Journal of Web Engineering, Vol. 5, No.4 (2006) 332-347

© Rinton Press

ADAPTATION OF MULTIMEDIA RESOURCES SUPPORTED BY METADATA

MULUGETA LIBSIE

Department of Computer Science, Faculty of Informatics

Addis Ababa University, Ethiopia

mlibsie@cs.aau.edu.et

HARALD KOSCH

Chair of Distributed Information Systems

University of Passau, Germany

harald.kosch@uni-passau.de

Received July 25, 2005

Revised August 23, 2006

The Video adaptation is an active research area aiming at delivering heterogeneous content to yet heterogeneous devices under different network conditions. This paper presents a novel method of video adaptation called segment-based variation. It aims at applying different reduction methods on different segments based on physical content. The video is first partitioned into homogeneous segments based on the physical characteristics of motion, texture, and color. Then optimal reduction methods are selected and applied on each segment with the objective of minimizing quality loss and/or maximizing data size reduction during adaptation. In addition, the commonly used reduction methods are also implemented. To realize variation creation utilizing these methods, a unifying framework called the Variation Factory is developed. It is extended to the Multi-Step Variation Factory, which allows intermediary videos to serve as variations and also as sources to further variations. It creates a tree of variations and the associated metadata, which allow one to apply successive reductions by active network nodes. They also allow the server to easily switch from one stream to another depending on resource availability. Our proposals are implemented as part of a server component, called the Variation Processing Unit (VaPU) offering user interface to guide the generation of the different versions of the source and an MPEG-7 metadata document. The information contained in this document describes both the source and the variations and helps the system to identify the most appropriate version. It can also be used by active components on the network to carryout efficient adaptation. Such adaptation will take user preferences, including disability, into account.

Key words: Metadata, MPEG-7, Segment-based variation, UMA, Video adaptation, Content accessibility

1 Introduction

Media adaptation is a research issue that arose as a result of the need to deliver content-rich media to users of varying resources, preferences and network connections in adherence to the real-time delivery requirements of continuous data. In general, client devices vary in their display capabilities, processing power and memory size. Different multimedia content are also stored in different formats.

Hence, there is a growing need for applications to bring such diverse multimedia information to yet diverse devices under different network conditions and user preferences. Stored content has to be converted between different bit rates and frame rates since content is usually available in a single modality, resolution and format. It must also account for different screen sizes, decoding complexity and resource constraints of client terminals. This scenario is often referred to as Universal Multimedia Access (UMA) [1].

To enable ubiquitous access, content variations must be generated either prior to delivery or on-the-fly [2,3]. When content variations are created prior to delivery, different versions are created, stored, selected and delivered. Our proposal is based on a mix of both approaches. More specifically, the interest is on video variation supported by metadata as a means for adaptation.

1.1 Video Variation

In video variation, different versions of a source video are created and stored in a media-database. Although most of the techniques used for variation creation are applicable to a video resource regardless of its encoding format, we are interested more on MPEG-4 videos to take advantage of the extensive adaptation options provided by the standard including its object-based coding feature.

Descriptive information (both for the source and the variation videos) is modelled using MPEG-7 descriptors and maintained in a meta-database [4]. MPEG-7 was selected for metadata description since it defines the *VariationSet Description Scheme* (DS) that allows standardized communication of audiovisual data in different representations [5]. Such metadata are useful for content adaptation. For instance, metadata help the end-user to filter the content according to his/her preferences. They help the system to identify the most appropriate variation that meets the required Quality of Service (QoS). A QoS specification may include information about network connection (minimum bandwidth, maximum delay, and minimum jitter), end-user device capabilities, and viewing preferences of the end-user. A variation video can also be created in real-time by the server or the metadata and the algorithm can be sent to another component on the delivery path, such as a proxy, to carryout the adaptation. Hence, all active network nodes might benefit from metadata to carryout efficient adaptation.

In view of the above, we defined a unifying framework, called the *Variation Factory*, that utilizes various reduction methods (which we also call them *variation methods* or *variation products*) to generate a set of variations and an MPEG-7 metadata document. There is always one source video to all the variation videos in a variation set. Although a source video may be subjected to more than one variation method in successive steps, there are no intermediary videos that also serve as variation videos. We have extended this framework to what is known as the *Multi-Step Variation Factory* or *Variation Tree* that allows the intermediary videos to be variations in their own right and then serve as sources to further variations thereby creating different variations with finer granularity in terms of resource requirements. We will use the following two use case scenarios to show how such a tree of variations and the associated metadata can be used for adaptation.

Use Case Scenario 1

Assume that a distributed multimedia adaptation and delivery system is composed of a server, clients of varying resources, and a network with active nodes such as gateways, routers, and proxies (Figure 1a) which are capable of adapting content. In this example, the server does not carryout

adaptation but delegates active nodes on the network. Initially the server sends the source video and the metadata document, created during variation creation, to Node 1. The metadata document has a tree structure where each node contains descriptive information about the variation as shown in Figure 1b. The information in the metadata document will assist the active nodes on the network to carryout the adaptation. The following is a possible sequence of events.

a. Node 1 realizes that the path to Node 3 does not need any adaptation. Hence, it sends the source and the metadata document as it is. However, the path to Node 2 requires adaptation. Temporal reduction is applied, resulting with Var1. It also prunes the metadata tree, so that only the corresponding branch is sent to Node 2, together with the adapted content (Var1) so that Node 2 can further adapt it in the spatial and color domains, if required.

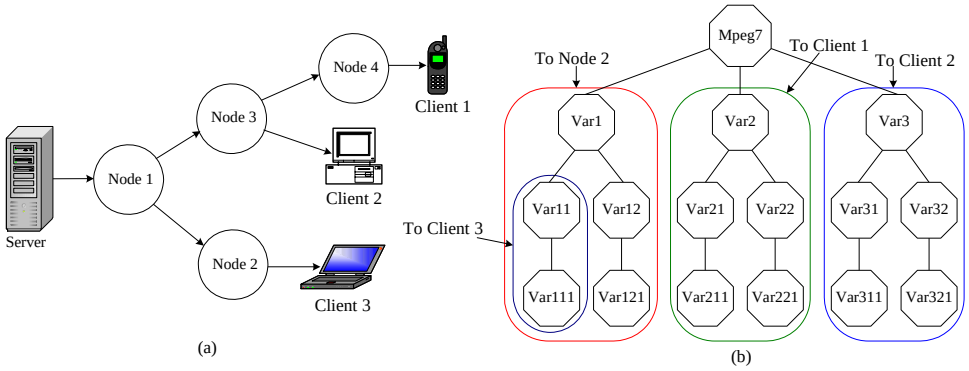

Figure 1: Distributed adaptation use case scenarios. (a) Distributed multimedia adaptation and delivery system (b) MPEG-7 metadata - Variation tree.

b. Node 2 realizes that the connection to Client 3 does not have enough bandwidth. Hence, it decides to adapt the content using color reduction. It also extracts the corresponding node from the new metadata tree and sends both the adapted content (Var11) and the metadata to Client 3 so that it can also further adapt it in the spatial domain, if required.

c. Node 3 realizes that Client 2 has no color capability. It therefore carries out color reduction and sends the adapted content (Var3) and the corresponding metadata to Client 2. It sends all what it has received from Node 1 to Node 4, since the path has sufficient bandwidth.

d. Node 4 identifies that Client 1 has limited display size. Therefore, it carries out spatial reduction and delivers the adapted content (Var2) and the corresponding metadata.

Use Case Scenario 2

Now consider situations where adaptation is to be carried out by the server. It will use variations that have been already created. Initially it selects a variation based on device capabilities and user preferences. Afterwards, it uses stream switching to account for shortages of bandwidth along the path. Figure 2 shows a possible stream switching path depending on resource availability. The following notations are used for the labels in Figure 2 and in later Sections. Let

- V_S denote the source video; T_V, S_V, and C_V denote temporal, spatial, and color variations.
- V_M denote a variation video obtained by applying the variation method(s) M. For instance, $V_{C_V T_V}$ denotes a variation video obtained by applying color and temporal variations.

Let us assume that a receiving device has no color capability. Therefore, the server initially tries to stream a variation without color (V_{C_V}). When further shortage of resources occurs, the server can successively switch to the next variation on the forward path ($V_{C_VT_V}$ then $V_{C_VT_VS_V}$). When resource availability improves, the server can switch to a better stream by tracing backwards to the source.

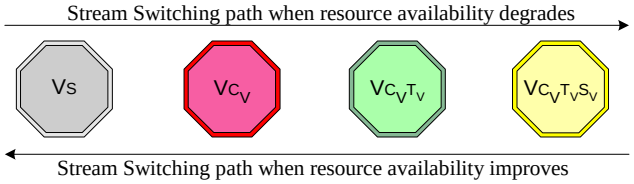

Figure 2: Possible stream switching path.

1.2 Reduction or Variation Methods

A source video can be subjected to different reduction methods in generating variation videos. These reduction methods can be categorised as *basic* or *composite*. The basic methods are applied by their own whereas the composite methods make use of a combination of the basic methods. The basic reduction methods include *Temporal reduction* (the frame rate of the visual stream is reduced), *Spatial reduction* (the size of each frame is reduced by encoding fewer pixels), *Color reduction* (reduces the color depth of each pixel), *Bit Rate reduction* or *Quality Scaling (*reduces the quality or details of a video by changing encoding parameters), *Syntax conversion* (re-encoding a video using a different encoding technique so that receiving devices that may not have the capability to decode the original can handle the new format), and *Extraction* (extracts key frames, audio or video).

There are two composite methods that utilize a combination of the basic methods: *object-based adaptation*, which is based on object-based coding features of encoders, and *segment-based variation*, which we introduce as a new adaptation method. *Object-based adaptation* is applicable for videos encoded using object-based coding techniques such as MPEG-4, where a scene can be made up of many objects. Hence, different reduction methods can be applied on each object. Entire objects can even be dropped if resource availability does not allow the delivery of the entire video.

Segment-based variation is a novel adaptation method proposed in this paper. It aims at applying different reduction methods on different segments based on physical content. The video is first partitioned into homogeneous segments based on the physical characteristics of motion, texture, and color. Then optimal reduction methods are selected and applied on each segment with the objective of minimizing quality loss and/or maximizing data size reduction.

The rest of the paper is organized as follows. Section 2 briefly surveys some of the important works related to the work in this paper. Section 3 details the proposed method of variation creation. Section 4 introduces a unifying framework of the variation methods called the *Variation Factory* and extends it to the *Multi-Step Variation Factory*. Implementation details and metadata representation are detailed in Section 5. Experimental results are presented in Section 6. Finally, Section 7 concludes the paper.

2 Related Work

There are many research efforts in video adaptation for delivery. Some works are devoted to individual reduction methods whereas others are more of on system level and architectural issues. This Section reviews some of the representative ones that have been reported in the literature.

Techniques for transcoding content on the Internet for heterogeneous devices was presented by Smith *et al.* in [1, 7, 8, 9]. The solution has two key components: a conceptual data representation framework that provides multi-modal and multi-resolution hierarchy for multimedia, known as the *InfoPyramid* and a *customizer* that selects the best representation from the InfoPyramids to meet client capabilities and other resources while delivering the most value. Steiger *et al.* [3] introduced a personalized multimedia content delivery system dealing with both user preferences and terminal/network capabilities to provide UMA within the PERSEO project. Dogan *et al.* [10] proposed a video transcoder bank to resolve congestion and/or bandwidth limitation for mobile communications. The authors proposed an architecture with a layered structure of multiple video rates as required by various networks.

Temporal adaptation is also widely researched. A two stage frame dropping for scalable MPEG video transmission over ATM was suggested by Zheng and Atiquzzaman in [11]. In the first stage, the server drops some frames in the case of network congestion. In the second stage, the server marks low priority frames to be dropped by the network in the case of severe congestion. Cha *et al.* [12] presented the design and implementation of an MPEG filtering mechanism which drops video frames dynamically and then reconstructs a valid MPEG system stream in real-time.

In the area of spatial reduction, Shen and Roy [13] proposed an algorithm that computes the motion vectors for the downscaled video sequence directly from the original motion vectors. Yin *et al.* [14] proposed an algorithm for video transcoding by reducing the spatial resolution, where a new MPEG stream is derived with half the spatial resolution from an MPEG input stream. Lei and Georganas [15] proposed an H.263 spatial resolution downscaling transcoder.

Other works include those in the area of syntax conversion, also called heterogeneous video transcoding. An evaluation of the performance of software implementations of compressed-domain processing for the problem of MPEG to motion-JPEG transcoding is presented in [16] by Acharya and Smith. In [17], Wee *et al.* presented an algorithm for transcoding high-rate compressed MPEG-2 bit streams to lower-rate compressed H.263 bit streams. Transcoding of pre-encoded MPEG-1/2 video into H.261/H.263 standards with lower bit rates and reduced spatio-temporal resolutions is reported in [18] by Shanableh and Ghanbari. Syntax conversion can also be employed to convert from one profile to another. Lin *et al.* [19] described a transcoding technique to convert multiple layer bit streams to a single-layer format. Specifically, their work targeted MPEG-4 FGS-to-Simple Profile transcoding.

Object-based adaptation has also attracted attention recently as a result of the introduction of the MPEG-4 standard. Vetro *et al.* [20] suggested the use of metadata called *transcoding hints* to guide adaptation of objects by selecting the quantization parameter for each object and frame skip. An adaptive streaming system that exploits the object-based coding capabilities of MPEG-4 by applying priorities to individual objects was proposed by Goor and Murphy in [21].

In all the related works assessed, there are some missing issues that require further investigation:

- Reduction methods are applied across the entire multimedia content without paying attention to the special circumstances of the constituent parts.
- The use of standardized metadata to govern the adaptation process is not extensively exploited.
- Although the theoretical framework of the use of variations for adaptation is well understood, especially within the MPEG community, no major implemented work was reported that uses video variation supported with metadata as a means for adaptation and that utilizes the extensive adaptation features provided by the MPEG-4 standard. No work is reported in the area of segment-based variation as proposed in this paper.
- Content accessibility by the disabled has been mainly researched for accessing Web pages [25, 26]. We couldn't however find works related to video adaptation that take into account disability of users.

3 Segment-Based Variation

A shortcoming of the widely known reduction methods such as temporal, color, and spatial is that they are indiscriminately applied on the entire video without paying attention to the particular characteristics of its parts or end-user preferences. For instance, some parts have fast moving regions while others have stationary ones; some parts are colorful while others have fewer colors. Instead of applying the same method across the entire video, it will be advantageous in terms of minimizing quality loss and/or maximizing the gain in data size reduction to apply different methods on different parts. For instance, applying temporal reduction on a fast moving segment will have a jerky effect degrading its quality. Hence temporal reduction is better applied on stationary segments. Similarly color reduction on a region with a higher color depth provides a significant reduction in data size.

Higher compression is achieved when segments exhibit lower motion, texture, and color characteristics because of spatial and temporal redundancies. For segments having higher motion, texture and color characteristics, the corresponding data sizes will be higher because of lack of redundancies, suggesting the use of the corresponding reduction methods on such segments. But, the use of temporal reduction on segments of high motion activity results in a significant reduction of quality and hence is not recommended. Applying spatial and color reductions on segments exhibiting higher texture and color features, respectively, maximizes the reduction in data size.

What is required is, therefore, a *content-aware* methodology that pays attention to differences in physical characteristics of segments. Hence, in segment-based variation, different reduction methods are applied on different parts of a video, called *segments*. This can be referred to as *local adaptation*, in comparison to *global adaptation* where a given adaptation scheme is applied on all parts of a video.

Another importance for segment-based variation can be identified in a variable bit rate encoded video. For such a video, local adaptation may be necessary on those parts of the video that exhibit bit rates that are more than the average. Segment-based variation may also be helpful to take end-user preferences into account. A viewer may be interested to watch more action for a given clip (hence less number of reduction methods are applied on that clip to maintain the highest quality) but less for another (say if the scene depicts some terrifying action), or not at all (except to listen to the audio) if one is scared of watching some scenes.

3.1 The Process of Segment-Based Variation

After motivating the rationale behind segment-based variation, we now provide the steps in applying the proposed method (Figure 3). First the video is partitioned into segments using segmentation techniques for segment-based variation. Then optimal methods that provide the best results are selected. This process also takes into account user preferences and can utilise transcoding hints metadata. Then the appropriate reduction methods are applied on each segment. Finally the variation video and the metadata document are created. Each of these steps is detailed in the sequel.

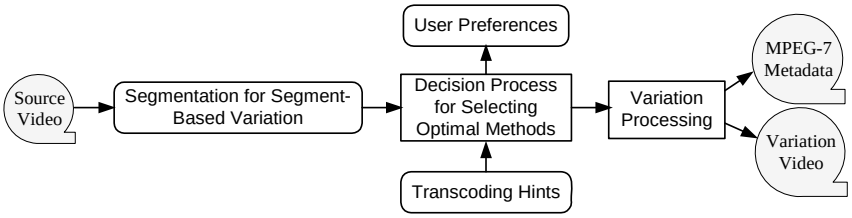

Figure 3: Sequence of steps in segment-based variation.

3.2 Video Segmentation

In general, segmentation can be done at a shot level, a scene level, or any other granularity based on application requirements. Segmentation methods are usually used in Content Based Video Retrieval (CBVR) applications. However, segmentation for the purpose of segment-based variation has different requirements than that for CBVR applications.

For CBVR applications, a video is segmented into semantically related scenes. All shots that are semantically related are grouped together to form a scene; indexing and retrieval is based on such semantically distinguished scenes. In segmentation for segment-based variation, a video is partitioned into segments based on physical characteristics such as motion, color, and texture, and segmentation does not depend on semantics. Shots that have similar physical characteristics are grouped together to form a segment. For this, a shot boundary is identified first. Then a decision is made whether or not the beginning of a new segment is declared or that this shot be merged with the previous segment. The following two factors affect the decision:

- In shot detection for segment-based variation, the physical characteristics of a shot are taken into account. If the current shot depicts similar color, motion, and texture characteristics as its predecessor, then the two will merge together and will be considered as a single segment, since there will be no reason to apply different reduction methods on such shots. This method can be generalized for any number of shots. Secondly, a segment can include shots from different scenes if they exhibit similar physical characteristics even though they differ in semantics.
- The other consideration is the length of segments. This is from the point of view of processing cost. If there are too many segments in a video, it will be time consuming to process them individually. This is specifically true if the end-user is to be involved in the decision-making process by specifying his/her viewing preferences and/or device capabilities.

3.3 Decision Process for Optimal Methods Selection

The three features of a segment that we consider will affect the decision as to which methods should be applied on which segments are the levels of motion, texture, and color. The data size of a segment is higher when these three features have higher values. For our analysis, two levels from each feature are chosen, viz., Low and High, resulting in eight combinations (see Figure 4a): LLL, LLH, LHL, LHH, HLL, HLH, HHL, and HHH, where the first letter represents the motion level, the second represents the texture level and the third represents the color level.

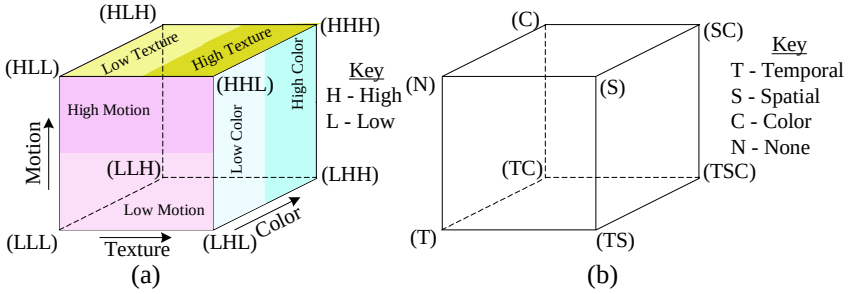

Figure 4: Level combinations of physical features and selection of optimal methods. (a) Level combinations of physical features, (b) Optimal reduction methods.

Motion can be measured in different ways. For instance, the percentage of interpolated macroblocks in B-frames can serve as a measure of motion [6]. A high number of interpolated macroblocks (inter-coded) means that a greater portion of the frame is similar to the reference frames and suggests low level of motion whereas a low number of interpolated macroblocks (intra-coded) implies that there are a greater number of changes between frames suggesting more motion. The texture level can be measured by taking the average magnitudes of the DCT coefficients for each frame of the luminance blocks and then averaging over the entire segment [22]. The color level can also be estimated similarly, but this time taking the chrominance blocks. These measures can be made during encoding or later in a post-production process during variation creation.

By considering two levels of values for each of the three physical characteristics, motion, texture, and color, Figure 4b depicts the optimum methods that are applicable on each segment. The considerations in choosing the optimum method combinations are the following:

- temporal reduction will be applied if motion is low (level combinations L**),
- spatial reduction will be applied if texture is high (level combinations *H*),
- color reduction will be applied if color is high (level combinations **H).

This will leave us with one combination, i.e., HLL, having no methods selected for it. Obviously temporal reduction is not appropriate since quality reduction will be high. However, color and spatial reductions can be applied on such a segment having in mind that the gain in data size reduction may not be relatively high. Fortunately, since the texture and color components are low, such segments are expected to have a moderate data size and applying the two methods suffices.

3.4 Metadata Support for Variation Creation

In the above discussion, the three physical characteristics were assumed to be estimated during variation creation. However, they can be estimated a priori, represented and stored as MPEG-7 transcoding hints metadata. *Transcoding Hints* are metadata that can be used to guide the transcoding process [2, 23], which in our case is equivalent to variation creation. MPEG-7 has a special description scheme for this, *MediaTranscodingHints* of the *MediaProfile* description scheme. The objective of this description scheme is to improve the quality of the transcoded video and also to reduce the complexity of the transcoding operations.

There are two descriptors currently defined in the standard that can be used in segment-based variation: *MotionHint* and *SpatialResolutionHint*, that describe motion hints for a transcoder and the maximum allowable spatial resolution reduction factor for perceptibility, respectively. However, there are no descriptors that correspond to the texture and color information that are used in this paper. Hence, we defined new descriptors, called *TextureHint* and *ColorHint*. They are elements that take real values in the unity interval [0, 1] and indicate the level of texture and color in a segment, respectively.

3.5 User-preferences and Accessibility Characteristics

User preferences and device characteristics are also important when users are involved in the decision-making process. For instance, if it is known in advance that the receiving device has display size or color limitations, then the corresponding reduction methods are not anymore optional and must be applied on all segments. Users can also specify their viewing preferences so that reduction methods from a set of methods that result with nearly the same bit rate can be selected to provide the user with the best viewing experience.

One important aspect of user preference is to take into account disability. Multimedia adaptation by means of meta-data helps users to specify their needs accordingly and the system to perform multimedia adaptation afterwards. We rely on the Source Preferences DS, as part of the User Preferences DS, to state the terminal requirements issued from disability, which is encoded along with the Variation DS in order to enhance the Variation Factory, described in the next Section. For instance, for a news-on-demand application, visually impaired people would be interested to listen to the audio. Hence, the bandwidth-hungry visual component can be removed and only the audio part delivered to the user. This will enable visually impaired users to have access to the repository of electronic information [25].

3.6 Algorithm for Segment-based Variation

The first step in applying segment-based variation is to partition the source video into its constituent segments as discussed earlier. Then a *method selection table* is constructed indicating which reduction methods are applied on which segments. The reduction methods to be applied on each segment are decided based on the motion, texture, and color levels of a segment. User preferences and device characteristics are also important considerations in the decision-making process.

It is assumed that threshold values for the three physical characteristics have been set a priori based on some heuristics, which are mostly engineering choices. The method selection table includes one entry per segment, each specifying the reduction methods to be applied on that segment.

An algorithm for discriminatory local adaptation is given in Algorithm 1. The method selection table is initialized at the beginning (line 4). The video is segmented using a shot detection method for segment-based variation (line 7) and estimates for feature values are done during this segmentation stage. Then entries for the method selection table are decided (lines 8-19). δ_M, δ_T, and δ_C that are used in the decision process are the threshold values for motion, texture, and color, respectively. The method selection table is stored in a file for later use (line 25), if required. Reduction methods are applied on each segment as per the entries in the method selection table (line 21) and the variation video is created (line 24). Finally, an MPEG-7 document is produced (line 26).

Algorithm 1: Optimal methods selection.

```
 1: begin
 2:     int segNum = 0
 3:     int tempCol = 0, spaCol = 1, colCol = 2
 4:     initializeMethodSelectionTable() // all entries of the table are initialized to false
 5:     while (not EndOFSourceVideo) do
 6:         begin
 7:             getNextSegment() // construct a segment and estimate feature values
 8:             if (texture > δT) then // i.e., texture is high
 9:                 methodSelectionTable[segNum, spaCol] = true // *H*
10:             end-if
11:             if (color > δC) then // i.e., color is high
12:                 methodSelectionTable[segNum, colCol] = true // **H
13:             end-if
14:             if (motion < δM) then // i.e., motion is low
15:                 methodSelectionTable[segNum, tempCol] = true // L**
16:             elseif (texture <= δT & color <= δC) then // HLL
17:                 methodSelectionTable[segNum, colCol] = true
18:                 methodSelectionTable[segNum, spaCol] = true
19:             end-if
20:             // apply reductions on this segment based on the look-up table
21:             applyReductionsOnThisSegment()
22:             increment(segNum)
23:         end-begin // of while
24:     createVariationVideo()
25:     saveMethodSelectionTable()
26:     createMPEG7Metadata()
27: end-begin
```

4 The Variation Factory

To realize variation creation utilizing the different variation methods under a unifying framework, a novel architecture called the *Variation Factory* is defined as shown in Figure 5. It is responsible for generating variations by making use of reduction methods or variation products and the corresponding MPEG-7 metadata document.

The input to the Variation Factory is an MPEG-4 video and the outputs are: (1) one or more MPEG-4 variation videos, and (2) an MPEG-7 metadata document that describes the source and the variations. Relevant descriptors that include low-level features and semantic information such as *fidelity* and *priority* values of variations are extracted automatically. The *fidelity* value specifies the value of the fidelity of the variation with respect to the source. It is similar, but not exactly the same as

quality. The *priority* value specifies the value of the priority of the variation with respect to the other variations that may be specified for the same source video. The chain of applying variation methods as shown in Figure 5 is not necessarily sequential. It is possible for a video to pass through only one method or more than one as shown by the vertical bi-directional hollow lines. For instance, it is possible to apply only spatial variation and bypass all the others. It is also possible for a video to pass through more than one variation method, say for example, spatial variation followed by temporal variation, etc. The order in which methods are applied is not also important. By applying reduction methods individually and in combination, many variations can be created from a single source. In general, given n variation methods that can be applied on a source video, V_S, the total number of possible variation videos is given by

$$T_n(V_s) = \sum_{i=1}^{n} \binom{n}{i} \quad (1)$$

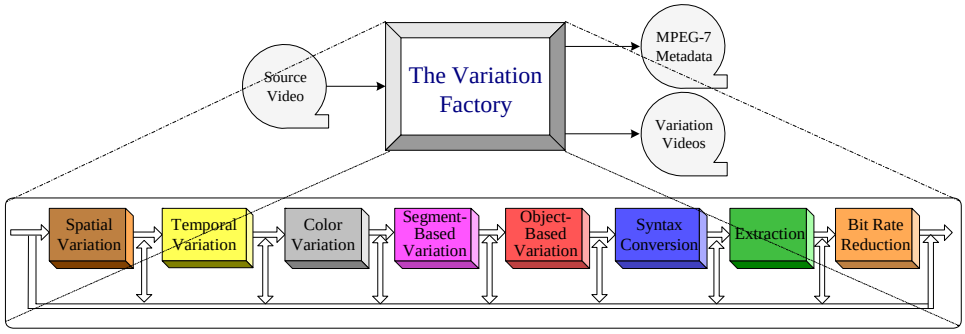

Figure 5: The Variation Factory.

If we take the three variation methods, namely temporal, spatial and color, the tree in Figure 6. shows all possible variations that would be obtained by applying all combinations of variation methods, where the root node is the source. Using Eq. 1, there are 7 variation videos.

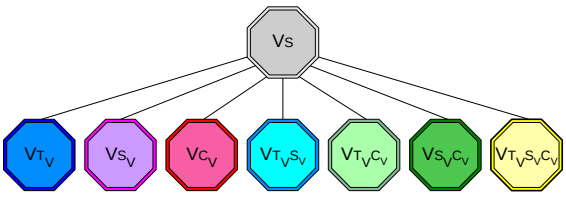

Figure 6: A tree of variation videos from the Variation Factory.

4.1 The Multi-Step Variation Factory

In the case of the *Variation Factory* introduced above, there is always one source video to all the variation videos in a variation set. A source video may be subjected to more than one variation method in successive steps, but there are no intermediary videos that also serve as variation videos. The *Multi-Step Variation Factory* or *Variation Tree* extends this concept by allowing the intermediary videos to be variations in their own right and then serve as sources to further variations thereby creating different variations with finer granularity in terms of resource requirements. In this case, variation videos within the same variation set will have different sources.

The *Multi-Step Variation Factory* is diagrammatically shown in Figure 7. It extends the *Variation Factory* and is, therefore, its superset. Once a variation video is created by piping it through the different variation methods in the *Variation Factory*, it can then serve as a source to further rounds of variation video creation. This is shown by the directed dashed line. Creating a tree of variations allows us to fulfil the two use case scenarios outlined in Section 1.

Unlike the *Variation Factory*, there is now a tree of variations as shown in Figure 8 for three variation methods. The major difference with the tree of variations created by the Variation Factory (Figure 6) is that internal nodes together with leaf nodes serve as variation videos and that some variation videos are used as source for other variations (hence multi-step). Nodes with the same color (shading) represent variations that are similar since they provide the same viewing experience to the viewer. This is because they are obtained from the same set of methods, but in different order.

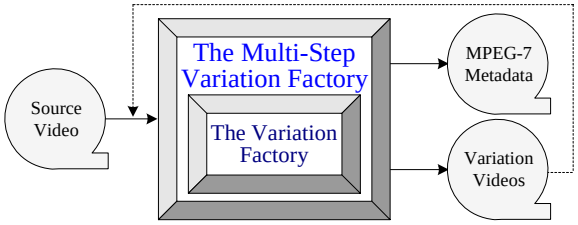

Figure 7: The Multi-Step Variation Factory.

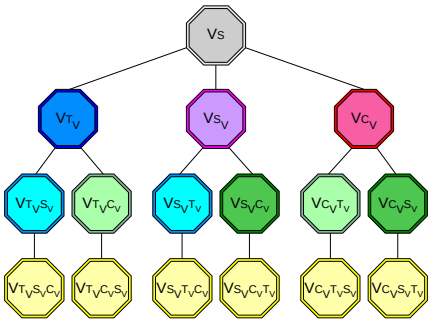

Figure 8: A tree of variation videos from the Multi-Step Variation Factory.

5 Implementation

Segment-based variation and the related products are implemented in a prototype system module for variation creation called the Variation Processing Unit (VaPU). It is a server component in charge of generating variations and the corresponding MPEG-7 metadata documents. A screenshot is shown in Figure 9. It logically lies between the media- and meta-databases and has four major components.

- The *User Interface* communicates with the user to get inputs. These inputs are passed as parameters to the *Variation Processor* and the *MPEG-7 Document Processor*.
- The *Variation Processor* creates the variations. It also extracts and gathers information about the source video and the variations and avails this information to the MPEG-7 Document Processor.

- The *MPEG-7 Document Processor* creates MPEG-7 documents with the appropriate MPEG-7 descriptors for the source and the variation videos. The values for the descriptors are collected during variation creation and supplied to this module by the Variation Processor.
- The *Output Sampler* is a temporary facility used to play/display variation processing results.

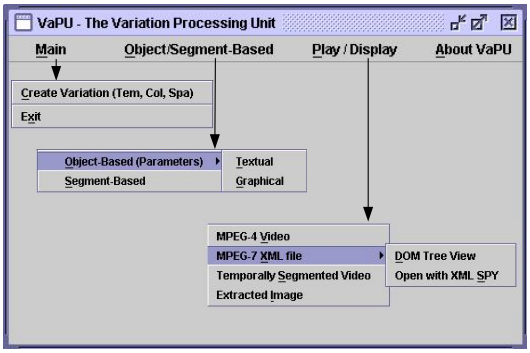

Figure 9: The entry screenshot of VaPU.

5.1 Metadata Description

The *VariationSet* DS of MPEG-7 is used to represent the associations between different variations of multimedia resources. The major objective of the *VariationSet* DS is to allow the selection of the most suitable variation which can be used instead of the original to adapt to the different capabilities of terminal devices, network conditions and user preferences [2].

A *VariationSet* is an aggregation of a source and one or more variations. A variation has the attributes of *fidelity*, *priority*, and *relationship*. The variations may result from various types of multimedia processing such as summarization, reduction, transcoding, etc. The important descriptions include fidelity, data size, priority and type of relationship. During delivery, the information contained in the MPEG-7 document will help the system to identify the most appropriate variation that meets the required QoS. They can also be transmitted to an adaptation engine on the network or to end-users so that they may carryout the adaptation more efficiently.

6 Experimental Results

Table 1 shows data size ratios and fidelity values for segment-based variation. The fidelity values were calculated using the methodology introduced by MPEG-7 [24]. It is based on the media attributes of the variation and the source audiovisual data and is defined as follows.

For the variation method X_V on the ith source video, let *a* denote the source video and *b* the variation video, then the *fidelity* of *b* with respect to *a*, denoted by $q_i^{x_v}$, is calculated as:

$$q_i^{x_v} = \frac{1}{7}\left(\frac{b.hasVideo}{a.hasVideo}+\frac{b.hasAudio}{a.hasAudio}+\frac{b.DataSize}{a.DataSize}+\frac{b.FrameRate}{a.FrameRate} + \frac{b.SampleRate}{a.SampleRate}+\frac{b.SpatialSize}{a.SpatialSize}+\frac{b.Colors}{a.Colors}\right) \quad (2)$$

where *hasVideo* and *hasAudio* are binary values ($\in\{0,1\}$) and it is assumed that the source video has both a video and an audio track: *a.hasVideo* = *a.hasAudio* = 1.

Since different reduction methods are applied on different segments, the fidelity values for segment-based variation are calculated differently. First fidelity values for each segment, denoted by q^{Si}, are calculated using Eq. 2. Then the overall fidelity is calculated as a weighted mean of the fidelity values of its segments. Let w_i be the weight assigned to each segment (relative to its size). Then,

$$w_i = \frac{number\ of\ \ frames\ in\ segment\ i}{total\ number\ of\ \ frames\ in\ the\ source\ video}$$

Then, fidelity of the variation is given by (k refers to the number of segments):

$$q_i^{X_v} = \sum_{i=1}^{k} w_i q^{si} \tag{3}$$

File size has decreased by an average of 30% while the average quality reduction is 26% (the last row of Table 1). The average values may not be representative since the extent of variation methods applied on each segment can vary significantly because of differences in the number of segments among source videos as well as the size of segments. Hence, the resulting file size and fidelity values also vary accordingly. For instance, the number of segments varies from 7 to 18 (Table 1). But the result is sufficient to demonstrate the gains that would be obtained from segment-based variation.

Table 1: File size ratio and fidelity values for segment-based variation.

Video	Source file size (KB)	Variation file size	File size ratio	Fidelity	Number of Segments
spiel04	32,217	17,176	0.53	0.72	18
spiel05	49,990	34,957	0.70	0.74	17
spiel06	23,400	17,860	0.76	0.75	8
spiel07	27,975	21,950	0.78	0.76	9
spiel08	31,061	22,668	0.73	0.74	7
average			0.70	0.74	

In order to evaluate the performance of segment-based variation, its results were compared with the results obtained when reduction methods are applied across the whole video separately. The result is shown in Table 2 for one of the source videos. Similar results are obtained for the others and are not shown here. Segment-based variation performs better than temporal and color reductions. It is rivalled only by spatial reduction which always results in higher file size reduction. This is at the expense of a high loss of visual perceptual quality. The importance of segment-based variation is that not all the segments have reduced spatial resolution, thereby reducing the loss in quality.

Table 2: File size ratio and fidelity values for the spiel04 video.

Variation Video	File size (KB)	File size ratio	Fidelity	Methods applied
spiel04_sbv	17,176	0.53	0.72	Segment-Based
spiel04_tem	22,976	0.71	0.80	Temporal
spiel04_col	27,305	0.85	0.80	Color
spiel04_spa	12,939	0.40	0.66	Spatial

7 Conclusion

This paper presented a novel method of variation creation, called segment-based variation. It aims to apply different reduction methods on different segments of a video based on physical characteristics. The objective is to minimize the reduction in quality and/or to maximize the reduction in data size. The first task in applying the method is to partition a video into homogeneous segments in terms of physical characteristics. Segmentation for segment-based variation is different from segmentation for CBVR applications. Whereas segmentation for CBVR applications focuses on semantics, segmentation for segment-based variation is based on physical characteristics. An algorithm for selecting the optimal combination of methods to be applied on each segment was presented.

The proposed method is implemented alongside other known methods. A unifying framework, called the Variation Factory, was developed. It is responsible for generating variations and the corresponding MPEG-7 metadata document. It is extended to the Multi-Step Variation Factory where intermediary videos serve as variations and also as sources to further variations to support certain application scenarios. A prototype called VaPU is developed to implement the proposed solutions.

Relevant metadata were automatically generated during variation creation and described using MPEG-7 descriptors. Such metadata are useful for content adaptation. MPEG-7 was selected as a metadata description standard since it provides extensive descriptors which allow standardized communication among components on the delivery path. Hence, important MPEG-7 description schemes and descriptors that are used to describe both the source and the variation videos were identified and implemented.

Future work includes consideration of more than two levels for the physical characteristics for more refined variations and how the end-user can be involved in the variation creation process by specifying his/her preferences, since user preferences are emulated in the current implementation.

References

[1] R. Mohan, J. R. Smith, and C.-S. Li, "Adapting Multimedia Internet Content for Universal Access," *IEEE Transactions on Multimedia*, Vol. 1, No. 1, 1999, pp. 104-114.

[2] Harald Kosch, "Distributed Multimedia Database Technologies supported by MPEG-7 and MPEG-21," CRC Press, ISBN 0-8493-1854-8, November 2003.

[3] O. Steiger, D. M. Sanjuán, and T. Ebrahimi, "MPEG-Based Personalized Content Delivery," in *Proceedings of the IEEE International Conference on Image Processing, ICIP 2003*, Barcelona, Spain, September 2003, pp. 45-48.

[4] H. Kosch, L. Böszörményi, M. Döller, A. Kofler, P. Schojer, and M. Libsie, "The life cycle of multimedia meta-data," *IEEE MultiMedia*, 12(1), January/March 2005.

[5] B. S. Manjunath, P. Salembier, and T. Sikora, "*Introduction to MPEG-7 Multimedia Content Description Interface*," John Wiley & Sons, New York, 2002.

[6] A. Tripathi and M. Claypool, "Adaptive Content-Aware Scaling for Improved Video Streaming," in *Proceedings of the Second International Workshop on Intelligent Multimedia Computing and Networking (IMMCN)*, Durham, March 2002.

[7] R. Mohan, J. R. Smith, and C.-S. Li, "Adapting Content to Client Resources in the Internet," *IEEE International Conference on Multimedia Computing and Systems*, Vol. 1, Florence, Italy, June 1999, pp. 9302-9307.

[8] J. R. Smith, R. Mohan, and C.-S. Li, "Transcoding Internet Content for Heterogeneous Client Devices," in *Proceedings of IEEE International Conference on Circuits and Systems (ISCAS)*, Monterey, June 1998, pp. 599-602.

[9] J. R. Smith, R. Mohan, C.-S. Li, "Scalable Multimedia Delivery for Pervasive Computing," in *Proceedings of ACM International Conference on Multimedia (ACM-MM)*, Orlando, FL, November 1999, pp. 131-140.

[10] S. Dogan, A. H. Sadka, and A. M Kondpz, "MPEG-4 video transcoder for mobile multimedia traffic planning," in *Proceedings of the IEEE Second International Conference on 3G Mobile Communication Technologies, 3G'2001*, London, March 2001, pp. 109-113.

[11] B. Zheng and M. Atiquzzaman, "TSFD: two stage frame dropping for scalable video transmission over data networks," *IEEE Workshop on High Performance Switching and Routing*, Dallas, TX, May 2001, pp. 43-47.

[12] H. Cha, J. Oh, and R. Ha, "Dynamic Frame Dropping for Bandwidth Control in MPEG Streaming System," *Multimedia Tools and Applications*, Vol. 19, No. 2, February 2003, pp. 155-178.

[13] B. Shen and S. Roy, "A very fast Video Spatial Resolution Reduction Transcoder," in *International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, May 2002.

[14] P. Yin, M. Wu, and B. Lui, "Video transcoding by reducing spatial resolution," in *Proceedings of IEEE International Conference on Image Processing*, Vancouver, October 2000.

[15] Z. Lei and N. D. Georganas, "H.263 Video Transcoding for Spatial Resolution Downscaling," in *Proceedings of IEEE International Conference on Information Technology: Coding and Computing 2002*, Las Vegas, USA, April 2002, pp. 425-430.

[16] S. Acharya and B. Smith, "Compressed Domain Transcoding of MPEG," in *International Conference on Multimedia Computing and Systems (ICMCS 1998)*, June 1998, pp. 295–304.

[17] S. Wee, J. Apostolopoulos, and N. Feamster, "Field-to-frame transcoding with spatial and temporal downsampling," in *International Conference on Image Processing (ICIP 1999)*, Vol. 4, October 1999, pp. 271-275.

[18] T. Shanableh and M. Ghanbari, "Heterogeneous video transcoding to lower spatio-temporal resolutions and different encoding formats," *IEEE Transactions on Multimedia*, Vol. 2, No. 2, June 2000, pp. 101-110.

[19] Y. C. Lin, C. N. Wang, T. Chiang, A. Vetro, and H. Sun, "Efficient FGS-to-single layer transcoding," in *Proceedings of IEEE International Conference on Consumer Electronics*, Los Angeles, June 2002, pp. 134-135.

[20] A. Vetro, H. Sun, and Y. Wang, "Object-based transcoding for adaptable video content delivery," *IEEE Transactions on Circuits and Systems for Video Technology*, Vol. 11, No. 3, March 2001, pp. 387-401.

[21] S. Goor and L. Murphy, "An Adaptive MPEG-4 Streaming System based on Object Prioritisation," in *Proceedings of Irish Signals and Systems Conference 2003*, Limerick, Ireland, July 2003.

[22] A. M. Dawood and M. Ghanbari, "Content-Based MPEG Video Traffic Modelling," *IEEE Transactions on Multimedia*, Vol. 1, No. 1, March 1999, pp. 77-87.

[23] P. Kuhn, T. Suzuki, and A. Vetro, "MPEG-7 transcoding hints for reduced complexity and improved quality," in *Proceedings of International Packet Video Workshop*, Kyongju, Korea, April 2001.

[24] MPEG MDS Group, "MPEG-7 Multimedia Description Schemes XM," *ISO/IEC JTC1/SC29/WG11 N3966*, Singapore, Mar. 2001.

[25] S. Harper and S. Bechhofer, "Semantic Triage for Increased Accessibility". In John J. Ritsko et al, editor, *IBM Systems Journal*, Vol.4, No. 3, August 2005, pp. 637-648.

[26] P. Plessers, S. Casteleyn, Y.z Yesilada, O. De Troyer, R. Stevens, S. Harper, and C. Goble, "Accessibility: A Web Engineering Approach". *In Proceedings of the 14th International Conference on World Wide Web (WWW 2005)*, Chiba, Japan, 2005, pp. 353-362.