Multimed Tools Appl (2011) 51:379–396
DOI 10.1007/s11042-010-0669-x

A survey of multimedia content adaptation for mobile devices

Velibor Adzic · Hari Kalva · Borko Furht

Published online: 9 December 2010

Abstract With continued increase in the use of smartphones, user expectations of content access have also increased. Most of the content that exists today is not designed for mobile devices. Mobile devices cannot directly access most of the content due to the mismatch in device capability and content playback requirements. Content adaptation is an essential tool that bridges the gap between device capabilities and content formats. In this paper we present an overview of content adaptation and survey recent papers on content adaptation for mobile devices. We introduce the when, where, and what of content adaptation to help classify the content adaptation techniques and to select the appropriate techniques for a given content delivery environment.

Keywords Content adaptation · Mobile devices · Video adaptation · Content semantics

1 Introduction

The ever increasing communication and computing capabilities of devices around us will enable new applications, a new way of life, and at the same time posing new challenges that require innovative solutions. The devices are gaining more multimedia capabilities and the communication among these devices is beginning to be dominated by multimedia data. Yet, the heterogeneous nature of these devices makes it necessary to adapt the multimedia information to enable processing with the available resources. The limited availability of resources puts constraints on the adaptation possible and optimal use of resources is necessary to maximize the end-user experience.

Figure 1 shows the key elements of a general purpose content delivery environment. Content is delivered from a sender to a receiver over a communication network. The capabilities of the devices in such an environment vary, requiring senders, receivers, and network nodes to adapt the content and bridge the mismatches between the content and the

V. Adzic · H. Kalva (✉) · B. Furht
Department of Computer & Electrical Engineering and Computer Science, Florida Atlantic University, Boca Raton, FL 33431, USA
e-mail: hari@cse.fau.edu

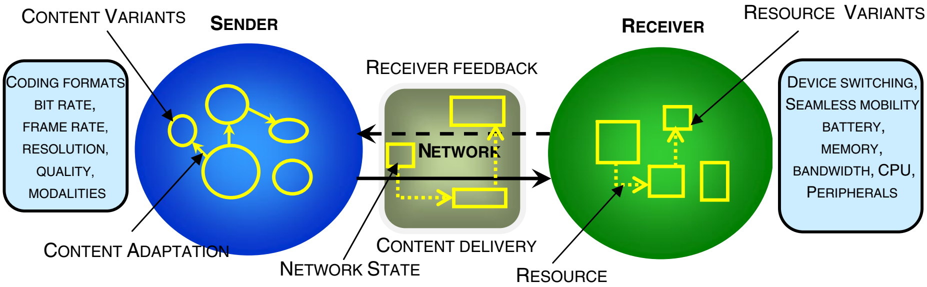


Fig. 1 Content delivery environment

device capabilities. The resources available on these devices would also change with the available battery, concurrently running applications, and available resources such as memory, bandwidth, and peripherals. As the available resources on a device change, the capability of the devices to process/playback content also changes. The resource variants shown in Fig. 1 represent the same device with changing capabilities. As the capabilities of a device change, the communication and content exchanged between the devices must be adapted without affecting the semantic significance of the content. When adaptation is not possible, the session may have to be terminated. The content available at a sender may be adapted dynamically to meet the changing resource capabilities or a discrete number of content variants could be created offline to serve the receivers. The available network resources such as bandwidth, latency, bit error rate, packet loss rate also vary during these content delivery sessions and the systems has to adapt gracefully to these varying conditions. The primary goal of content adaptation is to maximize the end user's quality of experience given the resource constraints at the receiver and the sender.

2 Components of content adaptation systems

Content adaptation is the process of adapting the multimedia content to meet the capabilities of the receiver while maximizing the end user experience. The sender, network, and the receiver work together such that the end user experience is maximized. Since end user experience is closely related to maintaining the semantics of the original content, a content delivery system should use adaptation methods that preserve the content semantics. This section presents the components of a content adaptation system.

2.1 The WWW of content adaptation

Given the dynamic nature of the content delivery environment, the key questions to answer are *where* does the adaptation take place, *when* is the adaptation performed, and *what* content is adapted. These when, where, and what of content adaptation are determined by content delivery environment and the context.

2.1.1 Where should adaptation take place?

The "*where*" attribute refers to the location where the adaptation takes place in the content delivery environment. There are three possible locations at which the adaptation can be performed: at the server which contains content, at the intermediary proxy server, and at the client side. The advantages and shortcomings of each of these approaches are discussed below.

Server-side adaptation involves committing additional resources and software on the server which stores content for delivery. Main advantage of this approach is the fact that the content at every location (web page) can be converted and tailored for specific needs of a receiving terminal. With this approach content provider also controls the way content is transformed and is able to preserve content semantics and protect their copyright. Also, compared to other two approaches this one leaves the smallest bandwidth footprint in the network (in case of downscaled content), since the transformed content is delivered all the way from server to the consumer terminal (whereas in other approaches original, larger amount of data is delivered at least from content server to proxy or all the way in the case of client-side adaptation). On the other hand, there are two major drawbacks—a content provider has to maintain and update comprehensive repository of client device types and has to implement all possible content adaptation engines on the server which is already burdened with delivering original content to desktop users.

On the opposite side of the spectrum we have client-side adaptation. The obvious advantage of this approach is the fact that user can define preferences and determine the type and scale of the adaptation process. With this approach, users with more powerful devices or need for higher quality content can override some usual transformation parameters (hence deliberately increasing complexity of the process or consumed bandwidth). The main disadvantage is possible limitation of the resources at the client terminal. Although many devices today have powerful processors and relatively large (and extendable) amount of memory, there is still high percentage of devices that are not able to cope with high complexity applications like the ones used for transformation of content. Other drawback is the fact that users need to install third party application or browser plug-ins which is not always regarded as a good practice (due to security and ease of usage factors). In addition to this, software providers have to prepare multiple variants of the program for different classes of devices and on top of that maintain and update the applications in all of the variants, which can be very tedious and complex task.

Proxy server or intermediary adaptation is performed between the content provider's server and client's terminal. It can be implemented by the content provider or by the third party. The only basic requirement for this implementation is the available connectivity between client and proxy on one side and proxy and content server on the other side. Advantages of this approach are apparent if we take into account the fact that the proxy server can be implemented independent of the content providers and can be applied to large sets of content types using general rules of transformation (for example, one rule can be to reduce the resolution of all images of the type JPG that go through the proxy script). Also, with this approach we are using specialized system for the sole task of transforming content with possibility of further segmentation of sub-tasks. This system is fairly easy to maintain and adapt to the needs of users or providers. It is very effective solution for situations where it is used to connect Intranet to Internet locations. On the negative side, there is the issue of availability of the proxy server to the client's terminal. Since the users can use terminals with broad mobility the only way to implement universal access to the proxy server is to make it available through the Internet (i.e. obtaining public IP address on the server with almost 100% uptime). To make this approach more efficient than the server-side implementation, the speed of the link between the client and the proxy server should be at least as high as the one available between the client and the content server. Also, as opposed to server-side solution, using this approach we cannot filter all the content from content providers but only the content that is considered in the general rules of transformation.

We have presented some objective advantages and drawbacks of all three approaches to determining *where* the adaptation is performed in a content delivery system. However, when choosing the best solution, different attributes of the environment in which the system is implemented play important role. Factors such as available bandwidth at the network links, tolerable latency and delay, computing resources—processing and memory limitations, acceptable costs and security and copyright issues will affect the adaptation location decisions.

2.1.2 When should adaptation take place?

The “*when*” attribute addresses the temporal relation between adaptation process and distribution process. There are two main approaches are—1) offline—adaptation performed before the transmission of content (transformed content is stored for future distribution) and 2) online—adaptation performed on-the-fly, during content delivery. Offline adaptation is possible only with server side adaptation as a set of content variants are created offline, stored at the server, and an appropriate variant selected at delivery time. The online adaptation, on the other hand, can be performed at the server, network, or the client. Online adaptation performed at the server could result in a hybrid model where content adapted during a session is cached and used for future sessions.

The offline approach, which implies transforming and storing the transformed content, has several advantages. Most obvious is the fact that the content is stored on the same server (or cluster of servers) where the original content is located which means that the availability of the content is same as for the original one. Also, there is the advantage of updating content at the same time as original content is updated, which means that the next transformation will be done only when content is changed. The major disadvantage is apparent if we think of how many versions of the content have to be prepared for the myriad of devices. This highlights two drawbacks—transformation process has to include updated repository of target devices and the amount of storage that has to be used for transformed content is very large, even if the version for one device is significantly smaller than the original content. This also brings up the issue of maintaining the storage system with enormous number of large, separate, files.

More popular and in some regards a better approach is the online adaptation. It can be implemented either on the content server or at proxy or the client terminal. The fact that this approach removes the need for storing different variants of content for every device is obvious advantage over previous method. It transforms only the information that is needed by the client at the time of request. The transformation is implemented on the content as it streams from content server and forwarded as a stream to the client. The transformation is done by pre-determined rules, which means that only the content that is specified in the rules is transformed. Transformation of every piece of content could raise the complexity of software and hardware requirements beyond feasible level. Also, this process has to be sensible to delays it introduces in the flow of information between server and client. The last drawback is the fact that not all the content can be transformed at real time—for certain formats of files, a whole file has to be downloaded by the proxy in order to be unpacked or decoded and some complex transformations such as semantic-preserving summarizations cannot be performed in realtime.

2.1.3 What should be adapted?

The "what" element of the adaptation process is the simplest one and depends only on the design of particular transformation framework. There are two approaches to transformation in this regard. We can choose to transform only presentation data (such as HTML and text) and discard the accompanying media (images, audio and video) or we can transform both presentation and attached media. The first approach was present in the first transformation systems which targeted devices that were severely limited in regard to screen size, displayed colors, capabilities and were operating in data networks with very low speeds. The complexity of such transformation is very low and the implementation is fairly easy. However, in today's world in which devices such as smartphones are emerging and the wireless networks are providing high speed access to Internet, this approach is not acceptable in most of the consumer markets. On the other hand, there are markets in which majority of devices are low end mobile phones that don't have necessary attributes for media consumption.

If we consider the amount of multimedia content that is consumed every day on the Internet and the need of the user to have the access to this content everywhere and at any time it is obvious that we need the system that can adapt all the content without discarding any important information. User experience is one of the most important factors in modern consumer systems, and the advantage of having the system that meets the needs of the user in the best possible way is clear. On the other hand, implementation of this system compared to former is much more complex. Not only does the system that is performing the transformation had to be more advanced in sheer computational power and storage capacity, but the complexity of the algorithm that has to tailor the data to specific user needs is very high.

2.2 The level of transformation

We will now discuss very important aspect of content adaptation—the level at which the information is transformed. Not only does the depth of the level at which the transformation is done affects the amount of resources needed for successful implementation, but is of essential importance in regard to semantic completeness of transformed information.

To better explain this concept we will illustrate the multilevel nature of content and its parts in Fig. 2.

The content of almost every multimedia document can be segmented into smaller components. The components have spatio-temporal relationships that define the semantics

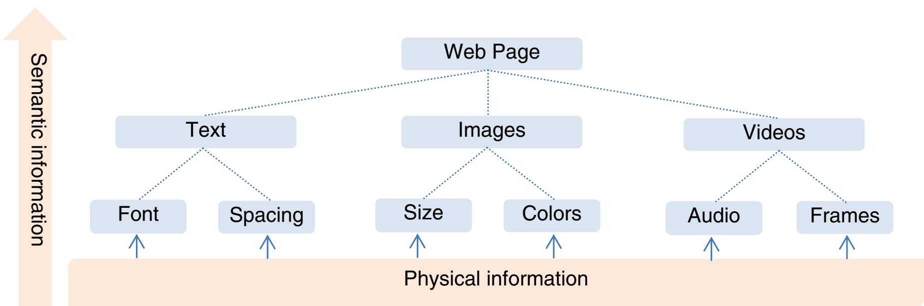


Fig. 2 Components of multimedia content

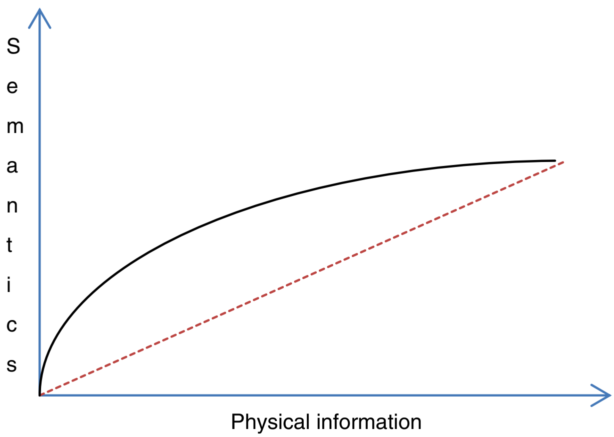
of the content. As we go down from the level of web page to the bottom level at which the physical information is presented, we are decreasing the level of the semantic information that is apparent; for example, the location of text in the final presentation will determine the semantics that may not be apparent when text alone is read. This means that we have to take into account the semantic aspect when we do transformation of information. Otherwise, no matter how much physical information we keep in transformed form we can lose semantic meaning and the resulting content will be useless from users' point of view. This is very important observation, and providers of transformation engines have to know that the target of transformed information is not the end terminal, but the end user. Having that in mind, we can define the goal of the transformation as providing the end user with equivalent quality of content no matter from which location it is accessed.

The important aspect of the relationship between physical and semantic information is shown in Fig. 3. The relationship between physical and semantic information usually is not linear or proportional. It is possible to achieve "more with less" results if the transformation algorithm contains procedures that involve user interaction. With end user involvement we can reduce the amount of physical information that has to be transmitted in the form of transformed information. This is essential aspect that needs to be considered when optimizing transformation process. It is possible to reduce the amount of transmitted information, but we have to be careful to keep important semantic components intact.

The best way to optimize the transformation process and keep semantic context is to allow end users to input relevant feedback (such as tags for images and video clips). Since the semantic meaning of transmitted information is not the same for every user, it is essential to allow interaction on the individual level. This can raise the level of complexity of implemented algorithm, but the end result is optimized transformation process that reduces amount of transmitted content. Also, with emerging social networks in which users provide personal information and feedback on daily basis, the process of including such information in the transformation algorithm would become simpler.

We have presented the "WWW" portion of the system implementation. It represents the basis of the framework architecture and should be considered as the first step in the process of constructing the transformation system.

Fig. 3 Semantics vs. physical information



2.3 Semantic preserving adaptation

Multimedia content has two key components: representation and presentation. The presentation layer is consisted of content primitives (for example, Paragraph and Font for Text or Resolution and Quality of Video). Presentation addresses the spatio-temporal relationships among the media elements of multimedia content. Representation is closest to the physical layer and consists of representing media elements with bits.

At the representation level there are two strategies for information transformation: content scaling/transcoding and modality conversion. As we move up to the presentation level, we can employ content selection as a third way of transformation. Thus a combination of presentation and representation adaptation is essential to maximize the quality of experience for end users.

The semantic information of content is highest at the presentation level and drops down in direction of separate representations. The only case where the semantic information is same is when presentation layer contains only one content item. However, most of the modern websites and multimedia files have at least two components (video and audio, video with captions, image with tags, etc.) and provide multiple choices for adaptation. A key challenge in content adaptation is preserving semantic information of authored content throughout adaptation process. The impact of adaptation strategy on the semantic information in discussed in the following sections.

2.3.1 Content scaling/transcoding

We will use the example of video transcoding to show the relation between changing the attributes of content and semantic context. Consider a video clip with following attributes that are changed during adaptation: Framerate (originally 30 fps) and Bitrate (originally 800 kbps).

Figure 4 shows an example of a relationship between the quality of representation and semantic integrity in video transcoding. Suppose that the setup for quality of representation "Q1" is to set parameters to 10 fps and 100 kbps. It is obvious that the resulting video will contain low level of semantic content since the user will be presented with the clip that displays 3 times less frames and we cannot determine which frames contain most of the semantic context. Also, some details could be lost due to the drastic reduction of bitrate. It is obvious that semantic level of the resulting video "S1" carries only a small portion of original context.

We can implement setup "Q2", setting framerate at 25 fps and bitrate at 600 kbps. The resulting video will be noticeably smaller in terms of file size. We can claim with high probability that the most of original semantic content is preserved. On the other hand, we produced the video that requires less storage space (and lower bandwidth to transfer).

We presented the relationship between QoR and Semantics in the graph (Fig. 4). We should notice that after achieving original semantic content level (by using original attributes in setup "Q3"), the increase in quality for each primitive or combination of primitives cannot increase semantic level (because original video contains 100% of semantic information).

The choice of setup "Q2" depends of requirements in regard to importance of semantic level of transformed content and computational constraints for transformation implementation. The complexity of transformation process (and required resources) rises proportionally to QoR. Choosing the best setup having in mind this trade-off is the crucial part of proper algorithm for content transformation.

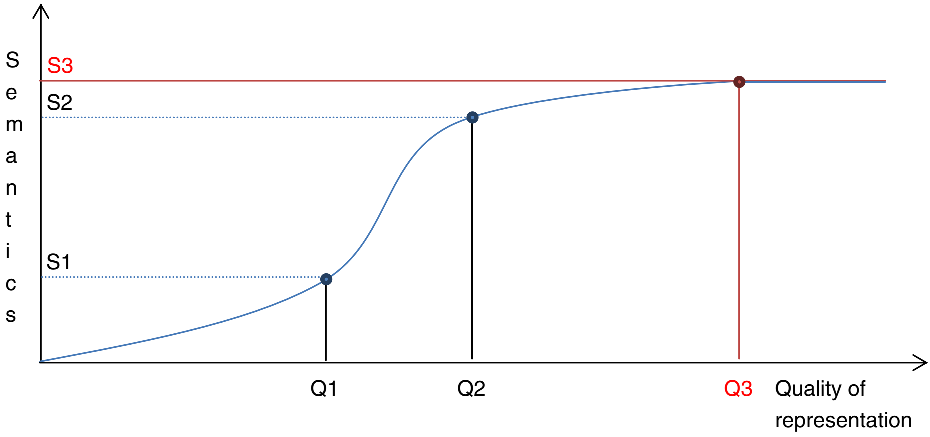


Fig. 4 Semantics preservation in video transcoding

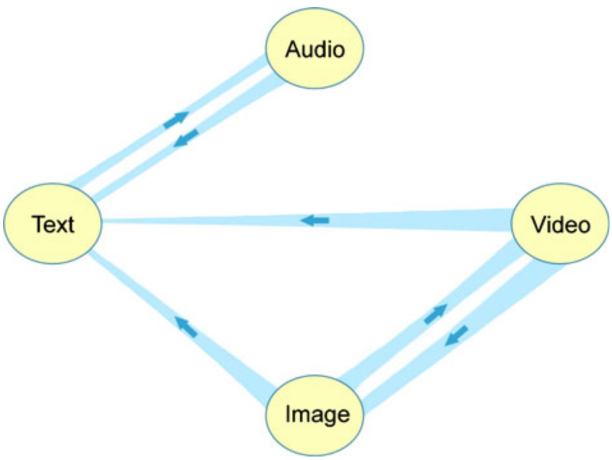
2.3.2 Modality conversion

In order to transform information to fit characteristics of end user terminal or needs of the end user, sometimes it is necessary to convert content into different multimedia modality. Conversions from video to text/audio for people with visual impairments and from audio to text for hearing impaired persons are examples of transformations that have to be implemented. Since different modalities carry different semantic levels, we have to be aware of relations between different modalities and semantic content level when changing modality of information.

Figure 5 shows the relationship between modality conversion and loss of semantics. Blue stripes that are connecting yellow ovals (presenting multimedia modalities) represent transitions between modalities in transformation process. Thickness of the stripes represents the semantic information contained within the modality—thinning of a line means that there is certain loss of semantic information in the process.

As we said in the previous section, there is no such thing as increasing the semantic information level of original content. This means that during any kind of transformation

Fig. 5 Loss of semantics due to modality conversion



(and hence modality conversion, too) the semantic information can only be lowered or maintained at original level. For example, it is obvious that during the conversion from video to text (in form of tags or video summarization) we lose significant amount of semantic information. No matter how detailed the text summarization could be, it can never produce same experience for the end user as presentation of original video.

On the other hand, the transition from text to audio (and vice versa for speech to text) can be implemented such that little or none semantic information is lost. This is in part due to the nature of those modalities (both use same dictionary of words and punctuation) and consequently due to very good algorithms that are implemented in the field of text—speech conversion. Of course, not all of the audio content is speech and hence the relation is not symmetrical.

2.3.3 Content selection

The presentation is the top layer at which we combine all the representations (original and/or transformed). Adaptation at this level is possible through content selection. Following system specification and terminal limitations or end user needs, adaptation engines can decide to discard some of the content objects in the final presentation. The relationship between the content objects and overall semantics of the transformed presentation is straight-forward. If we remove some of the objects, the semantic level of information will drop down. However, the relation between the number of objects that are kept in the final presentation and semantic level is not linear. This is because of the fact that not every object has same impact on overall semantics. Let's consider an example to clarify this relationship.

Figure 6 shows the typical relationship between the number of objects and content semantics. The final presentation in this example consists of four objects—video clip of certain event, audio track, textual explanation of the event captured in the video and image thumbnail of the video. These objects are presented in the graph as objects 1, 2, 3 and 4 in same order as listed. We are assuming that the loss in semantics due to any media transformation is negligible. Hence, having all four objects in the presentation means that there is 100% of semantic information. The video clip carries most of semantic information. If we would decide to discard the video as a result we would have presentation that contains only small portion of original semantic information. On the other hand, if we decide not to

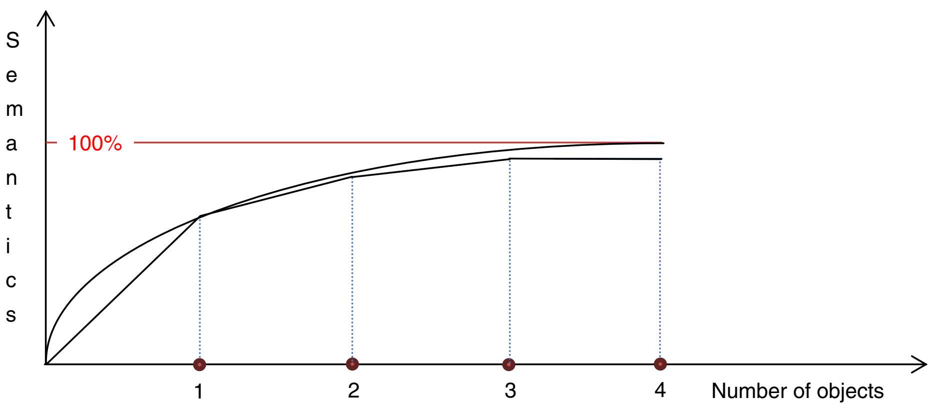


Fig. 6 Impact of number of objects on semantics

display the thumbnail image (i.e. one key-frame extracted from video) we would make only slight, if any, impact on semantic information.

For the presentations that have much more objects the model stays the same and follows the curve in the graph. There is increase of storage and computational resources with the increase of the number of objects (as was the case with content scaling/transcoding). This means that we have to deal with same trade-offs that we considered before. If we assume that during the transformation of representations we preserved most of the individual semantics, the task of choosing which objects (representations) to select for final presentation is related to choosing the objects that carry the most of semantic information. Not every presentation has such heterogeneous distribution of semantic information across objects, so choosing the right subset of objects is specific for individual implementations. This makes content selection hard problem to be solved and the one that can be further explored and presents an interesting research topic.

2.4 Receiver feedback

The importance of available feedback mechanism in the adaptation scheme is not only important from the aspect of understanding the user context but also allows a sender to understand the receiver capabilities. The content delivery environment is ever changing and the knowledge of these changes allows senders and proxies perform better content adaptation. The available bandwidth, currently available computing resources and location—those are all the factors that can change significantly during the transformation and transportation of the content. Availability of receiver feedback and the amount of feedback varied depending on the systems and protocols used. Content delivery using RTP has support to provide receiver feedback in the form of receiver reports that indicate the packet delay and packet loss. Capability exchange in a limited form is supported using Session Description Protocol (SDP). Comprehensive tools such as the MPEG-21 Digital Item Adaptation and Usage Environment Description exist that can provide information about end user environment and receiver capabilities. Another framework standardized by W3C is the Composite Capabilities/Preference Profiles (CC/PP) structures for describing receiver capabilities and user preferences. These frameworks are designed to perform capability exchange before a session starts and need to be adapted for HTTP based multimedia services.

2.5 User control

Full user control of transformation attributes is present in most of the client-side implementations. However, the addition of limited user control even in the case of server-side and proxy implementation can be useful. As we already mentioned, the main goal of transformation is maximizing the end user's quality of experience, so providing some means of overriding the automatic transformation parameters can be important. For example, the user may choose to wait a little bit more for watching the video in higher resolution. Or reduce the quality of content in order to run some additional applications on the terminal. However, when crafting such solutions one has to be careful not to give end user too much means to alter the transformation process as it can lead to unexpected behavior of the algorithm and affect performance of transformation system's software or hardware. By adding carefully planned features that would enable user to alter the system parameters, in conjunction with effective environment feedback can result in highly optimized transformation process.

2.6 Content caching

As is the case with any content distribution system, adding caching features to implementation of the adaptation engines can raise the quality and optimize system's performance. Storing the transformed content in the server-side implementation can be regarded as extreme case of caching strategy. Taking into account balance between storage capacity, computational expenses and available bandwidth we can develop caching scheme that can improve the performance of the system. Storing some portions of the transformed content (i.e. the most popular video clips that are frequently accessed) will increase the amount of used storage but it will also reduce significantly the usage of processing resources (which also means that the delay due to processing of transformation is removed).

In the case of proxy-based implementation it can also reduce the amount of network bandwidth consumption between proxy and content server, which, in some cases, is the most expensive segment of whole system (for example, if we are using proxy server to connect intranet with the content server that is outside local network).

3 Content adaptation for mobile devices

The components and strategies for content adaptation presented in Section 2 can be used in combination to provide adaptation services. In this section we survey the recent work on content adaptation for mobile devices and describe what adaptation components are used in the reported work. Table 1 summarizes the adaptation

Table 1 Summary of techniques for content adaptation for mobile devices (from surveyed papers)

Paper	Where	When	What	Level	User C	Feedback	Caching
[1]	C	R	T	S2	U2	0	0
[2]	P	R	T	S1	U1	0	0
[3]	P	R	T, I	S2	U2	1	1
[4]	P	R	T, I	S3	U1	0	0
[5]	P	R	T	S1	U1	0	0
[6]	P	R	T	S1	U2	1	0
[7]	C	R	T	S1	U3	0	0
[8]	P	R	T, I	S1	U1	0	1
[9]	C	R	T	S1	U1	0	0
[10]	S	R	T	S2	U1	0	0
[11]	S/P	R	T	S2	U2	0	1
[12]	S	R	T	S2	U1	0	0
[13]	S	P	M	S1	U1	1	0
[14]	C	R	M	S3	U2	0	0
[15]	S	R	M	S2	U2	1	0
[16]	S	R	M	S2	U1	1	0
[17]	C (P2P)	R	M	S1	U1	0	0
[18]	S	R	M	S3	U2	1	0

techniques used in the surveyed papers. The following attributes of the adaptation system are summarized:

Where: (S—Server, P—Proxy, C—Client)
When: (P—Preprocessed/Stored, R—Real time/On the fly)
What: (T—Text, I—Images, M—Multimedia)
Level: (S[1-3], S1 = Physical, S3 = Semantic)
User Controlled: (U[1-3], U1 = No User Control, U3 = Full User Control)
Feedback: 0/1—Is it resource adaptive?
Caching: 0/1—Implemented?

Ahmadi and Kong present a system for adapting presentations designed for desktops for mobile devices [1]. Implementation is done through mobile browser. Original pages are analyzed for both DOM structure and visual layout. Heuristic categorization is implemented so that the DOM elements are further divided into content-related parts. Every part is presented in the browser as a sub-page. A table of contents is provided for the whole page, and every sub-page has navigation with links to other sub-pages. In the user study authors show that their browser is slightly better than Opera mobile, in most of the categories (satisfaction, efficiency and aesthetics). The heuristics based approach to creating sub-pages and using a table of contents affects the semantics of the presentation as the users are not required to navigate the page using a menu. The proposed solution does not address dynamic solution such as Flash.

Blekas, Garofalakis, and Stefanis present a web content adaptation system using RSS feeds [2]. The browsing application is implemented at the proxy server. User enters the address of the Proxy in the browser, and gets to the page where the desired website address is entered in a text field. Proxy server parses the HTML of the desired page searching for RSS content. Users can choose RSS content page or normal page. Only links and content text are saved—all images and styling elements are discarded resulting in textual pages. While this method reduces the bandwidth required significantly, the need for user input explicit connection to a proxy make this difficult to use. The role of intermediary nodes should be transparent to the users allowing service providers to change the servers as needed.

Gupta et al. present a proxy server based solution that takes the desired web pages and transforms HTML into XML [3]. During the parsing and converting process, all non-standard and incomplete tags are discarded. The DOM tree is constructed using the resulting XML. The user is presented with a Table of Contents with links to sub-pages. The content is adapted, such that it occupies one page. If the content is more than one page, pagination is implemented. Images are resized using sliding-window wavelet based coding, such that the size fits to screen size. The use of table of contents to create sub pages creates another hierarchy not intended by the original content creator. This additional navigation step also can adversely affect the user experience. While only a negligible portion of the original content is discarded, the creation of sub pages can affect semantics.

Kim, Jang, and Kim, address the loss in semantics intended by original web content designers when performing content adaption for mobile devices [4]. The authors also address the importance of visual information by adapting images based on content. The adaptation engine is implemented on a proxy server where images in a multimedia document are analyzed and adapted according to the content. “Natural images” are only resized to fit the screen and “artificial images” such as logos are analyzed for edge detection and resized using that information. Adaptation of other elements is done in usual fashion.

Content based analysis can be resource intensive posing scalability problems in large systems.

Lam and Baudisch report a proxy based adaptation engine to mainly adapt the text portion of the content [5]. A web page is presented as a thumbnail, with original layout zoomed out, to fit the screen dimensions. Instead of just resizing text fields, font size is dynamically changed so that it is readable. This is accomplished by discarding some portions of the text. Most frequent words are removed, and further reduction is done, so that the text fits into the intended area, but still keeps meaningful structure. This adaptation technique overcomes shortcomings of thumbnail presentation (small, unreadable text—need for zooming) and single column layout (original content is modified, too much scrolling is needed).

Lee et al. report a device independent content adaptation system for mobile devices [6]. Using proxy server, a page is retrieved from original source and pre-processed by Vision-based Page Segmentation (VIPS) that segments a web page into blocks. Blocks are analyzed and filtered to discriminate useful blocks from the partitioned blocks. Blocks are divided into navigational and content categories. Once the user enters the website with navigation and content, clicks are monitored and used for feedback that provides information about user preferences. The learning process uses this information to produce proper links ranking in future adaptation.

Lee and Seo present a client-based interface adaptation technique for XML-based Web pages [7]. The authors implemented an adaptation engine on the device. Since adaptation is performed on the device, network and server resource requirements are unchanged. This method falls under presentation adaptation with representation of media elements unchanged. The client based adaptation also allows for better adaptation to user preferences. The proposed work is suitable for any XML based UI descriptions. Shortcomings of the proposed solution are that separate adaptation rules have to be pre-set for every page and a specialized browser compatible with adaptation engine has to be installed.

Lehtonen et al. introduce the Tut-Mobi system with a proxy server used as an adaptation engine and a browser instance [8]. On the mobile client, only UI of a page is displayed. User sends the request to the proxy server using Tut-Mobi protocol (TMP) and the server responds with the image screenshot of the original webpage, overlaid with coordinates and additional information of the page elements—links, text and images. Elements are highlighted and short information is presented when hovering. The proposed solution preserved the user experience of original page while reducing the bandwidth.

Roto et al. present a web page visualization method called Minimap to display adapted web pages [9]. The CSS parameters of the original page are modified, so that the area of the boxes of single elements is fitting the screen size. Also, Mini Map layout of the entire page is shown during the long scrolling action, so that user can have overview of current position on the page. A detailed user study conducted shows advances over Narrow layout method. The solution is integrated into browser and cuts down the actions needed by user to browse the page. The proposed solution is more usable than one column layout.

Xiao, Tao, and Li present a web page adaptation method for mobile devices [10]. A web page is separated to content and link blocks using DOM tree. During this process, "noise" is removed from the page and tree is traversed multiple times in order to group smaller blocks into two main groups. Implementation is done on the proxy server using Java Open source project HTMLParser. The implementation is tested more for efficiency of the block separating mechanism than for user experience or network resources consumption. Method of separating web page content into blocks has high percentage of accuracy. It can be implemented on a hosting server or on a proxy and reduces the amount of transferred data.

Yang et al. present a context aware service oriented content adaptation to enhance pervasive Web accessibility [11]. A web page is analyzed and tree is constructed and presented in the form of a linear sequence. The root of tree represents the content of the page and child nodes represent content objects such as text, images, audio, and video. Based on pre-specified rules and user preferences these content objects are transcoded to final form. The content style sheets are done by the script and final appearance fits small screen very well. The resource intensive nature of this process makes this adaptation difficult to use for mobile devices.

Yin and Lee present a link analysis based method to adapt the layout on mobile devices [12]. Algorithm that is proposed traverses the HTML code of the original page, extracting the tree structure of the page. Once the structure is obtained, the nodes (content segments) are connected via edges (links) that have some weight attributed. The authors use Google's PageRank algorithm to rank the segments. The segments with best score are delivered on the resulting page for the user. Authors are claim that using this system they are displaying the most relevant rectangle with main content from the main page. Also, in order to preserve the original structure they are adding links to other parts of the original page. The complexity of implementing this on a device could be high.

Bellinzona and Vitali present the Alembik framework for the transcoding of multimedia content based on receiver capabilities [13]. The proposed adaptation framework employs a tag library to drive adaptation of transform image, audio and video. Comprehensive library structure is proposed in order to cover as much as possible variants. For the recognition of a mobile client, authors use the WURFL library.

Davis et al. propose an adaptation system that exploits the relationships in social network to understand user preferences [14]. Exploiting the social signals had just begun and the proposed system is an presents an interesting direction. Video players are added with a "social" feature where users are allowed to add emotional tags during video playback. These tags are recorded in temporal order and the information is used in future for playing sequences of videos that have certain tags.

Hutter et al present a multimedia adaptation engine that is based on MPEG-21 DIA [15]. The system can be implemented either on the origin server or at the adaptation node between the server and the client in the network. In order to allow dynamic adaptation, the node is fed with data about the network conditions from client. All other (static) usage constraints are stored in the central database. Stream from the server is first decomposed, then adapted using parameters from database and client in addition to content-based metadata. Adapted content is delivered to the set of nodes that have similar constraints. Transport of the data is done with several protocols: HTTP for DIA context descriptors, SDP for BSDLink and gBSD Transformation and RTP for gBSD and AQoS. Authors also propose using RTP for most feasible transportation of content stream. This method can be implemented at server and at proxy. It allows dynamic adaptation that is sensitive to network changes in real time.

Kim and Yoon present an adaptation system that uses MPEG-7 and MPEG-21 based descriptors for content adaptation [16]. The proposed adaptation engine uses MPEG standards for various stages. Server side and client side are represented through MPEG-21 DI and DIA. MPEG-7 is used for linking content and description. For the middle part, video transcoding, authors suggest using MPEG-4 FGS and MPEG-21 SVC. While this is an interesting attempt to use standards based methods for adaptation, the complexity of the standards makes wider adoption difficult.

Rong and Burnett present a peer-to-peer architecture for content adaptation using MPEG-21 framework [17]. Authors use the MPEG-21 DI and DIA for media adaptation. Implementation is done on a P2P network with super peers. Super peers are divided into two groups—Category and Locality SPs. First group is responsible to store information about the categories of content, while second one stores information about local peers. Adaptation process is done in two stages. First client sends request for some content along with its DIA description. Then Category SP returns available content on the network. Adaptation of the content is done either by the available local idle peers or by the sending peer. Privacy and use of idle resources on peers for adaptation have to be managed carefully in order not to overwhelm the peers.

Zuffery and Kosch present an adaptation system that uses MPEG-7 semantic descriptors to preserve semantics and MPEG-21 digital item adaptation for driving content adaptation [18]. The downside is that the system will remain experimental because of lack of MPEG-7 and MPEG-21 adoption. Advances in automatic extraction of semantics will be necessary to effectively use such systems.

4 Conclusion

Content adaptation targeting mobile devices is receiving significant attention from the research community. The increased interest is largely due to the broad adoption of mobile devices with multimedia and web access capabilities. Content adaptation can be performed at the server, an intermediary such as a proxy server, or at the receiver. Proxy based content adaptation can scale to large number of users and support many content providers making this a candidate for most likely adoption. The server and client based adaptation also have important role for specific environments. Optimal adaptation will use hybrid models where adaptation strategy depends on the current playback environment. While there are efforts to standardize frameworks to simplify standardization, the complexity of the standards, and lack of adoption make standards based adaptation difficult in the short term.

References

1. Ahmadi H, Kong J (2008) Efficient web browsing on small screens. AVI ’08: Proceedings of the working conference on Advanced visual interfaces, ACM, pp 23–30
2. Blekas A, Garofalakis J, Stefanis V (2006) Use of RSS feeds for content adaptation in mobile web browsing. W4A ’06: Proceedings of the 2006 international cross-disciplinary workshop on Web accessibility (W4A), ACM, pp 79–85
3. Gupta A, Kumar A, Mayank, Tripathi VN, Tapaswi S (2007) Mobile web: web manipulation for small displays using multi-level hierarchy page segmentation. Mobility ’07: Proceedings of the 4th international conference on mobile technology, applications, and systems and the 1st international symposium on Computer human interaction in mobile technology, ACM, pp 599–606
4. Kim W, Jang D, Kim T (2007) Improved web content adaptation for visual aspect of mobile services. Signal-Image Technologies and Internet-Based System, 2007. SITIS ’07. Third International IEEE Conference on, Signal-Image Technologies and Internet-Based System, pp 402–408
5. Lam H, Baudisch P (2005) Summary thumbnails: readable overviews for small screen web browsers. CHI ’05: Proceedings of the SIGCHI conference on Human factors in computing systems, ACM, pp 681–690
6. Lee E, Kang J, Choi J, Yang J (2006) Topic-specific web content adaptation to mobile devices. Web Intelligence, 2006. WI 2006. IEEE/WIC/ACM International Conference on, Web Intelligence, 2006. WI 2006. IEEE/WIC/ACM International Conference, pp 845–848

7. Lee E, Seo J (2008) Contents adaptation of XForms pages using XPath filtering. Advanced Language Processing and Web Information Technology, 2008. ALPIT ’08. International Conference on, Advanced Language Processing and Web Information Technology, 2008. ALPIT ’08, pp 575–580
8. Lehtonen T, Benamar S, Laamanen V, Luoma I, Ruotsalainen O, Salonen J, Mikkonen T (2006) Towards user-friendly mobile browsing. AAA-IDEA ’06: Proceedings of the 2nd international workshop on Advanced architectures and algorithms for internet delivery and applications, ACM
9. Roto V, Popescu A, Koivisto A, Vartiainen E (2006) Minimap: a web page visualization method for mobile phones. CHI ’06: Proceedings of the SIGCHI conference on Human Factors in computing systems, ACM, pp 35–44
10. Xiao Y, Tao Y, Li Q (2008) Web page adaptation for mobile device. WiCOM ’08. 4th International Conference on Wireless Communications, Networking and Mobile Computing, pp 1–5
11. Yang S, Shao N, Sue A, Chung J-Y (2005) Pervasive content access for mobile commerce. CEC 2005. Seventh IEEE International Conference on E-Commerce Technology, pp 523–526
12. Yin X, Lee WS (2004) Using link analysis to improve layout on mobile devices. WWW ’04: Proceedings of the 13th international conference on World Wide Web, ACM, pp 338–344
13. Bellinzona D, Vitali P (2008) Multimedia content adaptation through tag libraries. Database and Expert Systems Application, 2008. DEXA ’08. 19th International Conference on, DEXA ’08. 19th International Conference on Database and Expert Systems Application, pp 716–720
14. Davis S, Cheng E, Burnett I, Ritz C (2010) Multimedia adaptation based on semantics from social network users interacting with media. Quality of Multimedia Experience (QoMEX), 2010 Second International Workshop on, (QoMEX), 2010 Second International Workshop on Quality of Multimedia Experience, pp 170–175
15. Hutter A, Amon P, Panis G, Delfosse F, Ransburg M, Hellwagner H (2005) Automatic adaptation of streaming multimedia content in a dynamic and distributed environment. Image Processing, 2005. ICIP 2005. IEEE International Conference on, ICIP 2005. IEEE International Conference on Image Processing, 3, III—716–719
16. Kim S, Yoon Y-I (2008) Universal multimedia access model for video delivery. Grid and Pervasive Computing Workshops, 2008. GPC Workshops ’08. The 3rd International Conference on, GPC Workshops ’08. The 3rd International Conference on Grid and Pervasive Computing Workshops, pp 201–205
17. Rong L, Burnett I (2006) Facilitating universal multimedia adaptation (UMA) in a heterogeneous peer-to-peer network. Automated Production of Cross Media Content for Multi-Channel Distribution, 2006. AXMEDIS ’06. Second International Conference on, 2006. AXMEDIS ’06. Second International Conference on Automated Production of Cross Media Content for Multi-Channel Distribution, pp 105–109
18. Zufferey M, Kosch H (2006) Semantic adaptation of multimedia content. Multimedia Signal Processing and Communications, 48th International Symposium ELMAR-2006 focused on, 48th International Symposium ELMAR-2006 focused on Multimedia Signal Processing and Communications, pp 319–322

Velibor Adzic is a PhD student at FAU (Florida Atlantic University). He received his Bachelor’s degree in Applied Computing from University of Montenegro. Currently he’s a member of Multimedia LAB at FAU and his research interests include computer vision, video transcoding and cloud computing.

Hari Kalva Dr. Kalva is an Associate Professor in the Department of Computer Science and Engineering at Florida Atlantic University. Dr. Kalva is an expert on digital audio-visual communications systems with over 15 years of experience in multimedia research, development, and standardization. He has made key contributions to the MPEG-4 Systems standard and also contributed to the DAVIC standards development. His current research activities include mobile video services, video transcoding, multi-view and 3D video coding, and multimedia programming. He is a recipient of the 2008 FAU Researcher of the Year Award and a 2009 ASEE Southeast New Faculty Research Award.

Dr. Kalva received a Ph.D. and an M.Phil. in Electrical Engineering from Columbia University in 2000 and 1999 respectively. He received an M.S. in Computer Engineering from Florida Atlantic University in 1994, and a B. Tech. in Electronics and Communications Engineering from N.B.K.R. Institute of Science and Technology, S.V. University, Tirupati, India in 1991.

Borko Furht is a professor and chairman of the Department of Computer Science and Engineering at Florida Atlantic University in Boca Raton. He is also Director of the NSF-sponsored Industry/University Cooperative Research Center at FAU. His current research is in multimedia systems, video coding and compression, 3D video and image systems, video databases, wireless multimedia, and Internet and cloud computing. He has been Principal Investigator and Co-PI of several multiyear, multimillion dollar projects including the Center for Coastline Security Technologies, funded by the Department of Navy, One Pass to Production, funded by Motorola, the NSF PIRE project on Global Living Laboratory for Cyber Infrastructure Application Enablement, and the NSF funded High-Performance Computing Center. He is the author of numerous books and articles in the areas of multimedia, computer architecture, real-time computing, and operating systems. He is a founder and editor-in-chief of *the Journal of Multimedia Tools and Applications* (Springer, 1993). His latest books include “Handbook of Multimedia for Digital Entertainment and Arts”

(Springer, 2009) and “Handbook of Media Broadcasting” (CRC Press, 2008). He has received several technical and publishing awards, and has consulted for many high-tech companies including IBM, Hewlett-Packard, Xerox, General Electric, JPL, NASA, Honeywell, and RCA, and has been an expert wetness for Cisco and Qualcomm. He has also served as a consultant to various colleges and universities. He has given many invited talks, keynote lectures, seminars, and tutorials. He serves on the Board of Directors of several high-tech companies.