INTEGRATING MOTION AND COLOR FOR CONTENT BASED VIDEO CLASSIFICATION

Markos Zampoglou[1], Theophilos Papadimitriou[2], and Konstantinos I. Diamantaras[3]

[1]Dept. of Applied Informatics, University of Macedonia, Thessaloniki, Greece, email: mzampog@uom.gr, [2]Dept. Int. Economic Relat. & Develop. Democritus University of Thrace, Komotini, Greece, e-mail: papadimi@ierd.duth.gr, [3]Dept. of Informatics, TEI of Thessaloniki, Greece, email: kdiamant@it.teithe.gr

ABSTRACT

How to achieve the goal of automatically classifying video shots by their content is still an issue under debate. In this paper we present a novel set of low-level descriptors for the classification of TV video shots into meaningful semantic classes which can then be useful when browsing a TV stations archives. The motion features we propose consist of a modified Perceived Motion Energy Spectrum descriptor for local motion and a Normalized Dominant Motion Histogram for camera motion. Since exclusively motion-based classification has a very limited applicability, we also add three normalized local HSV histograms, extracted from particular key-frames we select with a simple yet efficient approach, as color descriptors. Our experimental implementation is tested on real-world TV video shots using a binary classifier based on Support Vector Machines and the results demonstrate that the proposed features can achieve high success rates not only on narrow and specialized classes, but also on more generic ones.

Index Terms— Content-based video retrieval, Video indexing, Video Classification, Motion Descriptors.

1. INTRODUCTION

The field of content Content-Based multimedia Indexing and Retrieval is a relatively young field, attracting more and more interest. The main reason for this is the increasing amount of multimedia material that becomes available, and the need to introduce automatic processes of tagging, classification and retrieval in order to replace today's manual approaches, which are becoming increasingly inadequate in dealing with these issues. Content-Based Image Indexing and Retrieval has seen great progress in the recent years [1-4]. The potential of video material on the other hand is far from fully explored, as yet.

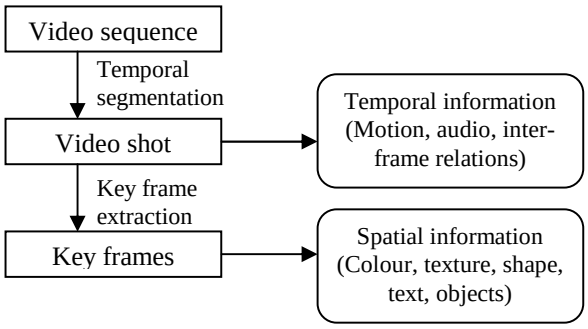

Figure 1: The feature extraction process for digital video.

Much like image indexing, video indexing is fundamentally based on the processing of a set of features extracted from a video sequence. However, because of the spatiotemporal nature of video data, the extraction process consists of multiple steps (Fig. 1). The first step is to segment a video sequence into the shots it consists of.

While a complete video sequence (such as a complete movie) may contain several camera changes and scene transitions, a shot is a piece of video taken from a single camera and containing no such changes. From each shot, we are then able to extract features to describe the temporal aspects of the shot, such as motion or audio. The following step is to select a limited number of characteristic frames (called key-frames) from each shot, and extract the spatial aspects of these frames, such as color, texture or text.

Each of these steps is an individual field of research. Temporal segmentation means all transitions between shots, being gradual or abrupt, have to be automatically detected. The segmentation of the temporal information goes beyond the scope of this text, and for our experimental applications we have resorted to manual segmentation. A review of the field can be found in [5].

Taking advantage of the text and audio information is a task which poses an entirely different set of problems. The field addressing these problems is called Multimodal Video

Retrieval. It has so far shown great potential, and the reader can turn to [6-7] for details.

In this paper, we will mostly focus on temporal information, and use a basic spatial information descriptor to achieve classification. Motion information in digital video usually comes in the form of vector fields. A motion vector is estimated by extracting a macroblock from a frame and then seeking its best match in another frame in the future, which can be the very next frame or N frames away. The vector field can either be dense, where the surrounding block of each pixel is used for estimation, thus assigning a vector on every pixel, or sparse, where the macroblocks are non-overlapping, thus assigning one vector per block. This raw motion information is often of low accuracy and offers little insight as to the content of the video, but it is, in most cases, the basis for the motion descriptors of a video shot. It is possible to use this low-level information as a descriptor for video motion, such as the vector magnitude and angle distributions [8], the histograms of the x and y components of the vectors [9-10], the histograms of the motion vector angles [11] or the spatial and temporal distribution of motion by locating series of nonzero vectors in space and time [12].

However, the motion vector fields express the complete motion content of a shot, and in this sense describe a mixture of the camera motion as well as the motion of the objects that appear in a shot. It is extremely difficult to extract meaningful descriptors based on this information. It would be more to our benefit to separate the two and model them separately, in order to form efficient video motion content descriptors.

In order to detect global motion, many models assume it is expressed by the majority of the motion vectors in a field [13-14]. The assumption is that the majority of pixels in every frame represent the background, and their motion vectors result exclusively from camera motion.

Following the detection of the dominant motion vectors, we can use that information to estimate the camera movements [15-16]. It is possible however, to estimate camera motion without seeking the dominant vectors. In [17], for example, a six-parameter camera motion model is estimated directly from the motion vectors. The following step is then to compensate for the camera motion in order to isolate the object motions. That information can then be used in order to form a local motion descriptor, which can either directly describe low-level information [18], or can be used for higher-level features, such as object trajectories [19].

A completely different approach is the "Luminance Field Trace", where each grayscale frame is treated as a point in a space whose dimensionality equals the number of pixels. A video shot is then represented as a trajectory in that space. Following dimensionality reduction, it is attempted to match trajectories so as to detect similar shots [20-21].

Our aim was to build a classifier, which, when trained with a number of videos, would be able to classify any given future video. We decided that camera and object motion should be treated separately for such a scheme. Also, as will be shown below, we added a basic color descriptor to increase our classifier's capabilities.

We start with video shots and proceed with the extraction of temporal information, key-frame selection and the extraction of spatial information. We present a set of features for the description of motion information, and mix them with color histograms we extract from key-frames selected with a fast and efficient approach.

2. OUR PROPOSED METHODS

2.1. Local Motion Features

Ma and Zhang, in [22] propose a descriptor for video motion called the perceived motion energy spectrum (PMES). It is a robust, fixed length descriptor which can convey both the spatial distribution and the overall intensity of object motion within a video shot.

The first step is to calculate the mixture energy of each macroblock, that is, the motion resulting from both camera motion and object motion. That is achieved by trimming and averaging over the vector magnitudes of each block through time. So, the mixture energy of block *(i, j)* is calculated by:

$$MixEn_{i,j} = \frac{1}{N - 2\lfloor \alpha N \rfloor} \sum_{n=\lfloor \alpha N \rfloor + 1}^{N-\lfloor \alpha N \rfloor} Mag_{i,j}(n), \quad (1)$$

where N is the total number of magnitudes (in our case, the number of vector fields extracted from a shot), α is the trimming parameter ($0 \le \alpha \le 0.5$) and $Mag_{i,j}(n)$ is the magnitude of element n. The mixture energy is then normalized to [0, 1].

The next step is to remove the effect of camera motion. This is achieved by filtering through the Global Motion Ratio (GMR). We can treat a block's motion vector angle through time as a stochastic variable and calculate the angle entropy through its probability distribution function. The mixture energy of blocks with consistent angles will most probably be due to camera motion, and the low entropy result for these blocks will allow us to weed them out.

To estimate the angle entropy, we first quantize the vector angles into n directions, and form a histogram. For each bin t, the probability distribution is calculated through

$$p(t) = AH_{i,j}(t) / \sum_{k=1}^{n} AH_{i,j}(k), \quad (2)$$

where $AH_{i,j}(t)$ denotes the histogram value for bin t. The angle entropy for the histogram can then be found by

$$AngEn_{i,j} = -\sum_{t=1}^{n} p(t) \log p(t), \quad (3)$$

The angle entropy is then normalized to [0, 1], which gives us the GMR. The perceived motion is then calculated by multiplying each block's mixture energy with its GMR.

$$PMES_{i,j} = GMR_{i,j} \times MixEn_{i,j}, \quad (4)$$

In our previous research, we applied the PMES descriptor for TV footage classification with significant success [23].

However, the PMES descriptor in its original form has a number of limitations. First, it uses a large number of features (one per macroblock) to describe motion patterns. This, in classification results in a very high-dimensional space which makes learning extremely difficult. Secondly, it only describes the magnitudes of local motion, which is only a small part of the information we can extract from the motion fields. Third, even if we added further motion descriptors, motion by itself has a limited classification potential. We will also need spatial descriptors, which will only increase the dimensionality.

In order to reduce the dimensionality of the classification problem, we decided to reduce the length of the descriptor. To achieve this, we average the PMES measure over neighborhoods of macroblocks. We can thus achieve a more coarse representation of the spatial distribution of local motion. In this manner, not only we reduce the actual dimensionality of the problem, but we also remove unwanted detail, leaving only a rough representation of the distribution of local motion intensity. In long shots, high PMES values do not appear concentrated on single blocks, but rather spread on broader regions. This information can be retained after averaging, even if we lose some detail.

A second modification we applied to the PMES measure was normalization. In its initial form, the PMES measure conveys both the intensity and the spatial distribution of local motion. However, in many cases, videos that belong to the same class present varying overall motion intensities but can be identified by their spatial distribution. To this end, we should isolate the spatial distribution aspect, regardless of the magnitude of the local motion. To achieve this, we can normalize the PMES values of a shot, to make them sum to 1, thus keeping only their relative values.

$$\overline{PMES_{i,j}} = \frac{PMES_{i,j}}{\sum_{i,j} PMES_{i,j}}, \quad (5)$$

2.2. Local Motion Features

As we mentioned before, the PMES measure completely removes the effects of camera motion and describes only local motion. However, camera motion patterns often carry important information about the content of a video shot. In order to take them into account we built a simple descriptor. First we quantize all motion vector angles into 8 directions, plus one for zero-magnitude vectors. To deal with noise, we threshold all vectors below a certain magnitude to count as zero-magnitude. Then, under the assumption that the majority of pixels belong to the background, we extract the dominant angle from each vector field. We can assume this to be the direction of the camera motion for the current vector field.

Since the length of the video shots varies, we have a different number of motion fields for every different shot. To convert it into a fixed-length descriptor, we form the Dominant Direction Histogram by counting the number of times each dominant direction appears. We group the initial 9 directions into four groups: Horizontal (0° and 180°), Vertical (90° and 270°), Diagonal (45°, 135°, 225° and 315°) and Static, since it does not make any difference whether, for example, a horizontal motion is leftward or rightward. Such detail does not usually give any useful information on the content of a shot, and keeping the number of features as low as possible is extremely important.

Another result of the varying video length is the scale issue: The Dominant Direction Histogram as described above has a varying sum, and even two videos which both have exclusively horizontal motion but different lengths will have different histograms. To alleviate it, we proceed to normalize each histogram to [0, 1].

The Normalized Dominant Direction Histogram (NDDH) is a compact descriptor of the camera motions of a shot which allows for multiple different camera motions to be taken into account, thus being significantly more robust and general than using a descriptor which allows for a single camera motion to be modeled.

2.3. Color Features

The features we have presented so far give us an overview of the motion patterns in a video shot: The PMES measure gives us the spatial distribution of the local motion magnitudes, while the direction histogram gives us the temporal distribution of the camera motion directions. However, most classes that we seek can also be distinguished by their color distributions. We will thus also need a color descriptor for classification to be effective.

The most typical color distribution descriptor is the color histogram. To construct a color histogram we quantize the color spectrum into a number of cells and count the number of pixels that fall into each cell. Several variations of the color histogram exist, as well as various proposals as to the quantization. In our approach, since we are dealing with TV footage, we know that usually the middle part of the frame contains the items and events of interest. In trying to isolate the central part, we split the frame in three (upper, middle and lower) and extract three separate histograms. As for the quantization, we use the HSV color spectrum and

simply take 8 cells for H, completely ignoring S and V. Colorwise, the H parameter carries most information, and since we have increased dimensionality by taking three histograms from each frame, it is in our best interest to keep these histograms as compact as possible. After extraction, each area's histogram is also normalized so that its 8 elements sum to one.

A very important issue when dealing with spatial features from video shots is key-frame selection: We can extract a number of histograms equal to the number of frames in the shot (multiplied by three, in our case), but that would lead to too much redundancy. We have to choose a small number of frames (preferably one) from which to extract the spatial information which will describe the video shot.

Ferman et al [24] proposed a simple method for selecting a keyframe from a shot. The way to achieve this is by estimating the sum of each frame's histograms' absolute differences from every other one and choosing the one that minimizes the function.

$$KeyH = \arg\min_{H_k} \left\{ \sum_{l \neq k} |H_l - H_k| \right\}, \quad (6)$$

where H_n is the histogram of the n^{th} frame of the shot.

A problem with this approach is that it is computationally expensive, since each shot contains a large number of frames. However, we know that there is continuity in the color distribution between frames in a shot, since objects do not appear or disappear abruptly between frames. We can thus chose to extract a small number of frames from each shot (in our case, five), evenly distributed through time, and choose one of them as the most representative.

Thus, after applying (6) to these five frames, we end up with a single frame's histograms whose color distributions roughly represent the overall color distribution of the shot.

2.4. Classifier

Having established a set of features to describe both temporal and spatial features of a shot, we proceeded to test these features in a number of classes. For classification, we used a Support Vector Machine, a very popular and well-established binary classifier [25-26].

Support Vector Machines have excellent generalization capabilities, and have been successfully applied so far for relevance feedback on Content-Based Image and Video retrieval [27-29]. In our case, where we deal with video classification in predetermined classes, our approach was to train an SVM with a number of videos from a certain class, to have it distinguish between future videos that belonged to that class and those that didn't. We are working towards a system applicable on a TV station's archives, and, to this end, we user real world TV data to evaluate its efficiency.

3. IMPLEMENTATION

3.1. The video database

We were offered a part of the archive of the Omega TV channel in Thessaloniki, Greece. The database was manually cut into 1074 single shot videos of varying content, from newscasts to sports, to talk shows and theatrical plays. As mentioned in the introduction, we bypassed the issue of temporal segmentation and focused on classification. The fact that the database came from the real world offered a number of challenges, since we had to deal with videos of low quality, whose content had to be classified in a number of classes according to the channel's needs. On the other hand, this gave us the opportunity to test our approach on the basis of its real world implementation potential.

3.2. Feature Extraction

To calculate the motion features, we had to extract the motion fields from the given videos. To this end, we applied a block-matching algorithm, extracting the motion fields not over consecutive frames, but over a temporal distance of 8 frames each. A large temporal distance for the motion fields means that the motion vectors have correspondingly increased magnitudes. This helps eliminate potential camera shakes, and also significantly reduced the effects of noise, since both of these will remain small in magnitude and thus easily detectable. The vector field was sparse and consisted of 9×11 vectors, each corresponding to a 64×64 pixel block. For the PMES measure, this led to 99 features per shot, but after averaging over neighborhoods of 3×3 blocks, this was reduced to 12 features. The motion vectors, after applying a threshold of 10 pixels, were also used to extract the Normalized Dominant Direction Histogram, consisting, as described above, of 4 features.

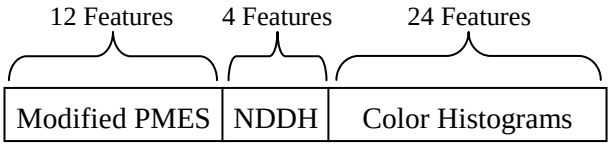

Figure 2: The proposed feature set.

Finally, five frames were extracted from the videos, namely the first, the last and three more, evenly distributed through time. From each of these frames, the H histogram of the HSV spectrum was extracted for the upper, middle and lower part of the frame, using 8 cells for each. Of these five triplets of histograms, one was chosen to minimize the sum of absolute differences from the others. Thus, the final

feature vector length was 12 features for the modified PMES feature, 4 for the NDDH and 3×8=24 for the color histograms, amounting to a total of 40 features for each of the 1074 shots (Fig. 2).

3.3. Classification

The SVM application that we chose to use was Thorsten Joachims' SVM-light implementation [30], being fast, efficient and user-friendly. Through direct experimentation with linear, multinomial and radial basis kernels, we came to the conclusion that we should use a linear kernel, since it proved to be the most resistant to overfitting for the experiments described below.

Concerning the classes used, we tested four different ones: Two of them were extremely narrow ones that the station could use as a primary classification, namely "Newscast" (consisting of 16 shots), which were shots of the stations daily newscast and shots from a particular weekly talk show (19 shots, labeled as "Interview"). These did not seem to provide any problems for the features. Since shows like that are usually captured in the same studio, the color patterns are mostly identical and the camera motion patterns as well. As a result, successful classification results after training were expected be quite high. More interesting results concern the two more abstract classes, namely "soccer" (174 shots), which contained shots of various soccer game, with varying lighting condition and color patterns, and "speaker" (157 shots), containing all shots where a speaking person was standing in front of a mostly (but not exclusively) static camera, with his head occupying at least 15% of the frame. The content varied from interviews from the streets to political statements. Figure 3 shows a sample of each class as well as four videos that didn't belong to any class. The classes were at some cases overlapping, and the SVM for each one was trained and evaluated independently.

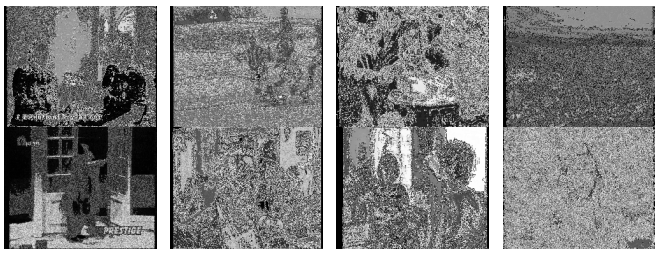

Figure 3: Top row: Four examples, one for each class: "Newscast" "Interview", "Soccer", "Speaker". Bottom row: four examples that didn't belong to any class.

Finally, it should be noted that, since SVMs are sensitive to the number of positive and negative training examples of each class [31], and given that in all classes the positive examples were fewer than the negative ones, we had to impose a weight factor on the positive examples. This was achieved by a simple trick [23]: We inserted multiple instances of each positive training video in the training set until the number of positive examples became at least 1.2 times the number of negative examples. The small advantage on the positive examples reflected the fact that false positives are more serious than false negatives, and it is thus preferable to slightly favor positive examples over negative.

4. RESULTS

The training set we used was about 35% of the total dataset, leaving 65% for evaluation. The results presented are the mean results of 100 repetitions for each class. At each repetition, a training set was formed by randomly picking 35% of the positive examples for a particular class and 35% of the negative examples for that class. Multiple instances of the positive examples were inserted, to balance the positive and negative set sizes. After training an SVM classifier, the rest of the examples were used as an evaluation set.

As can be seen in Table 1, the success rate for the narrow classes was virtually perfect, with almost no misclassifications on the positive labeled shots and very few misclassifications on the negative ones. As for the more generalized classes, the combination of color, object and camera motion patterns did give very good classification results. In the results presentation, 'True Positives' express the number of Correct Positives as a percentage of all the Positive examples in the evaluation set, and the 'True Negatives' are defined correspondingly.

Table 1: The classification results for the four classes

	Newscast	**Interview**	**Soccer**	**Speaker**
True Positives	100%	99.0%	92.6%	93.8%
True Negatives	99.35%	96.6%	88.4%	80.7%
Overall Success	99.36%	96.7%	89.1%	82.7%

5. CONCLUSIONS

We presented a set of features for the content-based classification of video shots based on motion and color. The application of our proposed feature set upon part of the archives of a TV station demonstrated that, for a number of different classes, our features were very successful in capturing the fundamental characteristics of the training examples and achieving high levels of generalization.

Towards building a complete system for classifying TV stations archives, our future work will focus on both the refinement of our existing features as well as the incorporation of further spatial features for texture and shape, to make it possible to extend our classification scheme to a broader range of classes.

REFERENCES

[1] R. Datta, J. Li and J.Z. Wang, "Content-based image retrieval-approaches and trends of the new age," *Proceedings of the 7th International Workshop on Multimedia Information Retrieval, in conjunction with ACM International Conference on Multimedia*, pp. 253-262, 2005.

[2] Y. Rui, T.S. Huang, and S.F. Chang, "Image Retrieval- Current Techniques, Promising Directions And Open Issues," *Journal of Visual Communication and Image Representation*, pp. 39-62, April 1999.

[3] A.W.M. Smeulders, M. Worring, S. Santini, A. Gupta, and R. Jain, "Content-based image retrieval at the end of the early years," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, pp. 1349-1380, Dec 2000.

[4] S. Z. Xiang, and T.S. Huang, "Relevance feedback in image retrieval- A comprehensive review," *Multimedia Systems*, pp. 536-544, April 2003.

[5] I. Koprinska, and S. Carrato, "Temporal Video Segmentation: A Survey," *Signal Processing: Image Communication*, Elsevier, pp. 477-500, January 2001.

[6] J. Calic, N. Campbell, S. Dasiopoulou, and Y. Kompatsiaris, "A Survey on Multimodal Video Representation for Semantic Retrieval," *Computer as a tool (Eurocon 2005), the Third International Conference on*, IEEE, pp. 135-138, November 2005

[7] C.G.M. Snoek, M. Worring, "Multimodal Video Indexing, a Review of the State-of-the-art," *Multimedia Tools and Applications, Springer Netherlands*, pp. 5-35, January 2005.

[8] D. Zhong, H.J. Zhang, and S.F. Chang, "Clustering Methods for Video Browsing and Annotation," *Proceedings of SPIE*, pp. 239-246, March 1996.

[9] Chen, J. F, Liao, H. Y. M., and Lin, C. W., *Knowledge-Based Intelligent Information and Engineering Systems*, Springer Berlin / Heidelberg, 2005, chapter "Fast Video Retrieval via the Statistics of Motion Within the Regions-of-Interest," pp. 381-387.

[10] A. K. Jain, A. Vailaya, and X. Wei, "Query by Video Clip," *Mutimedia Systems*, Springer Berlin / Heidelberg, pp. 369-384, September 1999.

[11] Ardizzone, E., Gatani, L., La Cascia, M., Lo Re, G., and Ortolani, M., *Advances in Multimedia Modeling*, Springer Berlin / Heidelberg, 2006, chapter "A P2P Architecture for Multimedia Content Retrieval," pp. 462-474.

[12] A. Divakaran, A. Vetro, K. Asai, and H. Nishikawa, "Video browsing system based on compressed domain feature extraction", *Consumer Electronics, IEEE Transactions on*, pp. 637-644, August 2000.

[13] R. Fablet, P. Bouthemy, and P. Pérez, "Nonparametric motion characterization using causal probabilistic models for video indexing and retrieval," *Image Processing, IEEE Transactions on*, pp. 393-407, April 2002.

[14] R. Fablet, and P. Bouthemy, "Statistical motion-based object indexing using optic flow field," *Pattern Recognition, IEEE International Conference on*, pp. 287-290, vol. 4, 2000.

[15] W.N. Lie, and W.C. Hsiao, "Content-based video retrieval based on object motion trajectory," *Multimedia Signal Processing, IEEE Workshop on*, pp. 237-140, 2002.

[16] W. Pan, F. Deschenes, "Interpreting Camera Operations in the Context of Content-based Video Indexing and Retrieval," *Computer and Robot Vision, IEEE Third Canadian Conference on*, pp. 7-7, June 2006

[17] M. Gelgon, P. Bouthemy, "Determining a Structured Spatio-Temporal Representation of Video Content for Efficient Visualization and Indexing", *Proceedings of 5th European Conference on Computer Vision*, Vol I, pp. 595-609, 1998.

[18] G. Piriou, P. Bouthemy, and J.F. Yao, "Recognition of dynamic video contents with global probabilistic models of visual motion," *Transactions in Image Processing*, IEEE, pp. 3418-3431, November 2006.

[19] W.N. Lie, and W.C. Hsiao, "Content-based video retrieval based on object motion trajectory," *Multimedia Signal Processing, IEEE Workshop on*, pp. 237-140, 2002.

[20] Li Gao, Zhu Li, and A. K. Katsaggelos, "Robust Video Retrieval with Luminance Field Trace Indexing and Geometry Matching", *Content-Based Multimedia Indexing, 2007. CBMI '07. International Workshop on*, IEEE, pp. 99-105, June 2007.

[21] Koskela, M., and Laaksonen, J., *Artificial Intelligence Applications and Innovations*, Springer Boston, 2006, chapter "Semantic Concept Detection from News Videos with Self-Organizing Maps," pp. 591-599.

[22] Y.F. Ma, and H.J. Zhang, "A new perceived motion based shot content representation," *Image Processing, International Conference on*, pp. 426-429, 2001.

[23] M. Zampoglou, Th. Papadimitriou, and K. I. Diamantaras, "Support Vector Machines Content-Based Video Retrieval based solely on Motion Information", *Proc. 17th Int. Workshop on Machine Learning for Signal Processing (MLSP-2007)*, IEEE, Thessaloniki, Greece, August, 2007

[24] A. M. Ferman, A. M. Tekalp, and R. Mehrotra, "Robust Color Histogram Descriptors for Video Segment Retrieval and Identification," *Image Processing, IEEE Transactions on*, May 2002, pp. 497-508.

[25] Vapnik, V, *The Nature of Statistical Learning Theory*, Springer-Verlag, New York (1995).

[26] J. C. Burges, "A Tutorial on Support Vector Machines for Pattern Recognition," *International Journal on DMKD*, pp. 121-167, 1998

[27] Lei Zhang, Fuzong Lin, Bo Zhang, "Support vector machine learning for image retrieval." *2001 International Conference on Image Processing*, pp:721-724, 7-10 Oct. 2001.

[28] Dacheng Tao, Xiaoou Tang, "Random sampling based SVM for relevance feedback image retrieval." *Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp: 647- 652, 27 June-2 July 2004.

[29] V. Mezaris, I. Kompatsiaris, N.V. Boulgouris, M.G. Strintzis, "Real-time compressed-domain spatiotemporal segmentation and ontologies for video indexing and retrieval." *IEEE Transactions on Circuits and Systems for Video Technology*, pp: 606-621, May 2004.

[30] Joachims, T., Schqilkopf, B., Burges, C., and Smola, A., (ed.), *Making large-scale SVM learning practical. Advances in Kernel Methods -Support Vector Learning*, MIT-Press, 1999.

[31] Chu-Hong Hoi, M.R. Lyu, "Group-based relevance feedback with support vector machine ensembles", *Pattern Recognition, Proceedings of the 17th International Conference on*, pp. 874-877, Vol. 3, Aug. 2004.