

# Action Scene Detection with Support Vector Machines

Liang-Hua Chen<sup>1</sup>, Chih-Wen Su<sup>2</sup>, Chi-Feng Weng<sup>1</sup> and Hong-Yuan Mark Liao<sup>2</sup>

<sup>1</sup>Department of Computer Science and Information Engineering, Fu Jen University, Taipei, Taiwan

<sup>2</sup>Institute of Information Science, Academia Sinica, Taipei, Taiwan

**Abstract**—To entice the target audience into paying to see the full movie, the production of movie trailers is an integral part of movie industry. Action scene is the main component of a movie trailer. In this paper, we propose an automatic action scene detection algorithm based on the analysis of high-level video structure. The input video is first decomposed into a number of basic components called shots. Then, shots are grouped into semantic-related scenes by taking into account the visual characteristics and temporal dynamics of video. Based on the filmmaking characteristics of action scene, some features of the scene are extracted to feed into the support vector machine for classification. Compared with related works which integrate visual and audio information, our visual-based approach is computationally simple yet effective.

**Index Terms**—Video content analysis, support vector machine, scene classification

## I. INTRODUCTION

As recent advances in computational power and storage capacity, the potential for large digital video libraries is growing rapidly. Owing to the sheer volume of data and unstructured format, efficient access to video is not an easy task. It is imperative to give users necessary summarizing and skimming tools so that they can quickly find the content they interest. The *abstract* of a video sequence is a short synopsis which preserves the essence of the original video content. If effective abstraction tools were available, the users could evaluate ten hours of video in tens of minutes and determine quickly which portions to examine in further detail. In movie industry, it is common to produce a trailer (short abstract) of a movie to get people interested. However, manual production of trailer is time-consuming and costly. As the production of movie trailer is a creative process, it is difficult to automatically generate a trailer that is very similar to the manually produced trailer, unless we fully understand the grammar behind what makes both a movie and a trailer. According to the guideline film grammar provides, there are three basic types of scenes in a movie: dialogs without action, dialogs with action and actions without dialog [1]. In particular, action clips are often more interesting and carry more content in a short time than calm clips. Action scenes such as gunfire, explosions and car chases, attract attention and make viewers curious. Therefore, automatic extraction of action scenes from a movie is an important issue for the promotion of a movie. On the other hand,

the action scene detection algorithm can be extended to implement automatic blocking of inappropriate violence in movies watched by children.

Relatively little research has addressed the problem of action scene detection in video. Chen et al. proposed a rule-based model to extract simple action scenes [2]. Through analyzing video editing rules and observing temporal appearance patterns of shots in action scenes of movies, they deduced a set of rules to recognize action scenes. However, the type of action events that can be detected is restricted to one-on-one fighting only. Nam et al. detected violent events in a movie by searching for visual cues such as flames or blood pixels, or audio cues such as explosions and screaming [3]. Their system needs manual intervention for creation of audio samples to detect sounds associated with violence. Lehane et al. examined the filmmaking conventions that are inherent in the action scene [4]. Based on these conventions, a finite state machine was used to integrate several visual features and detect the action events. In the work of Geng et al. [5], visual and audio features are combined to compute the *film rhythm* of a movie. Action scenes are detected at the peaks of the film rhythm curve. Then the probability neural networks is employed to classify the detected scenes into fight, chase and uncertain scenes. However, satisfying results may not be obtained from video with a sound track containing more than just speech (such as music and environmental sound), or video clip which is silent.

While previous approaches use domain knowledge and film production rules to extract action scenes, they do not fully exploit the information contained in video structure. In this paper, we propose a novel technique based on the construction of high-level video structure. To be simple yet effective, only visual features of video are used. Support vector machine (SVM) is also employed to classify the extracted scenes into action scenes and non-action scenes. As SVM is known to generalize well even in high dimensional spaces under small training sample conditions, it has been successfully applied to face authentication/recognition[6], [7], object recognition[8], character recognition[9], speech recognition[10], image retrieval[11] and so on. To our knowledge, our work is the first one that applies SVM to achieve the task of action scene detection.

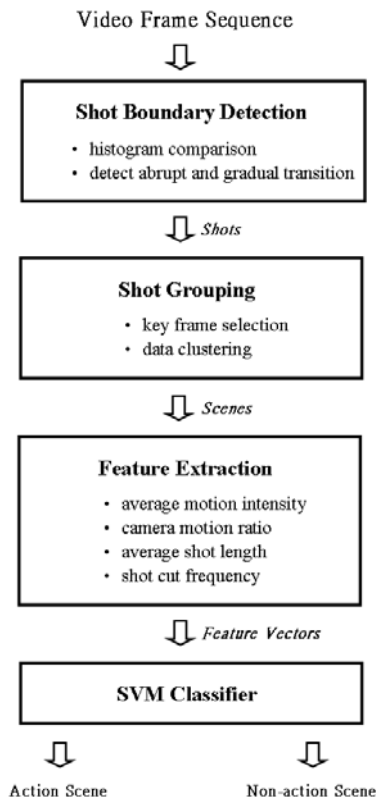


Figure 1. Flowchart of the proposed approach.

II. MOTIVATION

A video is physically formed by shots and semantically described by scenes. A shot is a sequence of frames that was continuously captured by the same camera. A scene is basically a story unit and consists of a small number of interrelated shots that are consecutive or not. Shots in video are analogous to words in language as they convey little semantic information in isolation. On the other hand, scenes allow event to be fully presented and reflect the dramatic and narrative structure of a video. It is noted that previous approaches on action scene detection only work on shot level. To fully exploit the information contained in video structure, the proposed approach is made up of three main steps which are described in the following sections.

- (1) Segmentation of video into shots.
- (2) Grouping of shots into scene.
- (3) Scene classification using SVM.

Figure 1 gives an overview of the proposed approach.

III. SEGMENTATION OF VIDEO INTO SHOTS

Shot is the fundamental unit of a video. Shots can be joined together by either an abrupt transition (cut) or a gradual transition. In abrupt transition, two shots are simply concatenated, while in the gradual transition, additional frames may be introduced using editing operations such as fade in, fade out, dissolve and wipe. A good video segmentation technique should be able to detect shots

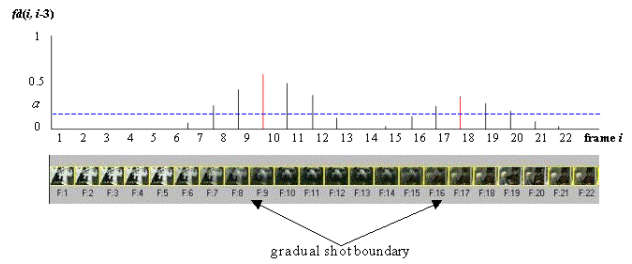


Figure 2. The detection of gradual transition.

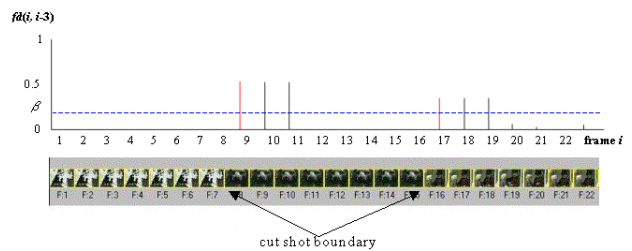


Figure 3. The detection of abrupt transition.

with both types of transition. The existing shot boundary detection techniques can be classified into five categories: pixel based, statistics based, transform based, feature based, and histogram based. Several researchers claim that the histogram based approach achieves good trade off between accuracy and speed[12]. The histogram based algorithm is implemented by comparing the histogram of two consecutive frames. If the frame difference is greater than a threshold, an abrupt transition is detected. A problem arises when the transition is gradual; the shot does not change abruptly but over a period of few frames. The difference between two frames is not so large to declare it a shot boundary. We have proposed an improved algorithm to detect both types of shot boundary[13]. Here, we briefly describe our shot boundary detection algorithm. For more details, please see the work of Chen et al.[13]. The basic idea is that the frames before and after a gradual transition are usually markedly different. Instead of the difference between two consecutive frames, we compute the difference between two frames which are  $k$  frames apart. Let  $H(f_m, i)$  denote the number of pixels of gray value  $i$  in the  $m$ -th frame  $f_m$ , the histogram difference between the frame  $f_m$  and its  $k$ -th predecessor is defined as:

$$fd(m, m - k) = \frac{1}{2N} \sum_i |H(f_m, i) - H(f_{m-k}, i)|, \quad (1)$$

where  $N$  is the number of pixel in a frame. A similar formulation is defined for color image. Given a sequence of frames  $f_1, f_2, \dots, f_n$  and a fixed number  $k$ , the frame difference sequence  $fd(k + 1, 1), fd(k + 2, 2), \dots, fd(n, n - k)$  can be determined. Frame  $f_m$  is detected as a gradual transition boundary (see Figure 2), if  $fd(m, m - k) > \alpha$  ( $\alpha = 0.2$ , in our setting) and  $fd(m, m - k)$

is a local maximum of the frame difference sequence. Frame  $f_m$  is detected as an abrupt transition boundary (see Figure 3), if  $fd(m, m-k)$  is the first element of the subsequence  $fd(m, m-k), \dots, fd(m+s, m+s-k)$  ( $s > 0$ ) that are greater than  $\beta$  ( $=0.1$ , in our setting). If  $k$  is too large, the algorithm will miss some short shot boundary. To determine appropriate  $k$  value, we consider the minimal length of a video shot. In filmmaking, each shot lasts at least for 1/3 second to impress audience. Therefore, for 30 frames/sec video,  $k$  should not be larger than 10.

#### IV. GROUPING OF SHOTS INTO SCENES

After video is segmented into shots, key frames can be extracted from each shot. Key frame is the frame which can represent the salient content of the shot. For any shot after an abrupt transition, a natural and easy way is to choose the first frame of that shot as the key frame. However, for a shot after a gradual transition, it is possible that the first frame is part of a dissolve effect at the shot boundary, which strongly reduces its representative quality. Therefore, for a shot after a gradual transition, we choose frame  $f_m$  as the key frame if  $fd(m, m-k)$  is the first element of the subsequence  $fd(m, m-k), \dots, fd(m+s, m+s-k)$  ( $s > 0$ ) that are smaller than a threshold. If no such  $f_m$  exists, we choose a frame  $f_m$  with the minimum  $fd(m, m-k)$  as the key frame.

Since people watch the video by its semantic scenes not the physical shots, shots can not convey meaningful semantics unless they are purposely grouped into semantic-related scenes. In our approach, shots are grouped on the basis of their visual contents and temporal localities. Two shots are *similar* (or semantic-related) if they are visually similar and temporally close. On the other hand, two shots that are far apart in time but similar in visual content should belong to two different scenes. Given two shots  $S_i$  and  $S_j$  with respective key frames  $f_{k_i}$  and  $f_{k_j}$ , a similarity measure between these two shots is defined as

$$D(S_i, S_j) = \begin{cases} fd(k_i, k_j) & \text{if } |k_i - k_j| < T \\ \infty & \text{otherwise} \end{cases}, \quad (2)$$

where  $T$  is a threshold. With this similarity measure, we apply classic data clustering technique (complete-link method [14]) to group shots that are similar together into a cluster (scene). The choice of threshold  $T$  is also critical. A too large  $T$  can render two distinct scenes to be grouped into one story unit, while a too small  $T$  can cause a scene to be broken into several story units. For application likes video abstraction, we prefer over-segmentation rather than under-segmentation. It is less detrimental to have several story units represent a scene than to have one story unit represent several scenes - these scene can not be recovered in subsequent analysis. In our experiment, we set  $T$  to be 90 seconds (or 2700 frames for 30 frames/sec video). The data clustering algorithm is briefly described below for completeness.

1) Initially, there are  $N$  clusters, one for each shot.

- 2) Stop when the dissimilarity between every two clusters is greater than a threshold  $\delta$ .
- 3) Find the most similar pair of clusters:  $R$  and  $S$ .
- 4) Merge  $R$  and  $S$  into a new cluster.
- 5) Go to step 2.

where the threshold  $\delta$  is set to be 0.3.

#### V. SCENE CLASSIFICATION USING SVM

For each scene of the video sequence, some features are extracted. Then, a binary classifier (SVM) is employed to detect the action scene. The details are described in the following two subsections.

##### A. Overview of Support Vector Machines

Unlike traditional learning techniques such as neural networks which minimize the empirical training error, the support vector machines (SVMs) are based on the structural risk minimization principle. The basic idea is closely related to regularization[15]: for a finite set of training samples, the search for the best model or approximating function has to be constrained by an appropriately small hypothesis space. If the space is too large, functions can be found which fit exactly the training data, but they will have poor generalization capabilities on new test data. Instead, the minimization of the structural risk is equivalent to minimizing the sum of the error on the training set and the complexity of the hypothesis space, expressed in terms of VC-dimension. Consequently, the solutions obtained with SVMs are more likely to generalize well on new data points. We now go through the SVMs for a two-class classification problem. For a comprehensive and rigorous account of SVMs, please see[16].

Given a training set  $S = \{(x_1, y_1), \dots, (x_n, y_n)\}$ , where feature vector  $x_i \in R^d$  and label  $y_i \in \{1, -1\}$ , the goal of SVM is to construct a hyperplane that maximizes the margin while minimizing a quantity proportional to the misclassification error. The optimal separating hyperplane  $w^* \cdot x + b^* = 0$  can be found under the following constraints:

$$\min_{w, b, \xi} \frac{1}{2} w \cdot w + C \sum_{i=1}^n \xi_i \quad (3)$$

subject to

$$y_i(w \cdot x_i + b) \geq 1 - \xi_i, \quad \xi_i \geq 0, \quad i = 1, \dots, n \quad (4)$$

where  $C$  is the penalty parameter that controls the tradeoff between the margin and the misclassification errors  $\xi = (\xi_1, \dots, \xi_n)$ . This is a quadratic programming problem which can be solved by standard technique such as Lagrange multipliers. The classification of a new data with feature vector  $x$  is determined by the evaluation of the decision function

$$f(x) = \text{sign}(w^* \cdot x + b^*). \quad (5)$$

**B. Feature Extraction**

To extract proper features for classification, we consider the following film production rules[1]:

- In action scenes, the filmmaker often uses a series of shots with high motion activity to create tense and strong atmosphere.
- Fast edits are frequently used to build a sense of kinetic action and speed.
- Two action scenes with high film rhythm may not be juxtaposed together.

Using these guideline, the director and editor control the pace of a movie to grasp the attention of the viewers. Thus, most of the action scenes consist of a consecutive sequence of short shots with high motion activity. This type of video sequences will provide a lot of rapidly changing visual information displayed on screen to excite the viewers. Based on these action scene characteristics, the following features are extracted.

(1) Average Motion Intensity:

Motion is a visual feature which is essential to capture temporal variation of video. It also reveals the correlations between frame sequences within a video scene. To characterize the degree of motion within a scene, the *average motion intensity* is computed based on the motion vectors encoded in the MPEG-1 video stream[17]. In MPEG video, each frame is partitioned into blocks of size  $16 \times 16$  pixels called macro blocks (MBs). MPEG defines motion vector as the displacement from the Target (current frame) MB to the Prediction (reference frame) MB. In MPEG format, there are three types of frames: I, P and B frames. I frames are skipped because they are intra-coded and no motion information is available. P frames have forward motion prediction and B frames have both forward and backward motion prediction. In our system, only the forward motion vectors encoded in P frames are extracted. For a given P frame, the motion intensity matrix is defined as

$$M(i, j) = \sqrt{u_{i,j}^2 + v_{i,j}^2}, \quad (6)$$

where  $(u_{i,j}, v_{i,j})$  is the motion vector associated with  $(i, j)$ th macroblock. Assuming there are  $m \times n$  macroblocks in the frame, then the average motion intensity of the frame is

$$\bar{M} = \frac{1}{mn} \sum_{i=1}^m \sum_{j=1}^n M(i, j). \quad (7)$$

Then, the average value of  $\bar{M}$  over all frames within a scene is obtained. Finally, this value is normalized to be in the interval  $[0, 1]$ .

(2) Camera Motion Ratio:

If a frame has less than 10% motion vectors are zero, then this frame has camera motion. Assuming a scene  $S$  consists of  $m$  frames, and  $k$  frames have camera motion. The camera motion ratio is defined as

$$C = \frac{k}{m}. \quad (8)$$



Figure 4. Some action scenes in “007: Casino Royale”.



Figure 5. Some action scenes in “Blood Diamond”.

(3) Average Shot Length:

Assuming a scene  $S$  consists of  $n$  shots, and their corresponding shot length is  $L_i, i = 1, \dots, n$ . The average shot length of  $S$  is defined as

$$\bar{L} = \frac{1}{n} \sum_{i=1}^n L_i. \quad (9)$$

Likewise,  $\bar{L}$  is normalized to be in the interval  $[0, 1]$ .

(4) Shot Cut Frequency:

Assuming a scene  $S$  consists of  $n$  shots, then the shot cut frequency of  $S$  is defined as

$$F = \frac{1}{n}. \quad (10)$$

Thus, a 4-dimensional feature vector for classification is constructed.

VI. EXPERIMENTAL RESULTS

Four movies in MPEG-1 format are used in our experiment: (1)“007: Casino Royale”, (2)“Blood Diamond”, (3)“Fearless” and (4)“The Departed”. The ground truth of test video, i.e., the decision whether a given scene is action scene or not, is determined by human subjects. Some action scenes of movies (1) and (2) are shown in Figure 4 and Figure 5, respectively. In the SVM training process, 4-fold cross-validation is used to prevent the overfitting problem. The radial basis function is chosen as the kernel function of SVM:

$$K(x, y) = e^{-\gamma \|x-y\|^2}. \quad (11)$$

Finally, a grid-search procedure is employed to select the parameters of SVM[18]: penalty parameters  $C = 32$  and kernel parameter  $\gamma = 0.000488$ .

The experimental results are shown in Table I. The missed detection is due to the zoom out effect of camera. As the moving objects only cover relatively small area of image frame, their corresponding motion vectors also become small. Figure 6 shows a missed detection case. The false detection is mainly caused by the montage representation of film. As shown in Figure 7, the movie



Figure 6. A missed detection case.



Figure 7. A shot sequence that cause false detection.

director sometimes emphasizes the important content by controlling the camera to repeatedly alternate among several scenes. This will result in some shot sequences that have similar characteristics as action scene.

The performance of action scene detection is usually measured by the following two metrics:

$$\text{Recall} = \frac{D}{D + MD}, \quad (12)$$

$$\text{Precision} = \frac{D}{D + FD}, \quad (13)$$

where  $D$  is the number of action scenes detected correctly,  $MD$  is the number of missed detection and  $FD$  is the number of false detection. For performance comparison, we also implement the algorithm proposed by Geng et al.[5]. As shown in Table II, our approach is, in overall, better than Geng’s approach in term of recall and precision.

### VII. CONCLUSION

We have presented an effective method for automatically detecting action scene in the digital movies. While

TABLE I.  
ACCURACY MEASURES FOR FOUR TEST VIDEOS.

Movie ID No.	No. of Action Scenes	Correct Detection	Missed Detection	False Detection
(1)	21	17	4	6
(2)	16	14	2	3
(3)	14	13	1	3
(4)	19	16	3	5

TABLE II.  
PERFORMANCE COMPARISON FOR ACTION SCENE DETECTION.

Movie ID No.	Our Approach		Geng’s Approach	
	Recall	Precision	Recall	Precision
(1)	80.95%	73.91%	71.19%	72.72%
(2)	87.50%	82.35%	81.25%	72.22%
(3)	92.85%	81.25%	92.85%	76.47%
(4)	84.21%	76.19%	73.68%	73.68%

previous approaches addressed on shot level of video structure only, our approach construct more semantic-complete scene structure of video. Thus, more spatiotemporal information of video is exploited for analysis. Based on some features extracted from scene structure, a support vector machine is employed to detect action scene. Experimental results show that the proposed approach works reasonably well in detecting most of the action scenes. Compared with the related work[5], our approach is promising. Our approach can be applied directly to video abstraction and can be utilized to support high-level video indexing in movie databases. As dialogue scene is also an important component of movie trailer, our future work will be focused on how to extend current technique to dialogue scene detection.

### REFERENCES

- [1] D. Arijon, “Grammar of the film language.” Silman-James Press, Los Angels, 1991.
- [2] L. Chen, S. J. Rizvi, and M. T. Ozsu, “Incorporating audio cues into dialog and action scene extraction.” In *Proceedings of SPIE Conference on Storage and Retrieval for Media Database*, pp.252–264, San Jose, CA, 2003.
- [3] J. Nam, M. Alghoniemy, and A. H. Tewfik, “Audio-visual content-based violent scene characterization.” In *Proceedings of International Conference on Image Processing*, pp.353–357, Chicago, 1998.
- [4] B. Lehane, N. E. O’Connor, and N. Murphy, “Action sequence detection in motion pictures.” In *Proceedings of European workshop on the integration of knowledge, Semantic and Digital Media Technologies*, London, 2004.
- [5] Y. L. Geng, D. Xu, J. Z. Yuan, and S. H. Feng, “Two important action scenes detection based on probability neural networks.” In *Proceedings of International Symposium on Neural Networks*, pp.448–453, Chengdu, China, 2006.
- [6] K. Jonsson, J. Kittler, Y. P. Li, and J. Matas, “Support vector machines for face authentication.” *Image and Vision Computing*, 20(5-6):369–375, April 2002.
- [7] G. Guo, S. Z. Li, and K. L. Chan, “Support vector machines for face recognition.” *Image and Vision Computing*, 19(9-10):631–638, August 2001.
- [8] M. Pontil and A. Verri, “Support vector machines for 3D object recognition.” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(6):637–646, June 1998.
- [9] C. Bahlmann, B. Hassdonk, and H. Burkhardt, “On-line handwriting recognition with support vector machine – a kernel approach.” In *Proceedings of International Workshop on Frontiers in Handwriting Recognition*, pp.49–54, Ontario, Canada, 2002.
- [10] A. Ganapathiraju, J. E. Hamaker, and J. Picone, “Applications of support vector machines to speech recognition.” *IEEE Transactions on Signal Processing*, 52(8):2348–2355, August 2004.
- [11] S. Tong and E. Chang, “Support vector machine active learning for image retrieval.” In *Proceedings of ACM International Conference on Multimedia*, pp.107–118, 2001.
- [12] U. Gargi, R. Kasturi, and S. H. Strayer, “Performance characterization of video-shot-change detection methods.” *IEEE Transactions on Circuits and Systems for Video Technology*, 10(1):1–13, February 2000.
- [13] L. H. Chen, C. W. Su, H. Y. Liao, and C. C. Shih, “On the preview of digital movies.” *Journal of Visual Communication and Image Representation*, 14(3):357–367, September 2003.

- [14] A. K. Jain and R. C. Dubes. *Algorithms for Clustering Data*. Prentice Hall, Englewood Cliffs, NJ, 1988.
- [15] T. Evgenious, M. Pontil, and T. Poggio, "A unified framework for regularization networks and support vector machines." A.I. Memo 1654, MIT, Cambridge, MA, 1999.
- [16] V. N. Vapnik, "Statistical learning theory." Wiley, New York, 1998.
- [17] D. L. Gall, "MPEG: A video compression standard for multimedia applications." *Communication of ACM*, 34(4):46–58, April 1991.
- [18] C. W. Hsu, C. C. Chang, and C. J. Lin, "A practical guide to support vector classification." Technical report, Department of Computer Science, National Taiwan University, Taipei, Taiwan, 2003.

**Liang-Hua Chen** received the B.S. degree in information engineering from National Taiwan University, Taipei, Taiwan in 1983. He received the M.S. degree in computer science from Columbia University, New York, in 1988, and the Ph.D. degree in computer science from Northwestern University, Evanston, IL, in 1992. From March 1992 to July 1992, he was a senior engineer at Special System Division, Institute for Information Industry, Taipei, Taiwan. He is currently a full professor in the Department of Computer Science and Information Engineering at the Fu Jen University, Taipei, Taiwan. Dr. Chen's research interests include computer vision and pattern recognition.

**Chih-Wen Su** received the B.S. degree in mathematics and M.S. degree in Computer Science and Information Engineering, both from Fu Jen University, Taipei, Taiwan, in 1999 and 2001, respectively. He received the Ph.D. degree in Computer Science and Information Engineering from National Central University, Chung-Li, Taiwan, in 2006. Currently, he is a postdoctoral fellow in the Institute of Information Science, Academia Sinica, Taiwan. His research interest are in image and video analysis and content-based indexing and retrieval.

**Chi-Feng Weng** received the B.S. and M.S. degree in Computer Science and Information Engineering from Fu Jen University, Taipei, Taiwan, in 2005 and 2007, respectively. His current research interests include video analysis and image processing.

**Hong-Yuan Mark Liao** received the B.S. degree in physics from National Tsing-Hua University, Hsin-Chu, Taiwan, in 1981, and the M.S. and Ph.D. degrees in electrical engineering from Northwestern University, Illinois, in 1985 and 1990, respectively. Currently, he is a research fellow of the Institute of Information Science, Academia Sinica, Taiwan. He is also jointly appointed as a professor of the Computer Science Department of National Chiao-Tung University, Taiwan. Dr. Liao's current research interests include multimedia signal processing, wavelet-based image analysis, content-based multimedia retrieval, and multimedia protection. He was an Associate Editor of the IEEE Transactions on Multimedia during 1998-2001. Dr. Liao is on the Editorial Boards of Journal of Visual Communication and Image Representation; the Acta Automatica Sinica; and the Journal of Information Science and Engineering. He is a senior member of the IEEE Computer Society.