

Efficient Measuring of Various Motions in MPEG Videos

JungHwan Oh and Praveen Sankuratri
Department of Computer Science and Engineering
University of Texas at Arlington
Arlington, TX 76019-0015 U. S. A.
email: oh@cse.uta.edu

Wallapak Tavanapong
Computer Science Department
Iowa state University
Ames, IA 50011 U. S. A.
email: tavanapo@cs.iastate.edu

ABSTRACT

There is a need to analyze motions in video sequences to characterize them since *motion* is a representative feature that distinguishes videos from other multimedia. However, the existing approaches can only provide very preliminary classification of motions. To support in-depth measurement of various motions, we need to distinguish the different motions generated by the different sources. In this paper, we explicitly distinguish camera and object motions in a shot. In addition to distinguishing motions themselves, we propose new techniques to measure not only *amount* but also *direction* of each motion distinguished, and express them into comparable forms. Our preliminary experimental studies indicate that the proposed techniques are promising.

KEY WORDS

Image Processing and applications, Video Processing, Camera and object motion detection, MPEG Video.

1 Introduction

With the expansion of software and hardware technologies for data compression, storage devices and communication networks, video media, has become an integral part of many applications in the fields including education, entertainment, business and medicine [1]. These have spurred the demand for the researches about content analyzing and indexing of videos which are essential for effective browsing, retrieving, filtering, and summarizing [2, 1]. Analyzing and indexing video data, however, are very complex due to the enormous size of video files and their semantically rich and unstructured contents. The first step managing video data, therefore, is to segment automatically each video stream into a set of basic units which are relevant to the structure of the video. In general, the most widely used basic unit is a *shot* which is defined as collections of frames recorded from a single camera operation. For shot characterization, we extract various features of video contents such as color, object, motion, etc. from decomposed shots.

Among these features, *motion* defined as the temporal intensity change between successive frames, is a unique character that distinguishes videos from other multimedia (i.e., image or audio). The motions in a video sequence are caused by two different sources which are camera and object(s). In other words, these motions can be divided into

camera and object motions. Camera motion detection has been investigated using motion vectors computed by optical flow or MPEG macro block compensation [3, 4]. Object motion detection in video sequences with fixed or moving camera has been studied using Bayes decision test, Gaussianity detection test, multiscale Markov random fields or morphological filtering [5, 6]. The overall motions are manipulated for content analyzing and indexing purposes in the existing techniques [7, 8, 9, 3, 10, 11, 12]. To determine the motion activity level of video segments, they are using a combination of image and audio features [7], a mode of motion vector magnitudes [8], a tangent distance between consecutive frames [9], mean, variance and median of motion vector magnitudes [10], or tensor applied to a video sequence represented as a 3D volume with one temporal (z) and two spatial (x,y) coordinates [11, 12].

However, the existing approaches can only provide very preliminary classification of motions. To support in-depth classification of various motions, we need to distinguish the different motions generated by the different sources. In this paper, we explicitly distinguish camera and object(s) motions in a shot. In addition to distinguishing motions themselves, we propose new techniques to measure not only *amount* but also *direction* of each motion distinguished, and express them into comparable forms.

The main contributions of the proposed techniques can be summarized as follows.

- The proposed technique to compute various motions is very cost-effective because it uses the motion vectors previously computed when the video was made in MPEG format.
- We present various motions using two-dimensional matrices. They are showing not only the amounts of motions but also the direction of them. Therefore, we can get more accurate and richer information of motion contents of shot.
- Because the proposed matrix comparison algorithm is very efficient and scalable, it can provide various ranges of clustering [13, 14, 15] for shots which is essential tool for content analyzing, indexing, browsing, and querying of video data.

The remainder of this paper is organized as follows. In Section 2, we describe an algorithm to detect camera motion changes, and a technique for automatic measurement of

various motions. The experimental results are discussed in Section 3. Finally, we give our concluding remarks in Section 4.

2 Motion Feature Extraction

In this section, we explicitly distinguish camera and object(s) motions in a shot. In addition to distinguishing motions themselves, we propose new techniques to measure not only *amount* but also *direction* of each motion distinguished, and express them into comparable forms.

2.1 Detection of Camera Motion Changes

We present a technique to detect the exact camera motion changes for each frame in a shot. In a MPEG video, each frame consists of a number of (i.e., 10×8 in our case) macroblocks as shown in Figure 1. Each macroblock consists of 16×15 pixels.

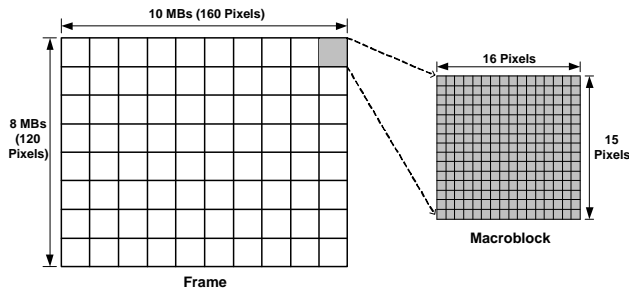


Figure 1. A Frame and its Macroblocks

Each macroblock is associated with a *motion vector* (mv). The motion vector is extracted from a pair of consecutive frames. The motion vector represents the displacement of macroblock between two consecutive frames as shown in Figure 2. Here the point p which is a center of macroblock is moving from frame k to frame $k+1$. The motion vector is represented as an arrow starting from its initial position (P) in frame k and ending at the final position (p') in frame $k+1$. The displacement is given by (mvx, mvy) of the points between two frames (see Figure 2). For calculating the camera motions, we need to extract this motion vector for all the macroblocks in the frames (see Equations (1) and (2)). These motion vectors are already present in the video because the MPEG videos are encoded using these motion vectors. This can save a significant computation.

$$\begin{pmatrix} mvx_{1,1}^k & mvx_{1,2}^k & mvx_{1,3}^k & \dots & mvx_{1,10}^k \\ mvx_{2,1}^k & mvx_{2,2}^k & mvx_{2,3}^k & \dots & mvx_{2,10}^k \\ \dots & \dots & \dots & \dots & \dots \\ mvx_{8,1}^k & mvx_{8,2}^k & mvx_{8,3}^k & \dots & mvx_{8,10}^k \end{pmatrix} \quad (1)$$

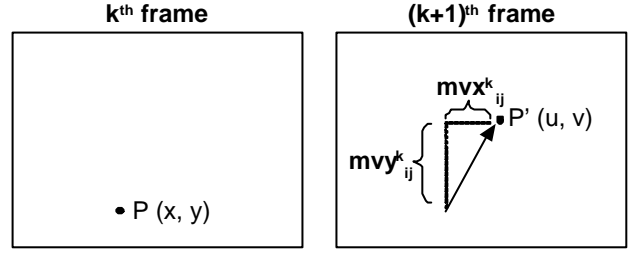


Figure 2. Motion Vector Calculation

$$\begin{pmatrix} mvy_{1,1}^k & mvy_{1,2}^k & mvy_{1,3}^k & \dots & mvy_{1,10}^k \\ mvy_{2,1}^k & mvy_{2,2}^k & mvy_{2,3}^k & \dots & mvy_{2,10}^k \\ \dots & \dots & \dots & \dots & \dots \\ mvy_{8,1}^k & mvy_{8,2}^k & mvy_{8,3}^k & \dots & mvy_{8,10}^k \end{pmatrix} \quad (2)$$

The affine model [3] is employed to compute camera motions from these mv s for the following considerations even if there are more accurate models of higher order. First, the affine model is more resilient to the noisy and sparse motion vectors. Second, we only need to compute the camera motion values of pan, tilt, rotation and zoom as shown in Figure 3 for analyzing different types of motions to be used in content-based indexing.

In the affine motion model, the motion vector (u, v) of a macroblock located at position (x, y) is expressed as

$$\begin{pmatrix} u \\ v \end{pmatrix} = \begin{pmatrix} a_2 & a_3 \\ a_5 & a_6 \end{pmatrix} \times \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} a_1 \\ a_4 \end{pmatrix} \quad (3)$$

where $\Psi (= a_1, a_2, a_3, a_4, a_5, a_6)$ is the parameter vector estimated from the motion vectors of the underlying frame by using the Gauss Elimination (GE) method [3]. After the estimation of the *parameter vector*, we can exploit this information for camera motion characterization aimed at detection of the predefined classes of camera motion. The parameters can be expressed in another basis of elementary fields as in [4, 16], which are more directly related to the physically meaningful camera motion as follows.

$$\begin{aligned} pan &= a_1, \quad tilt = a_4 \\ div &= \frac{1}{2} \times (a_2 + a_6), \quad rot = \frac{1}{2} \times (a_5 - a_3) \end{aligned} \quad (4)$$

These terms of the transformed *parameter vector* namely *pan*, *tilt*, *div* and *rot* represent each component of the motion vector induced by camera operation of pan (or horizontal tracking), tilt (or vertical tracking), zoom (or forward/backward tracking), and rotation, respectively.

The overall procedure of the proposed technique to detect camera motions consists of the following steps.

1. From the motion vectors which are seen in Equation (1) extracted from a frame pair, three *Most Dominant Motion Vector (MDMV)*s are selected. A *MDMV* is

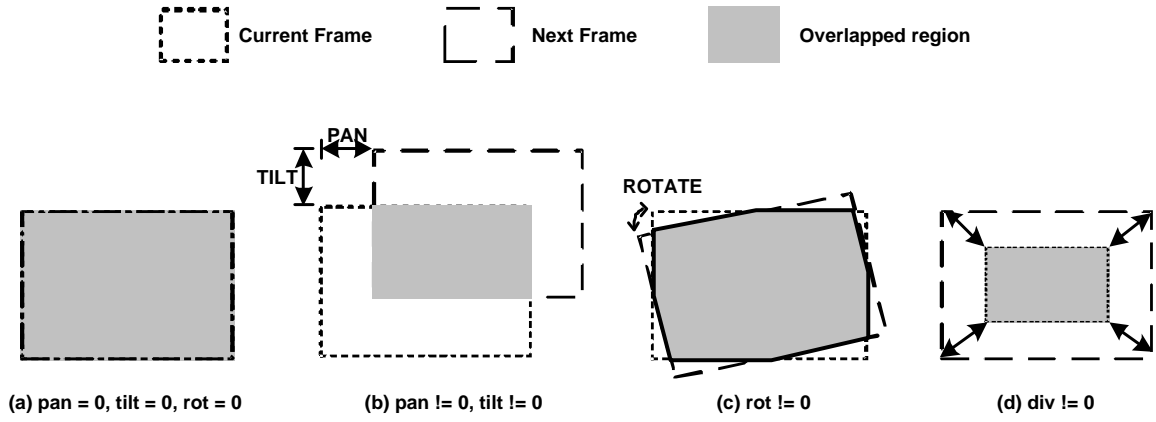


Figure 3. Pan, Tilt, Rotation and Zoom

defined as a most frequently occurred mv . If we assume that the selected $MDMVs$ are $mvx_{i,j}$, $mvx_{i',j'}$, $mvx_{i'',j''}$, we can have the corresponding $mvx_{i,j}$, $mvx_{i',j'}$, $mvx_{i'',j''}$ from Equation (2).

- Since the degrees of dominance for these $MDMVs$ are different from each other, the $MDMVs$ are normalized with the different weights ($\omega_1, \omega_2, \omega_3$) as follows.

$$\begin{aligned} \overline{mvx_{i,j}} &= mvx_{i,j} \times \omega_1, & \overline{mvy_{i,j}} &= mvy_{i,j} \times \omega_1, \\ \overline{mvx_{i',j'}} &= mvx_{i',j'} \times \omega_2, & \overline{mvy_{i',j'}} &= mvy_{i',j'} \times \omega_2, \\ \overline{mvx_{i'',j''}} &= mvx_{i'',j''} \times \omega_3, & \overline{mvy_{i'',j''}} &= mvy_{i'',j''} \times \omega_3, \end{aligned}$$

where $\omega_1 = \frac{\text{Number of the first MDMV}}{\text{Number of Total MVs}}$,
 $\omega_2 = \frac{\text{Number of the second MDMV}}{\text{Number of Total MVs}}$,
 $\omega_3 = \frac{\text{Number of the third MDMV}}{\text{Number of Total MVs}}$.

- From these three $MDMVs$ which are $(\overline{mvx_{i,j}}, \overline{mvy_{i,j}})$, $(\overline{mvx_{i',j'}}, \overline{mvy_{i',j'}})$ and $(\overline{mvx_{i'',j''}}, \overline{mvy_{i'',j''}})$, we can compute three displacement coordinates such as $((x, y), (u, v))$, $((x', y'), (u', v'))$ and $((x'', y''), (u'', v''))$ since we already know the coordinates of (x, y) , (x', y') and (x'', y'') .
- From these three displacement coordinates, we compute Ψ (a_1, a_2, a_3, a_4, a_5 and a_6) using Equation (3).
- Using these parameters a_1, a_2, a_3, a_4, a_5 and a_6 computed in the previous step and Equation (4), we compute $pan, tilt, div$ and rot values.
- The signs of the pan and $tilt$ indicate the directions of camera motions. If a value of pan is positive, the camera motion is towards *right*. Otherwise, it is towards *left*. If a value of $tilt$ is positive, the camera motion is *upward*. Otherwise, it is *downward*.
- For each pair of frames in a shot, we compute the $pan, tilt, rot$ and div values. Let p_k, t_k, r_k and d_k represent

the $pan, tilt, rot$ and div calculated for the k and $k + 1$ frame. Since we assume that a shot has a single camera motion, we return the average $pan, tilt, rot$ and div values as the camera motion for a shot with n frames.

$$\bar{p} = \sum_{k=0}^{n-1} p_k, \quad \bar{t} = \sum_{k=0}^{n-1} t_k, \quad \bar{r} = \sum_{k=0}^{n-1} r_k, \quad \bar{d} = \sum_{k=0}^{n-1} d_k \quad (5)$$

2.2 Computation of TM, OM and CM

In this subsection, we describe a technique for automatic measurement of the overall motions in not only two consecutive frames but also whole shot which is a collection of frames. As a result, accumulated motions of shot are represented as a *two dimensional matrix*.

This *two dimensional matrix* which is called *Total Motion Matrix (TMM)* for a shot A with n frames is computed using the following algorithm.

- Step.1:** Two empty two dimensional matrices (having size (10×8) same as that of motion vector matrix) are created and their elements are all initialized with zeros.
- Step.2:** These two matrices ($TMMX$ for x-axis and $TMMY$ for y-axis) form the two components of *Total Motion Matrix (TMM)*, and can be seen as follows.

$$TMMX_A = \begin{pmatrix} ax_{1,1} & ax_{1,2} & ax_{1,3} & \dots & ax_{1,10} \\ ax_{2,1} & ax_{2,2} & ax_{2,3} & \dots & ax_{2,10} \\ \dots & \dots & \dots & \dots & \dots \\ ax_{8,1} & ax_{8,2} & ax_{8,3} & \dots & ax_{8,10} \end{pmatrix} \quad (6)$$

$$TMMY_A = \begin{pmatrix} ay_{1,1} & ay_{1,2} & ay_{1,3} & \dots & ay_{1,10} \\ ay_{2,1} & ay_{2,2} & ay_{2,3} & \dots & ay_{2,10} \\ \dots & \dots & \dots & \dots & \dots \\ ay_{8,1} & ay_{8,2} & ay_{8,3} & \dots & ay_{8,10} \end{pmatrix} \quad (7)$$

where $ax_{i,j}$ and $ay_{i,j}$ are the horizontal and the vertical total motion matrix components of the shot for the

macroblocks in the same position, and can be computed as follows.

$$ax_{i,j} = \sum_{k=0}^{n-1} mvx_{i,j}^k, \quad ay_{i,j} = \sum_{k=0}^{n-1} mvy_{i,j}^k \quad (8)$$

Now, we compute *Object Motion Matrix (OMM)* using the similar technique used for computing *TMM* in the above. Similarly as in the result of the computation of *TMM*, the *OMM* of shot is represented as a *two dimensional matrix*. The *Object Motion Matrix (OMM)* for a shot *A* with *n* frames is computed using the following algorithm.

- **Step.1:** Two empty two dimensional matrices (having size (10×8) same as that of motion vector matrix) are created and their elements are all initialized with zeros.
- **Step.2:** These two matrices (*OMMX* for x-axis and *OMMY* for y-axis) form the two components of *Object Motion Matrix (OMM)*, and can be seen as follows.

$$OMMX_A = \begin{pmatrix} bx_{1,1} & bx_{1,2} & bx_{1,3} & \dots & bx_{1,10} \\ bx_{2,1} & bx_{2,2} & bx_{2,3} & \dots & bx_{2,10} \\ \dots & \dots & \dots & \dots & \dots \\ bx_{8,1} & bx_{8,2} & bx_{8,3} & \dots & bx_{8,10} \end{pmatrix} \quad (9)$$

$$OMMY_A = \begin{pmatrix} by_{1,1} & by_{1,2} & by_{1,3} & \dots & by_{1,10} \\ by_{2,1} & by_{2,2} & by_{2,3} & \dots & by_{2,10} \\ \dots & \dots & \dots & \dots & \dots \\ by_{8,1} & by_{8,2} & by_{8,3} & \dots & by_{8,10} \end{pmatrix} \quad (10)$$

where $bx_{i,j}$ and $by_{i,j}$ are the horizontal and the vertical object motion matrix components of the shot for the macroblocks in the same position, and can be computed as follows.

$$bx_{i,j} = \sum_{k=0}^{n-1} (mvx_{i,j}^k - \bar{p} - \cos \bar{r}),$$

$$by_{i,j} = \sum_{k=0}^{n-1} (mvy_{i,j}^k - \bar{t} - \sin \bar{r}) \quad (11)$$

The above equations compute the object motion by compensating camera motions from overall motion. The ‘zoom’ camera operation is not included for the computation since it is considered as a special case.

We compute *CMM_A* (Camera Motion Matrix) by subtracting *OMM_A* from *TMM_A* as follows (see, Equations (12) and (13)) By doing this, we can distinguish the motions caused by objects movements and camera operations.

$$CMMX_A = \begin{pmatrix} ax_{1,1} - bx_{1,1} & ax_{1,2} - bx_{1,2} & \dots & ax_{1,10} - bx_{1,10} \\ ax_{2,1} - bx_{2,1} & ax_{2,2} - bx_{2,2} & \dots & ax_{2,10} - bx_{2,10} \\ \dots & \dots & \dots & \dots \\ ax_{8,1} - bx_{8,1} & ax_{8,2} - bx_{8,2} & \dots & ax_{8,10} - bx_{8,10} \end{pmatrix} \quad (12)$$

$$CMMY_A =$$

$$\begin{pmatrix} ay_{1,1} - by_{1,1} & ay_{1,2} - by_{1,2} & \dots & ay_{1,10} - by_{1,10} \\ ay_{2,1} - by_{2,1} & ay_{2,2} - by_{2,2} & \dots & ay_{2,10} - by_{2,10} \\ \dots & \dots & \dots & \dots \\ ay_{8,1} - by_{8,1} & ay_{8,2} - by_{8,2} & \dots & ay_{8,10} - by_{8,10} \end{pmatrix} \quad (13)$$

From the above three matrices, we compute three different motion features (*TM*, *OM* and *CM*) using the following simple formulas.

$$TMX_A = \sum_{i=1}^8 \sum_{j=1}^{10} ax_{ij}, \quad OMX_A = \sum_{i=1}^8 \sum_{j=1}^{10} bx_{ij},$$

$$CMX_A = \sum_{i=1}^8 \sum_{j=1}^{10} (ax_{ij} - bx_{ij}) \quad (14)$$

$$TMY_A = \sum_{i=1}^8 \sum_{j=1}^{10} ay_{ij}, \quad OMY_A = \sum_{i=1}^8 \sum_{j=1}^{10} by_{ij},$$

$$CMY_A = \sum_{i=1}^8 \sum_{j=1}^{10} (ay_{ij} - by_{ij}) \quad (15)$$

Thus by using these computations of *TMX*, *TMY*, *OMX*, *OMY*, *CMX* and *CMY*, we can also get the directions of motion by taking into consideration the signs of those features. These six values are the motions features indicating the amounts and the directions of various motions in a shot.

3 Experimental Results

Our experiments in this paper were designed to assess the effectiveness of the proposed algorithms to detect camera motions and to measure total, object and camera motions. Our video clips were originally digitized in MPEG format at 30 frames/second. Their resolution is 160×120 pixels. Our test set has 192 shots which consist of total 64,350 frames, and consists of documentaries and test videos made by us using two people as objects. They have various camera motions and object movements. As shown in Table 1, we classify these shots into 13 categories on basis of the camera and the object motions present in the shots. We discuss our performance results as follows.

Since the general concepts ‘recall’ and ‘precision’ are not applicable, we measure the performance of our camera motion detection technique using the *correct detection rate (CDR)* which is in the interval of [0,1], and defined as follows.

$$\text{Correct Detection Rate} = \frac{\text{Number of shots correctly detected}}{\text{Total Number of relevant shots in database}}$$

The experimental results for camera motion detection are summarized in Table 2. The descriptions of the first row

Category No.	Category Description
1	No Camera Motion and No Object Motion
2	No Camera Motion and Object Moving Left
3	No Camera Motion and Object Moving Right
4	Camera Pan Left and No Object Motion
5	Camera Pan Left and Object Moving Left
6	Camera Pan Right and No Object Motion
7	Camera Pan Right and Object Moving Right
8	Camera Tilt Up and No Object Motion
9	Camera Tilt Up and Object Moving Up
10	Camera Tilt Down and NO Object Motion
11	Camera Tilt Down and Object Moving Down
12	Camera Zoom out
13	Camera Zoom in

Table 1. Categories of Video Shots present in Database

(Category) in the table can be found in Table 1. The second row is showing the total number of shots for each category. The third and the fourth rows indicate the number of correctly detected shots for each category in term of camera motions, and their ratios (CDRs) respectively. As seen in this table, the CDRs for the shots in the categories such as 1, 4, 6, 8 and 10 in which there is no object motions are perfect (100%). The CDRs for the shots in the other categories such as 2, 3, 5, 7, 9, 11, 12 and 13 are a little less than perfect. We found that the incorrectly detected shots have usually very large object(s). We will investigate how to distinguish these incorrect cases. The overall average of CDRs is acceptable (84%).

The fifth through the tenth rows in Table 2 shows the experimental results of measuring the amount and the direction for the various motions such as camera motion (CM), object motion (OM) and total motion (TM). The fifth and the sixth columns show the amounts and the directions of camera motions (CMs). The numbers show the exact camera motion contents described in Table 1. For example, the values of $CMXs$ and $CMYs$ for the categories 1, 2, and 3 are negligible, which indicates that there is no camera motion in these categories. The values of $CMXs$ and $CMYs$ for the other categories (4 through 11) show the amount and the directions of the camera motions, pan left (Categories 4 and 5), pan right (Categories 6 and 7), tilt up (Categories 8 and 9), and tilt down (Categories 10 and 11). The seventh

and the eighth columns show the amounts and the directions of object motions (OMs). Similar as $CMXs$ and $CMYs$, $OMXs$ and $OMYs$ show the amounts and the directions of object motions. For example, the values of $OMXs$ and $OMYs$ for the categories 1, 4, 6, 8, and 10 are negligible, which indicates that there is no object motion in these categories. The values of $OMXs$ and $OMYs$ for the other categories (2, 3, 5, 7, 9, and 11) show the amount and the directions of the object motions, moving left, right, up and down. The total motion (TM) which is combination of the object and the camera motions can be seen in the ninth and the tenth rows. The values of the categories for 12 and 13 indicate that our technique can also distinguish two camera motions which are zoom out and in.

4 Concluding Remarks

For automatic and accurate content-based analyzing and indexing of digital videos, we need to classify motions whether they come from camera or object(s) movements. In this paper, we first, measure overall motion and detect camera motion changes in a shot, then distinguish object and camera motions from overall motions using the proposed techniques. The main contributions of the proposed techniques can be summarized as follows. The techniques to compute TM or OM are very cost-effective because they use accumulation of quantized pixel differences, and expensive computation (i.e., optical flow) is not necessary. The matrices representing TM and OM are showing not only the amounts of motions but also the exact locations of motions without any overhead. Therefore, we can get more accurate and richer information of motion contents of shot in an inexpensive way. Because we extract and use various motion features from shots, numerous ranges of hierarchical clustering can be provided for shots, which are essential for content analyzing, indexing, browsing, and querying of video data. We will investigate this hierarchical clustering in near future. Although our experimental data set has very limited number of shots, the results are showing that the proposed techniques are effective in capturing different motions and comparing shots based on them. We will perform further experiments in the future to study the effectiveness of the proposed techniques when they are applied to various kinds of different videos.

References

- [1] R. Jain. Content-based multimedia information management. In *Proc. of 14th Int'l Conf. on Data Engineering*, pages 252–253, Orlando, Florida, February 1998.
- [2] P. Aigrain, H. Zhang, and D. Petkovic. Content-based representation and retrieval of visual media : A state-of-the art review. *Multimedia Tools and Applications*, 3:179–202, 1996.

Category	1	2	3	4	5	6	7	8	9	10	11	12	13	Total
Total Shots	12	17	15	15	15	18	19	17	6	16	6	18	18	192
Correctly Detected Shots	12	13	10	15	13	18	14	17	3	16	4	14	13	162
CDR	1.0	0.77	0.67	1.0	0.87	1.0	0.74	1.0	0.5	1.0	0.67	0.78	0.72	0.84
CMX	0.01	0.01	0.01	-1.59	-0.89	1.12	0.38	-0.00	0.06	0.07	0.05	0.45	-0.35	
CMY	-0.01	0.00	0.00	0.02	0.02	0.01	0.01	0.41	0.53	-0.75	-0.7	0.40	-0.37	
OMX	0.01	-0.57	0.62	-0.05	-0.68	0.03	0.77	0.02	0.08	-0.00	0.00	-0.01	0.01	
OMY	-0.01	0.02	0.01	0.01	-0.00	0.01	-0.06	-0.01	0.70	-0.05	-0.50	-0.01	0.01	
TMX	0.01	-0.58	0.62	-1.64	-1.57	1.15	1.15	0.02	0.08	0.09	0.04	0.45	-0.36	
TMY	-0.01	0.02	0.02	0.03	0.02	0.02	0.02	0.40	1.23	-0.8	-1.2	0.40	-0.38	

Table 2. Performances for Different Categories

- [3] J. G. Kim, H. Chang, J. W. Kim, and H. Kim. Efficient camera motion characterization for mpeg video indexing. In *Proc. of 2000 IEEE International Conference on Multimedia and Expo., 2000*, pages 1171–1174, New York, NY, July 2000.
- [4] P. Bouthemy, M. Gelgon, and F. Ganansia. A unified approach to shot change detection and camera motion characterization. *IEEE Trans. on Circuits and Systems for Video Technology*, 7(10):1030–1044, 1999.
- [5] R. Zaibi, A.E. Enis, and Y. Yardimci. Small moving object detection in video sequences. In *Proc. of ICASSP 2000*, pages 2071–2074, 2000.
- [6] J.M. Odobez and P. Bouthemy. Detection of multiple moving objects using multiscale mrf with camera motion compensation. In *Proc. of Int'l Conf. on Image Processing*, pages 257–261, 1994.
- [7] S. Pfeiffer, R. Lienhart, S. Fischer, and W. Effelsberg. Abstracting digital movies automatically. *Journal of Visual Communication and Image Representation*, 7(4):345–353, December 1996.
- [8] W. Wolf. Key frame selection by motion analysis. In *Proc. of ICASSP 96*, pages 1128–1231, 1996.
- [9] N. Vasconcelos and A. Lippman. Towards semantically meaningful features spaces for the characterization of video content. In *Proc. of ICIP 97*, 1997.
- [10] K. Peker, A. Divakaran, and T. Papathomas. Automatic measurement of intensity of motion activity of video segments. In *Proc. of SPIE conf. on Storage and Retrieval for Media Databases 2001*, pages 341–351, San Jose, CA, Jan. 2001.
- [11] C.W. Ngo, T.C. Pong, and H.J. Zhang. On clustering and retrieval of video shots. In *Proc. of ACM Multimedia 2001*, pages 51–60, Ottawa, Canada, Oct. 2001.
- [12] G. Kuhne, S. Richter, and M. Beier. Motion-based segmentation and contour-based classification of video objects. In *Proc. of ACM Multimedia 2001*, pages 41–50, Ottawa, Canada, Oct. 2001.
- [13] D.A. Forsyth. Benchmarks for storage and retrieval in multimedia databases. In *Proc. of SPIE conf. on Storage and Retrieval for Media Databases 2002*, pages 240–247, San Jose, CA, Jan. 2002.
- [14] S.H. Kim and R.H. Park. An efficient video sequence matching using the cauchy function and the modified hausdorff distance. In *Proc. of SPIE conf. on Storage and Retrieval for Media Databases 2002*, pages 232–239, San Jose, CA, Jan. 2002.
- [15] S. Park, J. Cho, and K. Hyun. Indexing technique for similarity matching in large video databases. In *Proc. of SPIE conf. on Storage and Retrieval for Media Databases 2002*, pages 214–222, San Jose, CA, Jan. 2002.
- [16] E. Francois and P. Bouthemy. Derivation of qualitative information in motion analysis. *Image Vis. Computing*, 8(4):279–287, 1990.