An intelligent video categorization engine

G.Y. Hong
Institute of Information and Mathematical Sciences, Massey University, Auckland, New Zealand

B. Fong
Department of Electrotechnology, Auckland University of Technology, Auckland, New Zealand, and

A.C.M. Fong
School of Computer Engineering, Nanyang Technological University, Singapore

Abstract

Purpose – We describe an intelligent video categorization engine (IVCE) that uses the learning capability of artificial neural networks (ANNs) to classify suitably preprocessed video segments into a predefined number of semantically meaningful events (categories).

Design/methodology/approach – We provide a survey of existing techniques that have been proposed, either directly or indirectly, towards achieving intelligent video categorization. We also compare the performance of two popular ANNs: Kohonen's self-organizing map (SOM) and fuzzy adaptive resonance theory (Fuzzy ART). In particular, the ANNs are trained offline to form the necessary knowledge base prior to online categorization.

Findings – Experimental results show that accurate categorization can be achieved near instantaneously.

Research limitations – The main limitation of this research is the need for a finite set of predefined categories. Further research should focus on generalization of such techniques.

Originality/value – Machine understanding of video footage has tremendous potential for three reasons. First, it enables interactive broadcast of video. Second, it allows unequal error protection for different video shots/segments during transmission to make better use of limited channel resources. Third, it provides intuitive indexing and retrieval for video-on-demand applications.

Keywords Cybernetics, Video, Neural nets

Paper type Research paper

Introduction

The ability of machines to extract semantically meaningful objects or events out of video footage has tremendous potential applications including synchronization of audio and visual (A-V) information (e.g. lip reading), optimal transmission of video over band-limited channels, and object-based video clip indexing and retrieval (e.g. finding specific frames in video footage).

In A-V data synchronization, the goal is to provide the viewer a natural feel in terms of A-V sensation, e.g. when the video shows a person speaking. Unequal error protection has been applied to transmission of scalable video to ensure an optimum quality of service is provided at all times. With the advent of MPEG-4, which treats a video frame as a composition of video objects (VOs), it is possible to apply unequal error protection to different VOs according their relative importance. Also, one can

Kybernetes
Vol. 34 No. 6, 2005
pp. 784-802

0368-492X
DOI 10.1108/03684920510595490

apply unequal error protection to different scenes according to their relative importance.

Object-based indexing and retrieval can be used to facilitate video-on-demand (VOD) applications allowing viewers to use high-level search queries. Currently, typical search queries for video clips are by means of specification of low-level features (e.g. color, brightness and texture information) (Deng and Manjunath, 1998) or by example (Kung and Hwang, 1998).

Clearly, current technology is inadequate and machine understanding of video footage could bring automation to the object-based indexing and retrieval processes.

However, achieving full machine understanding of video footage is far from trivial. Since artificial neural networks (ANNs) can be trained to exhibit human-like intelligence in object recognition and classification, we investigate two suitably trained ANNs to categorize incoming video footage into a finite number of predefined scenarios. In particular, we describe an intelligent video categorization engine (IVCE) that enables a computer to attempt to make sense of scenes captured on video near-instantaneously. For experimental purposes, we have predefined a number of scenarios such as vehicle movement and certain human motions. A computer is trained to categorize a temporal video segment into one of the predefined scenarios (events), where a temporal video segment is defined as all frames F_i such that i is a finite integer in the range $L \leq i \leq N$, and a scene change is detected between frames F_{L-1} and F_L and between frames F_N and F_{N+1}. These temporal video segments are commonly referred to as video shots in the literature.

As shown in Figure 1, our IVCE comprises an offline training process and an online categorization process. The purpose of the offline training process is to train the ANNs using some training video footages to generate suitable ANN models for subsequent categorization. Two ANNs are investigated: Kohonen's self-organizing map (SOM) (Kohonen, 1995) and Fuzzy Adaptive Resonance Theory (Fuzzy ART) (Carpenter *et al.*, 1991). Both ANNs are used to form clusters based on a similarity measure between different temporal video segments. In particular, members within each cluster have maximal similarity among themselves and minimal

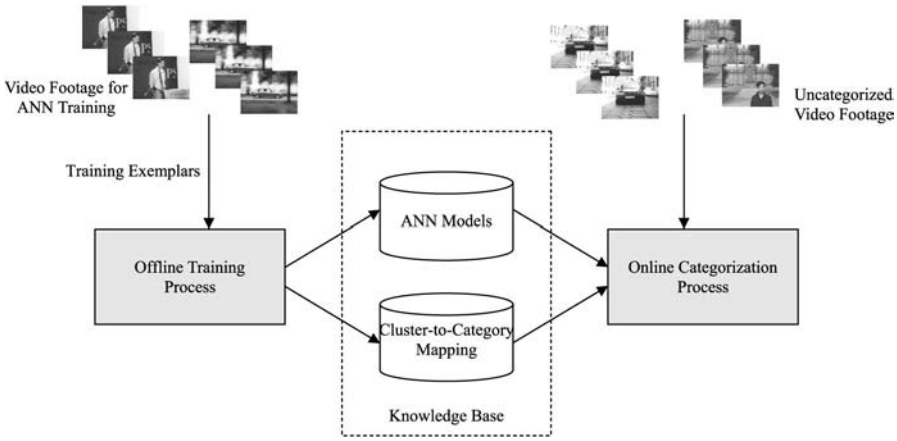


Figure 1.
Intelligent video categorization engine

similarity between them and other clusters. Once the clusters are formed, it is necessary to provide cluster-to-category mapping for subsequent categorization. During the online categorization process, incoming video footages are temporally segmented and individually categorized into a predefined scenario using the trained ANNs.

The rest of the paper is organized as follows. In the next section, we review existing VO segmentation and motion tracking technologies reported in the literature. We then survey statistical and neural network techniques for classification. The details of our IVCE are then presented. This is followed by a discussion on experimental results. Finally, we conclude the paper.

VO segmentation and motion tracking

Intra-frame VO segmentation and inter-frame motion tracking of the segmented objects are key techniques required for a computer to extract semantic meanings from the video footage. Current VO extraction techniques reported in the literature include spatial and temporal operations. Virtually all techniques require some measure of homogeneity, typically in terms of some low-level features such as grayscale, color and texture. Spatial VO segmentation techniques, e.g. (Salembia *et al.*, 1995) treat each video frame as a two-dimensional (2D) function $f(x,y)$ and are generally derived from classical image processing. However, a fundamental limitation of these techniques is that homogeneous regions segmented by these techniques do not necessarily correspond to semantically meaningful objects.

Compared to 2D still images, video provides a 3D of useful information in the time domain. It is generally assumed that semantically meaningful objects have coherent and homogeneous motions between video frames. Motion tracking provides important information for segmentation of semantically meaningful objects across video frames, which amounts to following the trajectory of a feature temporally across frames. It can also provide useful information to estimate the current location of the tracked feature based on previous frames. Sometimes, an entire region of interest (ROI) can be tracked, but this approach tends to be compute-intensive. It is generally preferable to track selected feature points, e.g. head top points (Shao *et al.*, 2000) instead.

In addition, if the camera moves in relation to the scene it captures, the resultant motions will become more complex for analysis. Under tightly controlled studio environments, it is possible to use high-precision camera rotation and zoom parameters to determine global motion due to camera movement, and the global motion can then be cancelled accordingly (Zheng *et al.*, 2001). However, we have not adopted this approach in our research for generality. We apply global motion compensation instead.

From the above discussion, we observe that both spatial and temporal methods are useful for VO segmentation. In fact, most techniques reported in the literature employ a combination of both approaches. Further, it appears that human intervention is mostly necessary. The fundamental problem is that while humans can easily determine what is/is not semantically meaningful, it is not a trivial matter for a machine to perform such tasks. For example, NeTra-V (Deng and Manjunath, 1998) is reported to perform well using low-level content description, but still lacks high-level content description capabilities. Overall, fully automated VO segmentation methods reported only operate in very restrictive conditions and semiautomatic methods appear to be most promising

(Meier and Ngan, 1999). We therefore adopt a semiautomatic framework for VO segmentation similar to (Gu and Lee, 1998), which involves some human intervention for very reliable segmentation. In particular, the method requires active human supervision during the segmentation of the *I*-frame for each shot. Thereafter, tracking of the segmented objects is performed automatically for the remaining *P*-frames in the shot. We take a different approach so that human intervention is limited to confirming the correctness of the automatic spatio-temporal VO segmentation process. This high level of reliability in VO segmentation is crucial for subsequent tracking and scenario categorization.

Statistical and neural classification techniques

So-called intelligent classification techniques, such as K-nearest neighbor (KNN) (Yang, 1999) and linear discriminant analysis (LDA) (Koehler and Erenguc, 1990), have been developed to classify objects into different groups mainly according to the statistical occurrence of sets of features. However, most of these techniques have been developed for textual information. So, a challenge is to extract features to characterize video footage. In addition, real-world video data maybe noisy and ill defined, they cannot always be describable with linear or low-order statistical models. Thus, we need some models that can tolerate data with noise while giving high performance in classification. ANNs are robust enough to fit a wide range of distributions accurately and have the ability to model any high degree exponential models (Dalton and Deshmane, 1991). They can also exhibit human-like intelligence through generalization of knowledge during a training process. In this research, we study the Kohonen's SOM (Kohonen, 1995; Flexer, 2001) and Fuzzy ART (Carpenter *et al.*, 1991) ANN models for video shot classification. In particular, we extract important features from training video shots to form training exemplars for adapting the ANN to form ANN models suitable for subsequent classification.

Kohonen's self-organizing map

SOM (Kohonen, 1995; Flexer, 2001) is a competitive ANN that provides a topological mapping from the input space to clusters. It provides a non-linear projection of the input pattern space of arbitrary dimensionality onto a 1 or 2D array of neurons. The array exists in a space that is separate from the input space, and any number of inputs may be used as long as the number of inputs is greater than the dimensionality of the output space.

Essentially, the SOM algorithm is a stochastic version of K-means clustering method (Balakrishnan, 1994). The fundamental difference is that in the case of SOM, the neighboring clusters are updated in addition to the winner clusters. The projection makes the topological neighborhood relationship geometrically explicit. The network tries to find clusters such that any two clusters that are close to each other in the output space have input vectors that are close to each other as well. This is achieved by finding the best output neuron while at the same time activating its spatial neighbors to react to the same input vector. The SOM training algorithm is given in Figure 2.

Fuzzy adaptive resonance theory

Fuzzy ART (Carpenter *et al.*, 1991) is an adaptive clustering technique originally inspired by neurophysiology. Figure 3 shows the structure of the Fuzzy ART ANN, which consists of three layers of nodes:

Step 1: Initialize the weight vector of all the output neurons.

Step 2: Determine the output winning neuron m by searching for the shortest normalized Euclidean distance between the input vector and the weight vector of each output neuron.

$$| \mathbf{X} - \mathbf{W}_m | = \min_{j=1\ldots M} | \mathbf{X} - \mathbf{W}_j |$$

where $\mathbf{X}$ is the input vector,

$\mathbf{W}_j$ is the weight vector of output neuron j, and

M is the total number of output neuron.

Step 3: Let $N_m(t)$ denote a set of indices corresponding to a neighborhood size of the current winner neuron m. The neighborhood size needs to be slowly decreased during the training session. The weights of the weight vector associated with the winner neuron m and its neighboring neurons are updated by

$$\Delta \mathbf{W}_j(t) = \alpha(t)[\mathbf{X}(t) - \mathbf{W}_j(t)] \qquad \text{for } j \in N_m(t)$$

where α is a positive-valued learning factor, $\alpha \in [0, 1]$. It needs to be slowly decreased with each training iteration.

Thus, the new weight vector is given by

$$\mathbf{W}_j(t+1) = \mathbf{W}_j(t) + \alpha(t)[\mathbf{X}(t) - \mathbf{W}_j(t)] \qquad \text{for } j \in N_m(t)$$

Steps 2 and 3 are repeated for every exemplar in the training set for a user-defined number of iterations.

Figure 2.
Training algorithm for SOM

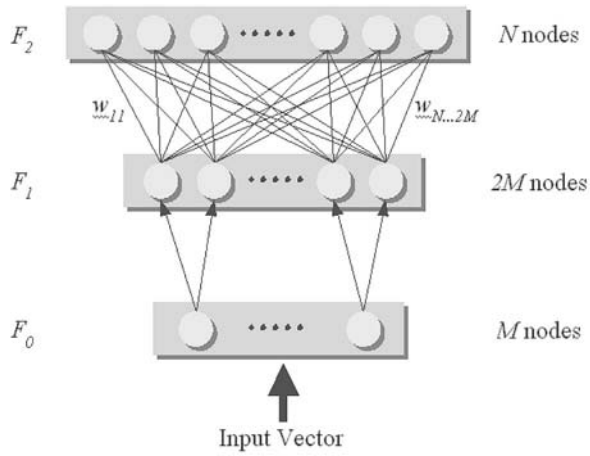


Figure 3.
Structure of the Fuzzy ART network

- preprocessing layer F_0, which transforms the input pattern using complement coding;
- category representation layer F_2, in which clusters are formed at the committed nodes; and
- input layer F_1, which receives both bottom-up input pattern from F_0 layer and top-down weights representing cluster seeds from F_2 layer.

For an input vector of M-dimensions, the number of nodes at the F_0 layer is M. After going through complement coding, the input vector becomes $2M$-dimensional. Thus, the F_1 layer will need $2M$ nodes to receive the complement-coded vector. The F_2 layer comprises N nodes, where N is the maximum number of clusters that can be accommodated by the network. Each of the N nodes in F_2 layer has an associated weight vector, denoted by $\mathbf{W}_j$ for the F_2 layer node j, of which the weights are those that emanate from that F_2 layer node and converge to all of the F_1 layer nodes. During the learning process, a node at the F_2 layer is committed if it has previously coded an input pattern. Otherwise, it is uncommitted. Before training starts, the weights in the weight vector of all F_2 layer nodes are initialized to one and all F_2 layer nodes are uncommitted. When the learning process ends, clusters containing coded input patterns are formed at all committed F_2 layer nodes.

Like other ART networks, Fuzzy ART can operate between *plastic* and *stable* modes. Thus, it can still be trained whenever a new input pattern is fed to the network, even the training session has already been completed previously. There are three parameters that describe the dynamics of the network:

- choice parameter, $\alpha(\alpha > 0)$, which alters the bottom-up inputs from all of the F_1 layer nodes produced at each F_2 layer node;
- vigilance parameter, $\rho(\rho \in [0, 1])$, which indicates the threshold level of how close an input pattern must be to a stored cluster seed before a match is said to occur. Higher values of ρ will result in more precise categorization of objects and thus more F_2 layer nodes will become committed; and
- learning parameter, $\beta(\beta \in [0, 1])$, which manipulates the adjustments to the weight vector $\mathbf{W}_J$, where node J is the chosen cluster which fulfils the vigilance match.

If $\beta = 1$, the learning process is considered fast. Otherwise, it is called slow learning. A special type of learning process, which is termed as fast-commit slow-recode, is characterized by applying fast learning for uncommitted F_2 nodes and slow learning for committed F_2 nodes. The training algorithm for the Fuzzy ART network is given in Figure 4.

Intelligent video categorization engine

In an attempt to extract semantic meaning from video footage, Haering *et al.* (2000) have proposed a three-stage algorithm for detecting hunts in wildlife video. In the first stage, they extract low-level features from the video and use a back-propagation ANN to classify image regions into objects such as sky, grass, animal, etc. Next, they generate shot descriptors that attempt to link objects with their temporal and spatial relationships. Finally, they manually design an event inference mechanism based on a finite state machine to specifically detect hunting sequences in video.

In this research, we take a significantly different approach towards achieving some form of machine understanding of video footage. In particular, we use ANN to perform event inference (categorization). Prior to that, we apply spatio-temporal VO segmentation techniques to isolate main objects from the background at the beginning of each shot. Low-level features, such as color and texture, provide spatial cues and regions of homogeneity. At the same time, motion information obtained after global motion compensation provides important temporal cues for segmentation. In

Step 1: Initialize the weights of all weight vectors to 1 and set all the F_2 layer nodes to uncommitted.

Step 2: Apply complement coding to the M-dimensional input vector. The resultant complement coded vector $\mathbf{I}$ is of $2M$-dimensions.

$$\mathbf{I} = (a, a^C) = (a_1, \ldots, a_M, a_1^C, \ldots, a_M^C)$$

where $a_i^C = 1 - a_i$, for $i \in [1, M]$.

Step 3: Compute the choice function value for every node of the F_2 layer. For the complement coded vector $\mathbf{I}$ and node j of F_2 layer, the choice function T_j is defined as

$$T_j(\mathbf{I}) = \frac{|\mathbf{I} \wedge \mathbf{W}_j|}{\alpha + |\mathbf{W}_j|}$$

where W_j is the weight vector of node j, and
α is the choice parameter.

The fuzzy AND operator $\wedge$ is defined by $(\mathbf{P} \wedge \mathbf{Q})_i \equiv \min(p_i, q_i)$, and the norm $|\ |$ is defined as $|\mathbf{P}| \equiv \sum_{i=1}^{M} |p_i|$.

Step 4: Find the node J of the F_2 layer which gives the largest choice function value.

$$T_J = \max\{T_j : j = 1 \ldots N\}$$

The output vector $\mathbf{Y}$ of the layer F_2 is thus given by $y_J = 1$ and $y_j = 0$ for $j \neq J$.

Step 5: Determine if resonance occurs by checking if the chosen node J meets the vigilance threshold.

$$\frac{|\mathbf{I} \wedge \mathbf{W}_J|}{|\mathbf{I}|} \geq \rho$$

where ρ is the vigilance parameter.

If resonance occurs, the weight vector $\mathbf{W}_J$ is updated and the new $\mathbf{W}_J$ is given by:

$$\mathbf{W}_J^{new} = \beta(\mathbf{I} \wedge \mathbf{W}_J^{old}) + (1 - \beta)\mathbf{W}_J^{old}$$

Otherwise, the value of the choice function T_J is set to 0. Repeat Steps 4 and 5 until a chosen node meets the vigilance threshold.

For every input pattern in the training set, perform Steps 2 to 5.

Repeat the whole process until the weights in all weight vectors remain unchanged.

Figure 4.
Training algorithm for Fuzzy ART

our experiments, we track up to two objects of interest in each video shot. The segmented objects are then verified by the human operator. This represents the only human intervention in the online categorization process, and is necessary because subsequent processing relies heavily on successful VO segmentation at this initial stage. Vital information about the objects such as color, texture, etc. is stored for subsequent event categorization. After segmentation, the objects are tracked to provide trajectory information for subsequent event categorization. If occlusion occurs at any time during VO tracking, a linear predictor is used to complete the trajectory of the object in question.

The above pre-ANN steps, which are summarized in Figure 5, are employed to process each video shot for both offline ANN training and online categorization. In fact, the goal of the above steps is to characterize each video shot using a feature vector. For offline ANN training, the vectors obtained from the video shots are used to train the ANNs. This results in clusters of similar events. To complete the offline training process, a simple cluster-to-category mapping is performed to label the clusters. During

```
Segment video footage into video shots based on a color histogram analysis

For each video shot do
 Repeat
  Perform global motion compensation, extract real motion information and low-level features
 Until Human verification of VO segmentation OK

 Begin VO tracking
 Repeat
  If occlusion detected
     Activate linear prediction
  Else
     Continue tracking
  Update trajectory information
 Until end of shot

 Collect all relevant information (motion, low-level features, trajectories, etc.) to form feature vector

 If video footage = training exemplar
  Use feature vector for offline ANN training
 Else
  Use feature vector for online categorization
```

Figure 5.
Pre-ANN processing algorithm

online categorization, the incoming video shots are also subjected to the pre-ANN steps to obtain a feature vector for each video shot. Categorization then matches the best category for the video shot based on the feature vector.

Scene change detection

Scene change detection divides a video footage into distinct scenes (video shots) as shown in Figure 6. There are two reasons for treating each video shot as a basic video unit. First, from an implementation point of view, objects may totally change in a scene change, so in most cases meaningful tracking is not achievable across shots. From the user's point of view, each video shot describes a semantically meaningful event. We employ a simple color histogram based analysis to determine when scene change occurs. In addition, a new boundary marker is also inserted if the global motion changes significantly. This ensures that excessive global motion also triggers a new categorization process, as this is likely to indicate semantically significant changes.

Global motion compensation

In general, video scenes contain foreground objects that move in relation to a relatively stationary background. However, the video capture process also introduces global motion due to camera movement, etc. In order to track semantically meaningful objects, it is necessary to isolate individual objects' movement from the background with global

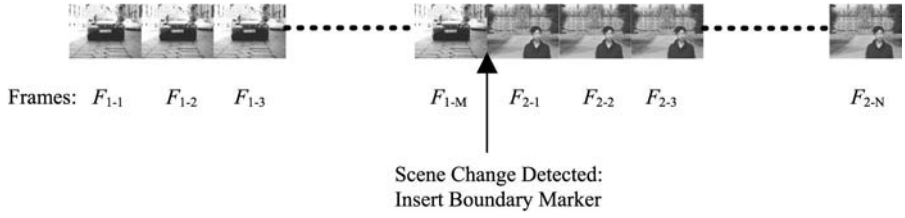


Figure 6.
Scene change detection and insertion of video shot boundary markers

motion compensation. Suppose global motion has transformed a point in frame F_i from coordinates (x, y) to (x', y'), we employ a six-parameter affine model to estimate the global motion (Biering, 1988). Let $\hat{v}_x(x, y)$ and $\hat{v}_y(x, y)$ be the optical flow vectors at (x, y), then,

$$\hat{v}_x(x, y) = x' - x = a_1 x + a_2 y + a_3$$

$$\hat{v}_y(x, y) = y' - y = a_4 x + a_5 y + a_6$$

Based on this affine model, we estimate the six parameters $a_1, a_2, \ldots, a_6$ using the least median of squares method described in (Rousseeuw and Leroy, 1987).

In order to enhance the computational efficiency, we do not perform global motion calculations on the whole frames. Instead, we perform the calculations on four 32×32 patches arranged in a cross configuration as shown in Figure 7.

Feature extraction and spatio-temporal VO segmentation

Important features that we extract from each video shot include both inter-frame motion and intra-frame low-level image properties. Motion information is used for both temporal segmentation and subsequent VO tracking to obtain trajectory information of objects of interest. On the other hand, since our spatial VO segmentation relies on morphological operations, low-level image properties are not used in the segmentation process, but are extracted to characterize each shot. The information thus obtained is used to partially form the feature vector for each shot.

For VO segmentation, we adopt a spatio-temporal approach as follows. First, spatial segmentation is applied to individual frames to divide the entire 2D frame into homogeneous regions based on low-level properties such as intensities. Unnecessary details such as texture information are removed using morphological filters: the morphological opening operator $\gamma(F_i)$ is used to remove unwanted highlight details and the morphological closing operator $\phi(F_i)$, is used to remove unwanted shadow details.

$$\gamma(F_i(x, y)) = \delta(\varepsilon(F_i(x, y)))$$

$$\varphi(F_i(x, y)) = \varepsilon(\delta(F_i(x, y)))$$

Morphological opening and closing are in turn defined by morphological dilation and erosion: the $\gamma(F_i)$ operation is essentially a morphological erosion $\varepsilon(F_i)$ followed by a morphological dilation $\delta(F_i)$, whereas $\phi(F_i)$ is the reverse operation. In all cases, a flat structuring element is used for the morphological operations. $\delta(F_i)$ and $\varepsilon(F_i)$ are in turn defined as follows for all (p,q) in the ROI.

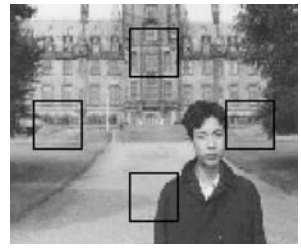


Figure 7.
Four patch locations for global motion estimation

$$\varepsilon(F_i(x,y)) = \min(F_i(x+p, y+q))$$

$$\delta(F_i(x,y)) = \max(F_i(x-p, y-q))$$

A morphological gradient operator is then used to obtain region boundary information. The morphological gradient is obtained simply by subtracting $\varepsilon(F_i\ (x,y))$ from $\delta(F_i\ (x,y))$. Morphological gradients obtained using symmetrical structuring elements are preferable to derivative gradients as they depend less on edge directionality than the latter. Next, the local gradient minima in each F_i are used as seeds for region growing. In this research, the watershed algorithm (Najman and Schmitt, 1994) is adopted for region growing. However, region-growing methods tend to produce over-segmented frames due to gradient noise. Typically, this results in isolated regions that should be connected to form a semantically meaningful object. Thus, it is necessary to apply region merging to the over-segmented frame. This is performed by a combination of similarity measure relaxation and morphological erosion.

Temporal segmentation relies on the isolation of background motion using global motion estimation described above. Then, to obtain relevant object motion information, we consider each pair of frames in turn and attempt to determine an inter-frame difference to isolate significant object movement. Suppose F'_{i-1} represents the previous frame that has received global motion compensation, then we can calculate:

$$\Delta = \left| F_i - F'_{i-1} \right|$$

such that for those regions where Δ is less than a threshold T_L, we consider the residue motion to be insignificant typically due to such physical phenomena as swaying plants or moving clouds. On the other hand, if Δ is greater than another threshold T_H, we consider the residue motion to be important. Based on this Δ, we adopt the histogram projection technique described in Haering *et al.* (2000) to obtain the relevant motion information, including the object center position (centroid), as well as its height and width. An advantage of this approach is that it allows iterative processing for multiple objects, which is important for us as we allow a maximum of two objects during tracking.

Low-level images properties are those originally used in classical image processing, such as color, intensity and texture information. For example, Brunelli and Mich (2000) have proposed the use of a dissimilarity measure on a multidimensional histogram for discrimination between different images. From our point of view, two important findings reported are:

(1) color and intensity are much better visual descriptors than edge information, particularly for videos; and

(2) the co-occurrence of color (or intensity), defined as a 2D histogram obtained by partitioning the image space into pairs of pixels by means of a binary spatial relation, is a more effective visual descriptor than color (or intensity) alone.

However, the improvement in using co-occurrence is much more pronounced with still images than videos. In fact, the improvement with videos is only marginal. Thus, in the interest of simplicity and computational speed, we only record color and intensity

information. In particular, we record the normalized red R, green G and blue B components, as well as the Intensity I. On the other hand, gray-level co-occurrence is effective for texture features. In fact, the use of a combination of models and measures has been reported to provide far better discrimination than using them singularly (Jain and Zongker, 1997). In this research, we use measures from three different models: local statistics, gray level co-occurrence matrix (GLCM), and lognormal random field model. For GLCM, we use six different measures as described in Haralick *et al.* (1973) and Conners and Harlow (1980). Suppose $P(i, j, d, \theta)$ represents the GLCM of pixels where i and j are the matrix indices, d is the distance between the pixels, and θ is the orientation. Then we can obtain $p(i, j, d, \theta)$, which is $P(i, j, d, \theta)$ normalized by $R(d, \theta)$ such that the entries of the normalized matrix sum to unity (Haering *et al.*, 2000; Haralick *et al.*, 1973; Conners and Harlow, 1980).

$$p(i,j,d,\theta) = P(i,j,d,\theta)/R(d,\theta)$$

Then, for example, the angular second moment (or energy $E(d, \theta)$) that gives more weight to textures having a sparse co-occurrence matrix is given by:

$$E(d,\theta) = \sum_j \sum_i (p(i,j,d,\theta))^2$$

The other GLCM measures are contrast, difference angular second moment, entropy, inertia, and correlation as defined in Haralick *et al.* (1973) and Conners and Harlow (1980). By varying the orientation θ in four regular intervals between 0 and π, we obtain a total of twenty-four GLCM features. The texture features used are summarized in Table I. These texture features together with color information contribute 33 entries towards the dimensionality of the feature vector representing each video shot.

VO tracking and occlusion handling

Following the interactive verification of VO segmentation, the segmented objects are tracked to obtain trajectory information. Our current implementation assumes that the objects of interest are either rigid (e.g. vehicles) or articulate and slightly deformable (e.g. persons). Thus, amorphous objects such as clouds could not be handled. The justification is that modeling only rigid and articulate objects would already provide

Model	Measure
Local statistics	Mean
Local statistics	Power-to-mean ratio
Local statistics	Skewness
GLCM	Angular second moment
GLCM	Contrast
GLCM	Difference angular second moment
GLCM	Entropy
GLCM	Inertia
GLCM	Correlation
Lognormal random field	Variance
Lognormal random field	Mean (logarithmic)

Table I.
Texture features from three models

enormous potential for practical applications (e.g. traffic monitoring, security surveillance).

For tracking and occlusion handling of objects, we focus on the inter-frame incremental displacement of both the centroid and head top point of each object. The head top point, which is defined as the highest point of a segmented object (Shao *et al.*, 2000), is used with the object centroid to account for a limited amount of deformation. To obtain the trajectory of each object, the start locations of the object centroid (x_{c0}, y_{c0}) and head top point (x_{h0}, y_{h0}) are first recorded. Up to ten additional locations are stored for each of the centroid and head top point. In particular, a new location is stored every time the actual pixel location deviates from the predicted location by an angle α (15° in our experiments) in the 2D plane. Figure 8 shows the tracking process, which shows a slight deformation as the centroid and head top point have slightly different trajectories.

To handle occlusion, we use a combination of linear prediction and nearest Euclidean distance matching to fill in the missing trajectory information as shown in Figure 9, where an object is taken as a whole for illustration purposes (actually the

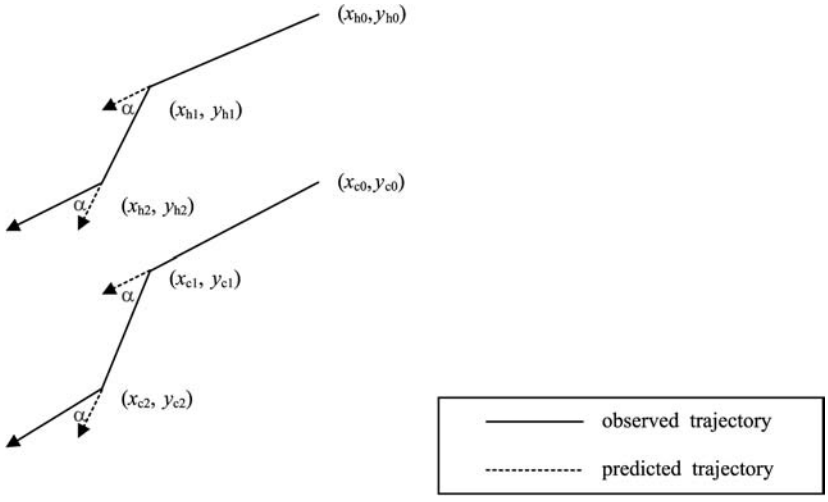


Figure 8.
Object tracking

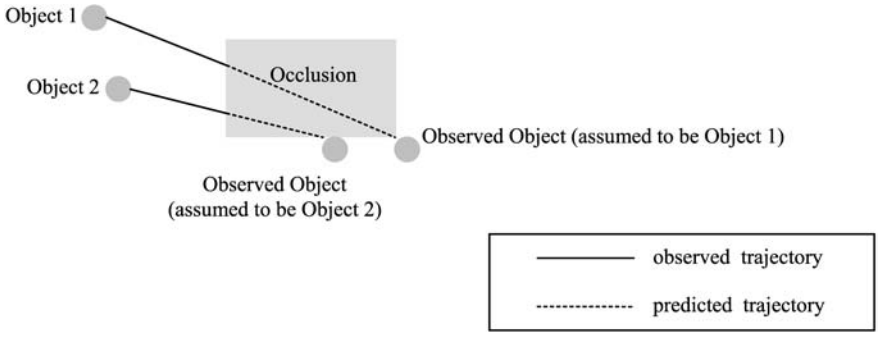


Figure 9.
Occlusion handling

centroid and head top point are considered separately). When a tracked object is obscured by another, its most recent position is recorded. As soon as it reemerges, a simple linear prediction algorithm is used to derive its missing trajectory. If an occlusion involves multiple objects, then matching is performed to compare the Euclidean distance between the predicted locations and the actual observed locations to distinguish between the objects. In addition, the distance between the centroid and head top point of each object is also taken into account when distinguishing between objects following an occlusion. When taken together, these strategies are reasonably robust and efficient in terms of computational effort.

Video shot feature vector generation

Having obtained all the necessary low-level descriptions and trajectory information, the next step is to produce a feature vector that characterizes the video shot in question. If the video shot comes from a training set intended for ANN training, then the vector is fed to the ANN (SOM or Fuzzy ART as the case may be) to generate the appropriate ANN model using the training algorithms presented above. Otherwise, the vector is used for cluster matching during online categorization.

From the above discussion, the basic feature vector for each video shot has 77 elements. The basic vector must first be suitably transformed using a weight multiplication step, followed by normalization. Weight multiplication is used to multiply each of the elements of the basic vector by a weight to indicate the relative importance of the elements. All the weights are then adjusted for each training session of the ANN. Normalization is then performed on the weighted vector according to a common Euclidean length. The resultant feature vector is then fed to the ANN for training/categorization.

ANN model generation

The two ANNs, SOM and Fuzzy ART, are trained to form ANN models for subsequent online categorization. Since the feature vector has a dimension of 77, the SOM network is constructed with 77 inputs. After several training experiments, we have decided to construct the network with a square array of output neurons with a dimension of 8×8 for a good balance between accuracy and training complexity. In particular, we found that dimensions smaller than 8×8 gave inadequate discrimination performance with many clusters containing mixtures of different events. On the other hand dimensions larger than 8×8 did not show any significant improvement. Also, to fulfil Kohonen's requirement (Kohonen, 1995), the network is trained for 32,000 iterations or 500 times larger than the total number of output neurons. The initial learning rate is set to 0.6 and is linearly decreased in each iteration towards the final learning rate of 0.01. The value of the initial neighborhood size is 5 and decreased once every 5,000 iterations.

The Fuzzy ART network is constructed with 77 nodes at its F_0 layer. Due to complement coding, there are twice as many nodes at the F_1 layer. To compare Fuzzy ART with the SOM, we give the Fuzzy ART network a total of 70 nodes at F_2 layer. To minimize the training time, we have chosen a small value of 0.1 for α (Carpenter *et al.*, 1991). ρ, which affects the number of clusters generated by the network, is set to 0.68 after several training experiments to produce a total of clusters close to 64. β is set to 1 for uncommitted F_2 node and 0.5 for committed F_2 node, since we use fast-commit slow-recode as our updating scheme

Cluster-to-category mapping

At the conclusion of each ANN training session, clusters of video shots are formed with the property that the similarity of intra-cluster members is maximized and the similarity between different clusters is minimized. The purpose of cluster-to-category mapping is to map each cluster to one of the predefined meaningful events. In this research, we have defined 12 events under two major topics: vehicle motion, and human motion. Each major topic is then subdivided into specific events as shown in Table II. In addition, multiple events are allowed concurrently. For example, if the second and third ranking events have a matching score within 5 percent of the top ranking event, then both of them are returned as the categorization result. Thus, a vehicle may be found to be moving away from the viewer and from left to right on the screen, or a person may be jumping up and down while moving towards the viewer. Currently, up to two different objects are accounted for.

Although it might appear advantageous to adopt a hierarchical categorization of events (Vailaya *et al.*, 2001) according to the two major topics, this approach would lead to error propagation. If the probability of a successful categorization of the three major topics is x, then the categorization error $\varepsilon = 1 - x$ will be propagated to the next lower level of event categorization. The accumulating effect of such errors means that the overall categorization accuracy will be reduced. We therefore use a flat data structure such that each event is equally distinct from any other. In further research, we would like to investigate linking related events with conditional probabilities.

Experimental results

To gauge the effectiveness of our IVCE, we have conducted experiments using both common test sequences such as "foreman" and "mobile calendar" and other motion video footage we have captured. All test sequences are color sequences in QCIF format (176×144) at 30 frames per second. In addition, we are interested in comparing the performance of the SOM and the Fuzzy ART ANNs, as well as analyzing successful and unsuccessful categorization.

Major topic	Event number	Event
Vehicle motion	1	Moving towards viewer
	2	Moving away from viewer
	3	Moving left to right
	4	Moving right to left
	5	Stationary
Human motion	6	Head and shoulder view of person speaking (includes activities such as news reading and video conferencing applications)
	7	Moving towards viewer
	8	Moving away from viewer
	9	Moving left to right
	10	Moving right to left
	11	Jumping up and down
	12	Stationary

Table II.
Predefined events

VO segmentation and motion tracking

First, we evaluated the effectiveness of our pre-ANN steps, in particular VO segmentation and motion tracking, as this has a direct impact on the subsequent ANN training and categorization. In the first test sequence "foreman", there is a person talking in front of a building. The person, which is the main object of interest, has relatively little movement compared to our second sequence "mobile calendar". In the first sequence, the camera also moves, so global motion estimation and compensation is necessary. In the second sequence, we are interested in the motion of a ball against a relatively cluttered background. The ball rolls as it moves across the screen. Figure 10(b) and (d) show the successful VO segmentation of the main object of interest corresponding to Figure 10(a) and (c), respectively.

The sequence in Figure 11 highlights the difficulties of tracking an object that changes direction frequently, and occasionally goes out of frame. It shows only the centroid's trajectory. In this case, as with most cases with single rigid objects, the head top point's trajectory provides little additional information as both trajectories are almost the same.

Comparison of the two ANNs

Using a training set of 100 video shots covering all the predefined events in roughly equal distribution, we evaluated the training performance of the two ANNs. It should be emphasized that we did not make use of any textual clues (such as captions) for event inference. Where available, the use of textual information would significantly improve the accuracy of event categorization (Babaguchi *et al.*, 2002). Instead, we rely on the processing of visual data alone. Since training is performed offline, we are primarily interested in the training accuracy attained by each ANN. The results are summarized in the top portion of Table III, which shows that SOM significantly outperformed Fuzzy ART. Since training only has to be performed once offline, it is better to use SOM for higher training accuracy than Fuzzy ART unless retraining is frequently required, for example, when the actual categories are changed frequently.

To measure the online categorization accuracy of the two ANNs, we used another set of 30 video shots distinct from the training set. The main advantage of using ANNs

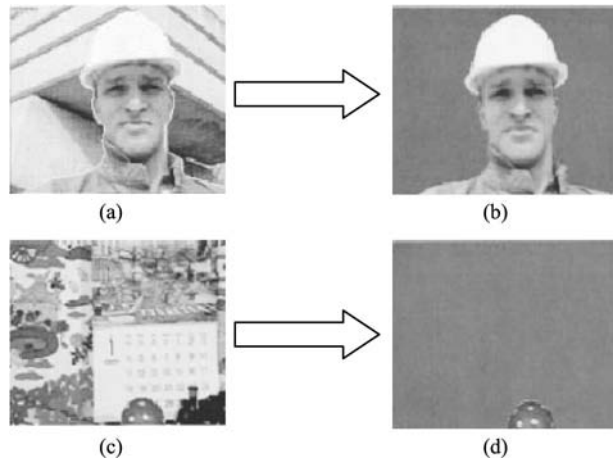
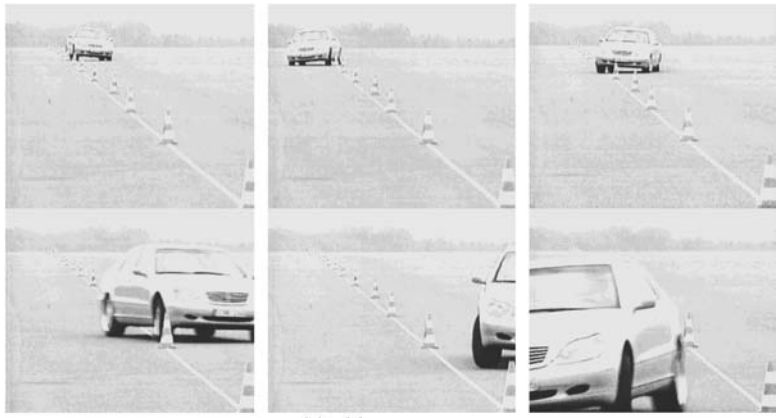
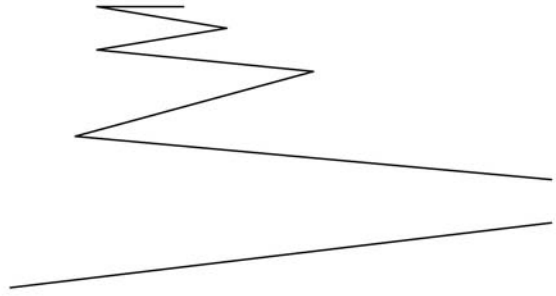


Figure 10.
VO segmentation



(a) video sequence



(b) object centroid's trajectory

Figure 11.
Illustration of object tracking

	ANN	Correctly categorized (percent)	Incorrectly categorized (percent)	Undetermined (percent)
Offline training	SOM	92	3	5
	Fuzzy ART	85	4	11
Online categorization	SOM	90	3	7
	Fuzzy ART	83.3	6.7	10

Table III.
Performance comparison between SOM and Fuzzy ART

in this context is the learning capability that they provide in generalizing information beyond what has been learned during the training process. The bottom portion of Table III summarizes the categorization accuracy of the two ANNs. SOM slightly outperformed Fuzzy ART and in both cases the response was near instantaneous.

Another important observation is that we achieved better results, both during offline ANN training and online categorization, with the "vehicle" shots and shots with little foreground object motion (event numbers 1-6 and 12). This suggests that our current implementation requires improvements for handling articulated objects. We

would need more than the tracking of two points (centroid and head top point) to accurately model the motion of such objects.

Online categorization

Figure 12 shows a number of representative test sequences used to gauge the effectiveness of the IVCE. The first sequence represents a relatively easy shot for analysis, which shows a single object of interest moving against a stationary background without any camera motion during the shot. The IVCE successfully inferred the shot as a vehicle moving away from the viewer. The second sequence is a little more difficult as there are two objects of interest that move fairly close to each other. In this case, the head top points' trajectories provide important information for tracking the two objects. In addition, there is also zooming and panning evident during the shot, so global motion compensation was necessary. Again, the intelligent categorization engine successfully inferred the shot as two vehicles moving from right to left.

The third sequence in Figure 12 shows a vehicle moving along a tree-lined road. The tree trunks occasionally obscure the view of the vehicle. The sequence therefore provides an opportunity to test the occlusion handling capability of our IVCE. In this event, it inferred the shot as a vehicle moving from right to left towards the viewer. From the user's viewpoint, this interpretation may or may not be acceptable. However, this was what the system could do given the restrictive set of predefined (allowable) categories. The forth sequence poses considerable challenge initially because the object of interest (vehicle) blends in with its surrounding when it is distant. Successful tracking began soon after the vehicle turned the bend and it was inferred as a vehicle moving towards the viewer. Again, some users might find this interpretation less than

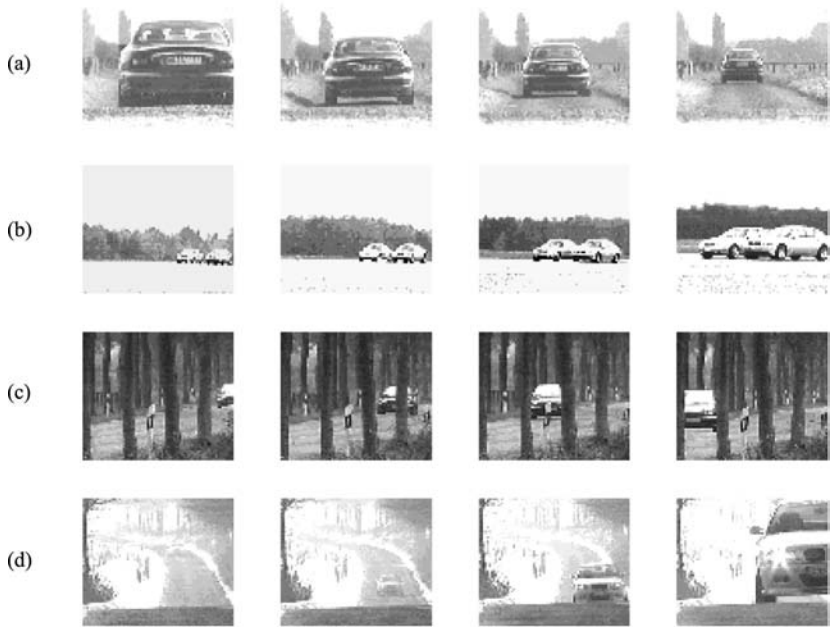


Figure 12.
Selection of the test sequences used in the experiments

adequate. Thus, sequences 3 and 4 have highlighted some of the shortcomings of the system. While the IVCE provided some useful information about sequences 3 and 4, more research is needed to extract more information from the video shots.

Conclusions

The ability of its machine to extract semantic understanding from video footage has tremendous potential for interactive broadcasting and related applications. However, this remains a difficult task as most traditional image processing and pattern recognition techniques are inadequate. We have presented a survey of related technologies, notably VO segmentation and motion tracking, that have been developed towards achieving this goal. This survey of current technologies has led us to the development of an IVCE that attempts to categorize video shots into a predefined set of meaningful events. In particular, we use the learning capability of ANNs, which can be trained to exhibit human-like intelligence, to cluster similar events such that intra-cluster similarity is maximized and inter-cluster similarity is minimized. A simple cluster-to-category mapping is then needed to complete the event inference process. While ANNs have been applied widely to classification textual information, the challenge we face is to characterize video shots using a finite set of features. We have also analyzed and compared the performance of two popular ANNs, Kohonen's SOM and the Fuzzy ART networks. Although it is much more efficient to train the Fuzzy ART than the SOM, the latter provides significantly better categorization results. Since our implementation of IVCE decouples the training phase (performed off line in advance) and the categorization phase, SOM is a better choice than Fuzzy ART for optimal performance.

Our results have shown that IVCE can perform well given the rather restrictive set of predefined events and the fact that only a maximum of two objects of interest are tracked. Further research will therefore focus on improving the IVCE by making it solve more generic problems with a larger set of predefined events. For example, finer granularity can be achieved by including more features into the algorithm, such as speed estimation and distance from camera that would, for instance, distinguish between walking and running. Also, linking related events with conditional probabilities might further improve the categorization results. In addition, tracking more object features points would likely improve the analysis of articulate objects. Another research direction is to investigate the application of ANN to VO segmentation by training an ANN to recognize semantically meaningful objects.

References

Babaguchi, N., Kawai, Y. and Kitahashi, T. (2002), "Event-based indexing of broadcast sports video by intermodal collaboration", *IEEE Trans. Multimedia*, Vol. 4 No. 1, pp. 68-75.

Balakrishnan, P.V. (1994), "A study of the classification capabilities of neural networks using unsupervised learning: a comparison with K-means clustering", *Psychometrika*, Vol. 59 No. 4, pp. 509-25.

Biering, M. (1988), "Displacement by hierarchical block matching", *Proc. SPIE*, pp. 942-51.

Brunelli, R. and Mich, O. (2000), "Image retrieval by examples", *IEEE Trans. Multimedia*, Vol. 2 No. 3, pp. 164-71.

Carpenter, G.A., Grossberg, S. and Rosen, D.B. (1991), "Fuzzy ART: fast stable learning and categorization of analog patterns by an adaptive resonance system", *Neural Networks*, Vol. 4, pp. 759-71.

Conners, R.W. and Harlow, C.A. (1980), "A theoretical comparison of texture algorithms", *IEEE Trans. Pattern Analysis Machine Intell.*, Vol. 2, pp. 204-22.

Dalton, J. and Deshmane, A. (1991), "Artificial neural networks", *IEEE Potentials*, Vol. 10 No. 2, pp. 33-6.

Deng, Y. and Manjunath, B.S. (1998), "NeTra-V: Toward an object-based video representation", *IEEE Trans. Circ. Syst. Video Tech.*, Vol. 8 No. 5, pp. 616-27.

Flexer, A. (2001), "On the use of self-organizing maps for clustering and visualization", *Intelligent Data Analysis*, Vol. 5 No. 5, pp. 373-84.

Gu, C. and Lee, M.-C. (1998), "Semiautomatic segmentation and tracking of semantic video objects", *Trans. Circ. Syst. Video Tech.*, Vol. 8 No. 5, pp. 572-84.

Haering, N., Qian, R.J. and Sezan, M.I. (2000), "A semantic event-detection approach and its application to detecting hunts in wildlife video", *IEEE Trans. Circ. Syst. Video Tech.*, Vol. 10 No. 6, pp. 857-68.

Haralick, R.M., Shanmugam, K. and Dinstein, I. (1973), "Textural features for image classification", *IEEE Trans. Syst. Man, Cybern.*, Vol. 3, pp. 610-21.

Jain, A. and Zongker, D. (1997), "Feature selection: evaluation, application, and small sample performance", *IEEE Trans. Pattern Analysis Machine Intell.*, Vol. 19 No. 2, pp. 153-8.

Koehler, G.J. and Erenguc, S.S. (1990), "Minimizing misclassifications in linear discriminant analysis", *Decision Sciences*, Vol. 21, pp. 63-85.

Kohonen, T. (1995), *Self-organizing Maps*, Springer-Verlag, Berlin.

Kung, S.-Y. and Hwang, J.-H. (1998), "Neural networks for multimedia processing", *Proc. IEEE*, Vol. 86 No. 6, pp. 1244-72.

Meier, T. and Ngan, K.N. (1999), "Video segmentation for content-based coding", *IEEE Trans. Circ. Syst. Video Tech.*, Vol. 9 No. 8, pp. 1190-203.

Najman, L. and Schmitt, M. (1994), "Geodesic saliency of watershed contours and hierarchical segmentation", *IEEE Trans. Pattern Analysis Machine Intell.*, Vol. 18 No. 12, pp. 1163-210.

Rousseeuw, P.J. and Leroy, A.M. (1987), *Robust Regression and Outliner Detection*, Wiley, New York, NY.

Salembia, P., Torres, L., Meyer, F. and Gu, C. (1995), "Region-based video coding using mathematical morphology", *Proc. IEEE*, Vol. 83, pp. 843-57.

Shao, H., Leung, M.K.H., Gao, Y. and Li, L. (2000), "Head-top detection in elevator monitoring", *Journal of Three-Dimensional Images*, Vol. 14 No. 4, pp. 117-22.

Vailaya, A., Figueiredo, M.A.T., Jain, A.K. and Zhang, H.-J. (2001), "Image classification for content-based indexing", *IEEE Trans. Image Proc.*, Vol. 10 No. 1, pp. 117-30.

Yang, Y. (1999), "An evaluation of statistical approaches to text categorization", *Journal of Information Retrieval*, Vol. 1 Nos 1/2, pp. 69-90.

Zheng, W., Shishikui, Y., Kanatsugu, Y., Tanaka, Y. and Yuyama, I. (2001), "A high-precision camera operation parameter measurement system and its application to image motion infering", *IEEE Trans. Broadcasting*, Vol. 47 No. 1, pp. 46-55.

Further reading

Hartigan, J.A. (1975), *Clustering Algorithms*, Wiley, New York, NY.