Multimed Tools Appl
DOI 10.1007/s11042-010-0575-2

Annotation based personalized adaptation and presentation of videos for mobile applications

Sarah De Bruyne · Peter Hosten · Cyril Concolato · Mark Asbach · Jan De Cock · Michael Unger · Jean Le Feuvre · Rik Van de Walle

Abstract Personalized multimedia content which suits user preferences and the usage environment, and as a result improves the user experience, gains more importance. In this paper, we describe an architecture for personalized video adaptation and presentation for mobile applications which is guided by automatically generated annotations. By including this annotation information, more intelligent adaptation techniques can be realized which primarily reduce the quality of unimportant regions in case a bit rate reduction is necessary. Furthermore, a presentation layer is added to enable advanced multimedia viewers to adequately present the interesting parts of a video in case the user wants to zoom in. This architecture is the result of collaborative research done in the EU FP6 IST INTERMEDIA project.

Keywords Annotation · Adaptation · Rich media presentation · Personalized multimedia

1 Introduction

Many situations exist where personalized multimedia is highly desirable in order to improve the user experience. Therefore, on the one hand, properties of multimedia need to match the current user situation such as the available network bandwidth,

S. De Bruyne (✉) · J. De Cock · R. Van de Walle
Department of Electronics and Information Systems—Multimedia Lab,
Ghent University—IBBT, Gaston Crommenlaan 8 bus 201, 9050 Ledeberg-Ghent, Belgium
e-mail: sarah.debruyne@ugent.be

P. Hosten · M. Asbach · M. Unger
Institute of Communication Engineering, RWTH Aachen University, 52056 Aachen, Germany

C. Concolato · J. Le Feuvre
Multimedia Group, Signal and Image Processing Department, Telecom ParisTech,
46 Rue Barrault, 75013 Paris, France

Published online: 28 July 2010

display device capabilities, etc. On the other hand, personal user preferences need to be taken into account to enable user-centric convergence of multimedia. This vision is generally known as Universal Multimedia Access (UMA) and is one of the research fields covered by the EU FP6 IST project Interactive Media with Personal Networked Devices (INTERMEDIA) [13]. In particular, one of the objectives of this project is to generate a common vision on user-centric multimedia services in shared content environments to provide users with content personalized to their (semantic) user preferences and usage environments [10].

To cope with the challenges imposed by the UMA paradigm, video content needs to be adapted to the capabilities of the terminal device, the network constraints, and the preferences of the user. Furthermore, when dealing with client devices characterized by small displays, additional techniques such as presentation layers are needed to optimally present the content. To make these different techniques aware of the actual content of the video files, annotation information is indispensable. The aim of this paper is to combine efforts from the video analysis, video adaptation, and multimedia presentation domains to customize multimedia content to the user preferences.

In particular, we propose a framework that illustrates how multimedia annotations can guide adaptation and presentation techniques to create personalized multimedia for applications with limited bandwidth and display constraints, such as mobile devices.

Firstly, in order to satisfy the bandwidth constraints imposed by the network or the decoding capability of the terminal devices, efficient adaptation techniques for reducing the bit rate are required [34]. Typically, these adaptation techniques will reduce the quality of the entire frame. However, by incorporating region-of-interest (ROI) information, more intelligent adaptations can be realized by assigning different priority levels to particular areas. Unfortunately, content collections often lack any metadata related to ROIs which can be used to steer context-aware adaptations. Therefore, automatic content analysis and annotation techniques are of paramount importance.

Secondly, in order to comply with the limited display constraints of mobile devices and the user preferences, presentation techniques are indispensable. According to Knoche et al. [15], it is important to offer people the possibility to individually adjust the viewing size of the content when dealing with mobile devices. Furthermore, they verify that up-scaling or zooming into the picture can lead to better user experience when consuming content on mobile devices as detailed information in video sequences can otherwise no longer be seen. Therefore, in this paper, a dynamic presentation layer is added which takes into account the user preferences by using interactions, the characteristics of the device, and the ROI information generated during the annotation process. As such, advanced multimedia viewers can present the ROIs in the adapted video streams in a suitable manner using this presentation layer.

By combining the three different research domains, both adaptation and presentation techniques can become more intelligent as the semantics of the underlying video are taken into consideration.

The different aspects of personalized video adaptation and presentation guided by annotations are further described in the remainder of this paper. First, related work on personalization of multimedia content is provided in Section 2. Next, as

the first building block in our architecture, the automatic metadata generation is discussed in Section 3. Sections 4 and 5 elaborate on adaptation techniques and rich media presentations respectively which are guided by ROI information. Performance results are discussed in Section 6 and conclusions are drawn in Section 7.

2 Related work

Multimedia customization is an essential aspect in the development of solutions for UMA [7]. Consequently, a wide variety of multimedia customization approaches have already been proposed, as described by Magalhães et al. [18]. These approaches can be divided into two major categories: media bitstream selection and adaptation. Media selection tries to identify the most adequate multimedia bitstream from a set of available bitstreams with different characteristics. Multimedia adaptation, on the other hand, involves the transformation of the content if the available variations provided by multimedia selection are not adequate enough. Depending on the scalability provisions present in the bitstream, the desired layers need to be extracted or transcoding operations need to be performed.

Depending on which target applications are considered, the adaptation operations can be divided into structural and semantic adaptations. Most publications deal with structural adaptation techniques, which are typically performed to adapt multimedia resources according to network and terminal characteristics of the end-user. To obtain different quality versions, the video streams are adjusted along different scalability axes. Temporal scalability determines the frame rate; spatial scalability decreases the resolution of a video stream; Signal-to-Noise Ratio (SNR) scalability adjusts the visual quality.

Besides structural adaptations, semantic adaptation operations are gaining importance. This type of adaptation typically involves the temporal and/or spatial reduction of a multimedia asset and can be realized by combining the user preferences with metadata revealing semantic knowledge of the resource. The most meaningful parts of the video may have different coding than others, so as to adapt video transmission to both user's requirements and device's capabilities.

One type of spatial semantic adaptation is attaching a higher priority to the ROIs during the adaptation process. Like most publications in this domain, Cavallaro et al. start with the extraction of the semantic metadata from the video by performing background subtraction to obtain the moving objects [6]. The different regions are then assigned to different classes of relevance, leading to different qualities when encoding the video streams. Applied to object-based coding standards such as MPEG-4, the different regions are coded using different video objects. For frame-based coding standards such as MPEG-1, the amount of transform coefficients in the areas corresponding to background is reduced or the background in the video is lowpass-filtered prior to encoding. Bertini et al. [2] and Cucchiara et al. [8] follow a similar approach by employing different quality levels during encoding. Baccichet et al. [1] make use of the more recent H.264/AVC video standard to code the video streams. They divide the foreground and background regions in different slice groups. When working with static background regions, this implies that only the slices corresponding to moving objects need to be transmitted to the client. In contrast to the aforementioned algorithms, the surveillance system proposed by Hata et al. [12]

does not require that the ROIs are known prior to encoding and are embedded in the original sequence. Instead, the system transcodes the original JPEG2000 video stream in the compressed domain based on the ROI information provided during adaptation. By reusing information from the original video stream, the complexity of this approach is significantly lower than the aforementioned techniques. However, special requirements such the presence of independently coded spatial regions are imposed to the incoming video in order to support spatial adaptation.

Another technique for spatial semantic adaptation involves the cropping and scaling of content by selecting a suitable, semantically meaning region in the video. On the one hand, this decision can be taken prior to encoding and involves no further interaction with the user [21, 33]. During the selection of the desired region, not only metadata, but also the display resolution of the target device are considered to achieve a reasonable playback of a video. On the other hand, to enable user interaction, client-server systems have been proposed that interactively stream the desired region of the original video sequence [11, 20, 25]. To reduce the bandwidth consumed, the original video sequence is typically divided into multiple, independently coded tiles. Only those tiles which correspond to the desired region need to be extracted and transmitted. Due to the interaction between client and server, some latency is introduced.

In this paper, both aforementioned spatial semantic adaptation types are combined. Firstly, compressed-domain transcoding of H.264/AVC sequences guided by ROI information is performed on the server in order to distribute the available bit rate over the different priority regions. Secondly, a dynamic presentation layer is added to the video which enables the user to easily zoom into the ROIs. By offering this flexibility, the user is able to decide whether he prefers the original version of the content or the suggested regions. As the original transcoded version is completely sent to the client, no further bandwidth reduction is achieved during transmission when only requesting a ROI. However, this also implies that no latency is encountered and that the original sequence does not need to support the tiled coding patterns. To offer these two types of spatial semantic adaptation, existing techniques in the domain of video analysis, video adaptation, and multimedia presentation are brought together in order to obtain a framework for personalized multimedia content combining two types of spatial semantic adaptation, which is in contrast to related work which generally focuses on one issue.

In this context, the importance of standardization cannot be underestimated. MPEG and other standardization bodies have already dedicated a lot of effort to the standardization of tools for this application field. The first dimension relates to content coding (e.g., MPEG-1, MPEG-2, H.264/AVC, SVC); the second to content description and metadata (e.g., MPEG-7 [19]); the third to all issues related to content delivery (MPEG-21 [5]). In particular, the goal of the MPEG-21 standard is to realize the UMA paradigm by making use of the aforementioned standards for content coding and content description. As elaborated on in [3, 10], the general INTERMEDIA architecture for multimedia adaptation is built on several components of the MPEG-21 framework as well as on MPEG-7 metadata descriptions. In this general architecture, the desired bit rate adaptation techniques and the corresponding optimal settings of the transcoding parameters for the current situation should be determined by the adaptation decision taking engine (ADTE) and are executed on the server or possibly on a proxy on the network. On the other hand, the scene

description adaptation techniques which guide the presentation are done at the client side based on user inputs.

3 Automatic content annotation

During the last years, the field of image understanding has made significant progress. Different tasks such as shot boundary detection, face detection, optical character recognition, and even matching existing scripts to dialogs can now be handled by autonomous systems. Typically these techniques are used and evaluated in the context of information retrieval, i.e. searching digital libraries of stored media. An overview about this can be found in TRECVid [29].

In the context of Universal Multimedia Access (UMA), a new application field for automatic image understanding has arisen. As described above, sensible and intelligent adaptation of media that originally has been authored for bigger screens like television or cinema needs annotation.

The INTERMEDIA content annotation tool chain has been designed with personalized media adaptation and presentation in mind. It therefore extracts only those media characteristics that can be evaluated based on the current viewing situation to form an adaptation decision. At first, temporal segmentation is applied to find individual shots with mostly uniform media characteristics. This information is necessary for the following processing steps, but it can also be used for easily skimming content, skipping blocks or automatically creating a simple table of contents. Every shot is then analyzed for spatial partitioning.

Without any further knowledge on the kind of media content, general criteria are necessary to differentiate between important and less important parts. For INTERMEDIA, we chose the concept of foreground versus background to identify ROIs. Based on such annotations, the adaptation process can be steered to assign higher priority to (hopefully) more important foreground objects than to the surrounding background parts.

In parallel, specific objects are detected and tracked. Faces are important parts of typical visual media. Other kinds of objects could be interesting for certain domains like a football or cars for sports, or certain animals for documentaries. If special objects are present, media presentations can be personalized even more.

However, since there is no perfect and complete set of object categories per se, the generic segmentation information is always kept as a fall-back. This general structure is depicted in Fig. 1.

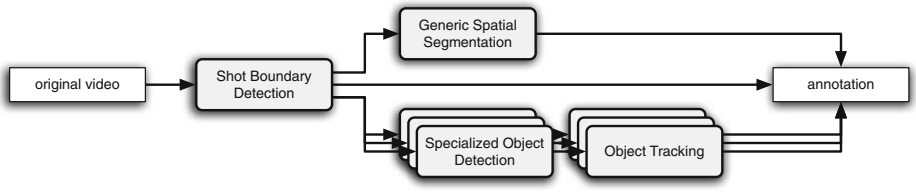


Fig. 1 Annotation pipeline with generic and specialized object detection concepts

3.1 Temporal segmentation

The first step of the automatic annotation process is the detection of different scenes. Via shot boundary detection, the temporal information in form of single shots is extracted. The detection of shot boundaries has been an area of active research for many years and techniques based on color histograms in HSV color space have been proven to be robust [4, 28].

By averaging over a couple of frames, small jumps in histogram entries are smoothed and only non-transient changes result in a jump in histogram differences that indicate a shot boundary. Also, a long-term comparison with the start of the current shot allows detecting gradual changes resulting from transition effects like wipes or dissolves.

3.2 Generic spatial segmentation

For every temporal segment, spatial segmentation information is extracted. In INTERMEDIA, generic object detection is based on motion compensated background subtraction. Background subtraction, being a standard approach for static cameras in surveillance scenarios, can be applied to general video content by compensating for camera motion.

The authors have presented an approach for motion compensated background subtraction that relies on global motion estimation and artificial background generation in [32]. We generate an artificial background image for every frame of a shot by following pixel trajectories into future and past frames as predicted by the estimated global motion model. Each resulting artificial background image is subtracted from its corresponding original frame.

As the resulting difference image contains small artifacts, we use segment-based diffusion for post-processing [31]. A color segmentation is performed grouping nearby pixels to segments that either belong to the foreground or background. The diffusion process exploits then the relationship between adjacent segments and propagates the difference energy. That way the decision whether a pixel belongs to the foreground or background is transferred to the segment level. Compare Fig. 2 for an exemplary frame.

This approach delivers pixel-accurate contours and masks for all those spatial regions of a video shot that cannot be described by a background model. It does not rely on any information about the objects and it is not restricted to special characteristics of a shot other than that there are objects that move relative to a background. Moreover, its underlying assumption (i.e. that there is foreground and background) seems very natural, as the reason for spatial segmentation is often to find ROIs.

By using background subtraction techniques, the detection of multiple objects moving differently to the global motion can easily be achieved. However, when multiple objects overlap or occlusion is present, additional information about the appearance of each object is required to correctly segment and track each of them. Current research comprises the extension of the generic spatial segmentation in order to support the detection of multiple objects by automatically learning their a priori unknown appearance.

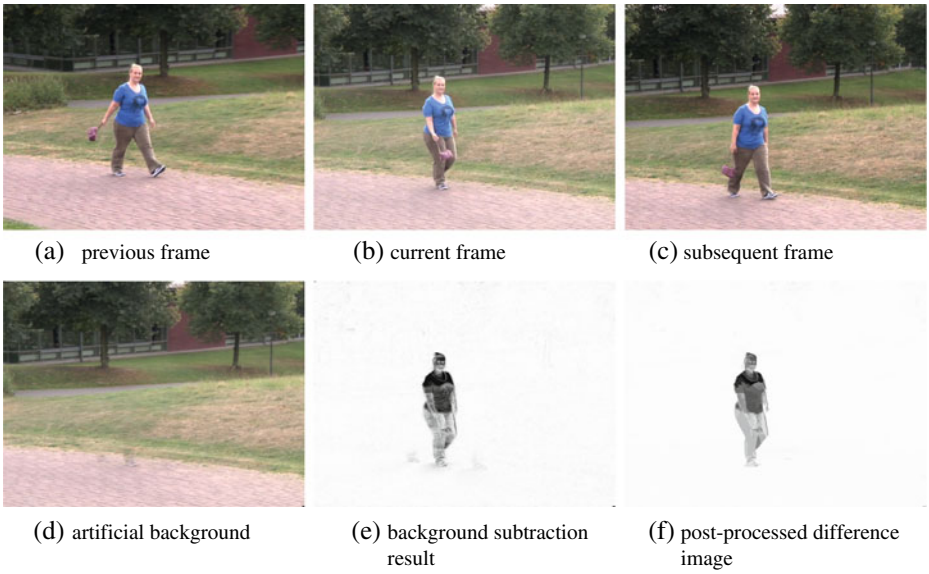


Fig. 2 Motion compensated background subtraction

3.3 Specialized object detection and tracking

In addition to this generic approach to spatial segmentation, the INTERMEDIA framework detects and tracks specific object categories. Knowing that an object represents a face for example, gives a lot more information that can be exploited for semantic adaptation. We use an object detection scheme based on a boosted cascade of simple features [36].

The detector itself is represented by a degenerate decision tree (i.e., the cascade). Using the scanning window approach, an image is sampled at multiple positions and scales. Every sampled window is passed to the root of the tree and each node (called "stage") has the task to reject a certain percentage of non-objects but pass nearly all real object candidates on to the next stage, as illustrated in Fig. 3. True positives (i.e. image windows that truly contain the object) have to pass all classifier stages. The majority of the sampled windows will however not contain the object nor anything that looks similar to some degree. Since these regions are rejected at early stages, the average processing power spent per sampled window is very low.

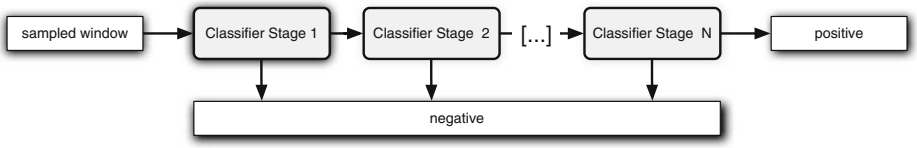
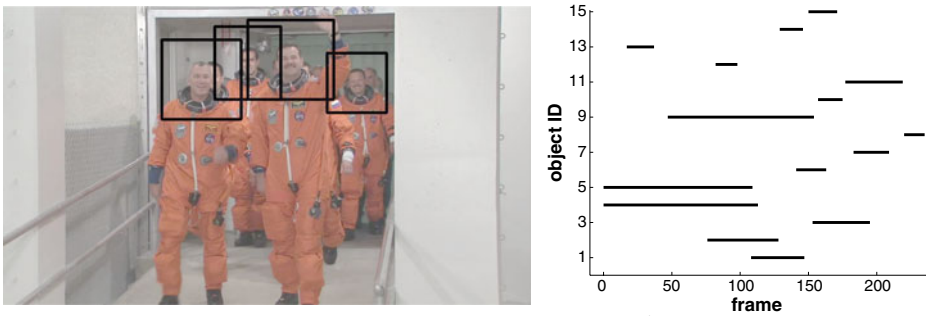


Fig. 3 Cascade of weak classifiers



(a) detected faces (b) visibility over time

Fig. 4 Specialized object detection results on the Crew sequence

However, only object categories that have been previously learned can be detected. As it is unfeasible to learn every possible object category, we have focused on the detection of faces. In principle this detector scheme can be extended by any kind of object, as long as the underlying training set covers all variations of the object's appearance [30].

Unfortunately, the detection scheme is only moderately accurate regarding the localization quality. Over subsequent frames of a video, the detected location and scale of the same object may vary markedly. For that reason a Kalman filtering approach is used for object tracking, modeling the location, velocity and size of the object. It is also used to correct missing detections if for some reason the object could not be found in a single frame. This way an object can be properly detected and tracked until it disappears, providing the information about size and location of very specific ROIs.

Figure 4a illustrates the result of the face detection and tracking process on the Crew sequence. While the astronauts approach the camera, most of their faces are visible. A camera pan follows and many faces turn away, while bystanders become visible. Appearance and disappearance of faces 1 to 15 in the Crew sequence are depicted in Fig. 4b.

4 Region-of-interest-based video adaptation

As described above, multimedia adaptation is required to for example match the bit rate of the video signal to the available network bandwidth. Adaptation techniques will typically reduce the quality of the entire frame to comply with the constraints [34]. However, by incorporating ROI information which is derived from the annotations, as described in Section 3, more intelligent adaptations can be realized.

Although scalability provisions at the encoder side might allow easy adaptation of video streams, such as with the scalable extension of the H.264/AVC video coding standard (SVC) [26], practical video encoders are likely to output single-layer video streams. Hence, adaptation of coded video content remains a challenging task. This is only reinforced by the high complexity of state of the art video coding algorithms.

As a straightforward solution of video adaptation, a coupled decoder and encoder might be used, where the output of the decoder is fed to the encoding process. Given

the high computational complexity of both modules, and in particular the encoder, such a solution is not viable in typical use cases. In order to reduce the computational burden of the adaptation, it is pivotal that information from the incoming bitstream is reused during adaptation.

Transcoding solutions provide fast adaptation by reusing data of the input stream such as motion vectors and prediction modes. As a result, the search space is reduced during transcoding when compared to recoding, hereby allowing a significant increase in processing speed. The presented video transcoding module is able to reduce the bit rate of the incoming coded video signal to comply with the constraints imposed by the environment, such as the available network bandwidth. Typically, the bit rate of the video stream is determined by the coarseness of the quantization during encoding [9, 22]. When a reduction in bit rate is desired, this can be accomplished by requantizing the prediction error coefficients with a coarser quantization step size, which is indexed by the quantization parameter (QP, which can take values from 0 to 51 for H.264/AVC).

Traditional transcoding techniques will reduce the quality of the entire frame [34]. However, when watching a video sequence, one will typically pay more attention to the important parts in the sequence. By assigning higher priority to the ROIs, as extracted in Section 3, more intelligent adaptations can be realized. Hata et al. [12] already investigated several object-aware approaches to transcode JPEG 2000 surveillance sequences. In this paper, we will take a closer look at the block-based H.264/AVC video coding standard [38].

In this ROI-aware transcoder, the quality of the picture after transcoding remains high in the ROI(s), while the background quality will be reduced, resulting in a lowered bit rate for the overall video sequence. In this way, the data in the bitstream will be apportioned to the relevant regions in the video sequence, while overhead and quality of the less important background regions will be reduced. This is demonstrated in Fig. 5a, where the high quality is only maintained for the ROIs detected in Fig. 2, while other regions are heavily quantized, leading to a significant bit rate reduction.

In case the original quality of the video sequence is very high, the difference in quality between the ROIs and the background can be experienced as disturbing.

(a) original quality in ROI (572 kbps)

(b) trade-off in quality for ROI and background (563 kbps)

Fig. 5 Examples of ROI-based adaptation

Furthermore, as the ROIs remain in the highest quality, the amount of bits needed to code these regions stays unaffected. Therefore, when certain bit rate constraints must be met, the reduction in bit rate needs to be compensated completely by the background regions. In order to better divide the available bit rate, a trade-off can be made between the different QPs corresponding to the ROIs and the background. Roughly, the bit rate associated with the coefficients will be halved by increasing the QP by 6 in H.264/AVC. As a result, it is clear that slightly reducing the quality of the ROIs will have a significant impact on the remaining quality of the background. Figure 5b illustrates a more realistic version of a ROI-based transcoded video, where approximately the same bit rate as in Fig. 5a is obtained. The QP of the ROI is increased by 2, whereas the QP for the background is raised by 16 instead of 24. The slightly reduced quality of the ROI is hardly noticeable, whereas the artifacts in the background resulting from the quantization are clearly less disturbing.

A high-level overview of the used transcoder architecture is given in Fig. 6. The first component of the transcoder is a decoder loop, which reconstructs the pictures to the pixel domain, and stores these pictures in the buffer. For these decoded pictures, object detection can be applied, resulting in the ROIs. The macroblock indices associated with the ROIs are passed on to the encoder loop. For these macroblocks, it is possible that no change in QP is incurred. Nonetheless, recalculation of the prediction error is necessary, since the prediction values may have changed. For the background macroblocks, requantization is executed with an increased QP. A second motion estimation step is avoided by passing the motion parameters from the incoming bitstream to the encoder loop. In this way, motion vectors, reference picture indices, macroblock partitioning, and prediction modes are reused and passed on to the output bitstream without additional computational complexity. This 'short cut' results in significant computational complexity savings when compared to a coupled decoder-encoder with full motion (re-)estimation.

Two strategies can be followed to resolve the issue of which QP to use during transcoding. On the one hand, a fixed increase in QP can be used, so that the output bit rate is a priori unknown. On the other hand, a rate control algorithm

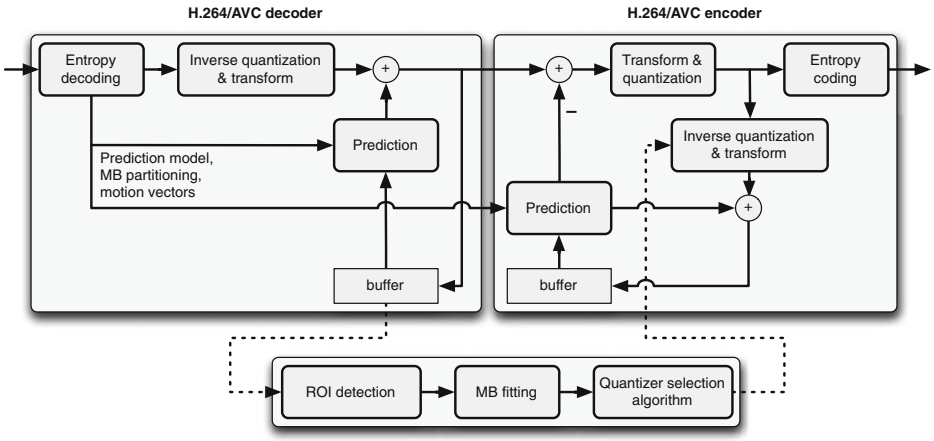


Fig. 6 Overview of ROI-based adaptation (transcoder) tool

can steer the QP selection so that the appropriate reduction in bit rate is achieved after transcoding. During rate-controlled transcoding, the available bit rate for each frame can be divided over the different detected objects, depending on the size of the area of the objects. Furthermore, the rate control algorithm can base itself on the amount of bits that were spent on the object region in the input bitstream, as an indication of the 'complexity' of the area to be coded. This information will be an asset when compared to encoder-side rate control algorithms, where the algorithms are typically based on a prediction of the texture complexity (expressed as mean absolute distortion (MAD) values) [35].

If desired, motion information can be changed to better reflect the updated information in the bitstream. Such a motion refinement step can help improve coding efficiency of the output bitstream, hereby helping to further improve video quality given the available bandwidth. In particular, in the case that ROI macroblocks are predicted based on non-ROI macroblocks, or vice versa, it is likely that prediction will benefit from an update in motion vectors or prediction modes. While this step can increase computational complexity, intelligent algorithms can be designed that benefit from the information in the input bitstream. This means that exhaustive motion estimation can still be avoided.

5 Rich media presentation

Section 3 presented how interesting objects can be located in a video, and Section 4 showed how this video can be adapted intelligently based on this information. This section describes how a presentation layer is generated to interactively present the adapted video to a user when dealing with mobile devices with constrained displays, based on this ROI information.

In order to achieve a suitable presentation, the following requirements should be met. The presentation system should:

1. be backward compatibility with simple audiovideo players, so that the content can be displayed on every device;
2. be able to present multiple ROIs at the same time;
3. be able to present ROIs of rectangular;
4. be able to present dynamic ROIs, synchronized with the video;
5. allow a user to interact with ROIs;
6. and enable adapted presentation according to the ROI aspect ratio and to the viewing device characteristics: screen size (in inches), screen resolution (in pixels), and screen aspect ratio.

Additionally, from a user perspective, we can add that the presentation layer offering the ability to zoom on a ROI should be as intuitive to use as possible. It should not disrupt the video viewing experience and should accommodate different types of videos: videos where the number of ROIs is low and quasi-constant such as in Fig. 5 and videos where the number of ROIs is changing rapidly, with possibly several overlapping ROIs, as in Fig. 4b.

These requirements lead us to the use of a scene description to provide presentation instructions. These presentation instructions indicate, to advanced multimedia players (also called rich-media players), where the ROIs are, how and

when to display them on top of the video, how they change over time, and how the user may interact and view them. When packaged properly, these instructions may be ignored by traditional audio-video players such as VLC, thereby fulfilling requirement 1. There are many candidate scene description technologies to fulfill the other requirements. We can cite the Scalable Vector Graphics (SVG) language and its extension, Lightweight Application Scene Representation (LASeR); Flash, the de facto web standard for animated graphics and video presentation (e.g. as on YouTube); the Binary Format for Scenes (BIFS), or the Synchronized Multimedia Integration Language (SMIL).

In our scenario, the description of the presentation instructions is tightly coupled with the video and the video content is described as a stream. We therefore naturally decide to choose a stream-based description language. Additionally, since we require a packaging format capable of storing separately the scene description and the video (to fulfill requirement 1), we are therefore left with either MPEG-4 LASeR or MPEG-4 BIFS. Both languages are stream-based, can be created using XML or simply plain text, then compressed or not, and finally streamed over IP or stored along the video in an mp4 file, both allowing individual presentation of the video. In terms of expressiveness of the presentation, even though the detected ROIs are currently rectangular, we require a language capable of representing arbitrary shaped ROIs. Although both MPEG languages could allow it, we choose to create our presentation instructions using the MPEG-4 BIFS language [14] since this language supports texture mapping.

We present now the structure of these instructions, which consist of an initial scene (presented at $T=0$) and scene updates. Based on the ROI information extracted during the analysis, we first compute the maximum number n of ROIs per frame for the whole video duration. With this information, we build an initial scene which consists of a video (*Shape*, *Bitmap* and *MovieTexture* nodes) on top of which n clickable rectangles (*Shape*, and *Rectangle* nodes), initially invisible, are drawn. We also define $n+1$ viewports (*Viewport* nodes) for each of the ROIs and for the non-zoomed version, used as the initial viewport. Upon a click (detected by a *TouchSensor* node) on one of the ROI rectangles, the associated viewport is bound (using a *Route*, a *Conditional* node, and the *set_bind* event of the *Viewport* node), and the video is therefore zoomed to show the appropriate ROI, as illustrated in Fig. 7.

(a) (b)

Fig. 7 Illustration of user-driven presentation when zooming into one of two ROIs

The viewport also allows indicating if the pixel aspect ratio is to be preserved or not and, if it is, how to fill the rest of the viewport. An example is provided below, using the BIFS textual syntax. Note that the body of the prototype *RegionOfInterestProto* is omitted for brevity.

```
PROTO RegionOfInterestProto [
  exposedField SFInt32 hidden 1
  exposedField MFString keyword [""]
  exposedField SFVec2f position 0 0
  exposedField SFVec2f size 0 0
  eventOut SFBool activate
  eventOut SFBool deactivate
] { ... }
OrderedGroup {
 children [
  Shape {
   geometry Bitmap {}
   appearance Appearance { texture MovieTexture
                           { url "video.mp4" } }
  }
  DEF VP_MAIN Viewport { f\/it 1 size 1280 720 }
  DEF ROI_MAIN TouchSensor {}
  DEF C_MAIN Conditional
    { buf\/fer { REPLACE VP_MAIN.set_bind BY TRUE } }
  DEF VP1 Viewport { f\/it 1 }
  DEF C1 Conditional
    { buf\/fer { REPLACE VP1.set_bind BY TRUE } }
  ...
  Transform2D {
   translation -360 216
   children [
    DEF ROI1 RegionOfInterestProto {}
    ...
]}]}
ROUTE ROI_MAIN.isActive TO C_MAIN.activate
ROUTE ROI1.activate TO C1.activate ...
```

Finally, we build a new scene update for each frame where the ROI changes. Each update contains commands to hide/ show and set the position and size of the clickable rectangles, and to set the position and size of the corresponding viewports. Each update can contain a command to set the title of each ROI in order to include semantic information into the presentation. An example of a scene update is provided below.

```
AT 40.04 { # time in milliseconds
REPLACE ROI1.hidden BY 0
REPLACE ROI1.keyword BY "Face1"
REPLACE ROI1.position BY 208 -160
REPLACE ROI1.size BY 48 48
REPLACE VP1.position BY -128 32
```

```
REPLACE VP1.size BY 48 48
REPLACE ROI2.hidden BY 0
REPLACE ROI2.keyword BY "Face2"
REPLACE ROI2.position BY 288 -160 ... }
```

The result of this generation process is then compressed into the BIFS binary format, packaged into an mp4 file together with the video and played with the GPAC Rich Media Player [16] on desktops or mobile devices.

It should be noted that even if this method relies on an initial scene with a defined maximum number of ROIs, the update mechanism could also be used to insert new ROIs dynamically. We can also remark that the purpose of this rather simple scene is to enable the zooming into ROI. However, depending on the number and dynamicity of the ROIs, the user interface in the scene needs to be carefully designed. First, if there are many ROIs that overlap, the user will not be able to click (easily) on all of them. To avoid this problem, one solution would be to move the selection of a ROI to buttons, menus or clickable thumbnails on the side of the video, or to link them to hardware buttons on the phone. Second, if the durations of ROIs are too short, the user may not be able to click on them or may see rapid changes of zooming factor. In particular, if the tracking is not handled correctly, sometimes a single object may create two short-running ROIs at different instants in time instead of a single long-running ROI. One way to solve this problem would be to filter out the ROIs that are too short. Another way would be to make continuous, smooth transitions when a ROI disappears and the zooming is deactivated. This could be done at the signal level or during the translation into BIFS instructions, but in any case, we see that the quality of the user interface is highly dependent on the quality of the tracking system.

6 Performance results

6.1 Automatic content annotation

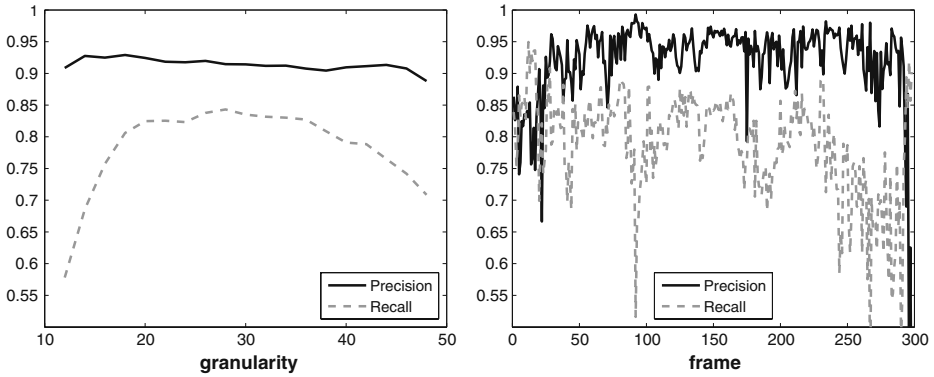
6.1.1 Accuracy results

As the generic spatial segmentation represents the main component of the annotation pipeline, its performance has been further investigated. For the evaluation we use the Stefan sequence of the official MPEG-4 video test set [23] and an additional sequence called CarII with pixel accurate object masks as ground truth. Hence the objective performance measures recall and precision can be computed:

$$Recall = \frac{\text{number of correctly detected foreground pixels}}{\text{number of all foreground pixels}}$$

$$Precision = \frac{\text{number of correctly detected foreground pixels}}{\text{number of all detected foreground pixels}}$$

A crucial factor of the algorithm is the post-processing, which transfers the decision whether a pixel belongs to the foreground or background to the segment level, as can be seen in Fig. 2f. Consequently, the underlying color segmentation algorithm that groups neighboring pixels to segments, has a significant impact. This aspect is illustrated in Fig. 8a, where the average recall and precision is shown in dependency of the granularity, i.e. the average size of the segments.



(a) Average recall and precision in dependency of the granularity, Stefan & CarII

(b) Recall and precision overtime, Stefan

Fig. 8 Results of the generic spatial segmentation

Obviously, too small segments as well as too large segments impair the performance. In the former case the diffusion process is not able to remove the artifacts in the difference image, whereas in the latter case the model assumption is violated, that the pixels in each segment either fully belong to the foreground or background.

In Fig. 8b the recall and precision results are shown for the Stefan sequence at a suitable operating point of the granularity. It can be observed, that especially at the end of the sequence (frame 250–300) the recall rate decreases. This can be explained by the fact that in this part of the sequence extreme camera panning as well as zooming are simultaneously present, which impede the estimation of the artificial background.

However, these outliers do not have a significant impact on the overall system for personalized adaptation and presentation as we use rectangular ROIs, whose trajectory and size are smoothed over time.

6.1.2 Complexity

The complexity of the presented system for personalized adaptation and presentation of multimedia data is unequally distributed over the whole processing chain. Whenever new multimedia data is added to a content repository, annotations have to be extracted. This process is done once and is usually an off line operation. During playback, the adaptation process has to be performed in real-time on a server in the network. In addition, the playback device itself has to process the rich media presentation in real-time.

Regarding the annotation process, it can be stated that the total complexity is currently slightly too high for real-time processing on a standard personal computer (e.g.: Intel Pentium 4, 14.4 GFlops). If the number of object categories is raised, the specialized object detection task demands linearly growing resources, although, by exploiting similarity of object features, the complexity can be reduced to logarithmic growth [30].

Moreover, all parts of the annotation process qualify for massive parallelization, which would allow execution on modern graphics processing units. Similar work has

been performed in [24]. An intrinsic latency will however always hinder real-time usage: generic spatial segmentation as well as Kalman filtering require a buffer of "future" frames for additional stability.

6.2 Adaptation and presentation

6.2.1 Rate-distortion results

The rate-distortion performance and the complexity of the transcoder were evaluated by transcoding several sequences and comparing the results with results obtained by a coupled decoder and encoder (i.e., a recoder). The original sequences were coded using the H.264/AVC Joint Model reference software (version 17.0) using default coding tools, Main profile, four reference pictures, full rate-distortion optimization enabled, and IPP GOP structures.

To determine the difference between recoding and transcoding guided by ROI information, the Joint Model reference software was adjusted to support ROI functionality. During the creation of the adapted bitstreams, the same settings as applied to the original sequences were used, and rate-distortion optimization was once enabled and once disabled. The transcoded and recoded streams were generated by increasing the original quantization parameters QP_{or} with fixed values (ΔQP_{ROI} and ΔQP_{BG}):

$$QP_{\text{ROI}} = QP_{\text{or}} + \Delta QP_{\text{ROI}}, \tag{1a}$$

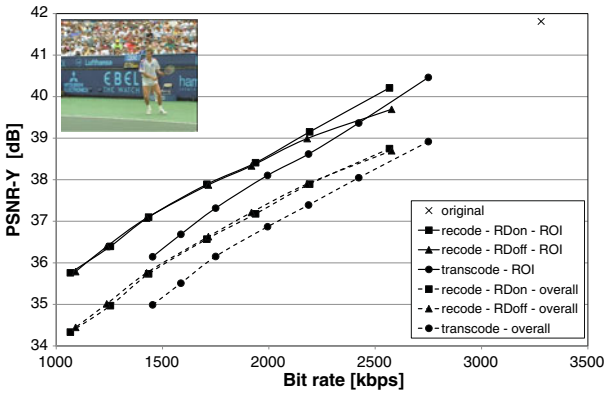
$$QP_{\text{BG}} = QP_{\text{or}} + \Delta QP_{\text{BG}}, \tag{1b}$$

$$\text{with } \Delta QP_{\text{BG}} - \Delta QP_{\text{ROI}} = c. \tag{1c}$$

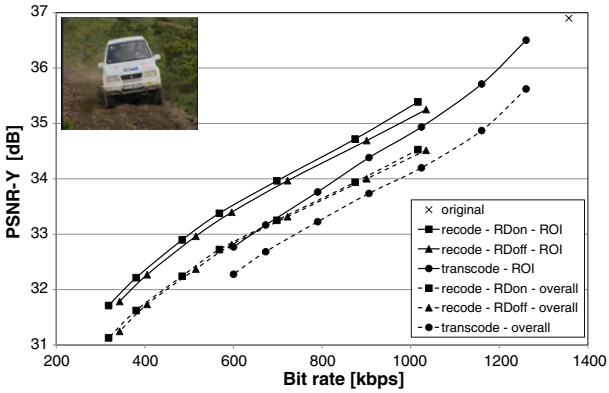
By using fixed quantization parameters, the influence of rate control on the performance results is eliminated. Large discrepancies in quality are avoided by setting the difference between ΔQP_{ROI} and ΔQP_{BG} to a constant value c.

In Fig. 9, rate-distortion results are shown for the Stefan and the CarII sequences. During the creation of these rate-distortion results, two scenarios were envisaged. Firstly, the dashed curves were generated by considering the bit rate and average PSNR-Y, where the latter is calculated by attaching equal importance to the ROIs and the background. Secondly, the overall bit rate and the PSNR-Y calculated using only the ROIs were combined, as indicated by the full lines. This configuration roughly corresponds to the situation where the user interacts with the presentation layer to zoom into the important regions. Consequently, these two scenarios correspond with the two extremes that can be obtained by using semantic PSNR (SPSNR [2]). In particular, by taking into account semantic relevance, this quality measure should better reflect the perceived quality compared to the general PSNR.

As can be seen from the rate-distortion curves, the recoder outperforms the transcoder in terms of rate-distortion. Whereas the transcoder adopts the macroblock modes, partitions, and the corresponding motion vectors from the original bitstream, the recoder searches for the optimal partitioning modes and motion vectors taking into account the requested quality. The gap between both approaches is about 0.5 dB for small reductions in bit rate and slightly increases to 1 dB when



(a) Stefan, CIF, 30Hz, $QP_{or} = 22$, $\Delta QP_{ROI} = 0, .., 6$, $c = 3$



(b) CarII, CIF, 25Hz, $QP_{or} = 27$, $\Delta QP_{ROI} = 0, .., 6$, $c = 3$

Fig. 9 Rate-distortion results for recoding and transcoding

the difference between the original and desired QP enlarges. This increasing gap is explained by the fact that larger macroblock partitions (and in general, coarser motion information) will be preferred at lower bit rates. While recoding evaluates the possibility of inserting larger partitions for the adapted bitstream, the transcoder will reuse the partitions from the input bitstream. This discrepancy will lead to an increasing rate-distortion gap as ΔQP_{ROI} and ΔQP_{BG} become larger.

As a final remark, on bit rate issues, we can also add that the cost of adding the BIFS instructions alongside the H.264/AVC video in the mp4 container is negligible. For example, we evaluated the mp4 file size increase when using BIFS instructions for ROI display and interactivity ranges from 1 to 1.5%.

6.2.2 Complexity

Although the transcoder is outperformed by the recoder in terms of rate-distortion, it is significantly computationally less demanding, as illustrated in Fig. 10. In particular, the complexity of the transcoding operation is mainly determined by the coding blocks indicated in Fig. 6, such as the prediction, (forward and inverse) transform, and quantization. When compared to full decoding and encoding, costly operations

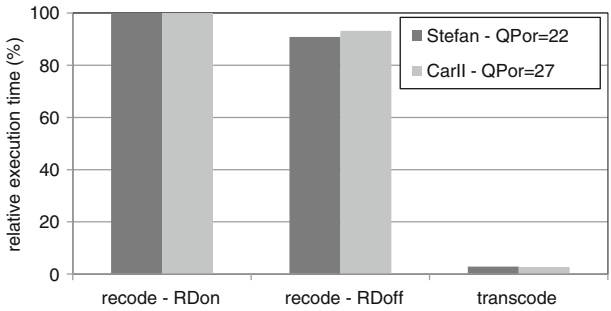


Fig. 10 Relative execution speed of the trancoder compared to recoding with rate-distortion optimization enabled and disabled

are avoided, such as motion estimation and mode decision [37]. Consequently, significant complexity savings can be accomplished by reusing the motion information. In this light, a compromise could be found by refining the partitioning modes and motion vectors to obtain higher rate-distortion results at the cost of increased computational complexity [17, 27].

For macroblocks in the ROIs, the same quantization parameter can be used as for the input bitstream (for the case that $\Delta QP_{\mathrm{ROI}} = 0$). Nonetheless, the residual data has to be recalculated for these macroblocks as well. This is necessary in order to avoid drift, since the prediction pixels might have changed. As a result, the same coding steps have to be executed for all macroblocks, and complexity remains identical for transcoding when more ROIs are added as well as when no ROI information is used as in traditional approaches.

During playback, the necessary processing power will be determined by the number of objects present at the same point in time. It can be assumed though, that to the user there is an upper limit of objects for the utility of the presentation anyway. And for typical cases, the cost of displaying the ROI on top of the video and of processing user interaction is negligible and the presentation can easily be achieved on mobile devices.

6.2.3 Visual observations

In order to compare ROI-based transcoding with traditional transcoding techniques, two adapted versions of the Crew sequence were created with approximately the same bit rate, as illustrated in the top row of Fig. 11. The quality of the traditionally transcoded sequence on the right is constant over the entire image. On the other hand, for the ROI-aware sequence on the left, the quality is higher for the important regions and lower for the remaining parts as the amount of bits is apportioned according to the priority levels of the regions. As a result, more artifacts can be observed on the walls and lower parts of the bodies, whereas the faces remain sharp and clearly recognizable.

When watching this content on small display devices, detailed information can no longer be distinguished. As explained in Section 5, dynamic presentation layers will make it possible to easily zoom into the important parts of the video. As a result, the majority of low quality regions will no longer be visible to the user and the displayed part of the sequence will mainly coincide with high quality ROI blocks. As a consequence, the average quality of the visible part of the ROI-based adapted

(a) ROI-based transcoded video (b) traditionally transcoded video

Fig. 11 Comparison of ROI-based and traditionally transcoding for the Crew sequence based on the annotations in Fig. 4a. The *first row* depicts the original resolution, the *second row* illustrates the user-driven presentation when zooming into the ROI covering all faces, whereas in the *third row*, we zoomed into the face of the person standing on the *left*

video will be higher compared to the traditionally transcoded bitstream, as depicted in the second and third row of Fig. 11.

7 Conclusions

This paper described an architecture for personalized adaptation and presentation of videos based on automatically extracted ROI information. The goal of this approach is to deliver content to users with mobile devices with limited display and network capabilities in a user-centric way in order to improve the user experience. First, by using ROI information, more intelligent adaptations can be achieved by degrading the quality of the different regions according to their importance. Furthermore, rich media presentations are included to enable interactivity with these ROIs. Performance results illustrated the advantages of combined ROI-based adaptation and presentation. To further prove the usefulness of the combination of ROI-based

adaptation and presentation in real-world applications, future work includes the evaluation of the system based on user studies.

Acknowledgements The research activities that have been described in this paper were co-funded by Ghent University, the Interdisciplinary Institute for Broadband Technology (IBBT), the Institute for the Promotion of Innovation by Science and Technology in Flanders (IWT), the Fund for Scientific Research-Flanders (FWOFlanders), the Belgian Federal Science Policy Office (BFSPO), and the European Union (within the framework of the NoE INTERMEDIA, IST-038419).

References

1. Baccichet P, Zhu X, Girod B (2006) Network-aware H.264/AVC region-of-interest coding for a multi-camera wireless surveillance network. In: Proceedings of the picture coding symposium
2. Bertini M, Cucchiara R, Del Bimbo A, Prati A (2005) An integrated framework for semantic annotation and adaptation. Multimedia Tools and Applications 26(3):345–363
3. Bolla R, Repetto M, De Zutter S, Van de Walle R, Chessa S, Furfari F, Reiterer B, Hellwagner H, Asbach M, Wien M (2008) A context-aware architecture for QoS and transcoding management of multimedia streams in smart homes. In: Proceedings of the IEEE international conference on emerging technologies and factory automation, pp 1354–1361
4. Boreczky J, Rowe L (1996) Comparison of video shot boundary detection techniques. J Electron Imaging 5(2):122–128
5. Burnett IS, Pereira F, Van de Walle R, Koenen R (eds) (2006) The MPEG-21 book. Wiley
6. Cavallaro A, Steiger O, Ebrahimi T (2005) Semantic video analysis for adaptive content delivery and automatic description. IEEE Trans Circuits Syst Video Technol 15(10):1200–1209
7. Chang S-F, Vetro A (2005) Video adaptation: concepts, technologies and open issues. Proc IEEE 93(1):148–158
8. Cucchiara R, Grana C, Prati A (2002) Semantic transcoding for live video server. In: Proceedings of the ACM international conference on multimedia, pp 223–226
9. De Cock J, Notebaert S, Lambert P, Van de Walle R (2010) Requantization transcoding for H.264/AVC video coding. Signal Process Image Commun 25(4):235–254
10. De Zutter S, Asbach M, De Bruyne S, Unger M, Wien M, Van de Walle R (2008) System architecture for semantic annotation and adaptation in content sharing environments. Vis Comput (Int J Comput Graph) 24(7–9):735–743
11. Feng W-C, Dang T, Kassebaum J, Bauman T (2008) Supporting region-of-interest cropping through constrained compression. In: Proceeding of the ACM international conference on multimedia, pp 745–748
12. Hata T, Kuwahara N, Nozawa T, Schwenke DL, Vetro A (2005) Surveillance system with object-aware video transcoder. In: Proceedings of the IEEE workshop on multimedia signal processing, pp 1–4
13. INTERMEDIA (2006) Interactive media with personal networked devices (ist-1-38419). In: European sixth framework programme (FP6) IST NoE co-funded project. http://intermedia.miralab.unige.ch/. Accessed May 2010
14. ISO/IEC 14496-11:2005 (2005) Information technology—coding of audio-visual objects. Part 11: scene description and application engine. ISO, Geneva
15. Knoche H, Sasse MA (2009) The big picture on small screens delivering acceptable video quality in mobile TV. ACM Trans Multimedia Comput Commun Appl 5(3):1–27
16. Le Feuvre J, Concolato C, Moissinac J-C (2007) GPAC: open source multimedia framework. In: Proceedings of the international conference on multimedia, pp 1009–1012
17. Lefol D, Bull D, Canagarajah N, Redmill D (2007) An efficient complexity-scalable video transcoder with mode refinement. Signal Process Image Commun 22(4):421–433
18. Magalhães J, Pereira F (2004) Using MPEG standards for multimedia customization. Signal Process Image Commun 19(5):437–456
19. Manjunath BS, Salembier P, Sikora T (eds) Introduction to MPEG-7: multimedia content description language. Wiley
20. Mavlankar A, Baccichet P, Varodayan D, Girod B (2007) Optimal slice size for streaming regions of high resolution video with virtual pan/tilt/zoom functionality. In: Proceedings of European signal processing conference, pp 1275–1279

21. MoCA project (2010) Movie content analysis. http://pi4.informatik.uni-mannheim.de/pi4.data/content/projects/moca/Project-ResolutionAdaptation.html. Accessed May 2010
22. Notebaert S, De Cock J, Beheydt S, De Lameillieure J, Van de Walle R (2009) Mixed architectures for H.264/AVC digital video transrating. Multimedia Tools and Applications 44(1):39–64
23. Pereira F, Alpert T (1997) MPEG-4 video subjective test procedures and results. IEEE Trans Circuits Syst Video Technol 7(1):32–51
24. Pinto N, Doukhan D, DiCarlo JJ, Cox DD (2009) A high-throughput screening approach to discovering good forms of biologically inspired visual representation. PLoS Computational Biology 5(11):e1000579
25. Quang Minh Khiem N, Ravindra G, Carlier A, Ooi WT (2010) Supporting zoomable video streams with dynamic region-of-interest cropping. In: Proceedings of the ACM SIGMM conference on multimedia systems, pp 259–270
26. Schwarz H, Marpe D, Wiegand T (2007) Overview of the scalable video coding extension of the H.264/AVC standard. IEEE Trans Circuits Syst Video Technol 17(9):1103–1120
27. Shen H, Sun X, Wu F, Li S (2006) R-D optimal motion estimation for fast H.264/AVC bit-rate reduction. In: Proceedings of the picture coding symposium
28. Smeaton A, Over P, Doherty A (2010) Video shot boundary detection: seven years of TRECVid activity. Computer Vis Image Underst 114(4):411–418
29. Smeaton AF, Over P, Kraaij W (2006) Evaluation campaigns and TRECVid. In: Proceedings of the ACM international workshop on multimedia information retrieval, pp 321–330
30. Torralba A, Murphy K, Freeman W (2004) Sharing visual features for multiclass and multiview object detection. In: Proceedings of the IEEE conference on computer vision and pattern recognition
31. Unger M, Asbach M (2008) Segment based diffusion—a post-processing step (not only) for background subtraction. In: International workshop on image analysis for multimedia interactive services, pp 167–170
32. Unger M, Asbach M, Hosten P (2008) Enhanced background subtraction using global motion compensation and mosaicing. In: Proceedings of the IEEE international conference on image processing, pp 2708–2711
33. Van Rijsselbergen D, Van De Keer B, Verwaest M, Mannens E, Van de Walle R (2009) Enabling universal media experiences through semantic adaptation in the creative drama production workflow. In: International workshop on image analysis for multimedia interactive services, pp 296–299
34. Vetro A, Christopoulos C, Sun H (2003) Video transcoding architectures and techniques: an overview. IEEE Signal Process Mag 20(2):18–29
35. Vetro A, Sun H, Member S, Wang Y (1999) MPEG-4 rate control for multiple video objects. IEEE Trans Circuits Syst Video Technol 9:186–199
36. Viola P, Jones M (2001) Rapid object detection using a boosted cascade of simple features. In: Proceedings of the IEEE computer society conference on computer vision and pattern recognition, vol 1, pp 511–518
37. Wiegand T, Schwarz H, Joch A, Kossentini F, Sullivan GJ (2003) Rate-constrained coder control and comparison of video coding standards. IEEE Trans Circuits Syst Video Technol 13(7):688–703
38. Wiegand T, Sullivan GJ, Bjøntegaard G, Luthra A (2003) Overview of the H.264/AVC video coding standard. IEEE Transact Circuits Syst Video Technol 13(7):560–576

Sarah De Bruyne received her M.Sc. degree in Computer Science from Ghent University, Belgium, in 2005. She joined the Multimedia Lab of Ghent University in 2005 where she is currently working toward the PhD degree. Her research interests are in the areas of content-based multimedia analysis, video coding technology, and content adaptation.

Peter Hosten received his Dipl.-Ing. degree in Electrical Engineering from RWTH Aachen University, Germany, in 2007. Since then, he has been working towards a PhD degree at the Institute of Communications Engineering, RWTH Aachen University. His research interests are video segmentation, image processing, and computer vision.

Cyril Concolato is Associate Professor in the Multimedia Group at Telecom ParisTech, Paris, France, where he received his master and doctoral degree in Computer Science in 2000 and 2007, respectively. His interests lie in multimedia scene descriptions and in interactive multimedia applications, in particular for the mobile domain. He is an active participant to the standardization bodies of MPEG and W3C. Finally, he is one of the project leaders of the Open Source project GPAC.

Mark Asbach received his Dipl.-Ing. degree in Electrical Engineering from RWTH Aachen University, Germany, in 2004. He has been working towards a PhD degree at the Institute of Communications Engineering, RWTH Aachen University. His research interests are pattern recognition, image processing, and computer vision, targeting multimedia content analysis.

Jan De Cock obtained the M.S. and Ph.D. degrees in Engineering from Ghent University, Belgium, in 2004 and 2009, respectively. Since 2004 he has been working at Multimedia Lab, Ghent University, and the Interdisciplinary Institute for Broadband Technology (IBBT). His research interests include video compression and transcoding, scalable video coding, and multimedia applications.

Michael Unger received his Dipl.-Ing. degree in Electrical Engineering from RWTH Aachen University, Germany, in 2004. Since then, he has been working towards a PhD degree at the Institute of Communications Engineering, RWTH Aachen University. His research interests are 3D reconstruction, image processing, and computer vision.

Jean Le Feuvre received his Ingénieur (M.Sc.) degree in Telecommunications in 1999, from Telecom Bretagne. He has been involved in MPEG standardization since 2000, and joined Telecom ParisTech in 2005 as Reasearch Ingeneer within the Signal Processing and Image Department. His main research topics cover multimedia authoring, delivery and rendering systems in broadcast, broadband and home networking environments. He is the project leader and maintainer of GPAC, a rich media framework based on standard technologies (MPEG, W3C, and Web3D).

Rik Van de Walle received his M.Sc. and PhD degrees in Engineering from Ghent University, Belgium in 1994 and 1998, respectively. After a visiting scholarship at the University of Arizona (Tucson, USA), he returned to Ghent University, where he became professor of multimedia systems and applications, and head of the Multimedia Lab. His current research interests include multimedia content delivery, presentation and archiving, coding and description of multimedia data, content adaptation, and interactive (mobile) multimedia applications.