

Raster Image Adaptation for Mobile Devices using Profiles

René Rosenbaum and Bernd Hamann

Institute for Data Analysis and Visualization (IDAV), Department of Computer Science,
University of California, Davis, Davis, CA 95616-8562, U.S.A.

ABSTRACT

Focusing on digital imagery, this paper introduces a strategy to handle heterogeneous hardware in mobile environments. Constrained system resources of most mobile viewing devices require contents that are tailored to the requirements of the user and the capabilities of the device. Appropriate image adaptation is still an unsolved research question. Due to the complexity of the problem, available solutions are either too resource-intensive or inflexible to be more generally applicable.

The proposed approach is based on scalable image compression and progressive refinement as well as data and user profiles. A scalable image is created once and used multiple times for different kinds of devices and user requirements. Profiles available on the server side allow for an image representation that is adapted to the most important resources in mobile computing: screen space, computing power, and the volume of the transmitted data. Options for progressively refining content thereby allow for a fluent viewing experience during adaptation. Due to its flexibility and low complexity, the proposed solution is much more general compared to related approaches. To document the advantages of our approach we provide empirical results obtained in experiments with an implementation of the method.

Keywords: Device adaptation, Scalable image coding, Progressive refinement, Profiles, Mobile environments

1. INTRODUCTION

Today, digital imagery is often available in high resolution and quality. As the small size of mobile hardware and existing limitations in wireless communication strongly constrain screen space and computing power as well as the data that can be transmitted, content reduction and adaptation is mandatory. Appropriate image adaptation is still an unsolved research question. Due to the complexity of the problem, available solutions are either too resource-intensive or inflexible to be generally applicable.

As shown by THIEDE AT AL.,¹ most limitations of existing image adaptation technology can be overcome by taking advantage of scalable image coding and progressive refinement. THIEDE AT AL. proposed an on-the-fly approach that is fully client-based. Thereby, it makes use of a single scalable image that is incrementally transmitted and displayed until it is predicted that the next refinement stage exceeds the capabilities of the viewing hardware. Although, this strategy does not require any prior knowledge to the shown image or the viewing environment adaptation accuracy is coarse.

In this publication, we propose an image adaptation strategy using profiles. Thereby, we follow THIEDE ET AL. and take advantage of scalable image coding. Based on a single "multi-purpose" data-stream and prior knowledge to the image, the viewing device, and user preferences, an image is refined as long as the device is able to provide the resources required for their proper transmission, decoding, and display. Only the data that can be handled is processed on the client device. If the resources required to enhance image detail can no longer be provided, the refinement is stopped and the best content representation for the client device is considered to be found. We describe this strategy in detail and provide an empirical evaluation of its properties. Our results demonstrate that it significantly reduces resource consumption and represents the content in high visual quality (see Figure 1). Available knowledge about the content and viewing environment is summarized in profiles. Detailed statements to all associated aspects, i.e., profile design, profile creation, and profile matching,

Further author information: (Send correspondence to René Rosenbaum)

René Rosenbaum: E-mail: rosenbaum@ieee.org

Bernd Hamann: E-mail: hamann@cs.ucdavis.edu

are provided. We also provide means to consider individual user preferences. These preferences support aspects crucial for fluent and resource-efficient image browsing - maximal response time and data volume as well as minimal visual distortion. We show how they are related to available computing power, bandwidth, and display space in order to determine a truncation point that is specific for the current user and device.

Section 2 discusses related work and open research questions. Section 3 introduces a profile-based image adaptation approach for digital raster imagery. The strength of this strategy is underpinned in Section 4 by empirical results obtained from its implementation. Section 5 discusses its properties. Section 6 closes this paper with conclusions and directions for future research.

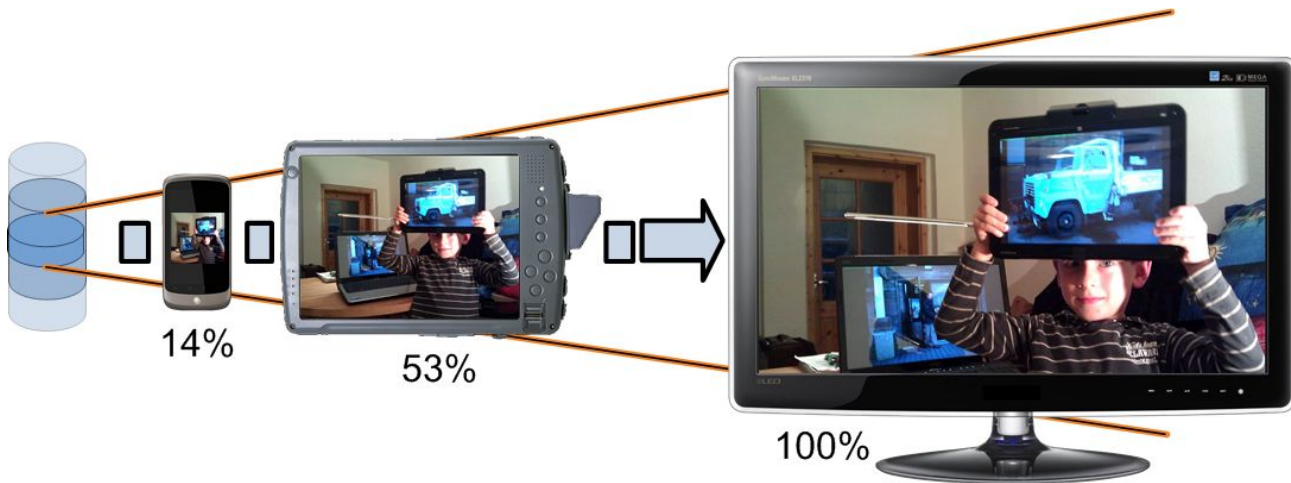


Figure 1. A raster image adapted to different viewing devices and the associated data volume required by the introduced adaptation strategy. Although the savings are high, the images show no visible degradation in quality.

2. RELATED WORK

In this section we provide a brief introduction to *scalable encoding and progressive refinement* whose beneficial properties are the foundations of the introduced framework. We also discuss related approaches in the field of *device adaptation*.

2.1 Scalable encoding and progressive refinement

Scalable encoding transfers an image into a representation that embeds multiple versions of the content at different detail levels. Modern schemes thereby allow one to restore the image with less details when not all of the encoded data is available for decoding.^{2,3} This is mostly achieved by transforming the image into a level-of-detail (LoD) hierarchy. Compression is applied in order to reduce space requirements. LoD hierarchy-building and compression are often combined in order to increase efficiency and to allow for an independent image decoding at individual detail levels.

Progressive refinement is a means for achieving scalable compression. It takes advantage of the fact that there are many redundancies at the different levels of the LoD hierarchy. Starting from the lowest detail level, only incremental data are stored and used to reconstruct the content at higher detail. This leads to a high compression performance. Typical codecs supporting and applying progressive refinement are SPIHT and JPEG2000. For further information to these codecs as well as scalable image encoding we refer to.⁴

One of the benefits of progressive refinement are the many different previews that result from an incremental reconstruction of the image. In order stream the LoD hierarchy, it must be sequenced. Dependent on the constraints of the applied compression, it can be traversed in many ways. Most of the proposed sequencing strategies are data-driven and provide a uniform content refinement by a level-ordered sequencing of the hierarchy. Related research to progressive imagery, e.g., RAUSCHENBACH AND SCHUMANN,⁵ also suggests the use of non-uniform refinement that can be used to satisfy individual user demands.

A typical communication system supporting progressive image refinement is shown in Figure 2. The source image to be displayed is labeled *original content*. *Encoding* and permanent *storage* of the resulting *progressive image* occur on *server side*. *Traversal* accesses the transformed image whenever it is requested and sequences and transmits the LoD hierarchy to the client. The client consists of a single *decoding* unit that manages the incremental content display. *Progressive selection* successively extracts the individual encoded detail levels from the data received. The associated data is then passed to a *decompression* and a *postprocessing* component and displayed. The remaining stages allow for profile-based adaptation and are described below.

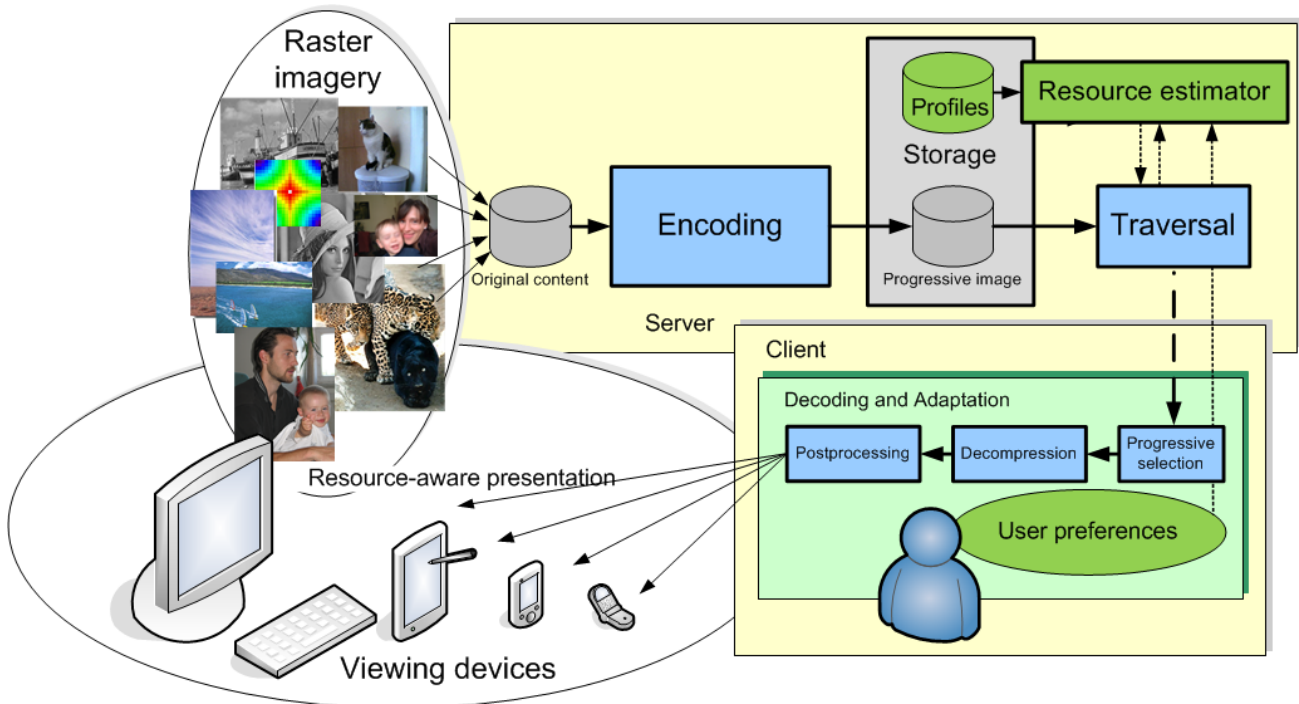


Figure 2. A common processing pipeline for progressive image communication enhanced by components for profile-based image adaptation (green).

2.2 Device adaptation

SKORIN-KAPOV ET AL.⁶ showed that the properties of the output device, such as its screen resolution, are important factors that must be taken into account to achieve appropriate image presentation. In the past, adaptation of visual content to the capabilities of the viewing set-up and user demands have mainly been accomplished by multi-resolution representations and the interactive or semi-automatic selection of an appropriate LoD of the image. Self-acting adaptation is considered to be an open research question and not widely covered in the scientific literature. This problem becomes complex due to the multiple factors influencing meaningful adaptation. Most of the related approaches focus on the problems imposed by constrained display size, such as solutions that strive for "plasticity" of the content (see⁷ for a detailed description of this principle), but do not aim at a general solution considering other or even multiple system constraints. One of the few exceptions is the proposal by MAO ET AL.⁸ focusing on the communication channel between two devices. Our approach considers screen space, bandwidth, and computing power available on client side, and thus provides a much more general solution.

Most adaptation strategies recently published are server-based. A related approach focusing on the adaptation of three-dimensional content has been introduced by KIM ET AL.⁹ Here, content customization is based on a generic schema of the MPEG-21 framework. The approach relies on a-priori knowledge about the output device and does not require any modification on client side. Although this is a meaningful strategy for selected application areas, frequent re-creation or storage of many different representations of the same image is resource-intensive. Furthermore, this approach does not allow for interactive changes in the representation once it has

been created. Our approach uses a single multi-purpose data structure that is created once and used for all requests allowing for user interaction combined with non-redundant data transmission.

New research directions that apply principles from scalable image coding recently led to proposals founded on progressive content. The adaptation strategy proposed by THIEDE ET AL.¹ is fully client-based and provides a number of benefits, such as low complexity and independence from any prior knowledge about data, device, or user. However, the adaptation strategy is *reactive*¹⁰ and requires additional computing power on client side. Thus, it cannot be meaningfully used in application domains characterized by low-power client devices. The adaptation strategy introduced in this paper is also based on progressive content, but is fully server-based and does not require additional computing power on client side. Although mentioned by ROSENBAUM & SCHUMANN³ and implemented for geometry data by ROSENBAUM ET AL.,¹¹ profile-based device adaptation has not yet been applied to raster imagery.

Summary Approaches for device adaptation already exist. As most of them focus on single aspects of the device only or require manual configuration, they are of limited applicability. Scalable image compression transforms an image into a representation that can be incrementally enhanced in detail. The advantage that refinement can be stopped whenever the capabilities of the device are used up to capacity has already been used for device adaptation, but issues still exist.

3. PROFILE-BASED ADAPTATION USING SCALABLE IMAGERY

In this section, a novel strategy for device adaptation taking advantage of scalable imagery is presented. After introducing the *main idea* of the strategy, we state the supported *user requirements and system resources* considered by this adaptation approach, and finally describe the details of *the adaptation procedure*.

3.1 The main idea - Create once, use many times

To overcome the various problems imposed by the complex creation or storage of many similar image representations for all potential viewing devices, we take advantage of scalable imagery. When requested an image is progressively refined until the system resources on client side are exhausted. The *main goals* are a *high degree of device adaptation, redundancy-free transmission*, and the requirement that *only data that can be processed and displayed are transferred*.

The adaptation concept introduced can be literally stated as: "Create once, use many times." A progressive image is created once and stored on a central server. If this image is requested, the associated data is incrementally streamed to the respective client device, and then displayed. By successively reconstructing the image with increasing LoD and resource utilization, it can be determined when the resources of the device are used to their full capacities. When this is predicted, the server truncates the progressive stream stopping the refinement process. This approach makes possible the multiple use of a single data source and a flexible, low-complexity adaptation. The fact that initial, low-resolution previews require few, subsequent, more highly refined views much resources supports this strategy.

The criteria used to determine when refinement should be completed include system resources crucial for fluent and efficient image display: *maximal response time* and *data volume* as well as minimal *primitive size*. They are specified by the users giving them the option to adapt the viewing experience to their liking. These requirements are then mapped to the available *computing power, bandwidth, and display space* in order to determine a truncation point that is specific to the current user and device. This combines the user and device perspectives into a single approach. Contrary to THIEDE AT AL., this truncation point is calculated on server side and before data transmission.

3.2 Considered user requirements and system resources

Individual users have individual demands. We consider the users as the most important factor in a computing system and align the definition and behavior of the introduced framework to their requirements and preferences.

User preferences are specified by a requirements tuple $lim(t, vol, size)$ which covers three aspects crucial for image presentation and browsing. The attribute t refers to the *response time* of the system ($0 \leq t < \infty$)

and is an important factor for usability. The attribute *vol* stands for the maximum *data volume* that can be transmitted ($0 \leq vol < \infty$) and is often used to limit transmission costs. *size* expresses the minimal *primitive size* ($0 \leq size < \infty$) that can still be identified on a small screen. The respective values can be interactively specified by the user, synthesized from prior or default user preferences, or can be based on empirical research, such as the "web stress index".¹²

In order to consider the requirements tuple for adaptation, its attributes are mapped to the main resource types of a mobile device as follows:

Computing power The provided computing power corresponds to the *response time*, *t*, of the system. It is assessed as the time required for updating the current content after new incremental data has arrived. It is constrained by the device's ability to decode and display the incoming data as quickly as possible.

Bandwidth Transmission via low-bandwidth channels also strongly influences the *response time* of the system. This especially applies when large indivisible data chunks must be transferred to obtain an image preview. It is assessed in the amount of time necessary to transfer a constant amount of data. Another requirement related to communication via low-bandwidth channels is the maximum *data volume*, *vol*, that is transferred to the client. It is assessed as the data volume that arrives on client side.

Display space Fitting visual content to a given screen estate usually causes undesirable effects for presentation that can be summarized by the term visual distortion. As this involves many semantic factors and varies dependent on the kind of content, it is difficult to quantify in a rigorous manner. Fundamentally, it can be seen as a measure assessing how well the image matches the available screen space. For the remainder of this paper, statements about display space refer to screen resolution.

Screen as well as image dimensions are stated by the number of available pixels in each dimension. The *primitive size* refers to the number of screen pixels per image pixel. Displaying a low-resolution image on a sufficiently large screen would lead to super pixels (large primitive size; $size > 1$), a high-resolution image to multiple image pixels assigned to a screen pixel (small primitive size; $size < 1$). The case of there being a single image pixel for each screen pixel ($size = 1$) is the best match.

The individuality of the user preferences and associated system resources requires their independent assessment and processing. Although the adaptation is considered completed when a single requirement can no longer be satisfied by the system's capabilities, it is meaningful to provide means to continue refinement in order to leave the final decision to the user.

3.3 The Adaptation Procedure

This section presents the details of the introduced approach. Adaptation is accomplished by taking advantage of prior knowledge about the current viewing device and the image to be displayed. This information is stored in profiles and used to calculate the most appropriate image detail. Due to the fact that resource consumption is estimated on the server side, the proposed strategy is *predictive*¹⁰ and thus does not require much additional computational resources on client side.

The adaptation process is illustrated in Figure 2. On the server side specific *profiles* are used to *establish the display context*. Profiles can be made available by storing them in a server side data base or by using individual XML-files that may also be provided by the client, e.g., by devices new to the environment. A *resource estimator* uses these profiles to *infer from the context* by matching them in order to find the optimal truncation point for the current preferences. As all requirements of the current *user* are also described by a profile, the resource estimator is able to perform a *compliance check* on all profiles. The client device only displays the adapted and transmitted data not requiring any modification to the traditional image handling.

3.3.1 Establishment of the display context

The display context is established via a-priori knowledge stored in profiles. In this section, we show how these profiles are structured and how they can be created for specific image and device entities. We distinguish between three different kinds of profiles: (1) device, (2) data, and (3) user profile (see Figure 3). The *device profile* describes the resources of the viewing device that correspond to the supported user requirements. The *data profile* serves as a means to consider properties of the image. The *user profile* allows for seamless integration of the user preferences into this strategy.

Device profile This profile captures the main system resources of the device. It summarizes *processing power*, *bandwidth* of the used communication channel, and *screen resolution* of the device. Due to the sheer variety of specifications and types, devices are categorized in classes. Aggregated values associated with each class are used to determine processing power and bandwidth. To select the value representing the capabilities of a specific class, we propose to conduct a series of empirical assessments and to use the resulting average values. Assessment strategies as described in¹ can be used as a meaningful foundation. Screen resolution is defined by the number of pixels in each direction.

Data profile This kind of profile mainly describes and stores the properties of all levels of the LoD hierarchy. It is specified for each image independently. *Chunk size* represents the amount of data associated with a certain level. *Chunk complexity* specifies the effort needed to integrate the associated incremental data in the current image resolution. In order to determine these values, their decoding and display times are assessed empirically for each device class. The *spatial extent* refers to the number of pixels that is associated with the respective resolution level. When adaptation is performed, this resolution is transferred into screen space in order to determine the detail improvement of the update on the respective display.

User profile The user profile has entries identical to the introduced user requirements. Thereby, response time is specified in milliseconds and bandwidth in bytes. Once specified, they are transferred into a profile that is specific to a certain user. Due to the fact that user preferences are often independent from the browsed image, user profiles can be stored for future use. To achieve this, established approaches, such as the WURFL Device Description Repository, can be used. Due to the specifics of the data and user profile, we propose to use schemes that allow for more generic descriptions, such as XML.

The introduced profiles do not represent all factors influencing appropriate image representation and browsing and strictly focus on attributes that are essential to illustrate the adaptation procedure. The dependencies between the different attributes and profiles are crucial for the success of the adaptation approach and are explained next as part of the actual adaptation procedure.

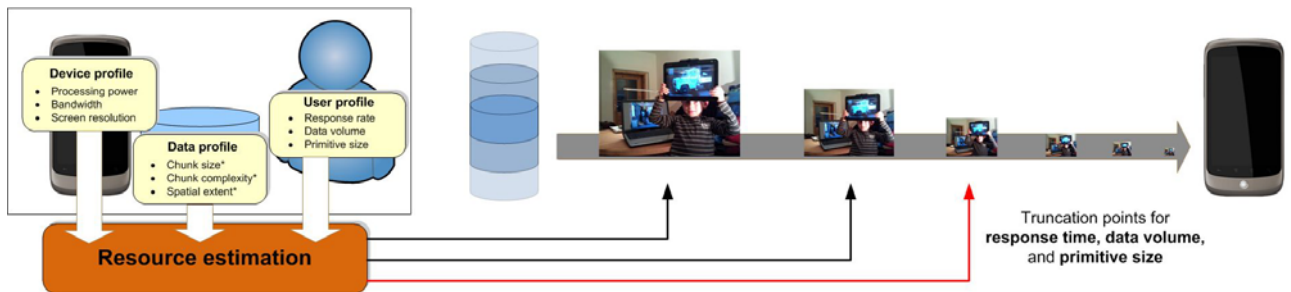


Figure 3. Profiles and attributes describe properties of device and image as well as user requirements. Resource estimation determines appropriate truncation of the progressive data stream for device adaptation (red arrow).

3.3.2 Interference from the context and compliance check

The established device context is used to estimate an appropriate LoD. Interference from the context and the compliance check are tightly coupled within a single process. Profile matching combines the information stored in the profiles in order to estimate the best truncation point for the current user: the data profile provides information about the inherent image detail levels and the device profile the details to the current device. The compliance check uses the information provided by the user profile to determine the upper limit for resource consumption. Following the general framework, the whole adaptation process considers each attribute of the requirements tuple separately leading to three different truncation points. In order to ensure compliance with all specified user requirements, the point leading to the shortest data stream (see red arrow in Figure 3) is chosen. The individual attributes are processed according to the strategies discussed in the following:

Response time Considering response time requires knowledge about the *processing power* of the viewing device and the *chunk complexity* of the image resolution as well as *chunk size* and provided *bandwidth*. The first inherent image detail that is too complex to be decoded and displayed within the required response time marks the truncation point with regard to processing power. The same applies to bandwidth; the corresponding truncation point is found as soon as a data chunk that is too large to be received within response time is detected. The point leading to the shortest stream is selected as the truncation point associated to response time.

Maximum data volume To adapt to data volume is straightforward; the *chunk sizes* of all already displayed detail levels are accumulated until the maximally allowed amount is reached. This determines the associated truncation point.

Minimum primitive size Matching for minimum primitive size is based on the *spatial extent* of the respective image resolution. As the spatial extent is based on a pre-assigned resolution, it must first be transferred into the domain of the viewing device. This can be achieved via its specified *screen resolution*. If the resulting *size* value is equal or smaller than one, the third truncation point is found.

Once the appropriate range of data is determined, the server selects and incrementally streams the progressive image until the truncation point is reached. The client only decodes the received data chunks and restores the image until no further detail is received. Once this is detected, the most appropriate detail level and image representation is considered to be found.

4. RESULTS

The introduced device adaptation strategy has been implemented in order to evaluate its performance. To introduce scalability, the imagery was encoded using the modern image compression standard JPEG2000. This section provides the results obtained from experiments with regard to adaptation accuracy to the three user requirements - *response time*, *data volume*, and *primitive size*. To demonstrate the wide applicability, the introduced strategies were tested with a wide range of user requirements and different hardware set-ups. A test suite of 20 images of different dimensions (1,000-3,000 pixels in both dimensions) and layouts was created. The following statements refer to a single image of dimensions $3,000 \times 3,000$ pixels in order to demonstrate details and specifics. The image was encoded by a seven-level discrete wavelet transform (DWT). To demonstrate the degree of adaptation, the term adaptation error stating the delineation of the obtained results from the user preferences is introduced.

4.1 Response time

Our strategy to assess response time was to measure the time needed to decode and display the image in the highest currently available detail level. We simulated the varying consumption of computing power for different classes of viewing devices by using a common laptop device at different degrees of capacity utilization (loads of 10%, 40%, and 80%). Data profiles were created for each hardware set-up.

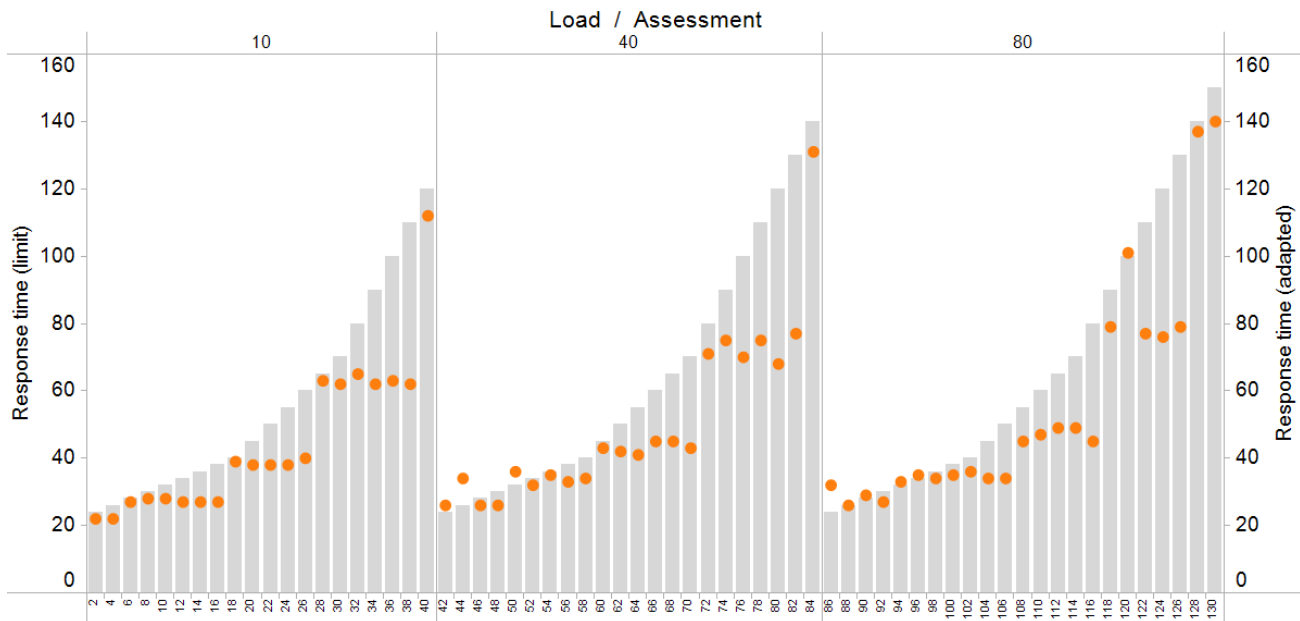


Figure 4. Adaptation accuracy regarding the maximum response time for different viewing devices: The gray bars indicate the largest accepted response times, the dots the times obtained for profile-based adaptation. The results are slightly better and vary less on hardware with a low computing load (left) relative to one with a high computing load (right).

Figure 4 illustrates the degree of adaptation by the delineation of the assessed values from the specified user demands. In Figure 4, left, it can be observed that the degree of adaptation is high, although an exact match could only be achieved for 5% of the test cases. Profile-based adaptation performs much better for lower levels which is due to the nearly accurate information about the image defined by the profiles. Contrary, performance decreases towards the higher detail levels. This is mainly due to the steadily increasing processing and transmission times allowing external factors to interfere with the establishment of the device context. The introduced strategy always complies with the user requirements and also reacts quickly to changes from one inherent resolution level to another.

Strong differences in the results obtained for different CPU capacity utilizations were expected. In the case of high CPU load the capacities are shared between different applications causing the system behavior to change unpredictably. This reduces adaptation accuracy. The obtained differences (see Figure 4, left to right), however, are surprisingly small. While the average adaptation errors are similar for all capacity set-ups, only small increases in the standard deviation (2%) and the maximal adaptation error (10%) were obtained.

4.2 Amount of data

Figure 5, left, illustrates the degree of adaptation for increasing user-defined data volumes. Due to the fact that only discrete resolution levels are supported by the DWT used, exact adaptation can only be achieved when the user preferences match the amount of data required to reconstruct one of the inherently supported resolutions. In those cases a "near-ideal" adaptation was obtained. Due to the fact that the amount of data required to reconstruct the inherent resolutions increases exponentially, the adaptation error is very small when only small amounts of data are accepted by the user.

The amount of data associated to the different hierarchy levels can be exactly assessed and accurately specified in the profiles. Thus, profile-based adaptation is very accurate. This also applies to requirements leading to truncation points located close to transitions between resolution levels. An over-estimation of the amount of data was not observed.

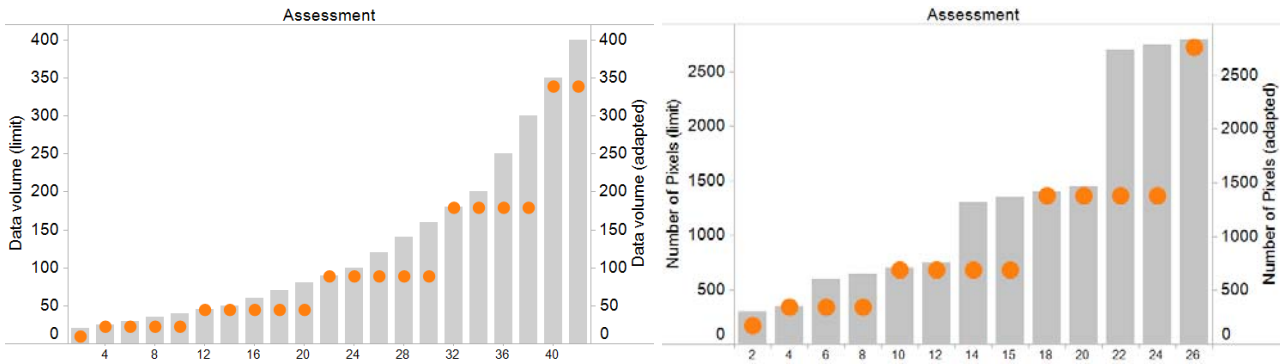


Figure 5. Adaptation regarding the maximum amount of data to be transmitted (left) and minimal primitive size (right): The gray bars indicate the user stated preferences (left: largest amount of data; right: minimal primitive size). The dots indicate the values obtained for the adaptation strategy. Although adaptation accuracy is high, only resolutions supported by the encoded image stream can be used for adaptation.

4.3 Primitive size

The degree of adaptation regarding primitive size was examined for many different user preferences (see Figure 5, right). Due to fact that the proposed adaptation strategy fully relies on the image resolutions inherent in the scalable image, similar results as stated for the amount of data - the more limited the device, the smaller the adaptation error - were obtained. If one of the inherent image resolutions does not match the desired resolution, the adapted resolution will be smaller. Due to the options for an accurate assessment and estimation of this system property, determination of upcoming resolutions is exact. Over-estimated resolutions or imprecisions are not possible.

5. PROPERTIES OF PROFILE-BASED IMAGE ADAPTATION

5.1 Advantages

Due to the use of a single predefined progressive image that is created once and used multiple times, many limitations of traditional adaptation approaches are overcome. Introducing scalability into the image is the most complex process but done in a preprocessing step on server side. During image transmission no more data than required for the adapted image representation is transferred and in a non-redundant manner. The proposed strategy is founded on a typical image communication pipeline and thus can be broadly applied. It can be migrated into many existing and future imaging systems. Due to its reliance on a single image source, it is most appropriate for systems characterized by central data storage and heterogeneous viewing devices.

Profile-based image adaptation also provides high adaptation accuracy. As it is possible to add more properties and profiles, this strategy offers the flexibility required to make it available to a broad variety of application domains. As the scalable image and the profiles can be created in a way that is most meaningful for a certain purpose, this strategy is most appropriate when potential viewing devices and images are known in advance. As this strategy is fully server-based, no additional resources are required on client side making adaptation available even for highly resource-constrained viewing devices.

5.2 Drawbacks

The use of a single scalable data structure also imposes drawbacks. The introduced adaptation strategy strongly relies on many and meaningful detail levels. When only a few levels are available, adaptation accuracy is coarse. Due to the exponential growth of the image data required to reconstruct levels of higher resolution, this becomes apparent especially for larger data volumes. The delineation from user demands requiring a small amount of data, however, is usually little. Due to the fact that a new LoD hierarchy of the image must be created when content changes, the concept loses some of its benefits when applied to dynamic content.

A general drawback of profile-based adaptation schemes is that prior knowledge about the device or image is required and profiles must be created and stored. Although this must be done only once, this process requires

some additional efforts before adaptation can take place. Thus, the introduced strategy is less suited for environments characterized by a non-predictable device ensembles. Due to the fact that profiles are static, profile-based adaptation is also not suited for environments and client devices that are subject to strong temporal changes in the available resources.

6. CONCLUSIONS

We have introduced a profile-based strategy for device adaptation of raster imagery. It takes advantage of scalability and progressive refinement as part of modern image compression schemes. An image is stored once and upon delivery refined until the system resources of the viewing device reach their capacity. This approach saves valuable resources, such as computing time and bandwidth, and also leads to a better visual representation compared to existing strategies. Empirical results obtained from an implementation of the approach show that it is well-suited for environments where devices are a-priori known and where high adaptation accuracy is mandatory. As adaptation takes place on server side, it is also available for low-power devices. When image content changes often or not much information about the viewing device or the image is available, the strategy loses some of its benefits.

Future research could explore other options for scalability, such as quality layers in JPEG2000-encoded imagery, to further increase adaptation accuracy. This also applies to non-uniform refinement schemes using region-of-interest approaches. One could also investigate in greater detail interactivity as a common part of image browsing systems. Here, an enhanced adaptation scheme could consider the fact that some data might already be available on client side from previous browsing stages and can be re-used to avoid redundant data transfer and processing.

REFERENCES

- [1] Thiede, C., Schumann, H., and Rosenbaum, R., “On-the-fly device adaptation using progressive contents,” in [*Proceedings of Intelligent Interactive Assistance and Mobile Multimedia Computing 2009*], (Nov. 2009).
- [2] Lin, E., Podilchuk, C., Kalker, T., and Delp, E., “Streaming video and rate scalable compression: What are the challenges for watermarking,” in [*Proceedings of SPIE Security and Watermarking of Multimedia Contents III*], **4314**, 116–127 (2001).
- [3] Rosenbaum, R. and Schumann, H., “Progressive raster imagery beyond a means to overcome limited bandwidth,” in [*Proceedings of Electronic Imaging - Multimedia on Mobile Devices 2009*], (Jan. 2009).
- [4] Taubman, D. and Marcellin, M., [*JPEG2000: Image compression fundamentals, standards and practice*], Kluwer Academic Publishers, Boston (Nov. 2001).
- [5] Rauschenbach, U. and Schumann, H., “Demand-driven image transmission with levels of detail and regions of interest,” *Computers and Graphics* **23**, 857–866 (Dec. 1999).
- [6] Skorin-Kapov, L., Komericki, H., Matijasevic, M., Pandzic, I. S., and Mosmondor, M., “MUVA: a flexible visualization architecture for multiple client platforms,” *Journal on Mobile Multimedia* **1**(1), 3–17 (2005).
- [7] Coutaz, J. and Thevenin, D., “Plasticity of user interfaces: Framework and research agenda,” *Proc Interact99 Edinburgh A Sasse C Johnson Eds IFIP IOS Press Publ* **99**, 110–117 (1999).
- [8] Mao, S., Bushmitch, D., Narayanan, S., and Panwar, S., “Mrtp: a multiframe real-time transport protocol for ad hoc networks,” *Multimedia, IEEE Transactions on* **8**, 356–369 (april 2006).
- [9] Kim, H. S., Joslin, C., Di Giacomo, T., Garchery, S., and Magnenat-Thalmann, N., “Device-based decision-making for adaptation of three-dimensional content,” *The Visual Computer: International Journal of Computer Graphics* **22**(5), 332345 (2006).
- [10] Funkhouser, T. A. and Squin, C. H., “Adaptive display algorithm for interactive frame rates during visualization of complex virtual environments,” in [*Proceedings of the 20th annual conference on Computer graphics and interactive techniques*], 247–254, ACM, Anaheim, CA (1993).
- [11] Rosenbaum, R., Gimenez, A., Schumann, H., and Hamann, B., “A flexible low-complexity device adaptation approach for data presentation,” in [*Proceedings of Electronic Imaging - Visualization and Data Analysis 2011*], 78680F–78680F–12 (2011).
- [12] Korsah, K., “2009 CA web stress index,” tech. rep., Computer Associates International (2009).