

Adapting Images on Proxies for Small Form Factor Devices

Xing Xie[‡], Xin Fan^{*†}, Wei-Ying Ma[‡], He-Qin Zhou[†]

[‡]Microsoft Research Asia
5F, Sigma Center, No. 49 Zhichun Road
Beijing, 100080, P.R.China
{xingx, wyma}@microsoft.com

[†]Department of Automation
University of Science and Technology of China
Hefei, 230027, P.R.China
van@mail.ustc.edu.cn, hqzhou@ustc.edu.cn

Abstract

Pictures have become increasingly common and popular in mobile communications. In our previous work, an attention model based scheme has been proposed to facilitate the browsing of large images on small displays. However, the image analysis components involved are quite costly in terms of power consumption and time, especially for thin clients. To deal with this problem, in this paper, we introduce our experience in using a proxy-based system for deploying image adaptation procedures. The images are processed while delivering from the server to the mobile clients and the adaptation results are stored as metadata in the image header. Experimental evaluations indicate that our approach is efficient and practical.

Keywords

Adaptive content delivery, form factor, attention model, content services network

1. Introduction

Handheld devices with diverse capabilities including embedded digital cameras are undergoing a considerable progress because of their portability and mobility. Now people can easily capture and share photos on these small-form-factor devices anywhere and anytime. One recent trend is “moblogging” [14], or mobile weblogging. It is the use of a phone or other mobile devices to publish content to the World Wide Web in real time, whether that content is text, images, media files, or some combinations of them.

In order to make people really enjoy the ease of mobile communications, many hurdles still need to be crossed. Among them, major crucial challenges include the limited accessing bandwidth and display sizes of mobile devices. Thanks to the galloping development of both hardware and software, the bandwidth condition is expected to be greatly improved. However, in the foreseeable future, the display, i.e. the form factor, will continue to be the major constraint on small mobile devices.

Many efforts have been put on image adaptation and related fields from quite different aspects [4][7][15]. Most of these works only focused on compressing and caching contents in order to reduce the data transmission for fast

delivery. Therefore, the results are often not consistent with human perception on small displays because of excessive resolution reduction or quality loss.

We have recently proposed an attention model based image adaptation approach in [2][6][9]. Instead of treating an image as a whole, we manipulate each region-of-interest in the image separately, which allows delivery of the most important region to the client when the screen size is small. In [2], an extensible image attention model has been introduced. A branch and bound algorithm was developed to find the optimal cropping region efficiently. Though this approach achieved satisfactory results in our user study, much other information which a user cares may be lost. In [6], we proposed to employ a widely-used presentation technique, Rapid Serial Visual Presentation (RSVP), in which space is traded for time [1]. An image is decomposed into a set of spatial-temporal information elements which are displayed serially, each for a brief period of time. In [9], we extended our previous model to include a time constraint and formulated the optimal browsing path problem based on the information foraging theory [16]. Experimental results indicated that this approach is an effective way for viewing large images on small displays.

The image attention model can be manually pre-assigned by authors or publishers, which however could be labor intensive. A more plausible approach is to detect each attention object automatically. A set of algorithms to generate the attention model has been discussed in [2][9].

Due to the very limited computational resource of current mobile devices, the model generation cost is still unacceptable for a real time user interface. As we will show later, even after optimization, it usually took a long time to detect all the attention objects on a typical Pocket PC. In order to deal with this problem, in this paper, we propose to use a proxy-based system to process images for small-form-factor devices and the adaptation results are stored as metadata in the image header.

The rest of this paper is organized as follows. Section 2 briefly introduces the image attention model and its performance. In Section 3, we present the detail of our system framework. We give the experimental results of the proposed scheme in Section 4. Finally, concluding remarks and discussions are provided in Section 5.

* This work was performed when the second author was a visiting student at Microsoft Research Asia.

2. Image Attention Model and Its Performance

In this section, we will first give a brief introduction to the image attention model and then discuss its computational cost in detail.

2.1 Image Attention Model

The image attention model is defined as a set of attention objects. Therefore, many image browsing tasks can be treated as manipulating attention objects to provide as much information as possible under resource constraints.

Definition 1: The visual attention model for an image is defined as a set of attention objects:

$$\{AO_i\} = \{(ROI_i, AV_i, MPS_i, MPT_i)\}, \quad 1 \leq i \leq N \quad (1)$$

where

AO_i ,	the i^{th} attention object within the image
ROI_i ,	Region-Of-Interest of AO_i
AV_i ,	attention value of AO_i
MPS_i ,	minimal perceptible size of AO_i
MPT_i ,	minimal perceptible time of AO_i
N ,	total number of attention objects

We assign four attributes to each attention object, which are *Region-Of-Interest (ROI)*, *attention value (AV)*, *minimal perceptible size (MPS)*, and *minimal perceptible time (MPT)*. The notion of '*Region-Of-Interest (ROI)*' is borrowed from JPEG 2000 [4], which is referred as a spatial region within an image that corresponds to an attention object. *Attention value (AV)* is a quantified value indicates the weight of each attention object in contribution to the information contained in the original image. *Minimal perceptible size (MPS)* represents the minimal allowable spatial area of an attention object. It is introduced as a threshold to avoid excessively sub-sampling during the reduction of display size. *Minimal perceptible time (MPT)* is a threshold for the fixation duration when browsing an attention object. If an attention object does not stay on the screen longer than *MPT*, it may not be perceptible enough to let users catch the information.

2.2 The Computational Cost

The total computational cost for image attention model based processing can be divided into three parts:

- Detection of attention objects in an image (i.e. the position and size of *ROI*).
- Calculation of model parameters for each attention object (i.e. *AV*, *MPS*, and *MPT*).
- Generation of the optimal cropping region or browsing path based on image attention model.

Our test bed throughout this paper is a Compaq iPaq 3670 with 64M memory, 206MHz CPU, 320x240 display and PocketPC 2002 as its operating system.

For the last part, though its complexity is exponential with the number of attention objects in the worst case, our approach can be conducted efficiently because the number of attention objects in an image is often less than a few dozens and the attention values are always distributed quite unevenly among attention objects. For example, in [9], we reported that the average time cost for optimal path generation is 230 microseconds, with variation from 150 to 350 microseconds. For the second part, the cost is even smaller since it is only linear with the number of attention objects. We separate it from the first part since the value of *AV*, *MPS* and *MPT* usually depends on user preference or application scenario while the value of *ROI* only depends on image content itself. Therefore, they can be performed on different locations.

The most costly computation lies in the detection of attention objects. In our current implementation, we consider three types of objects, i.e. saliency [12], face [8] and text [3]. The total detection time will be above 70 seconds if we directly port those detection algorithms to our test bed. Such performance is unacceptable for a real-time user interface. In addition, the power consumption of these computations would be very large.

We have tried lots of optimization approaches like reducing the number of floating operations and down-sampling the image to a small resolution before further processing. The time cost can be improved to 8-10 seconds after optimization. However, the detection precision will be sacrificed and the speed is still very slow. Therefore, a better approach should be designed to overcome the limited capability of mobile devices.

3. System Framework

In this paper, we will only focus on Web applications such as moblogging or online photo albums. For these Web images, different kinds of approaches can be employed to move the detection modules out from client devices. According to the location of processing, they can be classified into three types:

- Server-based solutions. All the images are pre-processed on Web servers. This requires adding of special modules to the server software, for instance, using ASP.NET mobile controls.
- Sync-based solutions. The images are processed when synchronizing between PC and mobile devices. This can be applied to some special scenarios where the mobile devices can be easily connected to a desktop PC.
- Proxy-based solutions. The images are processed while delivering from the server to the mobile clients. This approach does not require any modification on server or client.

Only proxy-based solution will be further discussed since it is more scalable and extensible. The basic idea is to extend the functionalities of a traditional caching proxy to value-added processing like image adaptation. In [7], the

authors proposed a framework for determining when/whether/how to transcode images in a HTTP proxy while focusing their research on saving response time by JPEG/GIF compression, which is determined by bandwidth, file size and transcoding delay. Since our approach does not focus on the bandwidth, the tradeoff between adaptation time and delivery time will not be studied. Instead, we present a scheme to instruct the Web proxies or so called service-enabled Web caches to perform adaptation task actions based on a subscription model.

3.1 Content Services Networks

Previously, we have proposed a subscription based system framework named content services networks (CSN) [10][11] which aim to make content delivery networks (CDN) capable of delivering content adaptation services to both content providers and content consumers. In this paper, we will apply this framework to provide our image adaptation functions on proxies.

Figure 1 shows the overall system, which constitutes two layers of network infrastructures: content delivery overlay (i.e. CDNs) and service delivery overlay. The content delivery overlay is constituted of a network of service-enabled Web caches which extend the functionalities of traditional Web caches for performing value-added processing. The service delivery overlay consists of a large number of application servers which act as remote call-out servers for service-enabled Web caches. These two overlays work together to provide content-oriented Web services.

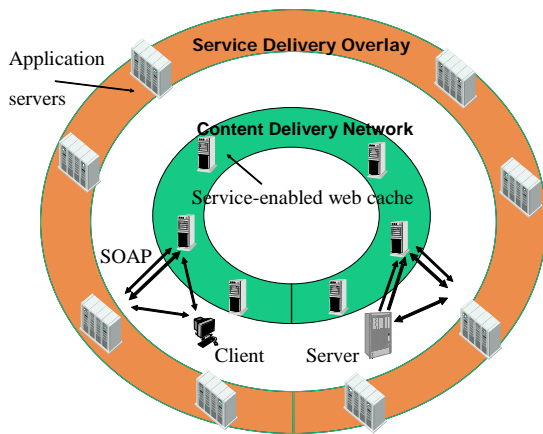


Figure 1. The architecture of content services network.

In [10][11], a service subscription and delivery protocol is defined to allow the management servers in CSN to upload service instructions to the service-enabled Web caches to control their behaviors. While trying to achieve this goal, we ensure that the proposed model interacts with other existing network elements seamlessly so as not to undermine the success of end-to-end nature of Internet client/server interactions.

Before the image adaptation service becomes available, it needs to be registered in the UDDI (Universal Description and Discovery Integration) [17] registry first. The received

components such as service specifications and binaries from service providers are stored the service database. It then publishes the service information to the UDDI for public discovery and access. In order to use the image adaptation service, a mobile client needs to first find and subscribe to the service via UDDI registries. Then the service instructions are generated and transferred from the management servers to the service-enabled Web caches that the subscriber is associated with. The service-enabled Web cache determines if a message needs services according to the service instructions. In our case, the instructions may simply be type comparison, i.e., whether the content is an image. If the message satisfies the condition specified in an instruction, the service-enabled Web cache will provide the service locally. The service can also be implemented as executing a remote callout service to the application servers. The processing results can be stored in the cache to avoid redundant computations for the same image.

3.2 Metadata Scheme

In order to deliver the detection results from proxy to the client, a metadata scheme is necessary to store the image attention model.

Currently, most of the digital cameras use Exif [5] format to insert image/camera information or thumbnail image to the photos. Exif format is designed to be compatible with JPEG specification. Therefore, Exif images can be viewed by JPEG compliant Internet browser or picture viewer as usual JPEG images.

We employ an approach similar to Exif format to insert the attention model information. First we briefly introduce the structure of JPEG files. Each JPEG file contains a set of data segments which starts with a “marker”. The marker defines the type of data segments that follows. A set of special markers are named as “application markers”. They can be defined and used by third-party applications. For example, Exif used a marker of 0xFFE1. In our system, we use 0xFFE3 as our application marker and the detail of the metadata scheme is shown in Table 1.

The metadata starts from application marker 0xFFE3. SSSS stands for the size of the data segment and it includes the size of SSSS itself. After SSSS, the first eight bytes are a special header which identifies whether it is an image attention model or not. In our scheme, ASCII character “MSRASCOT” has been used. Following the header, NNNN gives the number of attention objects. There can be at most 65535 attention objects for each image. It is sufficient for most images on the Web. Then for each attention objects, 16 bytes (DDDD...) are used to describe its properties like type, position and size. 6 the 16 bytes are reserved for future use. Currently, the value of type is defined as: 0-Face, 1-Text, 2-Saliency, 3-Other.

Table 1. The metadata scheme of image attention model.

Value	Description
FFE3	Application marker

SSSS		Data size
6D73 7261 7363 6F74		Header
NNNN		Number of attention objects
DDDD...	TTTT	Type
	LLLL	Left
	OOOO	Top
	IIII	Right
	BBBB	Bottom
	RRRR	Reserved
	RRRR	Reserved
	RRRR	Reserved

The total size of the metadata segment is $14+16*N$ bytes where N is the number of attention objects. For a typical image, it will be less than 200 bytes which is much smaller than the image size.

After the image and the metadata has been delivered to the client, the model parameters can be calculated and then the optimal cropping region or browsing path can be calculated based on the attention model.

4. Evaluation

We have developed a service-enabled Web cache based on Microsoft ISA Server 2000 [13]. Microsoft ISA Server 2000 is an extensible firewall and Web cache server. It provides Internet Server API that can be used to develop users' own extensions named Web filters. Web filters are dynamic-link libraries that are loaded when the ISA Server is started and stay in memory until the service shuts down. They can be configured to receive special filter-event notifications that occur with each HTTP request and response that the ISA server receives.

We implemented a special Web filter using ISAPI, which enables adaptation on HTTP messages containing images. This filter analyzes HTTP response messages passing by and performs the detection algorithms if it is an image file. The processing is executed locally on the proxy which is a Windows XP Server with two P4 3.1 GHz CPUs and 4G memory. The average model generation time for an image is 1.0 second.

5. Conclusions

Currently, the predominant methods for accessing large images on small devices are down-sampling or manual browsing by zooming and scrolling. We have proposed a novel image browsing strategy based on image attention model to facilitate scrolling and navigation of large pictures on devices with small displays. In this paper, we present our recent experience in deploying the image adaptation procedures on intermediary proxies. The images are processed while delivering from the server to the mobile clients and the adaptation results are stored as metadata in the image header. Experimental evaluations show that our approach is efficient and practical.

With the satisfactory results from our experiments, we plan to extend our work to other media types, such as

videos and Web pages. We will continue to investigate these directions in our future work.

6. Acknowledgements

We would like to express our special appreciation to Hao Liu and Mingyu Wang for their insightful suggestions and the Media Computing Group of Microsoft Research Asia for their generous help in building some of the image analysis modules.

7. References

- [1] O. Bruijn and R. Spence, Rapid serial visual presentation: a space-time trade-off in information presentation, Proc. of Advanced Visual Interfaces, pp189-192, 2000.
- [2] L.Q. Chen, X. Xie, X. Fan, W.Y. Ma, H.J. Zhang, and H.Q. Zhou, A visual attention model for adapting images on small displays, ACM Multimedia Systems Journal, to appear.
- [3] X.R. Chen and H.J. Zhang, Text area detection from video frames, Proc. of 2nd IEEE Pacific-Rim Conf. on Multimedia, pp222-228, Beijing, China, Oct. 2001.
- [4] C. Christopoulos, A. Skodras, and T. Ebrahimi, The JPEG2000 still image coding system: an overview, IEEE Trans. on Consumer Electronics, Vol. 46, No. 4, pp1103-1127, 2000.
- [5] Exif Version 2.2, JETA, Apr. 2002. <http://tsc.jeita.or.jp/avs/data/cp3451.pdf>.
- [6] X. Fan, X. Xie, W.Y. Ma, H.J. Zhang, and H.Q. Zhou, Visual attention based image browsing on mobile devices, Proc. of ICME 2003, Vol. I, pp53-56, Baltimore, USA, Jul. 2003.
- [7] R. Han, P. Bhagwat, R. Lemaire, T. Mummert, V. Perret, and J. Rubas, Dynamic adaptation in an image transcoding proxy for mobile web browsing, IEEE Personal Communications, pp8-17, Dec. 1998.
- [8] S.Z. Li, L. Zhu, Z.Q. Zhang, A. Blake, H.J. Zhang, and H. Shum, Statistical learning of multi-view face detection, Proc. of 7th European Conference on Computer Vision, Vol. 4, pp67-81, Copenhagen, Denmark, May 2002.
- [9] H. Liu, X. Xie, W.Y. Ma, and H.J. Zhang, Automatic browsing of large pictures on mobile devices, ACM Multimedia 2003, Berkeley, CA, USA, Nov. 2003.
- [10] W.Y. Ma, B. Shen and J. Brassil. Content services network: the architecture and protocols. Proc. of the 6th Intl. Workshop on Web Caching and Content Distribution, Boston, Jun. 2001, 83-101.
- [11] W.Y. Ma, X. Xie, C. Yuan, Y. Chen, Z. Zhang, and H.J. Zhang, Enabling multimedia adaptation services in content delivery networks, 3rd International Workshop on Intelligent Multimedia Computing and Networking, Cary, North Carolina, USA, Sep. 2003.

- [12] Y.F. Ma and H.J. Zhang, Contrast-based image attention analysis by using fuzzy growing, ACM Multimedia 2003, Berkeley, CA, USA, Nov. 2003.
- [13] Microsoft Internet Security and Acceleration Server, <http://www.microsoft.com/isaserver/>.
- [14] Mobile blogging resources. <http://www.moblogging.org>.
- [15] R. Mohan, J.R. Smith, and C.S. Li, Adapting multimedia Internet content for Universal Access, IEEE Trans. on Multimedia, Vol. 1, No. 1, pp.104-114, 1999.
- [16] P. Pirolli and S.K. Card, Information foraging, Psychological Review, Vol. 106, No. 4, pp643-675, 1999.
- [17] Universal Description, Discovery, and Integration (UDDI). <http://www.uddi.org/>