used for segmenting out objects of interest in a scene for applications such as surveillance. There are numerous methods in the literature [1]-[5]. The background estimation algorithm described in [2] uses a simple IIR filter applied to each pixel independently to update the background and use adaptively updated thresholds to classify pixels into foreground and background. This is followed by some post processing to correct classification failures. Stationary pixels in the video are the pixels of the background scene because the background can be defined as temporally stationary part of the video. If the scene is observed for some time, then pixels forming the entire background scene can be estimated because moving regions and objects occupy only some parts of the scene in a typical image of a video. A simple approach to estimate the background is to average the observed image frames of the video. Since moving objects and regions occupy only a part of the image, they conceal a part of the background scene and their effect is canceled over time by averaging. Our main concern is real-time performance of the system. In Video Surveillance and Monitoring (VSAM) Project at Carnegie Mellon University [2] a recursive background estimation method was developed from the actual image data. Let $I_n(x,y)$ represent the intensity (brightness) value at pixel position (x,y) in the n^{th} image frame I_n. Estimated background intensity value at the same pixel position, $B_{n+1}(x,y)$, is calculated as follows:

$$B_{n+1}(x,y) = \begin{cases} aB_n(x,y) + (1-a)I_n(x,y) & \text{if } (x,y) \text{ is non-moving} \\ B_n(x,y) & \text{if } (x,y) \text{ is moving} \end{cases} \tag{1}$$

where $B_n(x,y)$ is the previous estimate of the background intensity value at the same pixel position. The update parameter a is a positive real number close to one. Initially, $B_0(x,y)$ is set to the first image frame $I_0(x,y)$. A pixel positioned at (x,y) is assumed to be moving if the brightness values corresponding to it in image frame I_n and image frame I_{n-1}, satisfy the following inequality:

$$|I_n(x,y) - I_{n-1}(x,y)| > T_n(x,y) \tag{2}$$

where $I_{n-1}(x,y)$ is the brightness value at pixel position (x,y) in the $(n-1)^{st}$ image frame I_{n-1}. $T_n(x,y)$ is a threshold describing a statistically significant brightness change at pixel position (x,y). This threshold is recursively updated for each pixel as follows:

$$T_{n+1}(x,y) = \begin{cases} aT_n(x,y) + (1-a)(c|I_n(x,y) - B_n(x,y)|) & \text{if } (x,y) \text{ is non-moving} \\ T_n(x,y) & \text{if } (x,y) \text{ is moving} \end{cases} \tag{3}$$

where c is a real number greater than one and the update parameter a is a positive number close to one. Initial threshold values are set to an experimentally determined value. As it can be seen from (3), the higher the parameter c, higher the threshold or lower the sensitivity of detection scheme. It is assumed that regions significantly different from the background are moving regions. Estimated background image is subtracted from the current image to detect moving regions. In other words all of the pixels satisfying:

$$|I_n(x,y) - B_n(x,y)| > T_n(x,y)\ . \tag{4}$$

are determined. These pixels at (x,y) locations are classified as the pixels of moving objects.

3 Moving Region Detection in Compressed Domain

Above arguments and the methods proposed in [6] and [7] are valid in compressed data domain as well, [3]. In [3], DCT domain data is used for motion detection in video. Our paper covers both wavelet and DCT based compressed video. The wavelet transform of the background scene can be estimated from the wavelet coefficients of past image frames, which do not change in time, whereas foreground objects and their wavelet coefficients change in time. Such wavelet coefficients belong to the background because the background of the scene is temporally stationary. Non-stationary wavelet coefficients over time correspond to the foreground of the scene and they contain motion information. If the viewing range of the camera is observed for some time, then the wavelet transform of the entire background can be estimated because moving regions and objects occupy only some parts of the scene in a typical image of a video and they disappear over time. Similarly, DC-DCT coefficients of the background scene can be estimated from the corresponding coefficients of the past image frames. Stationary coefficients correspond to background whereas non-stationary ones over time belong to the foreground of the scene.

Let B be an arbitrary image. This image is processed by a single stage separable Daubechies 9/7 filterbank and four quarter size subband images are obtained. Let us denote these images as $LL(1), HL(1), LH(1), HH(1)$ [9]. In a Mallat wavelet tree, $LL(1)$ is processed by the filterbank once again and $LL(2), HL(2), LH(2), HH(2)$ are obtained. Second scale subband images are the quarter size versions of $LL(1)$. This process is repeated several times in a typical wavelet image coder. DCT compressed images used in this paper encode a 2-D image using the DCT coefficients of 8 by 8 image regions. Only the DC-DCT coefficients are used for motion detection. DC-DCT coefficients of 8 by 8 image blocks of an image and a three scale wavelet decomposition of the same image are shown in Fig. 1.

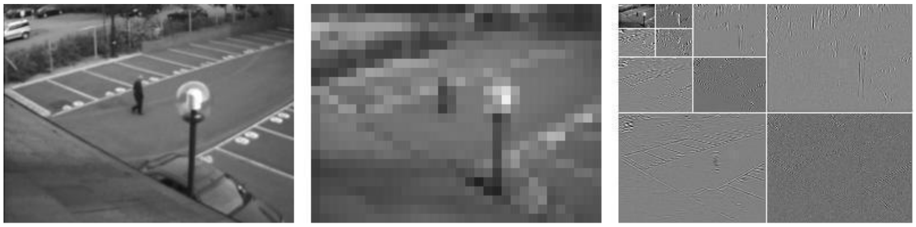

Fig. 1. Original image(left), the DC-DCT coefficients of 8 by 8 image blocks of the image(middle) and its corresponding three levels of the wavelet tree consisting of subband images (luminance data is shown)

Let D_n represent any one of the subband images of the background image B_n at time instant n. The subband image of the background D_{n+1} at time instant $n+1$ is estimated from D_n as follows:

$$D_{n+1}(i,j) = \begin{cases} aD_n(i,j) + (1-a)J_n(i,j) & \text{if } (i,j) \text{ is non-moving} \\ D_n(i,j) & \text{if } (i,j) \text{ is moving} \end{cases} \tag{5}$$

where J_n is the corresponding subband image of the current observed image frame I_n. The update parameter a is a positive real number close to one. Initial subband image of the background, D_0, is assigned to be the corresponding subband image of the first image of the video I_0. In Equations (1)-(4), (x,y)'s correspond to the pixel locations in the original image, whereas in (5) and in all the equations in this section, (i,j)'s correspond to locations of subband images' wavelet coefficients. In DCT compressed video, $D_n(i,j)$ and $J_n(i,j)$ represent the DC value of the $(i,j)^{th}$ block of the corresponding images at time instant n.

A wavelet coefficient at the position (i,j) in a subband image or a DC-DCT coefficient of the $(i,j)^{th}$ block is assumed to be moving if

$$|J_n(i,j) - J_{n-1}(i,j)| > T_n(i,j) \tag{6}$$

where $T_n(i,j)$ is a threshold recursively updated for each wavelet or DC-DCT coefficient as follows:

$$T_{n+1}(i,j) = \begin{cases} aT_n(i,j) + (1-a)(b|J_n(i,j) - D_n(i,j)|) & \text{if } (i,j) \text{ is non-moving} \\ T_n(i,j) & \text{if } (i,j) \text{ is moving} \end{cases} \tag{7}$$

where b is a real number greater than one and the update parameter a is a positive real number close to one. Initial threshold values can be experimentally determined. As it can be seen from the above equation, the higher the parameter b, higher the threshold or lower the sensitivity of detection scheme. Estimated compressed image of the background is subtracted from the corresponding compressed image of the current image to detect the moving coefficients and consequently moving objects as it is assumed that the regions different from the background are the moving regions. In other words, all of the coefficients satisfying the inequality

$$|J_n(i,j) - D_n(i,j)| > T_n(i,j) \tag{8}$$

are determined.

It should be pointed out that there is no fixed threshold in this method. A specific threshold is assigned to each coefficient and it is adaptively updated according to (7).

Once all the coefficients satisfying the above inequalities are determined, locations of corresponding regions on the original image are determined. For the wavelet compressed video, if a single stage Haar wavelet transform is used in data compression then a wavelet coefficient satisfying (8) corresponds to a two by two block in the original image frame I_n. For example, if $(i,j)^{th}$ coefficient of the subband image $HH_n(1)$ (or other subband images $HL_n(1)$, $LH_n(1)$, $LL_n(1)$) of I_n satisfies (8), then this means that there exists motion in a two pixel by two pixel region in the original image, $I_n(k,m)$, $k = 2i, 2i-1, m = 2j, 2j-1$, because of the subsampling operation in the discrete wavelet transform computation. Similarly, if the $(i,j)^{th}$ coefficient of the subband image $HH_n(2)$ (or other second scale subband images $HL_n(2)$, $LH_n(2)$, $LL_n(2)$) satisfies (8) then this means that there exists motion in a four pixel by four pixel region in the original image, $I_n(k,m), k = 4i, 4i-1, 4i-2, 4i-3$ and $m = 4j, 4j-1, 4j-2, 4j-3$. In general, a change in the l^{th} level wavelet coefficient corresponds to a 2^l by 2^l region in the original image. In DCT compressed video, if DC-DCT coefficient of $(i,j)^{th}$ block is found to be moving, then this means that there exists motion in an 8 by 8 region in the original image, $I_n(k,m), k = 8i, 8i-1, 8i-2, .., 8i-7$ and $m = 8j, 8j-1, 8j-2, .., 8j-7$.

In this paper, the wavelet compressed video is obtained using Daubechies' 9/7 biorthogonal wavelet. In this biorthogonal transform, the number of pixels forming a wavelet coefficient is larger than four but most of the contribution comes from the immediate neighborhood of the pixel $I_n(k,m) = (2i, 2j)$ in the first level wavelet decomposition, and $(k,m) = (2^l i, 2^l j)$ in the l^{th} level wavelet decomposition, respectively. Therefore, in this paper, we classify the immediate neighborhood of $(2i, 2j)$ in a single stage wavelet decomposition or in general $(2^l i, 2^l j)$ in the l^{th} level wavelet decomposition as a moving region in the current image frame, respectively.

Determining the moving pixels of the corresponding regions as explained separately for wavelet and DCT based compressed video above, the union of these regions on the original image is formed to locate the moving region(s) in the video. These pixels are processed by a region growing algorithm to include the pixels located at immediate neighborhood of them. This region growing algorithm checks whether the following condition is met for these pixels:

$$|J_n(i+m, j+m) - D_n(i+m, j+m)| > K\ T_n(i+m, j+m) \tag{9}$$

where $m = \pm 1$, and $0.8 < K < 1$, $K \in \mathbf{R}^+$. If this condition is satisfied, then that particular pixel is also classified as moving. After this classification of pixels, moving regions are formed and encapsulated by their minimum bounding boxes.

4 Experimental Results

The above algorithm is implemented using C++ 6.0, running on a 1500 MHz Pentium 4 processor. The PC based system can handle 16 video channels captured at 5 frames per second in real-time. Each image fed by the channels has the frame size of PAL composite video format, which is 720 pixel by 576 pixel.

The video data is available in compressed form. For the wavelet compressed video, only the lowest resolution part of the compressed video bit-stream is decoded to obtain the low-low, low-high, high-low, and high-high coefficients which are used in moving object detection. Higher resolution wavelet sub-images are not decoded.

The performance of our algorithm is tested using different video sequences and real-time data. 76 of the test sequences are reported in this paper. These sequences have different scenarios, covering both indoor and outdoor videos under various lighting conditions containing different video objects with various sizes. Some example snapshots of wavelet and DCT compressed domain methods are shown in Fig. 2.

The moving regions are also detected over 180 by 144 size images by using the hybrid method of VSAM [2]. Another widely used background estimation method is based on Gaussian Mixture Modelling [8]. However, this method is computationally more expensive than other methods.

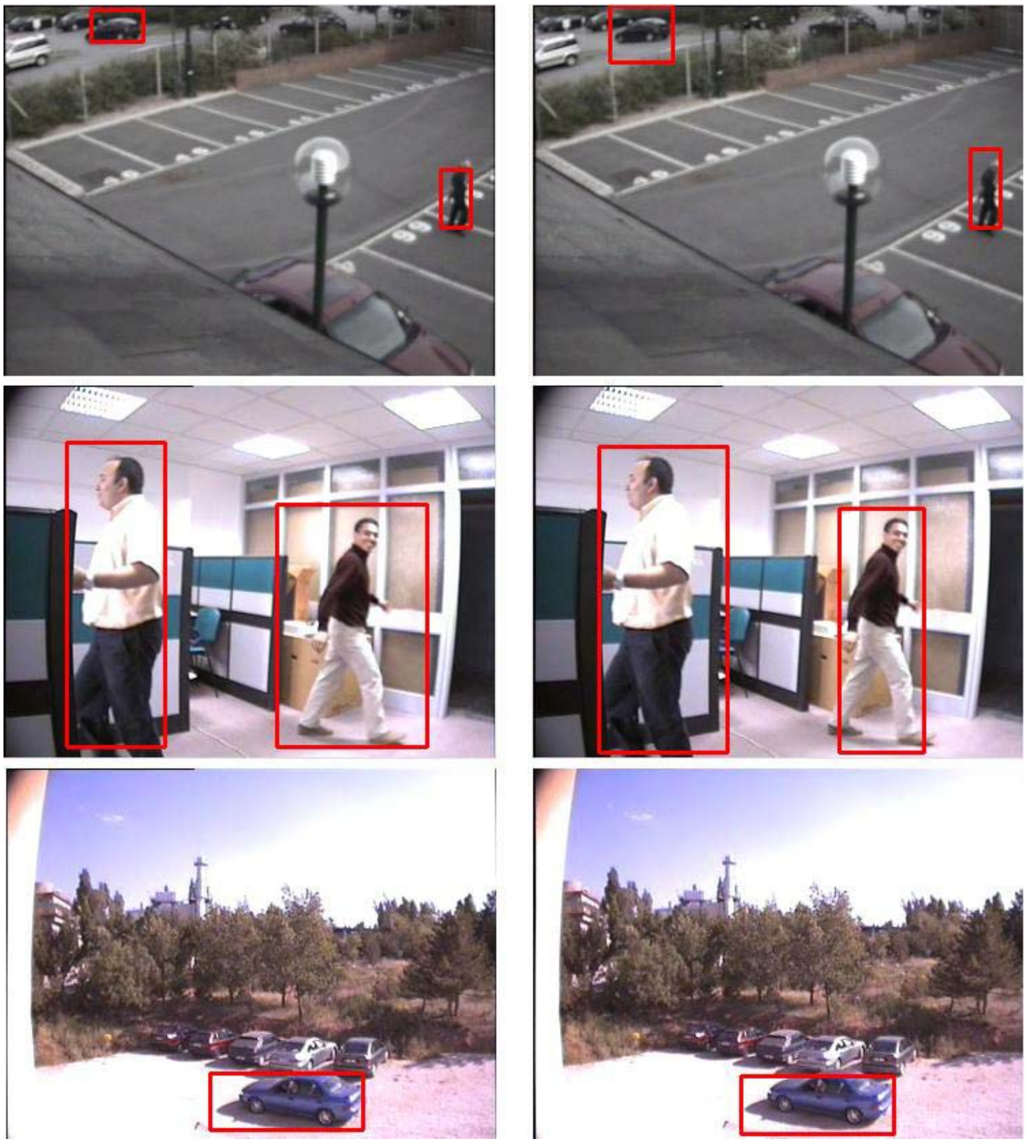

Fig. 2. Some detection results of DCT(left) and wavelet compressed domain methods

Moving objects of various sizes are successfully detected by these methods as summarized in Tables 1 and 2. The numbers listed in these tables are the frame numbers of frames in which detection took place. For example, MAN1 object in VIDEO-3 sequence in Table 1 is detected at the 15^{th} frame in all three methods, namely our methods utilizing the compressed data only and the method of VSAM [2].

Table 1. Comparison of motion detection methods with videos having large moving objects. All videos are captured at 10 fps except for VIDEO-4 which is captured at 5fps

Large Object Videos	Object	Compressed Domain Method		VSAM
		Wavelet	DCT	
VIDEO-1	MAN1	28	29	28
	MAN2	41	42	41
VIDEO-2	MAN1	19	19	19
	MAN2	75	75	75
VIDEO-3	MAN1	15	15	15
	MAN2	38	38	38
	MAN3	44	44	44
	MAN4	75	75	74
VIDEO-4	TRUCK1	6	6	4

Table 2. Comparison of motion detection methods with videos having small moving objects. VIDEO-5 is captured at 5 fps whereas the other videos are captured at 25 fps

Small Object Videos	Object	Compressed Domain Method		VSAM
		Wavelet	DCT	
VIDEO-5	MAN1	21	21	21
	MAN2	32	32	32
VIDEO-6	CAR1	55	55	55
	CAR2	62	62	62
	CAR3	63	64	63
	CAR4	98	100	98
VIDEO-7	CAR1	88	89	88

Motion detection results in videos containing objects with sizes ranging from 20 by 20 to 100 by 100 objects are presented in Table 1. Such large moving objects are detected about at the same time by all methods. In Table 2, motion detection results of the algorithms with videos containing objects having sizes comparable to 8 by 8 are presented. In these videos, there is not much difference in terms of time delay between the methods, as well.

Time performance analysis of the methods are also carried out. The method of VSAM is implemented using videos with frame-size of 180 by 144. This image data is extracted from the low-low image of the 2^{nd} level wavelet transform. Our method uses all the coefficients in the 4^{th} level subband image, including low-low, high-low, low-high and high-high subimages. For the DCT based method, 360 by 288 image frames are fed to our system. Macro image blocks of 8 by 8 are formed to obtain the DC-DCT coefficients. Hence, the data handled by the system are equal in amount for both of the compressed domain methods. Performance results show that compressed domain method is significantly faster than the method of VSAM. Our method processes an image in $1.1msec$, whereas ordinary VSAM method processes an image in $3.1msec$, on the average. It is impossible to process 16 video channels consisting of 180 by 144 size images simultaneously using the VSAM and GMM based motion detection methods in a typical surveillance system implemented in a PC.

In indoor surveillance applications, the methods does not produce false alarms. On the other hand, in outdoor applications, false alarms occur in both of the methods due to leaves and tree branches moving in the wind, etc., as shown in Table 3.

Table 3. Frame numbers of some outdoor videos at which false alarms occur when leaves of the surrounding trees move with the wind. Indoor videos yield no false alarms

Videos	Compressed Domain Method		VSAM
	Wavelet	DCT	
OUTDOOR-1	126, 163	126, 163	87, 126, 163
OUTDOOR-2	No false alarms	No false alarms	No false alarms
INDOOR-1	No false alarms	No false alarms	No false alarms
INDOOR-2	No false alarms	No false alarms	No false alarms

Motion sensitivity of our compressed domain method can be adjusted to detect any kind of motion in the scene, by going up or down in the wavelet pyramid for the wavelet compressed video and playing with the parameter b in equation (7) for both of the compression types. However, by going up to higher resolution levels in the pyramid, the processing time per frame of the compressed domain method approaches to that of the ordinary background subtraction method of VSAM. Similarly, false alarms may be reduced by increasing b in (7) at the expense of delays in actual alarms.

5 Conclusion

A method for detecting motion in compressed video using only compressed domain data without performing the inverse transform is developed. The main advantage of the proposed method compared to regular methods is that it is not only computationally efficient but also it solves the bandwidth problem associated with video processing systems. It is impossible to feed the pixel data of 16 video channels into the PCI bus of an ordinary PC in real-time. However, compressed video data of 16 channels can be handled by an ordinary PC and its buses, hence real-time motion detection can be implemented by the proposed algorithm.

References

1. Foresti, G.L., Mahonen, P., Regazzoni, C.S.: Multimedia Video-Based Surveillance Systems: Requirements, Issues and Solutions, Kluwer, (2000)
2. Collins, R.T., Lipton, A.J., Kanade, T., Fujiyoshi, H., Duggins, D., Tsin, Y., Tolliver, D., Enomoto, N., Hasegawa, O., Burt, P., Wixson, L.: A System for Video Surveillance and Monitoring: VSAM Final Report. Tech. Report CMU-RI-TR-00-12, Carnegie Mellon University (1998)
3. Ozer, I.B., Wolf, W.: A Hierarchical Human Detection System in (Un)Compressed Domains. IEEE Transactions on Multimedia. (2002) 283–300
4. Haritaoglu, I., Harwood, D., Davis, L.: W4: Who, When, Where, What: A Real Time System for Detecting and Tracking People. Third Face and Gesture Recognition Conference. (1998) 222–227
5. Bagci, M., Yardimci, Y., Cetin, A.E.: Moving Object Detection Using Adaptive Subband Decomposition and Fractional Lower Order Statistics in Video Sequences. Elsevier, Signal Processing. (2002) 1941–1947
6. Naoi, S., Egawa, H., Shiohara, M.: Image Processing Apparatus. U.S. Patent 6,141,435. (2000)
7. Taniguchi, Y.: Moving Object Detection Apparatus and Method. U.S Patent 5,991,428. (1999)
8. Stauffer, C., Grimson, W.E.L.: Adaptive Background Mixture Models for Real-Time Tracking. Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition. (1999) 246–252
9. Antonini, M., Barlaud, M., Mathieu, P., Daubechies, I.: Image Coding Using Wavelet Transform. IEEE Transactions on Image Processing. Vol.1, No.2. (1992) 205–220