Multimedia Tools and Applications, 23, 281–298, 2000

Adaptive Multimedia Presentation Strategies

B. PRABHAKARAN prabha@comp.nus.edu.sg
Department of Computer Science, School of Computing, National University of Singapore, Singapore 119260

Abstract. Multimedia presentations comprise various media objects such as text, audio, image, and video that are delivered to users according to certain temporal relationships. In stored multimedia presentations, these temporal relationships are explicitly formulated by the author(s) and stored along with the presentations. However, it is difficult to ensure that these temporal relationships are always strictly preserved in real-time, distributed multimedia presentations. This is due to the fact that various components of the run-time environment such as operating system and network may offer only *best effort* services, i.e., they may not be able to provide any real-time guarantees. In this paper, we survey the different approaches that can be used for adapting multimedia presentations to handle instances where temporal relationships cannot be preserved in a strict manner. We classify these approaches into three categories and discuss when these categories of adaptations can be used.

Keywords: multimedia presentations, temporal models, multimedia synchronization, flexible multimedia presentations, adaptive multimedia presentations, filters for multimedia presentations

1. Introduction

Multimedia presentations comprise diverse media objects such as text, image, audio, and video. In this work, we consider stored/orchestrated multimedia presentations where media objects are retrieved from secondary/tertiary storage systems. Figure 1 describes the system view of such a typical multimedia presentation and the author's view of a multimedia presentation is shown in figure 2. Media objects are retrieved from storage devices and delivered over communication network, according to presentation structure designed by author(s) of the presentation. This retrieval and delivery procedure requires certain resources such as disk bandwidth, buffer (both at server and client systems), and network bandwidth. Network connection might involve very high bandwidth channels such as optic fibres or very low ones such as modems/wireless access. A presentation server might be supporting several presentations and hence required disk bandwidth may not be available. In other words, support for a distributed multimedia presentation is constrained by resource constraints such as availability of disk bandwidth, network bandwidth, and buffer resources.

In addition, a multimedia presentation is also constrained by author's and user's preferences, as shown in figure 2. Author(s) can introduce access constraints for different categories of viewers. For example, a multimedia presentation on a war situation can have different categories such as the President, General, and soldier. Depending on the category, the structure and the content of the presentation may vary. Users' can also have their own set of preferences by choosing different window sizes, window positions, and by excluding certain media objects. For instance, a user might want to mute the audio channel or drop video channel. These set of preferences exercised by authors and users lead to different set of views of the same presentation for different users.

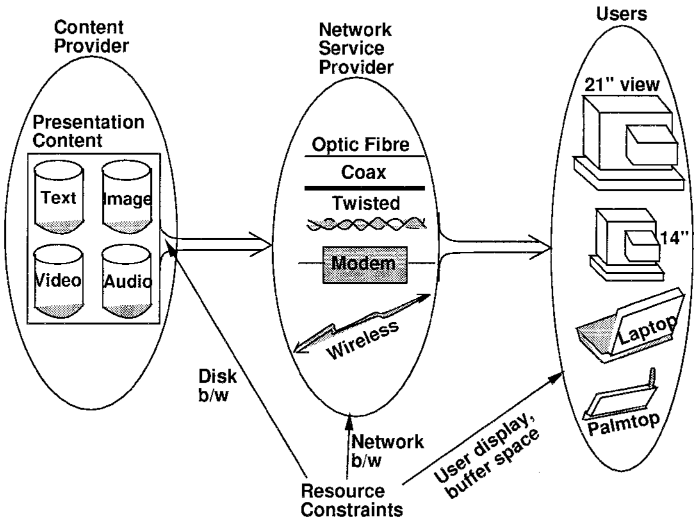

Figure 1. System view of multimedia presentation.

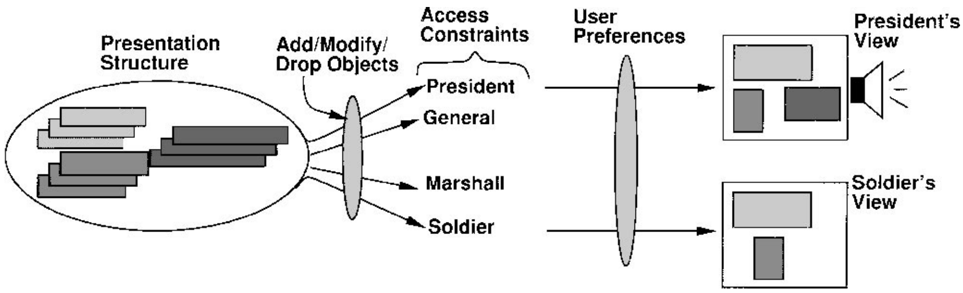

Figure 2. Author's view of multimedia presentation.

1.1. Adaptation approaches

A multimedia presentation, therefore, has to adapt to resource constraints as well as authors' and users' preferences. We identify three possible adaptation approaches, as described in figure 3.

1. *Structure adaptation*: helps to adapt the multimedia presentation structure based on resource availability, access constraints, and user preferences. This approach is generally feasible only when the author(s) of the presentation have allowed for such flexibility.
2. *Content adaptation*: is used for modifying media contents (e.g., MPEG-2/MPEG-4 video, JPEG image) based on resource availability and/or user preferences.
3. *View adaptation*: approach can be used for modifying the spatial/temporal view of the multimedia presentation based on access constraints and user preferences. This approach can also be used to adapt to resource availability, e.g., an object arriving later than its presentation time due to delay or lower network bandwidth.

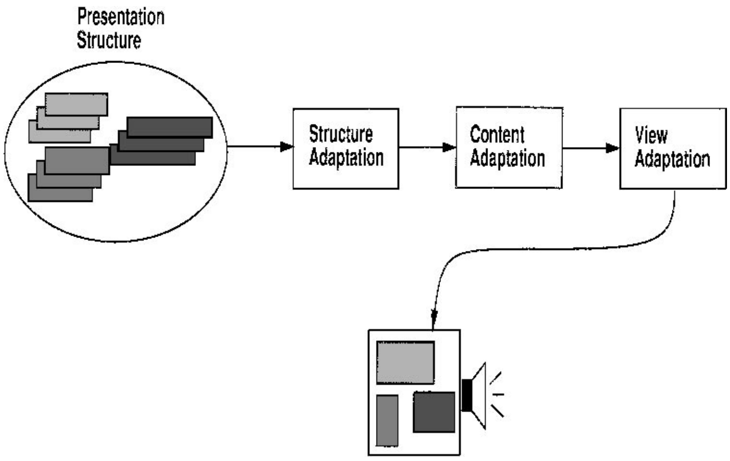

Figure 3. Adaptation approaches.

It should be noted there can be a degree of overlap among these approaches. For instance, view adaptation for handling user preference to mute audio delivery can be interpreted as structure adaptation. In this paper, we survey the different strategies that can used for the above approaches. The rest of the paper is organized as follows: structure adaptation is discussed in the following section. Content adaptation is described in Section 3, followed by view adaptation in Section 5. Possibilities for increasing resource availability are presented in Section 6 and then we summarize our discussion in Section 7.

2. Structure adaptation

We can consider a simple, example multimedia presentation. Figure 4 illustrates the media streams representation of the example, comprising two streams of information: audio (A) and motion video (V). The temporal characteristics of the presentation is such that audio and video objects are synchronized for every two audio objects. Multimedia presentations also have a spatial characteristic that describes the position of windows on user's monitor. Structure adaptation of a multimedia presentation can involve both temporal and spatial adaptation. Since spatial adaptation is relatively easier, we focus temporal structure adaptation in this adaptation. Temporal adaptation is easily possible when we have a flexible temporal specification. In a flexible temporal specification, the time instants and durations of presentations of objects are allowed to vary as long as they preserve certain specified

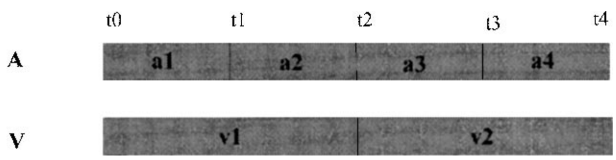

Figure 4. An example of multimedia presentation.

relationships. For instance, we can consider a specification such as: Start showing the image *sometime between* 9.58 am and 10.03 am and show it till the audio is played out. The advantage of a flexible temporal specification is that the temporal structure of a presentation can be adapted based on the system delays. For example, the image presentation (in the above specification) can be fixed at 10.03 am in case of high system delays and at 9.58 am in case of low system delays. System delays can be due to network load or server load. Now, we discuss some flexible temporal specification techniques for structure adaptation: Timed Petri nets and time flow graphs.

2.1. Time stream petri nets

Time Stream Petri Nets (TSPN) has been proposed in [17, 18] for modeling of adaptive multimedia presentations. In TSPN, time durations associated with object presentations are described using a *temporal validity interval* (TVI) specified by the 3-tuples (x, n, y), where x, n, y, denote the minimum, nominal, and maximum admissible durations of a multimedia object presentation. (For hard temporal specifications, TVI can be specified by the tuple (n, n, n)). Synchronization relationship among object presentations are described using a set of possible relationships such as:

1. *Strong-or* defining the earliest processing duration, i.e., the first arc that gets the maximum bound of its associated TVI.
2. *Weak-and* specifying the latest processing duration, i.e., the last arc that gets the maximum bound of its TVI.
3. *Master* denoting a selected processing duration, i.e., only one of the selected TVI is taken into account.

The entire set of synchronization relations and their influence on TVI for media object presentation are described in Table 2.1.

Synch. relationship	Time interval
1. *and*	$[\max_i(x_i), \max(\min_i(y_i), \max_i(x_i))]$
2. *weak-and*	$[\max_i(x_i), \max_i(y_i)]$
3. *or*	$[\min_i(x_i), \max_i(y_i)]$
4. *strong-or*	$[\min_i(x_i), \min_i(y_i)]$
5. *master*	$[x_m, y_m]$
6. *or-master*	$[\min_i(x_i), y_m]$
7. *and-master*	$[\max_i(x_i), \max(y_m, \max_i(x_i))]$
8. *strong-master*	$[x_m, \max(\min_i(y_i), x_m)]$
9. *weak-master*	$[x_m, \max_i(y_i)]$

Structure adaptation using TSPN: Synchronization relationships specified in TSPN can be used for describing structure adaptivity. Figure 5 describes different realization of presentations based on the TSPN specification.

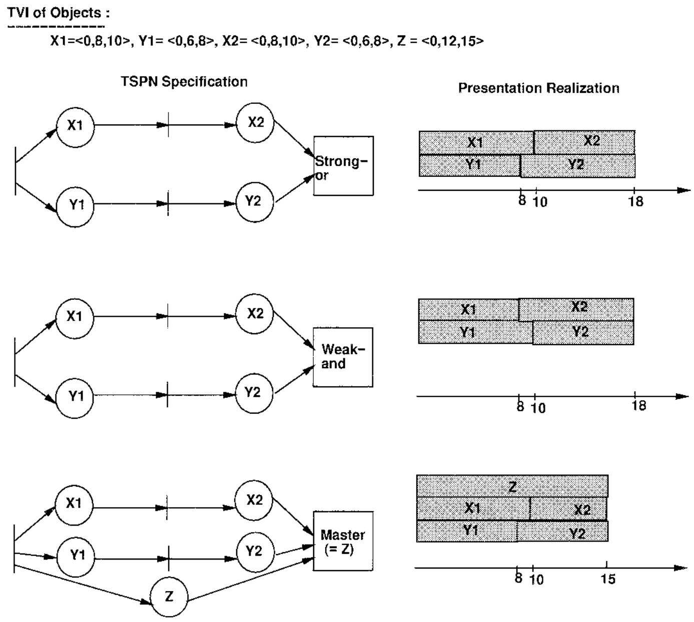

Figure 5. Adaptive presentations using TSPN models.

2.2. *Time-flow graph model*

A Time-flow Graph (TFG) model for describing fuzzy temporal relationships has been proposed in [14, 15] where *temporal interval* is taken as a primitive. The terms *object X* and *interval X* are considered synonymous. One temporal interval may contain several objects in different media, i.e., several concurrent multimedia intervals. There are thirteen relations (seven relations and their inverses) that are considered between any two temporal intervals [1]. These relations are described by the following primitives:

- '**m**eets', '**b**efore', '**e**qual', '**o**verlap', '**d**uring', '**s**tart', and '**f**inish'.
- Other relations are their inverse, e.g., '**di**' is the inverse of the relation '**d**' (during).

These primitives are described by the following set R $R = \{b, e, m, o, d, s, f, bi, mi, di, oi, si, fi\}$. The temporal relationships among the involved objects can be parallel or sequential.

2.2.1. Sequential Relations in TFG. The sequential relation between any two intervals can be either 'meets' (**m**) or 'before' (**b**). In teleorchestra applications, multiple intervals can be involved in sequential relation requirement. Hence, the sequential relation specification provided in TFG are:

1. $A\{B\}$: Interval(s) in B will start after all the intervals in A are finished.
2. $\langle A\rangle B$: Interval(s) in B will start when one of the intervals (the first) in A is finished.

2.2.2. Parallel relations in TFG. A subset R_f ($R_f \in R$) is defined as $R = \{e, o, d, s, f, oi, di, si, fi\}$. Considering two object intervals X and Y, X(r)Y describes the temporal relationship between the two intervals. When r = ‘s’, the relation X(s)Y specifies that object X and Y are to be displayed with the same start time. In a teleorchestra application scenario, the temporal relationships have to be specified despite the lack of duration information of the involved intervals. The presentation semantics are hence defined in TFG models as follows [14, 15].

1. X(d)Y(d)Z: All the other objects are displayed during the presentation of the object with the longest presentation duration.
2. X(e)Y(e)Z: The display of all the objects are started simultaneously. The presentation of all the other objects will be cut off when one of the objects (first) is finished.
3. ch(X,Y,Z) = Y: The presentation duration of all the involved objects should equal the one chosen. Objects are displayed according to the relations $r \in R_f$ specified between every two of them. Presentation of some of the objects might be cut off to equal the chosen duration.

Hence, three duration specifications $r \in R_m = \{d, e, ch\}$, can be applied to presentations involving concurrent multiple intervals.

Structure adaptation using TFG: Figure 6 describes different possible presentation structures depending on the TFG relations such as “{ }”, “<>”, “e”, “d”, and “s”.

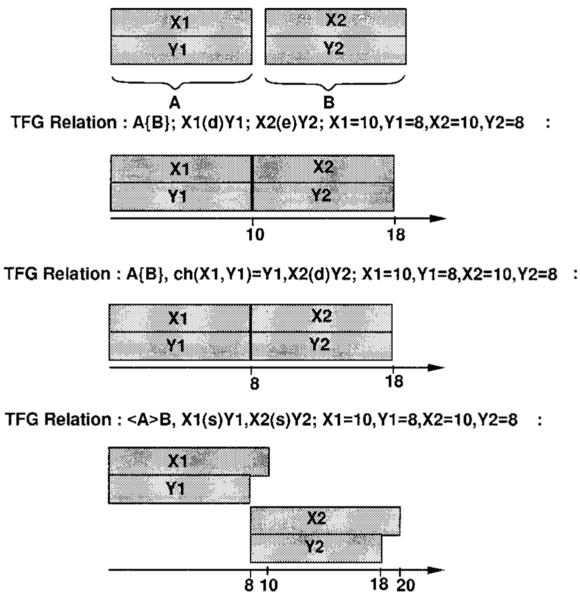

Figure 6. TFG example.

2.3. Summary

Structure adaptation is mainly achieved by flexibly specifying the temporal requirements of a multimedia presentation. Flexibility in the temporal requirements can be used to adapt to system delays such as delays due to network or server load. In the is section, we discussed techniques based on Petri nets and Time-flow graphs for structure adaptation. Techniques described in [5, 7–9, 16] also facilitate structure adaptation using approaches including difference constraints for flexible temporal specification. Now, we proceed to discuss content adaptation.

3. Content adaptation

Content adaptation is achieved by modifying the *quality* of a media object so that it can be delivered over the network with the available bandwidth. Quality of a media object refers to the resolution of the object and, if applicable, the rate of presentation (temporal quality) as in video objects. Content adaptation is normally resorted to only when structure adaptation is not possible for overcoming resource constraints. Sometimes access constraints (pertaining to different user categories) or user preferences might require content adaptation also. For instance, user can request for a HDTV quality video object or access constraint can force the viewer to have a low resolution video. Content adaptivity depends on the type of media compression technique employed. In this section, we discuss content adaptation possibilities for MPEG-2/MPEG-4 (Moving Pictures Experts Group standards) video and JPEG (Joint Photographic Experts Group) images.

3.1. MPEG-2/MPEG-4 video

MPEG-2/MPEG-4 encoding provides for simultaneous representation of video at different levels of quality. MPEG-2 uses block-based for spatial and temporal redundancy reduction involving the following steps, as shown in figure 7:

1. *Transform Coding*: converts pixel values into frequency coefficients. Transform coding is achieved by Discrete Cosine Transform (DCT).
2. *Vector Quantization*: scale frequency coefficients to help compression, e.g., divide low frequency coefficients by 10 and high frequency ones by 50.
3. *Entropy encoding*: (e.g.,) series of six 3's can be represented as !63.

MPEG-2/MPEG-4 makes available multiple qualities of video as multiple independent bitstreams. This is achieved through the use of hierarchical coding process: a coarse representation or *base layer* is first constructed and then successive enhancements are provided, as shown in figure 8/refscalenc. Enhancements are based on the coarse representation and only encode the incremental changes to improve the quality. MPEG-2/MPEG-4 offers the following scalability modes:

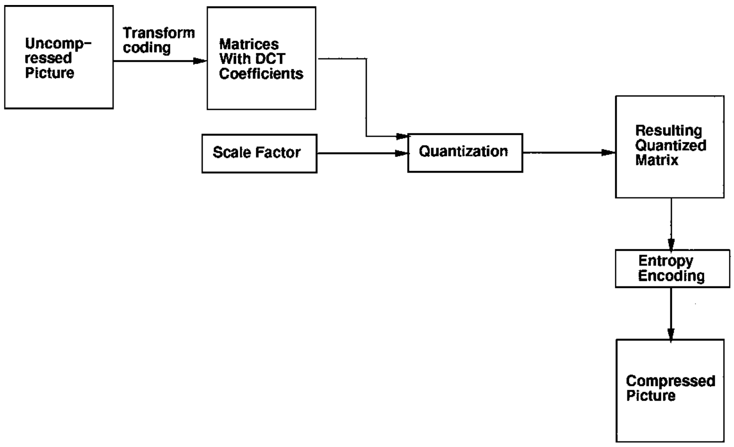

Figure 7. MPEG-2 quantization process.

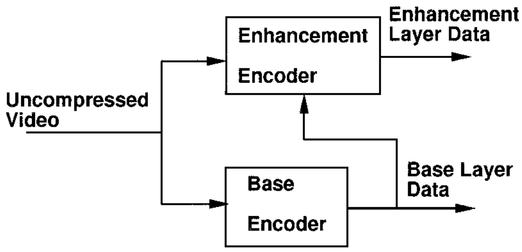

Figure 8. Layered MPEG-2 video.

- *Spatial*: Here, different spatial resolutions, i.e., frame sizes are provided.
- *Temporal*: Here, the frame rate (i.e., the number of frames per second) is scaled (e.g., from 30 frames/s to 60 frames/s).
- *SNR*: SNR scalability refines the quantisation process carried out in the base layer.
- *Depth*: scalability handles the stereoscopic vision process where base layer and enhancement layer carry the view of left and right eyes each.
- *Object*: MPEG-4 and above offer object scalability modes, where it is possible to scale the quality of individual objects in a video frame.

Spatial and temporal scalability: Spatial scalability supports different picture resolutions (on the X and Y axis), in a single video stream. For instance, MPEG-2 video can support spatial resolutions of 720×576 and 1440×1152 video on the same stream. Temporal scalability handles different frame rates in a single video stream. Enhancement layer uses base layer information to generate higher frame rates. Process mainly affected by spatial scalability is motion compensation, as shown in figure 9. For higher quality, enhancement layer data can be combined with base layer data. This can be done *after*

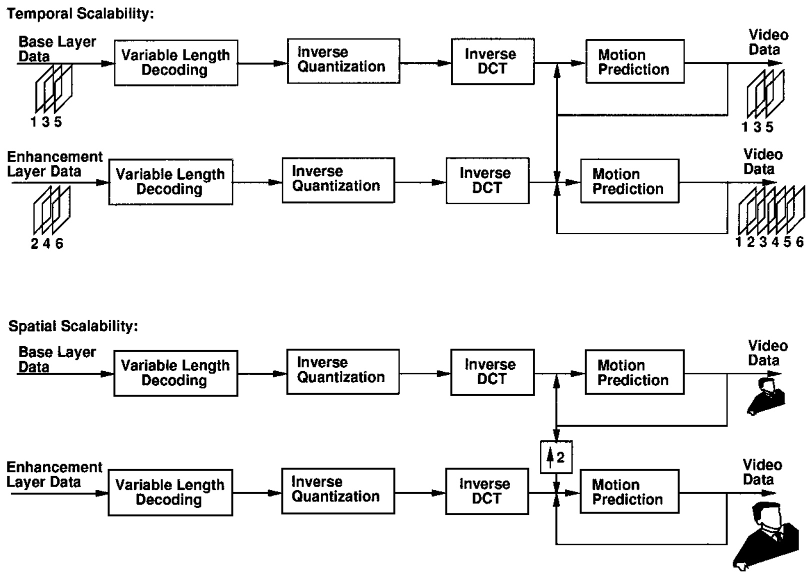

Figure 9. Spatial and temporal scalability process.

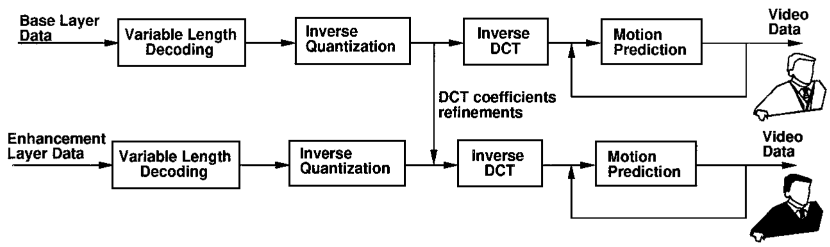

Figure 10. SNR scalability process.

Inverse DCT step. Temporal scalability is similar to spatial one, affects mainly the motion compensation process. Hence, combination is done after Inverse DCT.

SNR scalability: SNR scalability handles atleast 2 different video qualities with base and enhancement layer having the same spatial resolution. Enhancement process is done after the inverse quantization process, *before* inverse DCT, as shown in figure 10. Enhancement layer contains mainly DCT coefficients that are added to the one provided by the base layer.

Depth scalability: is achieved by emulating the stereoscopic vision of human eyes, as shown in figure 11. Base layer carries the left view and the enhancement layer carries the right view. Depth scalability is achieved using temporal scalability tools, and a hybrid prediction of motions and disparity. Depth scaling in MPEG-2 is also called *Multi-View Profile* (MVP) and can have 4 levels (high, high 1440, main, low). Depth scaling has applications in surgery, diagnosis, education, and entertainment.

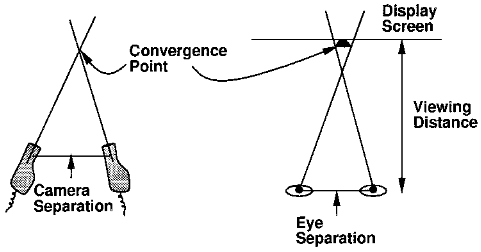

Figure 11. Depth scalability process.

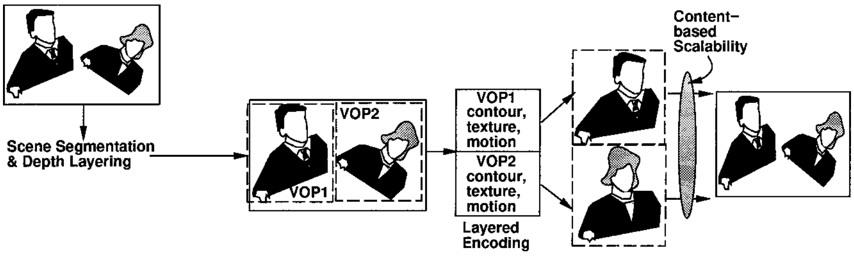

Figure 12. Object scalability in MPEG-4.

Object scalability: is provided as part of MPEG-4 standards. Here, a video frame is segmented to identify the object components. This segmented frame yields virtual object planes (VOPs), as shown in figure 12. These VOPs can be scaled independently based on the needs of an application. Object scalability can be combined with other types of approaches. For instance, figure 13 describes the combination with temporal scalability. Here, in Type 1, Layer 0 is entire frame at a lower rate and the enhancement layer is at a higher rate. In Type 2, background layer is at reduced rate and enhancement layer has objects at different frame rates. These approaches can involve spatial scaling too. These types of scalable approaches are especially useful in applications such as video telephony where background may have less importance compared to the person speaking. In such applications, objects can be selectively scaled to optimize the use of available bandwidth.

Table 1 gives an idea of the bit rates associated with different scalable modes in MPEG-2. The bit rates gives an idea regarding the type of profiles and scalable modes that can be

Table 1. MPEG-2 layers.

Layer name	Profile	Frame size	Bit rate	Visual QoS
Base layer (B1)	Main	304 × 112	0.32Mbps	VHS
Enhancement layer 1 (E1)	Spatial	608 × 224	0.83Mbps	Enhanced VHS
Enhancement layer 2 (E2)	SNR	608 × 224	1.85Mbps	Laser Disc

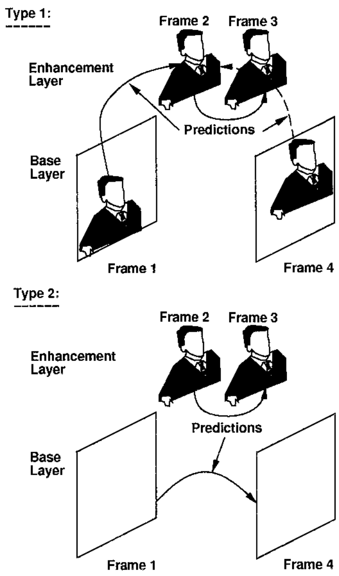

Figure 13. Object-based temporal scalability in MPEG-4.

used for different applications based on factors such as available network bandwidth and user system characteristics (resolution, size of window, etc.).

4. JPEG

JPEG provides for a hierarchical mode of operation by introducing a *pyramidal* structure of encoding. In this mode, an image is encoded using multiple resolutions, each differing from its adjacent encoding by a factor of two either in the horizontal or vertical dimension or both. Rest of the steps in hierarchical encoding are as follows (ref. figure 14).

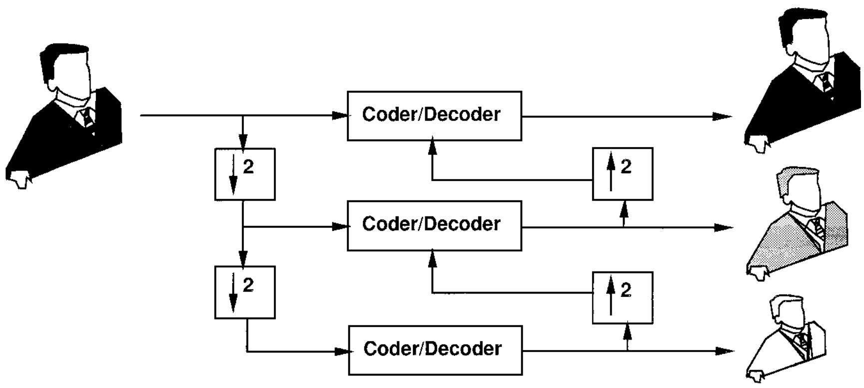

Figure 14. Scalability in JPEG.

1. First step in this process is to down-sample the image by the desired number of multiples of 2 in each dimension.
2. Second step is to encode the reduced-size image using DCT, as is normally done in JPEG.
3. This reduced-size image is then decoded and up-sampled by 2 horizontally and/or vertically. This up-sampled image is used as a prediction of the image at this resolution. The difference of this up-sampled image and the down-sampled image of the same resolution is then encoded using DCT. This step is repeated until the full resolution of the image has been encoded.

Hierarchical encoding of images in JPEG helps users to view a very high resolution image through low resolution devices. There is some reduction in resource requirement for delivering a lower resolution image also. However, this reduction may not be that significant.

4.1. Content adaptation approaches

The above techniques for content adaptivity are dependent on the characteristics of the media objects. Based on the media characteristics, content adaptation can be of the following types.

1. *Intrinsic Adaptation*: This is dependent on the encoding standard adopted for the particular media. Here, the encoder itself provides a set of options for different media quality. Implementation of intrinsic quality adaptation can be simple and straightforward. However, the disadvantage is that it provides only a limited (discrete) set of alternatives. Primarily, users can notice a visible change in viewing quality. This can especially be annoying if the viewing quality changes quite often. Also, it might be difficult to modify compression parameters once they are established.
2. *Extrinsic Adaptation*: The above disadvantages can be overcome to a reasonable extent by using extrinsic techniques to modify the quality of media objects. The price to be paid is the complexity involved in implementing the techniques. Extrinsic adaptation is provided by *filters* that operate explicitly on the encoded media objects. The advantage is that it can provide a broader (and continuous) range of possibilities.

Figure 15 describes the concept behind such a technique for video objects. Here, the dynamic adaptor takes as input the existing MPEG-2 stream and the resource constraints. It

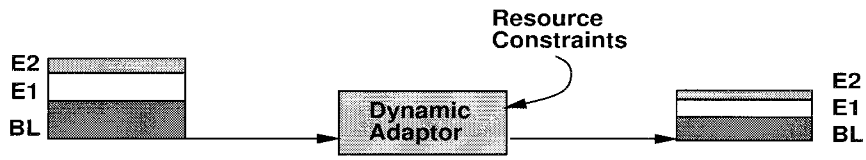

Figure 15. Dynamic MPEG-2 adaptation.

outputs a modified MPEG-2 stream that matches the given resource availability. Modification of the MPEG-2 quality can be made as follows.

- Modify the quantization factor used for quantizing the DCT coefficients. This is almost like recoding the MPEG-2 stream. The disadvantage is the time taken for the recoding process.
- Eliminate DCT coefficients. This is similar to selective transmission (of transform coefficients).

4.2. Content adaptation filters

External content adaptive techniques, similar to the ones discussed above, are normally employed in the form of *filters* that modify the flow of information. These filters can be of the following types.

- *Shaping Filters*: These filters manipulate the structural composition of compressed media objects, e.g., the quantization factor used in MPEG-2 for coding DCT coefficients. The manipulation is done in such a way that the modified output matches resource availability. Shaping filters are complex in nature (usually) and more expensive to implement.
- *Selection Filters*: are simpler in nature and require lesser computation. These filters operate by dropping portions of media objects (e.g., video frame dropping) or select a different encoding quality (e.g., BL or E1 or E2 in MPEG-2 stream).
- *Temporal Filters*: manipulate the temporal characteristics of media objects, e.g., video frame rate. Temporal filters typically require trivial computational effort and are easy to implement.

Filters can be employed at the sender or the receiver or the network figure 16. Filters employed by senders modify the media objects before they are delivered over the network.

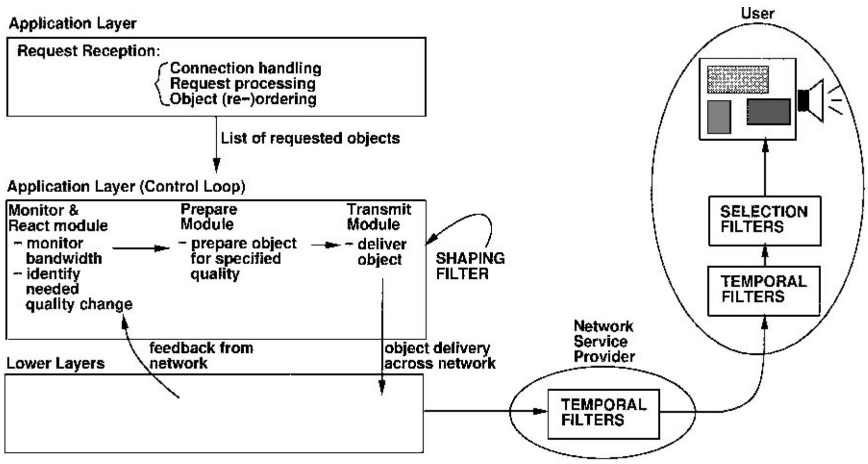

Figure 16. Filers for content adaptation.

In most cases, shaping filters are employed by the sender since any complex modification to the compressed object is done more easily at the encoding time. Filters at the receiver are used normally just before the presentation of media objects. Hence, receiver-oriented filters need to be computationally inexpensive most of the times. Selection and temporal filters are the ones used most at the receiving side. Network filters operate at intermediate switches and gateways. Needless to say, the filters at the network level have to use minimum computational effort. Hence, selection and temporal filters are used at the network level. In summary, content adaptation approach provides a set of possibilities for viewing media objects. These possibilities can be selected by employing one or more of the filters discussed above. Now, we proceed to discuss view adaptation approach.

5. View adaptation

Access constraints imposed by the author(s) of a presentation can lead to different sets of views for different users. In a similar manner, user preferences can request for dropping certain media objects leading different views of the same presentation. As described in figure 17, view adaptation can be of following types.

- *Access Constraints Adaptation*: that handles security levels imposed by the author(s) of a presentation. This may be achieved by:
  - Dropping objects from the presentation structure.
  - Replace an object by a *dummy* object. This approach is especially useful in handling access constraints. Viewer may not feel that they have missed some portions of the presentation.
- *Dynamic Interaction Adaptation*: that handles operations such as skip, reverse.
- *Viewer Preference Adaptation*:
  - Takes care of choice of window size, choice of media (e.g., presence/absence of audio), mode of rendering (e.g., language of text/audio).
  - User ability: views for hearing impaired, sight impaired.

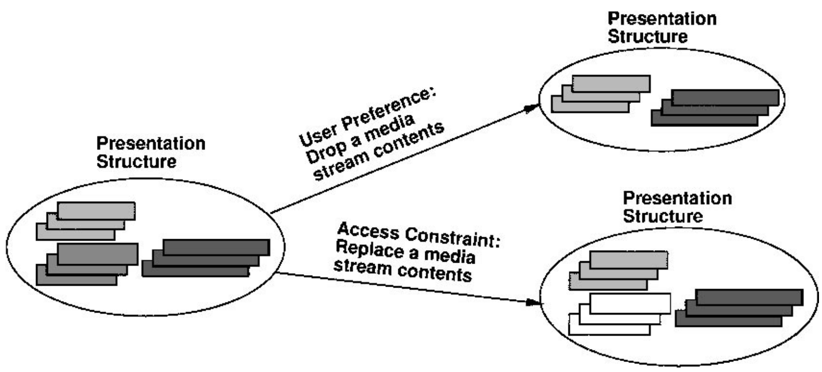

Figure 17. View adaptation possibilities.

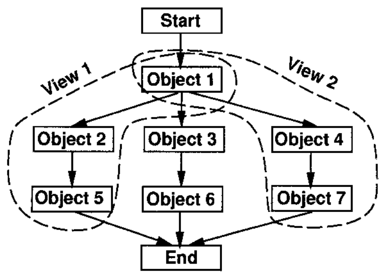

Figure 18. Dynamic interaction adaptation possibilities.

Dynamic interaction adaptation: Different types of user selections may give rise to different views, as shown in figure 18. Probabilistic pre-fetching might be needed to maintain continuity of multimedia presentation. This might be complex, consuming more resource and time. Also, possibilities for viewer selection has to be made clear during the authoring process itself. Synchronization Definition Language (SDL) is a language proposed in [4] for this purpose. SDL is a language is described by a grammar that allows the following synchronization specifications:

1. Non-temporal constraint: Expression consisting of constants, variables, etc.
2. Temporal constraint: Expression involving atleast one clock's value.
3. Combinations of above using and/or relations.

For instance, an author may specify that object Y will be displayed if the link is chosen within a time interval and object Z will be displayed if no link is chosen within a time interval, as follows:

When (X.clock $\geq$ 20 && link_chosen) {display Y}

When (X.clock $\geq$ 20 && !link_chosen) {display Z}.

It is obvious that authoring can become a complex programming process, for incorporating different possibilities for viewer selection and access constraints. This problem can be made transparent to authors by developing sophisticated user interfaces for authoring, that hides the laborious programming choices that go underneath.

Viewer preference adaptation: Facilities for dynamic interaction is part of the authoring process. Viewer preferences, e.g., language of audio/text, can be exercised only if choices are provided at the authoring stage. *Channel Abstraction* has been proposed by CWI, Amsterdam, for handling viewer preferences [6]. Channel is a grouping abstraction for a set of media objects that share some common attributes. Grouping can be based on window layout, rendering methodology (e.g., language used in audio, postscript/PDF text), and semantics (e.g., all audio streams are in one group). Apart from VCR controls, viewers may use channel options to select a desired mode of viewing. Facility for channel abstraction should be provided at the time of authoring itself. For instance, let us consider a multimedia presentation where video is designed to be played in parallel with audio and text. Audio and text have choice of two different languages, Dutch and English. The following channel specification describes the scenario in terms of the channel abstraction discussed in [6]. At presentation time, viewer can select audenglish and textenglish, or any other combination.

<channel> . . . <!definitions> </channel>
<par>
<video src=”xyz.mpg” channel=”mainV”. . . />
<audio src=”dutch.aiff” channel=”auddutch”. . . />
>audio src=”english.aiff” channel=”audenglish”. . . />
<text src=”dutch.html” channel=”textdutch”. . . />
<text src=”english.html” channel=”textenglish”. . . />
</par>

5.1. Summary

View adaptation approach deals with handling access constraints imposed by authors and the preferences exercised by viewers. View adaptation has to be explicitly taken care of during the authoring process. In this section, we discussed some possibilities for view adaptation.

6. Improving resource availability

While improving the availability of network bandwidth might be difficult in terms of cost and time, it is possible to improve server resources in terms of disk bandwidth. One approach is to replicate the presentation servers so that the cumulative bandwidth can handle more presentation requests. While this task of scalability can be considered as one of finding the best server or resource, the key issue of interest here is providing such a service in a transparent manner to the client. Figure 19 describes such a generic

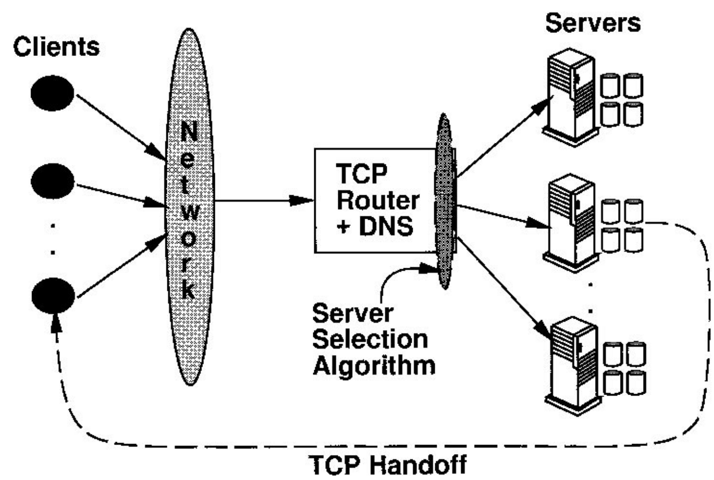

Figure 19. Scalable web servers.

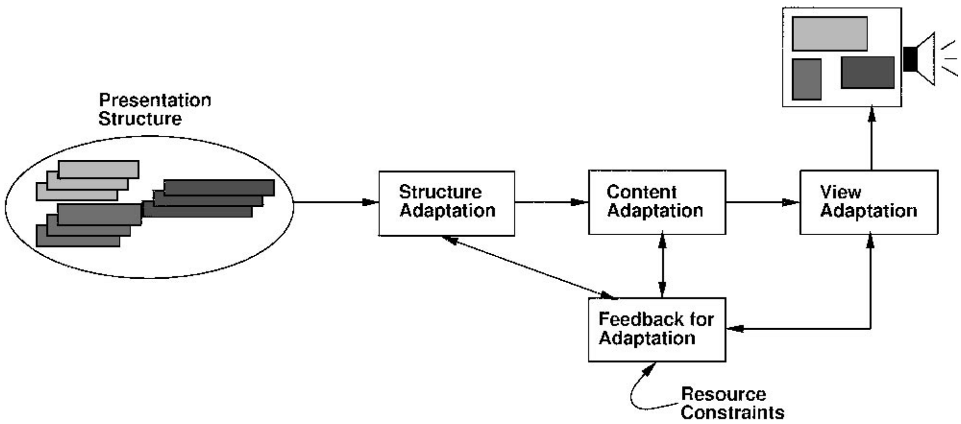

Figure 20. Adaptive multimedia presentation system components.

transparent approach using TCP handoff approach. Clients' requests normally reach a TCP router that also has Domain Name Server (DNS) functionalities. Using DNS functionalities, the router identifies the server(s) that can potentially serve a client's request. One of the possible servers is then selected using a server selection algorithm. Server selection algorithm can use a simple round-robin approach or a more complex greedy approach as in [3]. The TCP connection is then handed off by the router to the selected server.

7. Concluding remarks

Supporting distributed multimedia presentations requires resource commitments from server as well as network service provider. Depending on resource availability, multimedia presentation structure or content or both might have to be modified. Access constraints and user preferences might also necessitate modifications in multimedia presentations. In this paper, we surveyed different strategies for structure, content, and view adaptations. It should be observed here that for handling resource constraints, it is better to try structure adaptation first. If structure adaptation is not possible, then content adaptation can be tried. Also, view adaptation due to user preferences can modify resource requirements. For instance, user preference to shut off video presentation may result in a significant reduction of network bandwidth requirements. This reduction can help in adapting the structure or the content of the presentation in a better manner. Hence, a typical adaptive multimedia presentation system might use a feedback channel for adaptation as shown in figure 20.

Acknowledgment

B. Prabhakaran was supported in part by National University of Singapore Academic Research Fund Grant RP981669.

References

1. J.F. Allen, "Maintaining knowledge about temporal intervals," Communications of the ACM, Vol. 26, No. 11, pp. 832–843, 1983.
2. A.F. Ates, M. Bilgic, S. Saito, and B. Sarikaya, "Using timed CSP for specification, verification and simulation of multimedia synchronization," IEEE Journal on Selected Areas in Communication, Vol. 14, No. 1, 1996.
3. P. Bai, B. Prabhakaran, and A. Srinivasan, "Application-layer broker for scalable internet services with resource reservation," in Proceedings of ACM Multimedia'99, Orlando, November 1999.
4. B. Bailey, J.A. Konstan, R. Cooley, and M. Dejong, "NSync—A toolkit for building interactive multimedia presentations," in Proceedings of ACM Multimedia'98, 1998, pp. 257–266.
5. M.C. Buchanan and P.T. Zellweger, "Automatic temporal layout mechanisms," in ACM Multimedia'93, 1993, pp. 341–350.
6. D.C.A. Bulterman, "User-centered abstractions for adaptive hypermedia presentations," in Proceedings of ACM Multimedia'98, 1998, pp. 247–256.
7. K.S. Candan, B. Prabhakaran, and V.S. Subrahmanian, "CHIMP: A framework for supporting multimedia document authoring and presentation," in Proceedings of ACM Multimedia'96 Conference, Boston, November 1996.
8. K.S. Candan, B. Prabhakaran, and V.S. Subrahmanian, "Retrieval schedules based on resource availability and flexible presentation specifications," ACM Multimedia Systems Journal, Vol. 6, No. 4, pp. 232–250.
9. K.S. Candan, B. Prabhakaran, and V.S. Subrahmanian, "Collaborative multimedia documents: authoring and presentation," Journal of Intelligent Information Systems, to appear.
10. J.E. Coolahan Jr., and N. Roussopoulos, "Timing requirements for time-driven systems using augmented petri nets," IEEE Trans. Software Eng., Vol. SE-9, Sept. 1983, pp. 603–616
11. T.H. Cormen, C.E. Leiserson, and R.L. Rivest, Introduction to Algorithms, McGraw Hill Publishers.
12. S. Gibbs, "Composite multimedia and active objects," in Proc. OOPSLA'91, 1991, pp. 97–112.
13. A. Kay and J.N. Reed, "A relay and guarantee method for timed CSP: A specification and design of a telephone exchange," IEEE Transactions on Software Engineering, Vol. 19, No. 6, pp. 625–639, 1993.
14. L. Li, A. Karmouch, and N.D. Georganas, "Multimedia teleorchestra with independent sources: Part 1—Temporal modeling of collaborative multimedia scenarios," ACM/Springer-Verlag Journal of Multimedia Systems, Vol. 1, No. 4, pp. 143–153, 1994.
15. L. Li, A. Karmouch, and N.D. Georganas, "Multimedia teleorchestra with independent sources: Part 2—Synchronization algorithms," ACM/Springer-Verlag Journal of Multimedia Systems, Vol. 1, No. 4, pp. 153–165, 1994.
16. J. Schnepf, J.A. Konstan, and D.H. Du, "Doing FLIPS: Flexible interactive presentation synchronization," IEEE Journal on Selected Areas in Communication, Vol. 14, No. 1, 1996, pp. 114–125.
17. P. Senac, M. Diaz, and P. de Saqui-Sannes, "A formal environment for the specification and design of multimedia synchronization scenarios," in Proc. 4th Workshop on Network and Operating Support for Digital Audio and Video, Lancaster, November 1993.
18. P. Senac, M. Diaz, A. Leger, and P. de Saqui-Sannes, "Modeling logical and temporal synchronization in hypermedia systems," IEEE Journal on Selected Areas in Communications, Vol. 14, No. 1, pp. 84–103, 1996.