Video Analysis for Universal Multimedia Messaging
– Invited paper –

André Kaup
University of Erlangen-Nuremberg
Chair of Multimedia Communications and Signal Processing
Cauerstraße 7, D-91058 Erlangen, Germany
kaup@lnt.de

Siripong Treetasanatavorn, Uwe Rauschenbach, Jörg Heuer
Siemens Corporate Technology CT IC 2, D-81730 Munich, Germany
{*siripong.treetasanatavorn, uwe.rauschenbach, joerg.heuer*}*@mchp.siemens.de*

Abstract

Multimedia messaging is expected to become a major application for next generation mobile computing and communication devices. Heterogeneous capabilities of these devices, however, require adaptation of multimedia messages before rendering on any specific device. To achieve this in case of video messages, video analysis has to be performed to extract the structure of the video and control the message adaptation. This paper describes the universal multimedia messaging scenario and presents a short overview of existing video structuring approaches. A computationally efficient method for analyzing a video message in this scenario is introduced and a prototype for universal multimedia messaging conforming to the emerging MPEG-7 standard will be outlined.

1 Universal multimedia messaging

The rapid development of mobile devices and wireless communication networks will put devices capable of multimedia into the pockets of mobile people. Messaging applications which are today text-based will allow to send and receive multimedia content anywhere, anytime, using any device. Due to the heterogeneous capabilities of the different communication devices (e.g. screen size, computing power), the changing network characteristics while the user moves (e.g., bandwidth, delay) and the different situations in which a user is involved (e.g., in a meeting, driving), multimedia content will have to be adapted to devices, networks and user preferences. This concept is called *Universal Multimedia Access (UMA)*.

Integrating UMA functionality into an asynchronous multimedia messaging system will allow a sender to compose a message without having to know the technical characteristics of the recipient's messaging equipment. Similarly, a recipient can instruct the system how to transmit and present messages. We will call this a *Universal Multimedia Messaging (UMM)* system. UMM requires *meta data* to be stored and transmitted along with the message in order to provide the information needed for adaptation. In the UMM case, it is advantageous to use MPEG-7 [5] as a standardized meta data format. These meta data should be generated automatically as far as possible, but can also be provided by the user to emphasize or annotate parts of the message. In this paper, we will describe how UMM can be achieved in the case of asynchronous video messages. Video analysis methods are necessary to extract the meta data from the video.

We will first briefly review existing methods for video structuring and abstraction, which have originally been developed for video browsing and retrieval purposes to aid a user in rapidly understanding the essence of a video message without full replays. After that, a new analysis approach for UMM will be described. Finally, we put the analysis into the context of the M3Box, which is a prototype UMM system conforming to MPEG-7.

2 Related work

The process of generating a video structure with meaningful visual representations is at large comprised of two parts: temporal video segmentation and content abstraction or presentation.

Much of the work in temporal video segmentation con-

Proc. 5th IEEE Southwest Symposium on Image Analysis and Interpretation, SSIAI 2002, Santa Fe, NM, April 7-9, 2002, pp. 211-215.

siders global characteristics of the video sequence and detects video subunits (or so-called *shot*) boundaries, i.e., cuts [6, 7] and dissolves or fades [4]. The exploited features are typically obtained or derived either from the pixel domain, e.g. color histograms [14] and motion vectors [1], or from the compressed domain, e.g. Discrete Cosine Transform (DCT) coefficients and motion vectors [3, 13] of macroblocks. Moreover, a semantic attribute set is applicable in some domain-specific techniques, e.g. for news programs [8].

For video abstraction and presentation, there are two basic approaches. On the one hand, an indexing set of key information is used to present a video summary. For example, Toklu and Liou [11] proposed motion-parameter-based key-frame selection algorithms. On the other hand, representative synthetic images are computed from the video. For instance, Teodosio and Bender [10] proposed a technique to produce salient video stills reflecting the aggregate of the temporal changes with the salient features preserved, and Tanigushi *et al.* [9] used panoramic icons to represent the entire visible contents of a scene.

3 Video analysis in the context of UMM

For UMM, a video analysis technique is required which is able to detect the *video structure* and to extract an *abstraction* preserving the main information recorded in the video. This extracted description will be used to control the media adaptation. The proposed technique considers the imaged scenery at the background and independent object movements in front of it, and proposes

(a) a meaningful video structure or a group of segments of interrelating characteristic, i.e., coherent camera motion, and

(b) representative key-frame indices, where a group of single frames represents important events recorded by the message creator.

Video messages have some specific characteristics which have to be taken into account when detecting structure. Usually, they have been recorded from a hand-held camera or mobile videophone, where no zoom and no special cinematic tools like video merging, fading, and dissolving, are available. Thus, the temporal video structure can be derived considering two characteristics: *hard cuts* which are obtained directly via the device interface (e.g., Start/Stop button), and *camera work primitives* (e.g., still, pan) which describe segments of coherent camera motion.

To generate an abstraction, key-frames are selected in order to capture key events according to the following criteria:

- background: key-frames are selected to represent the background recorded during camera movements;

- object behaviors: key-frames are selected when the objects in front of the background create important events, i.e., their appearing sizes are at the largest, they are at the center-most position, or they are moving into or out of the operating camera frame.

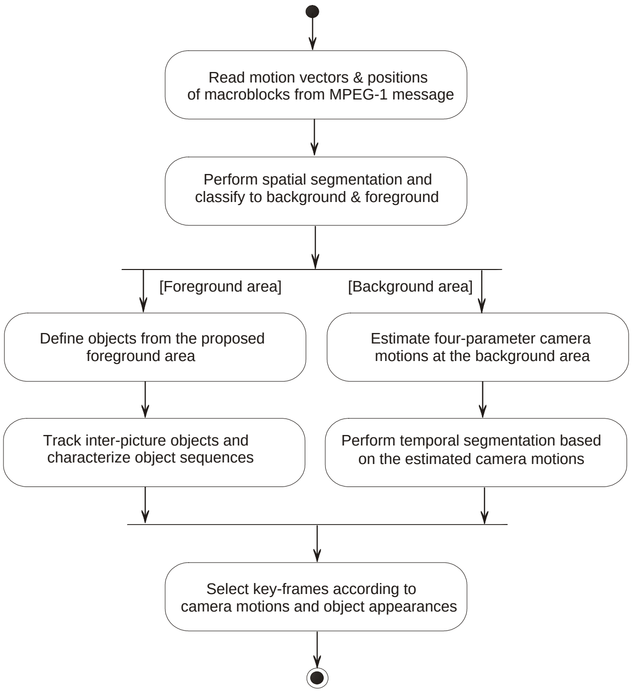

Read motion vectors & positions of macroblocks from MPEG-1 message
Perform spatial segmentation and classify to background & foreground
[Foreground area]
[Background area]
Define objects from the proposed foreground area
Estimate four-parameter camera motions at the background area
Track inter-picture 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 1. Video analysis sub-processes.

As the analysis tool will either operate on a user terminal with a slow CPU or on a heavily loaded server in the network, the algorithm's computational complexity should be as small as possible. We thus propose to exploit the encoded motion information in the compressed domain (MPEG-1). To further reduce the algorithm complexity, it considers only motion vectors encoded in P-type pictures. The overall system process (see figure 1) is summarized as follows: The system takes macroblock-based motion vectors as the input set. For each picture in a video sequence, a spatial segmentation process is carried out to classify spatial elements in macroblock units 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 executed. In the next section, some of these building blocks are described in greater detail.

4 The proposed analysis method and its building blocks

The video analysis system consists of the following components: spatial segmentation, background-foreground classification, object tracking, camera motion estimation, temporal segmentation, and key-frame selection. A more detailed description of the method will appear in [12].

4.1 Spatial segmentation and classification

The spatial segmentation functions as a preprocessing unit; it identifies which spatial unit of the input set belongs to the regions of background or foreground. To achieve this, the algorithm creates an overlaying layer on top of each picture. A layer is composed of a set of spatial regions of coherent characteristic explained in terms of the Jacobian matrix of the motion vectors, whereby the spatial relationship of each motion vector can be described with respect to those of neighboring macroblocks.

4.1.1 Region generation

The input set is classified spatially by evaluating the Jacobian matrix of the motion vector field for each macroblock. This 2x2 matrix describes the spatial relationship of motion vectors in terms of the changing rate of both motion vector's horizontal and vertical components:

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

where V_X and V_Y are the horizontal and vertical components of each motion vector at coordinate (X,Y). For every two adjacent macroblocks, the difference of their Jacobian matrices is computed by a quadratic distance measure and the macroblocks are labeled according to the following rules:

- of any two adjacent macroblocks, if the four elements in both Jacobian matrices are not sufficiently different according to a threshold t_{Jac}, they are labeled as belonging to the same region;
- otherwise, they are labeled as belonging to different regions.

4.1.2 Region classification

Each macroblock is then classified as background or foreground according to the overlaying regions as follows:

1. In each picture, the largest region is defined as background if it is sufficiently large according to a threshold t_{Bkg}
2. Foreground regions are defined as follows:
   (a) in case the background exists, other regions are defined as foreground (or intra-picture object) if and only if they are sufficiently big according to a threshold t_{Fg}
   (b) in case the background does not exist, there is no foreground;
3. Regions with a size less than t_{Fg} are considered as undefined.

4.1.3 Region precision improvement

Because the defined regions are sensitive to the chosen thresholds 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 of a macroblock is sufficiently similar to the average of the motion vectors at all adjacent macroblocks, the label of the overlaying region will be replaced with that of the neighbors' majority. After that, because each defined object region is frequently composed of multiple heterogeneous-motion regions, the object definition can be improved by an adapted *seed fill* algorithm. It groups all adjacent non-background macroblocks into a single region.

4.2 Camera motion estimation

After the spatial classification, the camera motion can be estimated from the motion vectors of the background macroblocks or from all macroblocks if no background has been found. The process is based on a four-parameter affine model, which is suitable because for most considered video sequences it is assumed that the camera rotation angles are small and the imaged scene 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 φ_z, and zooming factor C_F. The estimate of the four parameters (t_X, t_Y, C_F, φ_z) is determined by searching for the point where the derivatives with respect 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 macroblocks, and $R_1 = C_F - 1$ and $R_2 = C_F\varphi_z$), where i is an index into the set of macroblocks:

$$\sum_{i\in V_0}\Big[\big(V_{X,i}-R_1X_i+R_2Y_i-t_X\big)^2+\big(V_{Y,i}-R_2X_i-R_1Y_i-t_Y\big)^2\Big]$$

Since we are dealing with videos shot from a hand-held camera prone to camera shake, we have to reduce the jittering superimposed to the camera motion parameters. To achieve that, each of the four estimated parameter time-series is filtered with a low-pass filter.

4.3 Object tracking

This process tracks the classified intra-picture foreground regions (see section 4.1) among successive pictures. The algorithm tracks the calculated centroid of each object along the time axis, and then matches each object with the one of the nearest distance at the previous picture. The main characteristics of the objects (size, position, lifespan) are captured for the purpose of selecting key-frames later.

4.4 Temporal segmentation

Temporal segmentation detects segments of coherent camera motion. First, segments are classified into *still* and *moving-camera* ones by comparing the magnitude of the camera motions with a threshold t_{Mot}.

Next, the process further refines the result set of the defined *moving-camera* segments. It combines both translational motion components of every two adjacent frames into a single measure – translational angle. A temporal segment boundary is marked at the picture whose changing rate of the translational angle is considerably high, i.e. there is a sharp turn in camera movement. To do that, the derivative of the translational angle is compared with a threshold t_{Ang}.

4.5 Key-frame selection

The video sequence can be represented by key-frames, which are selected based on object characteristics and camera motion. Object-based criteria ensure, that each foreground object is visible in at least one key-frame. For objects with a long lifespan, additional key-frames are selected to depict their motion direction. For segments with camera motion, additional key-frames will be selected to show the complete background covered by a camera pan.

5 A UMM prototype - the M3Box

The MultiMedia Messaging Box *M3Box* is a UMM prototype which demonstrates the concepts described. The user can record a video message on a simulated multimedia mobile phone. The recorded video is then analyzed automatically to detect segments and select key-frames. After this analysis, the user can manually add text and audio annotations to the message and add or delete key-frames. Figure 2 depicts the easy-to-use interface to perform the annotation task. The timeline visualizes the result of the segmentation and keyframe selection as well as text annotations provided by the user. Segmentation results and additional annotations are encoded in MPEG-7 and sent to the M3Box server together with the video and audio data.

On the server, the video message is transcoded to a format which is understood by the device of the message recipient and which meets his/her current preferences. This may mean to play only selected video segments or to replace video segments by key-frames. Since the MPEG-7 message description follows the XML syntax, XSLT style sheets can be used to filter the message meta data. This filtering process generates the framework for controlling the presentation of the message on the device by means of HTML pages or SMIL scripts. Furthermore, the necessary transformations of the media data by external tools are initiated as well, e.g., to extract the image data of the key-frames from the video bitstream. A detailed description of the M3Box system can be found in [2].

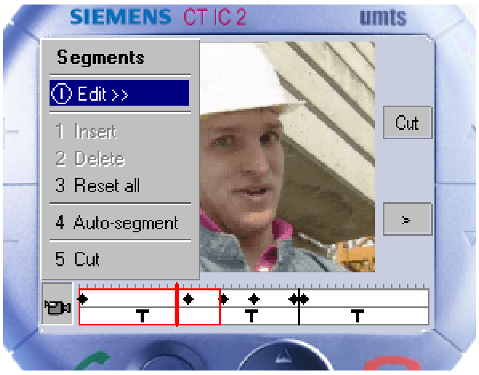


Figure 2. Video message creation interface. The user is supported by automatic analysis and timeline visualization.

Figure 3 shows the results of adapting a video message containing the 'foreman' video sequence to a device or situation where no video playback is possible. Three segments have been generated by the video analysis on the sender's device: a still segment at the beginning with an additional keyframe due to object motion, a pan-and-tilt segment depicting the range covered by the camera motion by three key-frames in the middle of the clip and another still segment with no object movements at the end of the clip. Note that no user intervention is required to generate the meta data necessary to perform the adaptation automatically.

6 Summary and outlook

We have introduced the concept of Universal Multimedia Messaging which allows to send multimedia messages from and to mobile devices. Because of the great variety of devices, network technologies and situation-dependent user preferences, messages have to be adapted accordingly. This adaptation relies on meta data which should be generated automatically as far as possible.

After briefly reviewing existing video structuring methods, we have described a low-complexity approach to structure a video sequence by partitioning it into meaningful segments and producing a representative key-frame index. The method has been evaluated in a prototypical UMM system.

By exploiting the motion information from the encoding process, the proposed method is computationally efficient. Because of the rough spatial unit (macroblock), the spatial segmentation and object tracking algorithms consume much less computation – compared to most pixel-based proposals – while still function acceptably for the task at hand. Since the applied temporal segmentation is based on a simple function (see section 4.4), only a small complexity is required here as well.

The proposed analysis method can be further improved by using additional information to enhance the precision of object detection and tracking, by exploiting other camera motion parameters, and by using a priori knowledge about the geometry of the objects in the video scene.

References

[1] A. Akutsu and Y. Tonomura. Video tomography: An efficient method for camerawork extraction and motion analysis. *Proc. ACM Multimedia*, pages 349–356, 1994.

[2] J. Heuer, J. L. Casas, and A. Kaup. Adaptive multimedia messaging based on MPEG-7 - the M3-Box. In *Proc. 2nd International Symposium on Mobile Multimedia Systems & Applications*, pages 6–13, Delft, 2000.

[3] V. Kobla and D. Doermann. Extraction of features for indexing MPEG compressed video. *Proc. of IEEE Workshop on Multimedia Signal Processing*, pages 337–342, 1997.

[4] R. Lienhart. Comparison of automatic shot boundary detection algorithms. *Proc. SPIE Storage and Retrieval for Image and Video Databases VII, Vol.3656*, pages 290–301, 1999.

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

[6] I. Sethi and N. Patel. Video shot detection and characterization for video databases. *Pattern Recognition, Vol.30, No.4*, pages 583–592, 1997.

[7] B. Shahraray. Scene change detection and content-based sampling of video sequence. *IS&T SPIE Proc. Digital Video Compression: Algorithms and Technologies, vol. 2419*, pages 2–13, 1995.

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

[9] Y. Tanigushi, A. Akutsu, and Y. Tonomura. PanoramaExcerpts: extracting and packing panoramas for video browsing. *Proc. ACM Multimedia*, pages 427–436, 1997.

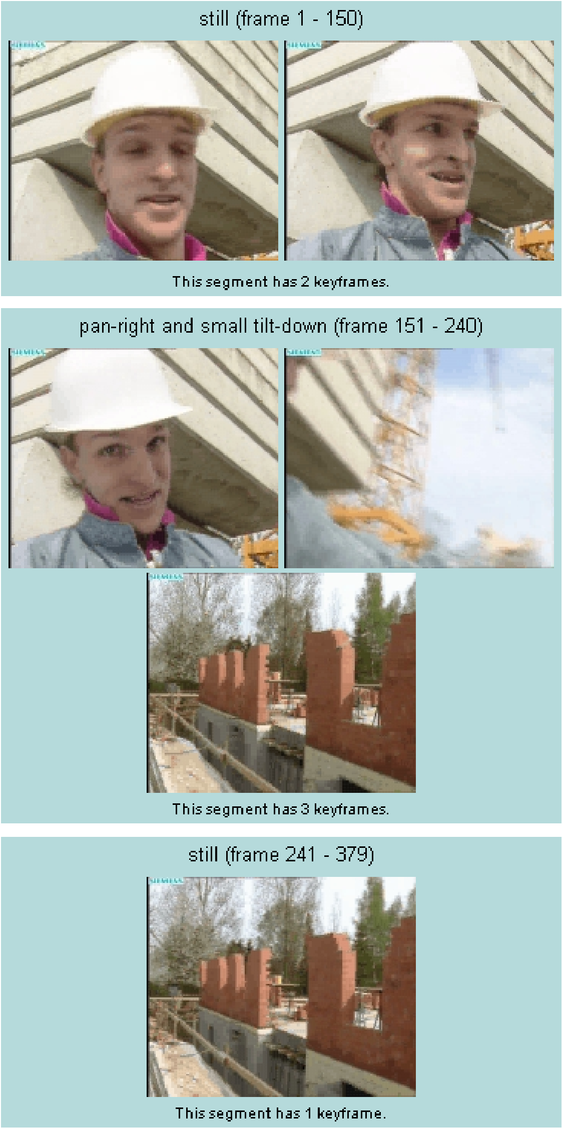
[10] L. Teodosio and W. Bender. Salient video stills: Content and context preserved. *Proc. ACM Multimedia Conference*, pages 39–46, 1993.

[11] C. Toklu and S. Liou. Automatic key-frame selection for content-based video indexing and access. *Storage and Retrieval for Media Databases; SPIE Vol. 3972*, pages 554–563, 2000.

[12] S. Treetasanatavorn, U. Rauschenbach, J. Heuer, and A. Kaup. Automatic video structuring for multimedia messaging. In *Proc. XI European Signal Processing Conference*, Toulouse, France, 2002.

[13] B. Yeo and B. Liu. Rapid scene analysis on compressed video. *IEEE Trans. on Circuits and Systems for Video Technology*, pages 533–544, 1995.

[14] H. Zhang, A. Kankanhalli, and S. Smoliar. Automatic partitioning of full-motion video. *ACM Multimedia Systems, 1(1)*, pages 10–28, 1993.



still (frame 1 - 150)

This segment has 2 keyframes.

pan-right and small tilt-down (frame 151 - 240)

This segment has 3 keyframes.

still (frame 241 - 379)

This segment has 1 keyframe.

Figure 3. Video segments and key-frames extracted from the 'foreman' sequence