# AUTOMATIC VIDEO STRUCTURING FOR MULTIMEDIA MESSAGING

*Siripong Treetasanatavorn*[1], *Uwe Rauschenbach*[1], *Jörg Heuer*[1], *and André Kaup*[2]

[1]Siemens Corporate Technology
Information and Communications CT IC 2
Otto-Hahn-Ring 6, D-81730 Munich, Germany
E-mail:{siripong.treetasanatavorn,joerg.heuer,
uwe.rauschenbach}@mchp.siemens.de

[2]University of Erlangen-Nuremberg
Chair of Multimedia Communications
and Signal Processing, Cauerstraße 7
D-91058 Erlangen, Germany
E-mail: kaup@lnt.de

## ABSTRACT

This paper proposes a low-complexity technique to structure video sequences and generate corresponding meta data supporting adaptive presentations for arbitrary communication terminals. The algorithm structures a video sequence into segments of coherent camera motion. The structured video is presented by representative key-frames, which are characterized by background and moving or stationary objects. The realized prototype incorporates algorithms for camera motion estimation, spatial-temporal segmentation, object tracking, key-frame selection, and meta data (MPEG-7) generation.

## 1 INTRODUCTION

Since the multimedia presentation becomes prevalent on most communication terminals, techniques enabling them to access rich multimedia content are therefore of increasing significance, e.g., as discussed under the concept of Universal Multimedia Access [11, 2]. From this motivation, the *Multimedia Message Box or $M^3$-Box scenario* [5] is formulated and being developed. As depicted in figure 1, multimedia messages are recorded at an input interface simulator, where the structural information or meta data is generated. The messages and corresponding meta data are transmitted to the message center, at which the messages can be adapted or transcoded according to the output's context description. The MPEG-7 standard [7, 8] plays an important role in the scenario because it allows to describe multimedia content at different abstraction levels. MPEG-7 defines content descriptions with respect to audiovisual characteristics and specifies an exchange format for describing, modelling, and indexing multimedia data.

The paper proposes to create the structured content of arbitrary video sequences because this is a prerequisite for the media adaptation. The targeted prototype automatically proposes a meaningful video structure represented by key-frames. As the tool runs on the user terminal, the computational complexity should be as small as possible to allow fast responsitivity to the user. Nonetheless, most existing techniques locating or generating key-frames require high computational efforts or are domain-specific—unable to apply to arbitrary video sequences as solicited. For instance, Tanaka *et al* [12] proposed an automatic indexing scheme for television news video, the classification process was based on the semantic attributes of captions. Teodosio and Bender [13] proposed a technique to produce salient video stills, reflecting the aggregate of the temporal changes with the salient features preserved. Tonomura *et al* [14] proposed panoramic icons to represent the entire visible contents of a scene. Moreover, much work on video temporal segmentation focuses on how to detect cuts [10] and dissolves or fades [6], while there is still not much work aiming to structure video recorded by amateur users, e.g., from a hand-held camera or portable videophone, where no sophisticated techniques to merge messages are available and video cuts are obtained directly from the device interface. Therefore, this paper proposes a technique to generate:

- meaningful video message structures or a group of segments, each of which contains interrelating characteristic i.e. coherent camera motions, and
- representative key-frame indices, where a group of them best represents key information preserved therein.

## 2 OVERVIEW OF THE SYSTEM

The fundamental criteria of the proposal was grounded by identifying which constituents of video content are integral to human understandings. Intuitively one video's main feature well observed by humans is *motion*, which is applicable to characterize both imaged scenery at the background and moving or stationary objects recorded in sequence within the frame of the operating camera. The technique was correspondingly constructed based on the two mentioned criteria.

The video structuring or temporal segmentation process considers background, which can implicitly be described by the notion of estimated camera motion. The proposed algorithm splits the video message temporally into a group of video intervals, each of which contains coherent camera motion. Key-frames are selected to preserve the main information both of the background and characterized by observable objects, e.g., size and location. Because the tool aims at the low-complexity technique, it thus exploits the

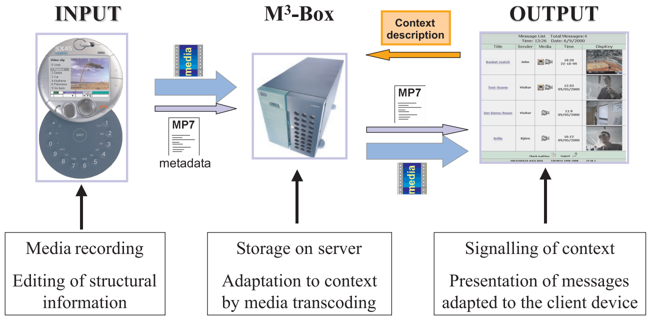
INPUT
M³-Box
OUTPUT
Context description
media
MP7
metadata
MP7
media
Media recording
Editing of structural information
Storage on server
Adaptation to context by media transcoding
Signalling of context
Presentation of messages adapted to the client device

Figure 1: M$^3$-Box scenario

readily-encoded motion information in the compressed domain (MPEG-1). The input set consists of macroblock-based motion vectors. To further reduce algorithm complexity, it considers only motion vectors encoded in P-type pictures. The overall system process (see figure 2) is summarized as follows:

For each picture in a video sequence, the spatial segmentation process is first carried out to classify spatial elements (in a unit of macroblock) to background or foreground. For the background set, the camera motion is estimated based on a four-parameter affine model, while the objects are defined and tracked according to the foreground set. The structuring and key-frame selection algorithms are then processed. The major components are described in the next section.

## 3 MAJOR COMPONENTS

The main components of the proposal are spatial segmentation, object tracking, camera motion estimation, temporal segmentation, and key-frame selection.

### 3.1 Spatial Segmentation

The spatial segmentation algorithm functions as a preprocessing unit to determine which spatial unit in each picture belongs to foreground or background. It groups inter-picture motion vectors into spatial regions in such a way that each region has coherent characteristic. As proposed by Heuer and Kaup [4], the Jacobi matrix can be applied to describe the spatial relationship of each motion vector with respect to those of the neighboring macroblocks.

#### *3.1.1 Jacobi Matrix*

The Jacobi matrix describes the spatial relationship of motion vectors in terms of the changing rate of both motion vector's horizontal and vertical components with respect to horizontal and vertical directions. The Jacobi matrix can be expressed in the following matrix of four spatial derivatives, where $V_X$ and $V_Y$ are the horizontal and vertical components of each motion vector in the image at coordinate $(X, Y)$:

$$\mathbf{J}_{\ V_X,\ V_Y}(X,Y) = \begin{bmatrix} \frac{\partial V_X}{\partial X} & \frac{\partial V_X}{\partial Y} \\ \frac{\partial V_Y}{\partial X} & \frac{\partial V_Y}{\partial Y} \end{bmatrix} = \begin{bmatrix} J(0,0) & J(0,1) \\ J(1,0) & J(1,1) \end{bmatrix}$$

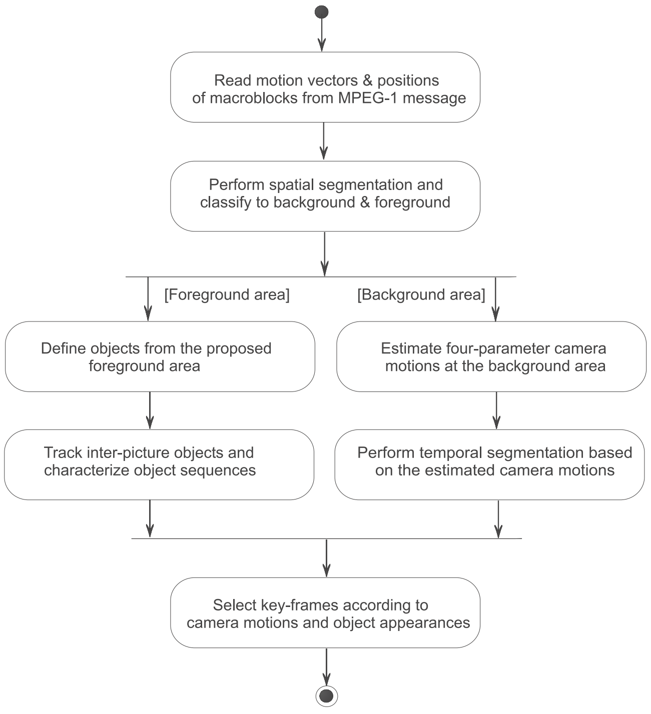
Read motion vectors & positions of macroblocks from MPEG-1 message
Perform spatial segmentation and classify to background & foreground
[Foreground area]
[Background area]
Define objects from the proposed foreground area
Estimate four-parameter camera motions at the background area
Track inter-picture objects and characterize object sequences
Perform temporal segmentation based on the estimated camera motions
Select key-frames according to camera motions and object appearances

Figure 2: overall system process in segmenting video and proposing key-frames

#### *3.1.2 Labelling Rules*

The labelling algorithm classifies motion vectors based on the estimated Jacobi matrices with the following rules:

- of any two adjacent motion vectors, if four elements in both Jacobi matrices are sufficiently similar, they are labelled to the same region;
- otherwise, they are labelled to different regions.

The *Euclidean metric*, $D(\mathbf{J}_1, \mathbf{J}_2)$, is used as a measure determining the degree of difference of any two Jacobi matrices. The measure can be expressed as follows, where $\mathbf{J}_1$ and $\mathbf{J}_2$ are the Jacobi matrices of the comparing motion vectors, and $J(i,j)_1$ and $J(i,j)_2$ are the elements of two considering Jacobi matrices:

$$D(\mathbf{J}_1, \mathbf{J}_2) = \sqrt{\sum_{i=0}^{1}\sum_{j=0}^{1}(J(i,j)_1 - J(i,j)_2)^2}$$

#### *3.1.3 Background and Foreground Findings*

Each motion vector is classified as background or foreground. The process treats each region (as defined in section 3.1.2) according to the following rules:

- The largest region is defined as background if it is sufficiently large; by experiment, the minimal size is configurable between 20% and 30% of the picture area.
- Other regions are defined as foreground (or intra-picture object) if their sizes are between two thresholds; the upper one is set at 55%, while the lower one is set at 5% of the picture area.

#### *3.1.4 Intra-picture Object Definition Improvement*

Because the defined regions are sensitive to the chosen Euclidean metric threshold and this situation particularly leads to imprecision of the object shape, a *Closing* algorithm is applied to improve the object outline. If the motion vector is sufficiently similar to the average motion vectors at all adjacent locations, its label will be replaced with the label of the neighbors' majority. Moreover, because each defined object region is frequently composed of multiple heterogeneous-motion regions, the object definition can then be improved by an adapted *Seed Fill* algorithm. It groups all adjacent non-background macroblocks into a single region.

### 3.2 Camera Motion Estimation

The process is based on a four-parameter affine model. Such a model is suitable because for most considered video sequences it is assumed that the camera rotation angles are small and the imaged screen is flat. The reference equation is expressed as follows:

$$\begin{bmatrix} V_X \\ V_Y \end{bmatrix} \approx \begin{bmatrix} C_F - 1 & -C_F\varphi_z \\ C_F\varphi_z & C_F - 1 \end{bmatrix} \cdot \begin{bmatrix} X \\ Y \end{bmatrix} + \begin{bmatrix} t_X \\ t_Y \end{bmatrix}$$

where the four parameters of the estimated camera motion are horizontal translation $t_X$, vertical translation $t_Y$, rotation angle $\varphi_z$, and zooming factor $C_F$. The estimate of the four parameters ($t_X$, $t_Y$, $C_F$, $\varphi_z$) is determined by searching for the point where the derivatives with respective to those four parameters of the following cost function (MSE) are equal to zero (considering only the motion vector set $V_0$ of the background, and $R_1 = C_F - 1$ and $R_2 = C_F\varphi_z$):

$$\sum_{i \in V_0} \left[ \left(V_{X,i} - R_1 X_i + R_2 Y_i - t_X\right)^2 + \left(V_{Y,i} - R_2 X_i - R_1 Y_i - t_Y\right)^2 \right]$$

### 3.3 Object Tracking

This process requires *a priori* knowledge of the intra-picture object regions (derived from the spatial segmentation process, see section 3.1) in order to track each particular object among successive pictures. For each object, the centroid is calculated, and then matched with the nearest one (via a Euclidean metric) of the previous picture. The main characteristics of particular objects imaged in different pictures are captured for the key-frame selection purpose. The pictures containing the largest or midmost object are among the key-frame candidates.

### 3.4 Temporal Segmentation

The algorithm combines both translational motion components of every two adjacent pictures into a single measure—translational angle—as defined:

$$\theta_n = \arctan\left(\frac{t_{Y,n} - t_{Y,n-1}}{t_{X,n} - t_{X,n-1}}\right)$$

where $n$ is a time variable representing the picture number. A segment boundary—temporal boundary indicating the change of camera motions—is located at the picture whose changing rate of the translational angle is considerably high. In practice a series of the absolute values of the derivatives of the translational angles, $\Theta_n = \left|\frac{\partial \theta_n}{\partial n}\right|$, is first estimated. A segment boundary is then located at the picture whose derivative value has a local maximum which is larger than the configurable threshold (set at 3°/picture).

### 3.5 Key-frame Selection

The content of the video sequence can be represented by key-frames, which are selected based on the rules according to the following criteria:

1. object characteristic:
   (a) the pictures containing the largest object, which has the longest lifespan (calculated from temporal difference between the first and the last picture containing that particular object),
   (b) the pictures containing the midmost object, which has the longest lifespan,
   (c) the pictures containing the left- and rightmost object if it moves horizontally, and
   (d) the pictures containing the upper- and lowermost object if it moves vertically.
2. estimated camera motion :
   (a) the first and last picture of a segment, and
   (b) every picture where the accumulated horizontal or vertical translational motion exceeds threshold, configured at 40%-60% of the horizontal or vertical camera frame size.

## 4 EXPERIMENTAL RESULTS

The chosen example introduces a typical video message in the scenario. The video content can be explained by the camera motion and the object appearance. The prototype correspondingly generates the segments and the key-frames as shown in figure 3. There are two segments, where the segment boundary is located at the moment the camera motion changes from panning-right to tilting-up (see the plot of camera-motion trajectory path in figure 4). The key-frames are selected according to the defined rules (see section 3.5); the largest object (frame 64, rule 1(a)), the midmost object (frame 55, rule 1(b)), large accumulative translational motion (frame 112, rule 2(b)), and new background information (frame 1 and 169, rule 2(a)).

Regarding the complexity issue, the prototype holds the following computational advantages. Firstly, it saves the motion-estimation computation thanks to the readily-encoded motion-vector set obtained directly from the MPEG-1 message. Secondly, despite the rough spatial unit (macroblock), the algorithms, i.e. spatial segmentation and object tracking, consume much less computation—compared to most proposals which are based on pixel-based approach—while still function acceptably. Lastly, since the applied temporal segmentation is based on a simple function (see section 3.4), only a small complexity is required.

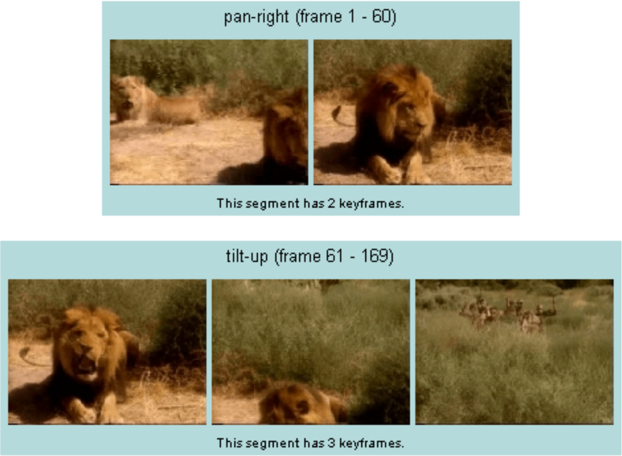

Figure 3: temporal segmentation and key-frame selection results from the lion example (adapted from [1])

## 5 SUMMARY AND OUTLOOK

This paper proposes a low-complexity technique to structure video sequences by splitting them into meaningful segments and to produce key-frame indices pointing to the pictures containing important information. The video is structured in such a way that each segment contains coherent camera motion. Each video segment is presented by key-frames, whose selection process considers the changing information of the background and observable objects.

Regarding the problems of how to structure, annotate, and present video content, the authors plan to further investigate which features and, in particular, which feature combinations could lead to a feasible criteria serving our purpose, e.g., as discussed in [3, 9].

## 6 ACKNOWLEDGEMENT

The authors would like to thank Klaus Illgner (Siemens AG München CT IC 2) for proofreading the article.

### References

[1] http://www.darim.co.kr/ftp/mpegs/lions.mpg.

[2] J. Bormans and K. Hill, *MPEG-21 overview*, ISO/IEC JTC1/SC29/WG11 N4318 (2001).

[3] R. Brunelli, O. Mich, and C.M. Modena, *A survey on video indexing*, IRSTTechnical Report 9612-06 (1996).

[4] J. Heuer and A. Kaup, *Global motion estimation in image sequences using robust motion vector field segmentation*, Proceedings ACM Multimedia 99, Orlando, Florida, pp. 261–264 (1999).

[5] J. Heuer, A. Kaup, and U. Rauschenbach, *Adaptive multimedia messaging: Application scenario and technical challenges*, Wireless World Research Forum Kick Off Meeting, Munich, Germany (2001).

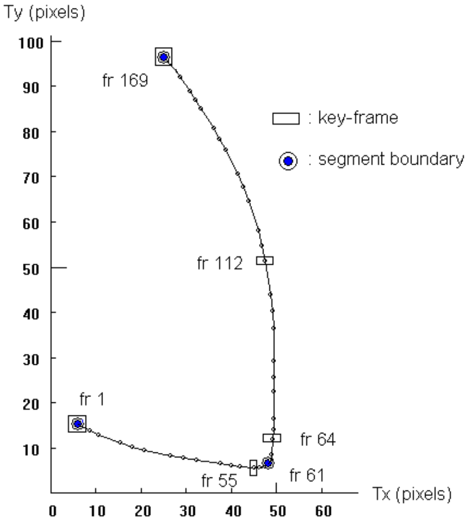

Figure 4: corresponding temporal segmentation and key-frame selection results marked on the camera-motion trajectory path from the lion example

[6] R. Lienhart, *Comparison of automatic shot boundary detection algorithms*, Proc. SPIE Vol. 3656, Storage and Retrieval for Image and VideoDatabases VII, pages 290–301, San Jose, CA (1999).

[7] J.M. Martínez, *Overview of MPEG-7*, ISO/IEC JTC1/SC29/WG11 N4031 (2001).

[8] F. Nack and A. Lindsay, *Everything you wanted to know about MPEG-7: Part 1-2*, IEEE Multimedia, 6(3)-6(4) (1999).

[9] Rosalind W. Picard, *A society of models for video and image libraries*, Technical Report 360, MIT Media Laboratory Perceptual Computing (1996).

[10] I.K. Sethi and N. Patel, *Video shot detection and characterization for video databases*, Pattern Recognition, vol. 30, no. 4, pp. 583–592 (1997).

[11] J.R. Smith, R. Mohan, and C.S. Li, *Adapting multimedia internet content for universal access*, IEEE Transactions on Multimedia 1(1) (1999).

[12] H. Tanaka, I. Ide, and K. Yamamoto, *Automatic video indexing based on shot classification*, Proc. 1st Intl.Conf. on Advanced Multimedia Content Processing (1998).

[13] L. Teodosio and W. Bender, *Salient video stills: Content and context preserved*, Proc. ACM Multimedia Conference (1993).

[14] Y. Tonomura, Y. Tanigushi, and A. Akutsu, *PanoramaExcerpts: extracting and packing panoramas for video browsing*, In the Fifth ACM Multimedia, pp.427–436, Seattle, WA (1997).