A GENERIC APPROACH FOR MOTION-BASED VIDEO PARSING

Martin Haller, Andreas Krutz, and Thomas Sikora

Communication Systems Group, Technical University of Berlin
Sekr. EN-1, Einsteinufer 17, 10587 Berlin, Germany
Email: {haller,krutz,sikora}@nue.tu-berlin.de
Web: `http://www.nue.tu-berlin.de/`

ABSTRACT

Motion-based video parsing methods segment video streams according to changes of camera motion types. They rely usually on compressed video streams, where motion vector fields are provided. Camera parameters can be derived from these motion vectors. There are a number of relevant video codecs where no motion information is included. For such video streams, camera parameters have to be estimated using a frame-to-frame image registration method. In our approach, we provide both techniques to estimate camera parameters. Enhanced feature extraction algorithms take advantage of estimated parameters. For classification, the method uses three multi-class Support Vector Machines (M-SVMs) to independently detect pan, tilt, and zoom camera motion as well as the direction of motion. Experimental results show a promising performance of our generic approach with test video streams from the TRECVid 2005 BBC rushes video corpus.

1. INTRODUCTION

Parsing video streams leads to structured video content. This is a requirement for content-based video analysis, especially in the application of video indexing and summarization.

Shot boundary detection techniques [1, 2] structure ideally video sequences with respect to transitions between groups of continuously recorded image frames. In addition, temporal segmentation of video sequences according to the appearance of camera motion types such as pan, tilt, zoom, rotation, dolly, and boom result in a motion-based temporal structuring on sub-shot level. The results of such a motion-based video parsing method are not only useful for extraction of sub-shot keyframes from edited video. They are even more important for unedited or barely edited video sequences like DV camcorder recordings, home videos, or rushes. Long shot durations are predominant for these video types. Further video analysis techniques that rely only on shot keyframes provide an insufficient temporal resolution of the visual content. To emphasize further the importance of camera motion, we would like to refer to [3] where Smeaton presented 8 challenges for video analysis, indexing, and retrieval and pointed out that information on camera motion amongst others is necessary to develop further video retrieval techniques that take full advantage of the temporal dimension.

Motion-compensated prediction is often used in video codecs for transmission and storage. Therefore, many methods for camera motion estimation and characterization use directly the motion vectors of block-based, motion-compensated prediction video coding techniques such as standardized by MPEG [4–9]. However, many consumer digital video devices as well as professional video recording and editing systems use intra-frame coding standards such as DV, DVCPRO, DVCAM, Motion JPEG, and Motion JPEG 2000. Motion vector fields (MVFs) are not available in such video streams. Thus, the motion has to be estimated before any successive motion-based analysis can be performed. Compressed domain methods depending on MVFs do not work for such intra-coded video streams.

Zhang et al. [1] used motion vectors for a threshold-based detection of panning and zooming. MVFs were determined with block-matching motion estimation or were extracted from the compressed domain [4]. In [5], Smith et al. detected different camera motion types using a set of rules based on the affine motion parameters and on the average flow. A least-squares error method was used to estimate the affine motion parameters from MVFs that were extracted from the MPEG compressed domain. Bouthemy et al. [10] used the affine motion model along with a robust M-estimator to estimate the parameter according to the dominant motion. The proposed method identifies significant camera motion with log-likelihood ratio tests on different parameters of the affine motion model. Then, the method segments video sequences according to the detected types of camera motion. Ngo et al. [11] made use of Hierarchical Hidden Markov Models to detect characteristic camera motion for stock, outtake, and shaky video sequences by using features derived from affine motion model Duan et al. [9] proposed a method for camera motion characterization using a non-parametric motion vector field representation based on mean shift filtering and mode-based feature extraction. The method obtains the motion characterization by using soft-margin Support Vector Machines (SVM) with a hierarchical taxonomy for the different types of camera motion. Experimental results were reported for 23191 MVFs. The MVFs were extracted from the MPEG-7 video dataset for motion activity analysis.

In this paper, we propose a generic approach for motion-based video parsing that works for arbitrary coded video streams. The estimation of affine motion parameters uses MVFs as input for compressed video streams with motion-compensated prediction. For all other video codecs, the estimation of affine motion parameter uses a frame-to-frame image registration algorithm. After the estimation of global motion parameters, the parameters are factorized using the Singular Value Decomposition (SVD) into scaling, rotation, and skewing components. For each camera motion type, suitable features for classification are extracted from these components and the translational parameters. Our approach uses three multi-class SVMs (M-SVMs) to recognize the camera motion types between successive image frames. Therefore, the camera motion types pan, tilt, and zoom can be detected independently. Each M-SVM distinguishes between the occurrence and the direction of each motion type. We evaluated our method with selected rushes videos from the TRECVid 2005 BBC rushes video corpus [12]. An overview of the approach is shown in Fig. 1.

This paper is organized as follows. The estimation of motion parameters is described in Section 2. Section 3 describes the used features, classification methods, and how the results are combined to form camera motion segments. Experimental results are presented in Section 4, which is followed by the conclusions and further work.

2. CAMERA PARAMETER ESTIMATION

Our approach for motion change detection and camera motion characterization relies on an accurate estimation of the physical camera motion parameters. To determine those parameters, e.g. translational parameters, camera angles, and zoom factors, we have to determine the motion model parameters frame by frame as the first

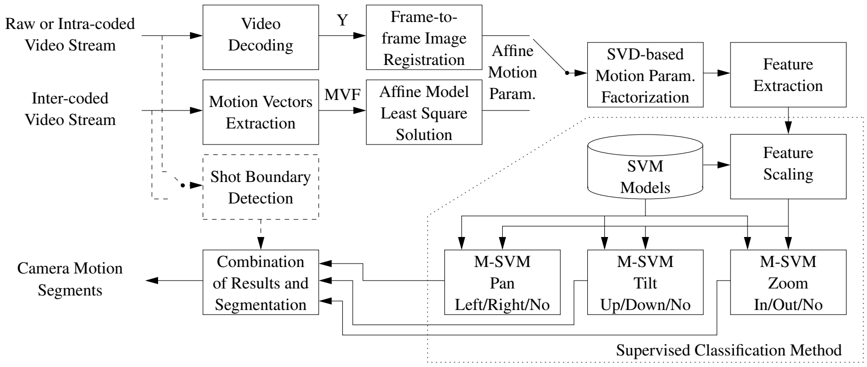

Raw or Intra-coded Video Stream
Inter-coded Video Stream
Video Decoding
Y
Frame-to-frame Image Registration
Motion Vectors Extraction
MVF
Affine Model Least Square Solution
Affine Motion Param.
SVD-based Motion Param. Factorization
Feature Extraction
SVM Models
Feature Scaling
Shot Boundary Detection
Camera Motion Segments
Combination of Results and Segmentation
M-SVM Pan Left/Right/No
M-SVM Tilt Up/Down/No
M-SVM Zoom In/Out/No
Supervised Classification Method

Figure 1: Overview of the proposed motion-based video parsing system

step. In this work, the affine motion model is used. Two methods for the frame-to-frame affine motion parameters estimation are used in our approach depending on the input video stream. If the input video stream contains MVFs, the translational motion parameters of macro-blocks can be used for computation of the global motion model parameters [8]. However, there are also video streams where motion parameters are not provided. Hence, the parameters for the frame-to-frame affine transformation have to be computed initially. We use an image registration algorithm [13]. Both methods and the extraction of the factorized camera parameters are described more in detail below.

2.1 Affine Parameters from MVFs

MVFs can be used to calculate the 6-parameter affine motion model. The approach used here was proposed by [7, 8] for the 4-parameter motion model. The relation between the affine motion parameters and the motion vectors of two successive image frames can be formulated as

$$\mathbf{v} = \mathbf{C} \cdot \mathbf{m}, \tag{1}$$

where

$$\mathbf{v} = \begin{pmatrix} v_x^1 + x^1 & v_y^1 + y^1 & \dots & v_x^N + x^N & v_y^N + y^N \end{pmatrix}^{\mathrm{T}} \tag{2}$$

contains the N motion vectors (v_x^n, v_y^n) and the coordinates x and y,

$$\mathbf{C} = \begin{pmatrix} x^1 & y^1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & x^1 & y^1 & 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ x^N & y^N & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & x^N & y^N & 1 \end{pmatrix} \tag{3}$$

includes the coordinates x and y, and

$$\mathbf{m} = \begin{pmatrix} m_1 & m_2 & m_3 & m_4 & m_5 & m_6 \end{pmatrix}^{\mathrm{T}} \tag{4}$$

contains the affine motion parameters. Equation (1) can be solved with respect to $\mathbf{m}$ using the pseudo inverse matrix for $\mathbf{C}$. Furthermore, the accuracy of the affine motion parameters achieved from a vector field can be influenced by several outliers of the motion vectors. To prevent this, a robust M-estimator with its diagonal weighting matrix $\mathbf{W}$ is used within the computation of the affine motion parameters as shown in [8]. This leads to

$$\mathbf{m} = (\mathbf{C}^T \cdot \mathbf{C})^{-1} \cdot \mathbf{C}^T \cdot \mathbf{W} \cdot \mathbf{v}. \tag{5}$$

2.2 Affine Parameters from Frame-to-Frame Image Registration

If MVFs are not available, we have to estimate the global motion from image sequences in a pre-processing step. Since we are more interested in global motion parameters rather than the corresponding blocks from image frame to image frame, the affine motion parameters are determined directly with a very common frame-to-frame image registration approach inspired by [14]. Figure 2 shows the block diagram of the approach.

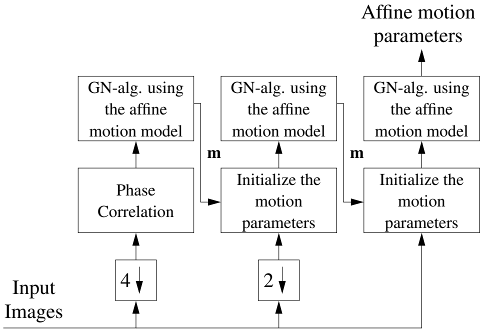

Affine motion parameters
GN-alg. using the affine motion model
GN-alg. using the affine motion model
GN-alg. using the affine motion model
m
m
Phase Correlation
Initialize the motion parameters
Initialize the motion parameters
4↓
2↓
Input Images

Figure 2: Global motion estimation algorithm (GN-alg. - Gauss-Newton algorithm)

The affine motion parameters are determined applying an energy minimization method. The Gauss-Newton gradient descent algorithm is used because of its very good performance if the start point is close to the minimum desired. Phase correlation is applied to ensure the initialization of the translational motion parameters as well as to decrease the computational complexity. An image pyramid is used to reduce essentially the computational costs. The phase correlation and gradient descent algorithm start on lower resolution versions of the input images. Afterwards, the obtained motion parameters initialize the Gauss-Newton algorithm at the upper stages until the original image size is reached. For downsampling, the low pass component of a wavelet decomposition is extracted. The affine motion parameters for the considered image pair are then computed and can be written as

$$\begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} m_1 & m_2 \\ m_4 & m_5 \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} m_3 \\ m_6 \end{pmatrix}, \tag{6}$$

where x and y are the coordinates of the original pixel and x' and y' are the coordinates of the corresponding pixel value of the frame to register.

2.3 Camera Parameter Factorization using SVD

Using the 6 affine motion parameters, the physical camera parameters can be calculated. The parameters m_3 and m_6 of (6) represent the translational shift of the pixels. Camera pan and tilt have a direct impact on these two parameters. For the zoom factor and the camera angles, the matrix

$$\mathbf{A} = \begin{pmatrix} m_1 & m_2 \\ m_4 & m_5 \end{pmatrix} \tag{7}$$

is considered. To obtain quantitative correct values for the effects of scaling, rotation, and skew, we are exactly interested in the factorization of the non-translational affine parameters. We assume that $\mathbf{A}$ is the result of the following product [15]

$$\mathbf{A} = \mathbf{R}_\phi \cdot \mathbf{R}_{-\theta} \cdot \mathbf{S} \cdot \mathbf{R}_\theta, \tag{8}$$

where the matrices $\mathbf{R}_\phi$, $\mathbf{R}_\theta$, and $\mathbf{S}$ contain the rotation angle ϕ (9), the skew angle θ (10), and the zoom factors s_x and s_y (11) related to the origin in the center of the camera lens .

$$\mathbf{R}_\phi = \begin{pmatrix} cos(\phi) & sin(\phi) \\ -sin(\phi) & cos(\phi) \end{pmatrix} \tag{9}$$

$$\mathbf{R}_\theta = \begin{pmatrix} cos(\theta) & sin(\theta) \\ -sin(\theta) & cos(\theta) \end{pmatrix} \tag{10}$$

$$\mathbf{S} = \begin{pmatrix} s_x & 0 \\ 0 & s_y \end{pmatrix} \tag{11}$$

To derive the angles ϕ, θ, and the zoom factors from the matrix $\mathbf{A}$, the SVD is used. SVD is applied to the matrix $\mathbf{A}$

$$\mathbf{A} = \mathbf{U} \cdot \mathbf{D} \cdot \mathbf{V}^\mathbf{T}. \tag{12}$$

Comparing equations (8) and (12) it can be seen that

$$\widehat{\mathbf{R}}_\theta = \mathbf{V}^\mathbf{T} \tag{13}$$

$$\widehat{\mathbf{S}} = \mathbf{D}. \tag{14}$$

The $\widehat{}$ sign indicates that these matrices are results of the factorization. Furthermore, the matrix $\mathbf{U}$ can be written as

$$\widehat{\mathbf{R}}_\phi \cdot \widehat{\mathbf{R}}_{-\theta} = \mathbf{U}. \tag{15}$$

The rotation matrix $\widehat{\mathbf{R}}_\phi$ can then be computed using $\widehat{\mathbf{R}}_\theta$ which is already determined.

$$\widehat{\mathbf{R}}_\phi = \widehat{\mathbf{R}}_\phi \cdot \underbrace{\widehat{\mathbf{R}}_{-\theta} \cdot \widehat{\mathbf{R}}_\theta}_{\mathbf{E}} \tag{16}$$

The inverse matrix of $\widehat{\mathbf{R}}_{-\theta}$ can be written as $\widehat{\mathbf{R}}_\theta$ due to the trigonometric functions. $\widehat{\mathbf{R}}_\theta$ is already known and multiplied with the matrix $\mathbf{U}$ results in the matrix $\widehat{\mathbf{R}}_\phi$. Thus, we obtain the rotation matrix with angle ϕ. The described factorization is valid only for the case $s_x > s_y$ due to the properties of the SVD. A proper case differentiation based on the comparison of the affine motion parameters m_1 and m_5 has to be applied during the computation of factorized camera parameters.

Based on these camera motion parameters, m_3 and m_6 for pan and tilt, s_x and s_y for zoom, ϕ for rotation relating to the center of the camera lens, and θ for skew, a scheme for a motion change detection algorithm is developed. Further components of the system are described in the next Sections.

3. FEATURE EXTRACTION, CLASSIFICATION, AND SEGMENTATION

The feature extraction stage determines suitable and useful features to support the classification of the different types of camera motion. For this, different sets of features determined from the factorized motion parameters are used for the camera motion categories pan, tilt, and zoom. Rotation as well as the skew parameters are not further considered in this work. After feature extraction, the classification stage uses pre-trained models to identify the type of camera motion between two successive image frames. Subsequently, classification results obtained from the three M-SVMs are combined to form an overall result. Camera motion segments are defined by their boundaries where the types of camera motion change. Recognized camera motion types are assigned to each respective segment.

3.1 Features for Pan and Tilt Camera Motion

All features extracted for pan and tilt camera motion rely only on the affine motion parameters $m_{3,l}$ and $m_{6,l}$, where l addresses the affine motion parameter for image frames l and $(l+1)$. Track and boom camera motion affects these parameters as well. With the complex normalized value

$$\underline{t}_l = \frac{m_{3,l}}{w} + j\frac{m_{6,l}}{h}, \tag{17}$$

the median $\phi_{\mathrm{t,med},l}$ of angle of translational motion and the medians $t_{\mathrm{x,med},l}$ as well as $t_{\mathrm{y,med},l}$ of $m_{3,l}$ and $m_{6,l}$ are computed with

$$\phi_{\mathrm{t,med},l} = \underset{s_l \le k \le e_l}{\mathrm{median}} \left(\arg\left(\underline{t}_k\right)\right) \tag{18}$$

$$t_{\mathrm{x,med},l} = \underset{s_l \le k \le e_l}{\mathrm{median}} \left(m_{3,k}\right) \tag{19}$$

$$t_{\mathrm{y,med},l} = \underset{s_l \le k \le e_l}{\mathrm{median}} \left(m_{6,k}\right), \tag{20}$$

where w and h are the image width and height and s_l as well as e_l are given as

$$s_l = l - W_{\mathrm{med}} + 1 \quad ; \quad e_l = l + W_{\mathrm{med}}.$$

The median filtering is necessary due to possible outliers introduced by the global motion parameter estimation methods. The used windowed median filter has a length of $W_{\mathrm{med}} = \lfloor R_f/2 \rfloor$ with R_f as frame rate per second of the video sequence.

To obtain more robust features for the direction of translational motion, a short-time translational angle histogram based on values for $\phi_{\mathrm{t,med},l}$ is determined. The scheme used for quantization of angles is shown in Fig. 3. The derived rates $R_{\mathrm{TAHPL},l}$, $R_{\mathrm{TAHPR},l}$, $R_{\mathrm{TAHTU},l}$, and $R_{\mathrm{TAHTD},l}$ represent the occurrence of angles for pan left/right and tilt up/down in the respective range of angles normalized to the window length W for the histogram computation. The used overlap of windows is extensive for a proper temporal resolution.

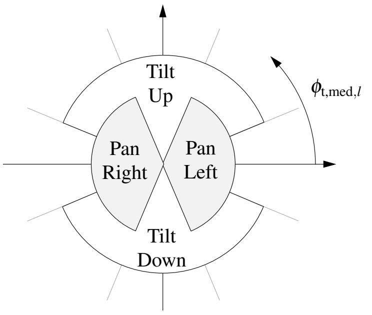


Figure 3: Quantization scheme for the translational motion angle histogram (TAH)

Furthermore, zero-crossing rates (ZCRs) for horizontal and vertical translational motion parameters are computed with

$$Z_{\mathrm{x},l} = \frac{1}{2W}\sum_{i=s_l}^{e_l} \left|\mathrm{sgn}\left(m_{3,i}\right) - \mathrm{sgn}\left(m_{3,i-1}\right)\right| \tag{21}$$

$$Z_{\mathrm{y},l} = \frac{1}{2W}\sum_{i=s_l}^{e_l} \left|\mathrm{sgn}\left(m_{6,i}\right) - \mathrm{sgn}\left(m_{6,i-1}\right)\right|. \tag{22}$$

These two features capture the reliability of intended translational camera motion. The complete feature vectors for classification of pan and tilt are as follows

$$\mathbf{x}_{\mathrm{pan},l} = \begin{pmatrix} t_{\mathrm{x,med},l} & R_{\mathrm{TAHPL},l} & R_{\mathrm{TAHPR},l} & Z_{\mathrm{x},l} \end{pmatrix}^\mathrm{T} \tag{23}$$

$$\mathbf{x}_{\mathrm{tilt},l} = \begin{pmatrix} t_{\mathrm{y,med},l} & R_{\mathrm{TAHTU},l} & R_{\mathrm{TAHTD},l} & Z_{\mathrm{y},l} \end{pmatrix}^\mathrm{T}. \tag{24}$$

3.2 Features for Camera Zoom

Zooming is a change of the focal length of camera rather than motion. However, also the dolly camera motion can result in a change of the multiplicative scaling motion parameters $s_{x,l}$ and $s_{y,l}$ derived from factorization of affine motion parameters. The affine motion model cannot distinguish between these two different effects. Therefore, zoom and dolly are both referred to as zoom in this work. The following features are used to describe the zoom effect. First, the centered and normalized joint zoom factor

$$z_l \quad = \quad \sqrt{s_{x,l}^2 + s_{y,l}^2} - 1 \tag{25}$$

is computed. The zoom factor z_l is greater than 0 for zoom in, less than 0 for zoom out, and equals 0 for no zoom. To have a robust feature, a median

$$z_{\text{med},l} \quad = \quad \underset{s_l \le k \le e_l}{\text{median}} \left(z_k \right). \tag{26}$$

is determined. The ZCR $Z_{z,l}$ for z_l

$$Z_{z,l} \quad = \quad \frac{1}{2W} \sum_{i=s_l}^{e_l} \left| \text{sgn}\left(z_i\right) - \text{sgn}\left(z_{i-1}\right) \right| \tag{27}$$

is computed to have also some reliability measure for intended camera zoom. The zoom feature vector can then be written as

$$\mathbf{x}_{\text{zoom},l} \quad = \quad \left(\begin{array}{cc} z_{\text{med},l} & Z_{z,l} \end{array} \right)^{\mathrm{T}}. \tag{28}$$

3.3 Classification

The approach of this work uses a supervised classification method to recognize the different types of camera motion as shown in Fig. 1. After scaling each dimension of the features described in the previous section, three multi-class support vector machines (M-SVMs) [16] are used independently to detect pan, tilt, and zoom as well as the direction. For instance, the classes defined for the camera motion type pan are pan left, pan right, and no pan. Several schemes are available to classify multiple classes with original binary SVMs. The one-against-one approach is used in this work.

3.4 Combination of Results and Segmentation

Each of the three M-SVMs provides a result with three possible states as exemplary described above for panning. The classifiers independently determine a result for pan, tilt, and zoom. Camera motion types pan left/right, tilt up/down, zoom in/out, and no camera motion can occur alone or in combinations between pan, tilt, and zoom. Changes between such combinations are identified as boundaries of segments with the same type or types of camera motion. This leads to a motion-based temporal segmentation of the analyzed video sequence on sub-shot level. Furthermore, separately detected shot boundaries can be included in the overall segmentation result.

4. EXPERIMENTAL RESULTS

We used selected videos from the development and test set of the TRECVid 2005 BBC rushes video corpus [12] for the evaluation of our approach. 19 training videos (37145 frames, 63 shot boundaries) from the development set and 13 test videos (16547 frames, 24 shot boundaries) from the test set were selected. Thus, we have approximately 70 % training data and 30 % available for evaluation purposes. The whole video data set has a total duration of about 35 minutes. The ground truth was created manually in two passes to reduce the number of possible annotation errors. The motion type of segments with shaky camera movements were labeled as undefined motion and ingored during training and evaluation phases. We noticed by comparison of the occurrence frequency of the different motion types that camera panning occurs more often and with higher motion intensity than camera tilting or camera zooming. Camera rotation is not considered in this evaluation due to the small number of camera rotations included in training and test data. Table 1 shows the number of segments and frame-to-frame motion parameter sets for all camera motion types.

	Pan		Tilt		Zoom		Rotation				
	PL	PR	TU	TD	ZI	ZO	RC	RA	UM	NM	Shots
TR-SEG	47	62	45	47	23	20	4	3	111	139	63
TR-AMP	3203	2865	1196	1456	431	706	207	95	7456	21973	–
TS-SEG	52	49	26	51	16	7	1	1	10	82	24
TS-AMP	1751	1939	501	2427	219	230	15	15	935	10389	–

Table 1: Number of camera motion segments (SEG) and of affine motion parameter sets (AMP) for the training (TR) and test (TS) database (PL/PR - Pan left/right, TU/TD - Tilt up/down, ZI/ZO - Zoom in/out, RC/RA - Rotation clockwise/anticlockwise, UM - Undefined motion, NM - No motion)

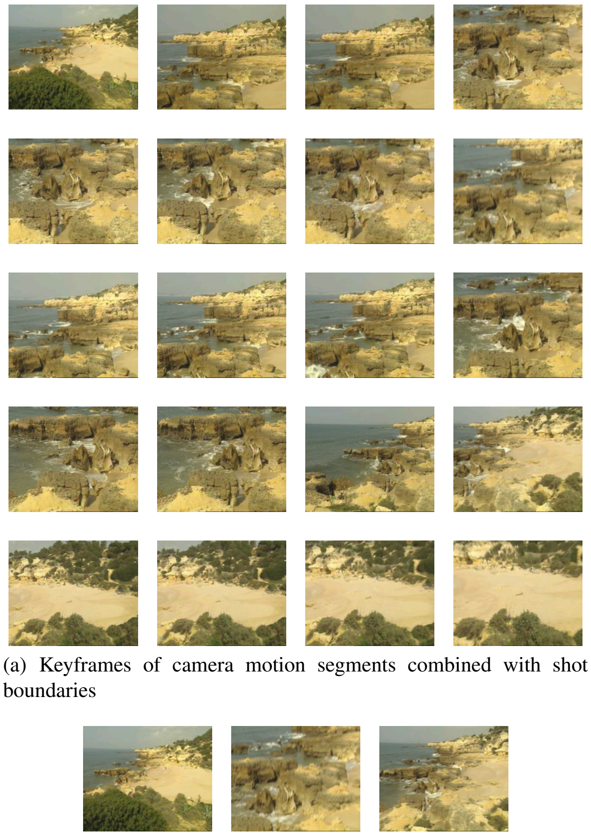
	Pan			Tilt			Zoom			
	PN	PL	PR	TT	TU	TD	ZM	ZI	ZO	NM
Frame-to-frame image registration										
M1 P	96.90	96.22	97.52	87.94	90.45	87.72	78.72	89.02	72.20	99.75
M1 R	83.72	84.36	83.14	73.15	35.93	80.82	74.00	66.36	81.30	99.83
M1 F_1	89.83	89.90	89.76	79.87	51.43	84.13	76.29	76.04	76.48	99.79
Motion vector field least-squared solution										
M2 P	63.77	88.39	53.60	19.24	50.81	18.55	01.82	05.95	00.33	99.92
M2 R	68.74	58.68	77.84	75.61	24.95	86.05	37.78	66.82	10.00	84.80
M2 F_1	66.16	70.53	63.49	30.67	33.47	30.52	03.47	10.93	00.64	91.74

Table 2: Frame-to-frame evaluation results for correct recognition of camera motion such as pan (PN) left (PL)/right (PR), tilt (TT) up (TU)/down (TD), zoom (ZM) in (ZI)/out (ZO), and no motion (NM) using frame-to-frame image registration (M1) and motion vector field least-squared solution (M2) in %

For this first evaluation of our approach, we used the one-against-one scheme for multi-class SVMs with linear kernels. The soft-margin parameter C was determined by 5-fold cross-validation grid-search in the range of $[0.5, 1, 5, 10, 20, 50, 100, 200]$. Precision P, recall R, and F_1-measure were used for evaluation of the results on frame-to-frame camera motion level and segment level. First, the camera motion was evaluated between successive image frames. Table 2 shows the results for using the frame-to-frame image registration method and the MVF least-squared solution as affine parameter estimation method for the feature extraction stage during the test phase. In both cases, the SVM models were used that were trained with features based on motion parameters estimated by the frame-to-frame image registration algorithm. To obtain the most correct motion vector fields, we determined the motion vectors using an exhaustive block-matching algorithm (full search) instead of using directly the motion vector fields from the MPEG-1 video files. Second, the results for motion change boundaries and resulting camera motion segments were examined. For this, camera motion segments with duration less than 6 frames (about a quarter second) were removed and a tolerance of 2 seconds was introduced for evaluation of the temporal accuracy of segment boundaries. For the second evaluation, we further distinguished between segments with at least one correct detected motion type and segments where

	M1		M2	
	Min1	All	Min1	All
P	98.55	75.12	67.53	21.10
R	97.14	83.71	94.86	46.57
F_1	97.84	79.19	78.90	29.05

Table 3: Evaluation results for correct detection of at least 1 camera motion type and correct recognition of all camera motion types using frame-to-frame image registration (M1) and motion vector field least-squared solution (M2) in %



(a) Keyframes of camera motion segments combined with shot boundaries

(b) Keyframes from shots

Figure 4: Examples of (a) keyframe images extracted from camera motion segments and (b) from shots

all motion types were correctly recognized. Table 3 lists the obtained results.

The estimation of the affine global motion parameters using frame-to-frame image registration leads to good results for the evaluation on a frame-to-frame basis as well as with segments. In particular, these results show that camera characterization can be performed using parametrical motion models even for complex camera movements included in the rushes videos. However, the use of affine parameters estimated from motion vectors results in poor information retrieval measures despite the application of an M-estimator. The detection especially fails for camera tilt and zoom. Here, object motion has a substantial influence on the estimated global parameters in contrast to the frame-to-frame image registration method.

Figure 4 shows keyframe images extracted from segments formed by motion change boundaries in comparison to shot keyframe images. It is obvious that the shot keyframes do not include a representative image for all camera views. Therefore some visual content could not be analyzed further with methods relying only on these shot keyframes.

5. CONCLUSIONS AND FURTHER WORK

A generic approach for motion-based video parsing for arbitrary video streams has been presented. Promising results were achieved in an experimental evaluation for global motion estimation using a frame-to-frame image registration method. Further work will examine additional methods for a more robust estimation of global motion parameters from motion vectors. Furthermore, additional features, several classification methods, and the influence of sliding windows on the temporal accuracy will be examined. In addition, the approach can be extended that shot boundaries can be detected by the RMS value of motion-compensated error images as well as the use of higher-order motion models.

Acknowledgment

This research was supported by the European Commission under contract FP6-027026-K-SPACE.

BBC 2005 Rushes video is copyrighted. The BBC 2005 Rushes video used in this work is provided for research purposes by the BBC through the TREC Information Retrieval Research Collection.

REFERENCES

[1] H.-J. Zhang, A. Kankanhalli, and S. W. Smoliar, “Automatic partitioning of full-motion video,” *Multimedia Systems*, vol. 1, no. 1, pp. 10–28, 1993.

[2] R. Brunelli, O. Mich, and C. M. Modena, “A survey on the automatic indexing of video data,” *J. Vis. Comm. Image Represent.*, vol. 10, no. 2, pp. 78–112, 1999.

[3] A. F. Smeaton, “Techniques used and open challenges to the analysis, indexing and retrieval of digital video,” *Information Systems*, vol. 32, pp. 545–559, June 2007.

[4] H.-J. Zhang, C. Low, and S. Smoliar, “Video parsing and browsing using compressed data,” *Multimedia Tools Appl.*, vol. 1, no. 1, pp. 89–111, 1995.

[5] M. A. Smith and T. Kanade, “Video skimming and characterization through the combination of image and language understanding techniques,” in *IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recogn. (CVPR)*, pp. 775–781, 1997.

[6] S. M. Bhandarkar and A. A. Khombhadia, “Motion-based parsing of compressed video,” in *Int. Workshop on Multi-Media Database Management Systems (IW-MMDBMS)*, pp. 80–87, 1998.

[7] R. Wang and T. S. Huang, “Fast camera motion analysis in MPEG domain,” in *IEEE Int. Conf. on Image Processing (ICIP)*, vol. 3, pp. 691–694, 1999.

[8] A. Smolic, M. Hoeynck, and J.-R. Ohm, “Low-complexity global motion estimation from P-frame motion vectors for MPEG-7 applications,” in *IEEE Int. Conf. on Image Processing (ICIP)*, vol. 2, pp. 271–274, 2000.

[9] L.-Y. Duan, J. S. Jin, Q. Tian, and C.-S. Xu, “Nonparametric motion characterization for robust classification of camera motion patterns,” *IEEE Trans. Multimedia*, vol. 8, no. 2, pp. 323–340, 2006.

[10] P. Bouthemy, M. Gelgon, and F. Ganansia, “A unified approach to shot change detection and camera motion characterization,” *IEEE Trans. Circuits Syst. Video Technol.*, vol. 9, pp. 1030–1044, 1999.

[11] C.-W. Ngo, Z. Pan, and X. Wei, “Hierarchical hidden Markov model for rushes structuring and indexing,” in *Int. Conf. on Image and Video Retrieval (CIVR)*, pp. 241–250, 2006.

[12] P. Over, T. Ianeva, W. Kraaij, and A. F. Smeaton, “TRECVID 2005 - an overview.” `http://trecvid.nist.gov/`.

[13] A. Krutz, M. Frater, M. Kunter, and T. Sikora, “Windowed image registration for robust mosaicing of scenes with large background occlusions,” in *IEEE Int. Conf. on Image Processing (ICIP)*, 2006.

[14] F. Dufaux and J. Konrad, “Efficient, robust, and fast global motion estimation for video coding,” *IEEE Trans. Image Processing*, vol. 9, no. 3, pp. 497–501, 2000.

[15] R. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision.* Cambridge University Press, 2nd ed., 2004.

[16] J. Weston and C. Watkins, “Multi-class support vector machines,” Technical Report CSD-TR-98-04, Dept. of Computer Science, Royal Holloway, Univ. of London, 1998.