

# A New Method for Moving Object Extraction and Tracking Based on the Exclusive Block Matching

Zhu Li, Kenichi Yabuta, and Hitoshi Kitazawa

Department of Electrical and Electronic Engineering, Tokyo University of Agriculture and Technology 2-24-16 Naka-cho, Koganei-shi, Tokyo, 184-8588 Japan  
lizhu@m.ieice.org, kyabuta@m.ieice.org, kitazawa@cc.tuat.ac.jp

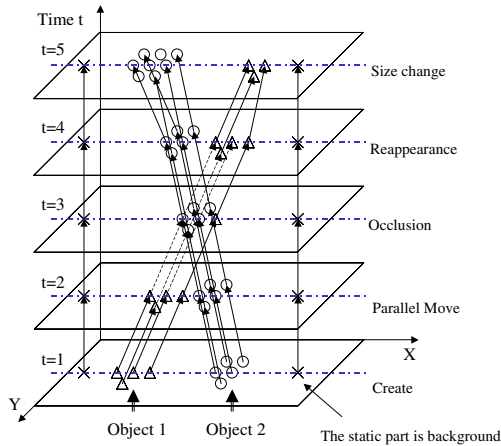
**Abstract.** Robust object tracking is required by many vision applications, and it will be useful for the motion analysis of moving object if we can not only track the object, but also make clear the corresponding relation of each part between consecutive frames. For this purpose, we propose a new method for moving object extraction and tracking based on the exclusive block matching. We build a cost matrix consisting of the similarities between the current frame's and the previous frame's blocks and obtain the corresponding relation by solving one-to-one matching as linear assignment problem. In addition, we can track the trajectory of occluded blocks by dealing with multi-frames simultaneously.

## 1 Introduction

Tracking objects is an omnipresent elementary task in online and offline image-based applications including traffic surveillance, motion capture, robot vision, etc. After years of researches, many efficient methods have been proposed. However, in order to achieve accurate motion analysis of moving object, it is necessary to obtain the corresponding relation of each part between consecutive frames. Background subtraction[1], which is one of the commonly used techniques for moving object extraction, can not provide the corresponding relation of each part. Mean-Shift[2] and Particle Filter[3][4] which are very popular for object tracking in present research do not obtain it in detail.

On the other hand, optical flow methods estimate pixel motion between two frames. It can obtain the corresponding relations on level of pixel by optical flow method such as Block Matching method, Horn-Schunck[7] method and Lucas-Kanade[8] method. However, aperture problem which is existing in optical flow estimation can give rise to a problem that many blocks match same or neighboring position in a area covered with uniform color. This problem causes error matching. Moreover, it is impossible to extract feature points in such area that intensity changes smoothly by Lucas-Kanade method.

In order to avoid the situation that destinations of matched blocks are too close or overlap, we assume block matches in such a way that destinations are mutually exclusive and propose a method to obtain the optimal matching using linear assignment. Different from the algorithms base on the graph matching[5][6],



**Fig. 1.** Spatio-temporal tracking

our method simply performs block matching which does not require the graph structures of moving objects. Moreover, compared with the feature of nodes and edges of graph, the features of small blocks shows higher robustness under some situations such as view point change caused by object motion and illumination change. The proposed method aims to achieve spatio-temporal continual object tracking even in the case of occlusion or shape change which is shown in Fig. 1.

## 2 Exclusive Block Matching

### 2.1 The Matching between Current Frame and Previous Frame

This section describes the basic method of block matching. Firstly, we scan the input images by block to convert the images into 1 dimensional data. If we assume that the block size is  $n \times n$  pixels, the width and the height of the image are  $w$  and  $h$ , respectively, the number of blocks  $N$  is given by the equation  $N = w/n \times h/n$  as shown in Fig. 2. We build an  $N \times N$  array consisting of the similarities (actually difference measure or distance) between the current frame's (Curr) and the previous frame's (Prev) blocks. It is required to perform all Curr blocks by assigning exactly one Prev block to each Curr block in such a way that the total cost of the assignment is minimized. Then this problem could be solved as a linear assignment problem.

However, it is impossible to archive the one-to-one assigning as long as the object moves in the scene. This is caused by various situations involving hiddenness and reappearance of background, occlusion, creation and vanishment of moving object. Therefore, this problem can not be simply solved as a linear assignment problem.

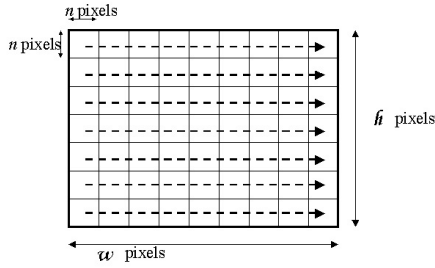


Fig. 2. Scan an image into 1 dimensional data

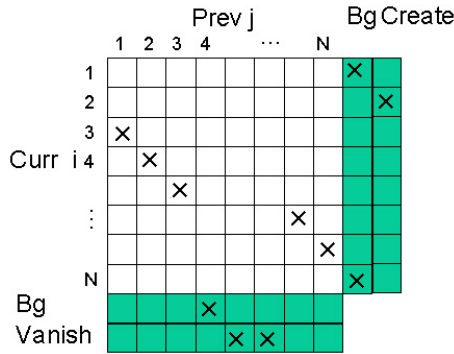


Fig. 3. Expand the matrix considering background, occlusion, creation, and vanishment

### 2.2 Expand the Matrix Considering Background, Occlusion, Creation and Vanishment

In order to solve this problem, we expand the basic matrix by adding 2 rows and 2 columns as shown in Fig. 3. The columns correspond to appearance of background and creation of blocks and the rows correspond to hiddenness of background and vanishment of blocks. We calculated the distances between current block and previous blocks to decide which column's block is the best match. If the distance between current frame's block and background's block is closer than the distance between current frame's and the previous frame's block, this block is matched with the Bg column. If neither previous frame nor Bg column can be matched, this block is matched with the Create column. In the same way, the blocks of previous frame which match neither the current frame's nor the Bg's blocks are matched with the Vanish row. This matching problem becomes the following linear programming problem.

Minimize

$$z = \sum_{i=1}^{N+2N+2} \sum_{j=1}^{N+2N+2} p_{ij} c_{ij}$$

Subject to

$$\begin{aligned} \sum_{j=1}^{N+2} p_{ij} &= 1 & i &= \{1, 2, \dots, N\} \\ \sum_{i=1}^{N+2} p_{ij} &= 1 & j &= \{1, 2, \dots, N\} \\ p_{ij} &= \{0, 1\} & \{i, j\} &= \{1, 2, \dots, N+2\} \\ \text{however, } p_{ij} &= \{0\} & \{i, j\} &= \{N+1, N+2\} \end{aligned}$$

$$c_{ij} = \begin{cases} \text{dist}\{Curr_i, Prev_j\} & i = \{1, \dots, N\}, j = \{1, \dots, N\} \\ \text{dist}\{Curr_i, Bg_j\} & i = \{1, \dots, N\}, j = N+1 \\ \text{dist}\{Bg_i, Prev_j\} & i = N+1, j = \{1, \dots, N\} \\ \text{penalty for creating} & i = \{1, \dots, N\}, j = N+2 \\ \text{penalty for vanishing} & i = N+2, j = \{1, \dots, N\} \end{cases}$$

$\text{dist}\{Curr_i, Prev_j\}$ : distance between the current frame's blocks  $i$  and the previous frame's blocks  $j$ .

$\text{dist}\{Curr_i, Bg_j\}$ : distance between the current frame's blocks  $i$  and the background's blocks  $j$ .

$\text{dist}\{Bg_i, Prev_j\}$ : distance between the background's blocks  $i$  and the previous frame's blocks  $j$ .

*penalty for creating*: fixed value chosen when there is no block similar with it. If this value is chosen, this block is regarded as creating of new block.

*penalty for vanishing*: fixed value chosen when there is no block similar with it. If this value is chosen, this block is regarded as vanishing of block or occlusion.

The part which consists of the  $N \times N$  array is an exclusive assignment problem. However, the part which is composed of the Bg row, Bg column, Create row and Vanish column is a partial assignment problem which is likely to be solved out with plural choices. There is no guarantee that such a problem can be solved out in a short time. For this reason, we ignore Bg and Vanish row to convert it to the linear assignment problem which is shown in Fig. 4. Only the diagonal elements can be selected in the part of Bg and Create. Since every row is assigned to exactly one column, only  $N$  columns are selected. Although the Bg and Vanish row are taken out, the blocks in previous frame which do not match with the blocks in the current frame (shown by  $\triangle$  in Fig. 4) are regarded as matched with Bg or Vanish.

### 2.3 Multi-frame Expanding

Although we can discern the movement of each block by matching between current frame's and previous frame's blocks, it is still impossible to discern some situations just like occlusion and reappearance. So we expand the matrix again

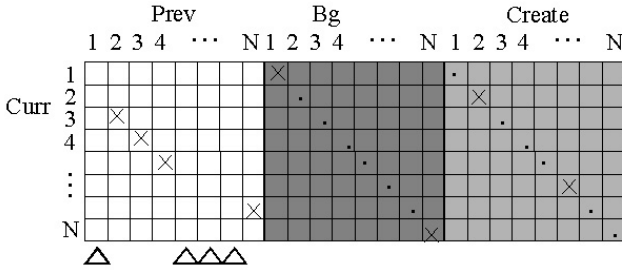


Fig. 4. Assignment problem considering background and creation

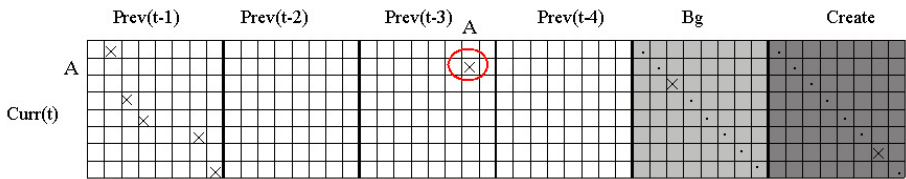


Fig. 5. Expand the cost matrix dealing with plural previous frames

by dealing with plural previous frames, as shown in Fig. 5. For example, block A in the current frame is matched with it in the frame at the moment t-3. The block A is regarded as occluded at the moment t-1 and t-2. The value in ( ) denotes the time of the frame.

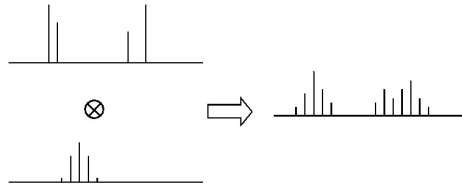
### 2.4 Similarity Measure

Beacuse the capability of matching is depend heavily on the calculation method of similarity, it is necessary to choose an appropriate measure. Through comparison of various experimental results, we adopt the Bhattacharyya coefficient[3] defining a distance on HSV histograms to measure the similarity between 2 blocks. The mathematical formulation of this measure is given by Eq. (1) and Eq. (2), where  $p$  and  $q$  represent 2 normalized HSV histogram.

$$\rho[p, q] = \sum_{u=1}^m \sqrt{p^{(u)}q^{(u)}} \tag{1}$$

$$d = \sqrt{1 - \rho[p, q]} \tag{2}$$

The HSV histogram is composed of  $m = N_h N_s + N_v$  bins and we set  $N_h$ ,  $N_s$ , and  $N_v$  to 10. So the  $m$  becomes 110[3].



**Fig. 6.** Smooth the histogram

### 3 Enhance Robustness for Real Data

In this section we describe our proposed method to enhance robustness for real data.

#### 3.1 Improvement of Similarity Measure

Due to noises, changed illumination environment and the small size of block, a small perturbation of RGB values will cause Bhattacharyya distance great change. As shown in Fig. 6, we calculate a convolution with Gaussian function the mean of which is 0 to smooth the S and V histogram. Considering H is invariant with respect to brightness variations, we do not perform the smooth processing on H. The histogram must be normalized after convolution processing. In addition, we can add some other feature quantities with invariance property in changed illumination environment to improve the robustness under some of the situations such as shadows. Here we use one of the texture features which is statistically represented as variance of color histogram.

Moreover, there is a contradiction between the precision in similarity calculation and the size of blocks. The smaller the block size is, the higher resolution which means more image details of moving object we can get. However, the less pixels each block contains, the lower the precision in similarity calculation becomes. In other words, the amount of blocks with same color information becomes higher. In order to solve this problem in some extent, when we calculate the HSV histogram of a block, not only do we calculate the color information of this block, but also calculate the sum of color informations of the 8 blocks around it for addition. In this method, the HSV histogram will be calculated repeatedly for overlapping blocks.

#### 3.2 Restriction of Block State Transition

In our system, each block corresponds to a situation such as moving object, static object, background, etc. We thus consider the state transition of blocks. In section 2.3, a block can become arbitrary state according to the optimal assignment. However, as shown in Fig. 7 state of block changes under some rules. Firstly, states of all blocks are set to Bg (background) by initialization.

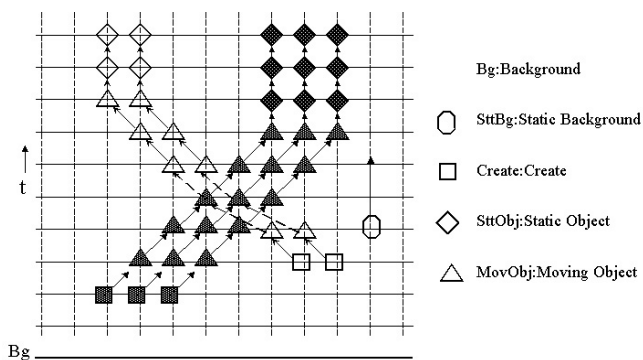


Fig. 7. States of blocks

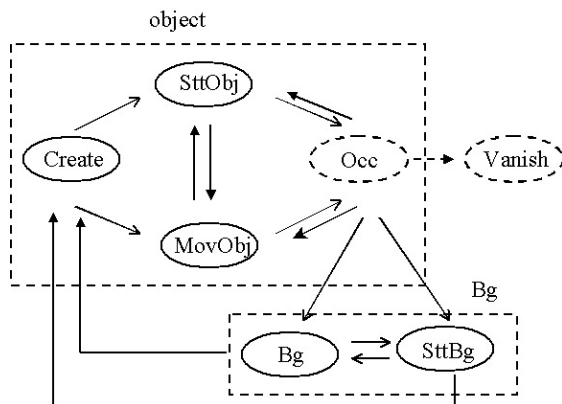


Fig. 8. Restriction of block state transition

Now we focus on the state of one block. As the block changes under assignment, if the distance from background is bigger than a pre-assigned threshold, this block's state will be set to Create which means the block belongs to an object. When it is matched with the next frame, if the block moves to another position, the state will be classified as MovObj (moving object). If it is matched with the block which is on the same position, the state is classified as SttObj (static object). Under condition that none of the blocks in the scene are matched with it, the state is set to Occ (occlusion). So long as it can not be matched with MovObj or SttObj again within a pre-assigned number of frames, it is classified as Vanish. Besides, the blocks which are matched with background are classified as Bg. Finally, if the block does not belong to object and matched with the block on same position, it will be classified as SttBg (Static background). Figure 8 shows the diagram of allowed block state transition.

### 3.3 Shape Preserving

Through the proposed method gives the exactly optimal matching, the proper flows what humans see can not always be calculated. It is impossible to track moving object accurately only using color information such as RGB values and HSV histogram. In the Horn-Schunck optical flow calculation method[7], it is assumed that motion vector changes smoothly almost everywhere in a scene. Unfortunately, it is impossible to transform this assumption to the linear assignment problem. We thus assume that the shape of object does not change abruptly and adjust the matching cost matrix. We obtain the mean vector from the initial matching result, then perform matching again with adding a penalty value to each block according to its distance from the mean vector, as shown in Fig. 9.

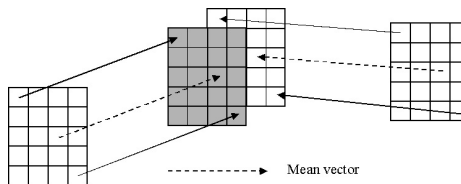
## 4 Experimental Results

### 4.1 Comparison with Optical Flow Methods

In order to prove the validity of our method, we show the experimental results of CG generated data at first. Two boxes move parallelly and occlusion occurs at the moment of frames 4 and 5. Figure 10 shows the experimental results of our method and 3 kinds of optical flow methods. The optical flow is calculated by OpenCV library[9]. We adjust the values and the numbers of parameters of optical flow methods to obtain the best results we can get. In the experimental results, red line represents the block which matches with block in previous frame (t-1). Yellow line represents the block matches with a block in frame (t-2). The green and blue line represent blocks which are matched with a blocks in frame (t-3) and (t-4), respectively. Experiment result indicates that the traditional optical flow methods induce unnatural flows because of pushing and entering of background. On the other hand, our proposed method avoid well this problem, and obtained flows are almost parallel.

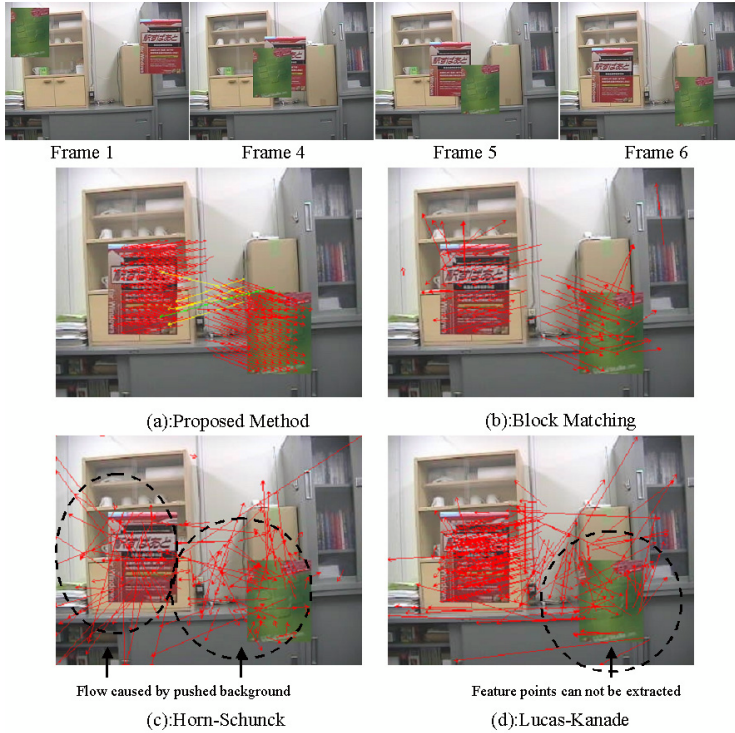
### 4.2 Magnification and Rotation

Although we assume blocks match exclusively, our method considering vanishing and creating is applicable for magnification and rotation as well. Figure 11 shows the experimental results of CG data. Accurate tracking was almost realized. As



**Fig. 9.** Shape preserving





**Fig. 10.** The comparison between proposed method and conventional optical flow methods

the boundaries of blocks change, the similarities between some blocks in current frame and in frame ( $t-2$ ) become higher than them between blocks in current frame and previous frame ( $t-1$ ). This causes the error matching represented as yellow flows which appear when occlusion does not occur.

### 4.3 CG Data with Occlusion

The next example shown in Fig. 12(a) is the trajectory of two boxes under occlusion in spatio-temporal space. We use the same data which was used in Fig. 10. The excellent result shows that the continuous tracking is not interrupted even under occlusion. In addition, flows are almost parallel that means the corresponding relation of each part between consecutive frames is obtained accurately.

### 4.4 Real Data

Finally, Fig. 12(b) gives a result of real data with occlusion. Tracking fails after occlusion, flows of blocks on the parts of clothes of which color is very similar



**Fig. 11.** Magnification and Rotation

mixed together. Figure 13 shows another real data with different actions of a person. Although there are some inaccurate flows and miss extraction, the obtained flow shows the motion of each part of the person and it will be useful for motion analysis.

#### 4.5 Linear Assignment Algorithm and Execution Time

In our experiment, we adopted the Munkres assignment algorithm[10] which is one of the implementations of Hungarian method to solve the linear assignment problem. We also adopted sparse-matrix calculation to improve the processing speed. In addition, the restriction of state transition avoid a great many unnecessary calculations and greatly improve the processing speed. The image sequences are processed with a Core 2 Duo 3.00 GHz PC under Window XP. The image size is  $320 \times 240$  pixels and the block size is  $8 \times 8$  pixels. The running time increases with the amount of flows. It is about  $0.2sec \sim 3sec$  (averagely  $1sec$ ) per frame. For example, the average processing time of Bhattacharyya calculation and linear assignment for the scenes which are shown in Fig. 13 are  $0.38sec$  and  $0.47sec$ , respectively. The linear assignment is repeated twice for the shape preserving. This speed is not enough to realize real time processing. However, it is possible to greatly improve the processing speed by various approximate calculations of linear assignment[11] or using hardware to solve it.

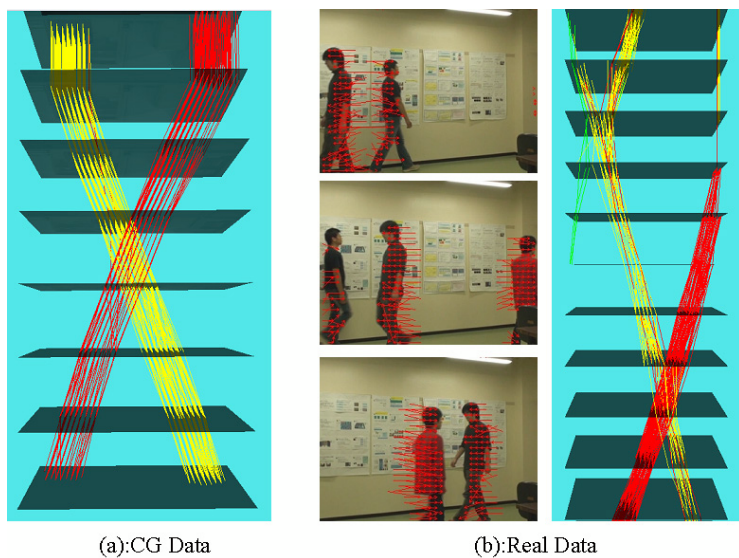


Fig. 12. Spatio-temporal space trajectory of objects when occlusion occur



Fig. 13. Real data with different actions

## 5 Conclusions

This paper has proposed a new method for moving object extraction and tracking based on exclusive block matching. This method has been successfully used to track moving object even in the case of occlusion and provides the corresponding relation of each part between consecutive frames. The assumption that block matches exclusively has improved the error matching caused by aperture problem. But it can not avoid it thoroughly when the colors of blocks are same or very similar. We, therefore, can make clear that tracking is limited if we merely use color information to calculate similarity. Our future work should focus on improving the approach considering shape similarity and connectivity in consecutive blocks.

## References

1. Stauffer, Grimson, W.E.L.: Adaptive background mixture models for real-time tracking. In: CVPR 1999, Fort Collins, CO, p. 2246 (June 1999)
2. Comaniciu, D., Meer, P.: A Robust Approach Toward Feature Space Analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 24(5), 603–619 (2002)
3. Perez, P., Hue, C., Vermaak, J., Gangnet, M.: Color-Based Probabilistic Tracking. In: Heyden, A., Sparr, G., Nielsen, M., Johansen, P. (eds.) ECCV 2002. LNCS, vol. 2350, pp. 661–675. Springer, Heidelberg (2002)
4. Nummiaro, K., Koller-Meierand, E., Van Gool, L.: A Color-based Particle Filter. *Image and Vision Computing* (2002)
5. Caetano, T.S., Cheng, L., Le, Q.V., Smola, A.J.: Learning Graph Matching. In: *IEEE 11th International Conference on Computer Vision, ICCV 2007*, 14–21 October 2007, pp. 1–8 (2007)
6. Jiang, H., Drew, M.S., Li, Z.-N.: Linear Programming Matching and Appearance-Adaptive Object Tracking. In: Rangarajan, A., Vemuri, B.C., Yuille, A.L. (eds.) *EMMVCPR 2005*. LNCS, vol. 3757, pp. 203–219. Springer, Heidelberg (2005)
7. Berthold, K., Horn, P., Schunck, B.G.: Determining Optical Flow. *Artificial Intelligence* 17, 185–203 (1981)
8. Bouguet, J.-Y.: Pyramidal Implementation of the Lucas Kanade Feature Tracker. The paper is included into OpenCV distribution
9. OpenCV, [http://opencv.jp/document/opencvref\\_cv.html](http://opencv.jp/document/opencvref_cv.html)
10. Munkres assignment algorithm, <http://cscslab.murraystate.edu/bob.pilgrim/445/munkres.html>
11. Trick, M.A.: A Linear Relaxation Heuristic For The Generalized Assignment Problem. *Naval Research Logistics* 39, 137–152 (1992)