CERIAS Tech Report 2005-146

Adaptable Web Browsing of Images in Mobile Computing Environment: Experiments and Observations

by A Kumar, A Bhargava, B Bhargava, S Madria
Center for Education and Research
Information Assurance and Security
Purdue University, West Lafayette, IN 47907-2086

Adaptable Web Browsing of Images in Mobile Computing Environment: Experiments and Observations

Atul Kumar[1], Anjali Bhargava[1], Bharat Bhargava[1], and Sanjay Madria[2]

[1] Department of Computer Science, Purdue University, West Lafayette, IN
[2] Department of Computer Science, University of Missouri-Rolla, MO

Abstract. In this paper, we report some experiments and observations to make browsing of images more adaptable using small devices. We highlight the usability of such an alternative in mobile e-commerce and bandwidth-constrained systems.

1   Introduction

The cellular phones, wearable PCs, personal digital assistants (PDA) being mobile, wireless and small, pose new challenges for browsing. Some issues include low bandwidth availability, limited memory, limited display area, frequent disconnection, fast changing locations etc [4]. In addition there is a problem of redundant information which becomes all the more costly in the context of limited resources. Full quality and high-resolution images may not be needed when the user just wants the stock quote of a company on his cell phone screen. Why burden a user with high quality thumbnail images for choices when the content provider is sure that the user will like to see in more detail and in better quality, the particular items he/she is more interested in?

New features started appearing in the widespread browsers including offline browsing and also some type of filtering. But they are not flexible enough in the sense that they can not download a web page with varying levels of compression on the fly so that the user can select what is suitable in current context nor can they offer choices with multi-resolution display of a page. Content based editing is difficult because of the inherent difficulty in understanding the semantics of the contents. The methodologies in this category offer text-only version or fixed pre-decided contents. An example is a text-based technology called "*Web Clipping*" developed by 3Com for its wireless Palm [16]. *Web Clipping* allows mobile users to download short bursts of text information from Web sites that have tailored content for 3Com's technology but it doesn't allow users to surf the Web at large.

We propose experimental observations [11] that can result in savings in download time and in easing the strain on the network bandwidth. We consider a novel approach to bridge the gap between the time required to download a website in its entirety and the time, the user is ready to wait for that website, by offering information at various levels of quality. These various levels of qualities are made possible by compressing the inline images at different ratios. Our solution is scalable as it can add on many other filtering criteria and can support various modes of compression as well as

selective compression. We didn't emphasize on algorithm of compression as many standards and effective compression techniques [5] are now supported in most of the high level languages. The solution is 'robust and scalable' in the sense that it can service a large number of these small devices in parallel without a loss of generality. It is 'transparent' in the sense that the user is totally unaware of the calculations going at the server side and it appears as if the user is directly connected to the actual web server. The scheme will work effectively with the new media streaming applications such as [4,5,9].

In trying to cut down the download time for the entire web site, we compress all the inline images in it. It warranties trapping all the multiple HTTP requests issued in effect while trying to access a single URL. We decided to enhance the existing applications by adding adaptability features in the browsing software (or more appropriately by changing the way the clients would browse a site) rather than building a complete new one. This decision is based on following pragmatic assertion: most users want to use their traditional browsers to surf even on their portable devices in a fashion similar to their desktop browsers.

2   Background and Relevant Work

Different research groups are developing prototypes to suggest the adaptable features and the tradeoff between the cost and applicability [1,13]. *NetBlitz* [1] is a multiresolution -based system for the WWW. This involves setting up a proxy server housing the URL cache and the multiresolution techniques. It dynamically generates different versions and delivers them to the users. It focuses on multiresolution as the parameter for quality of service. We concentrate on compressing many details and embedded information in fine quality images that can be taken off without any loss of generality of the image but in turn savings in the download time. [7] discusses about techniques to provide dynamic distillation of image and video over the web. Work is going on in the direction of transcoding images and content negotiation amongst the web community. We have not come across browsers that support vast adaptability criterion suitable for hand-held devices. [15] discussed methodologies to adapt multimedia traffic which is basically based on the idea of the application having some knowledge about the underlying network and the network having some knowledge of the kind of application currently running. It tries to fragment packet semantically rather than using the current IP fragmentation scheme. We observe that the above scheme when applied to our technique of compressing the images can help in guaranteeing much better QoS for the application.

2.1   Mobility Issues and Constraints

Providing Internet and WWW services on a wireless data network presents many challenges [2,3,8]. Mass-market, hand-held wireless devices present a more constrained computing environment compared to desktop computers. The wireless data networks present a more constrained communication environment compared to wired networks. Because of fundamental limitations of power, available spectrum, and mobility, wireless data networks tend to have less availability, stability, and

bandwidth as well as greater latency. Mobile networks are growing in complexity and the cost of providing new value-added services to wireless users is increasing.

3   Experimental Set Up

The experimental set up consists of setting our own proxy server. This proxy server ideally should be a powerful workstation servicing many mobile devices in parallel. All these devices here configure their client browser to talk to this server instead of directly connecting to the actual web server. The paradigm is a standard client-server model [1].

The compression engine, CE, is responsible for compressing the in-line images on the fly. The cache is supported in proxy to speed up the computation and performance of our adaptable web browser in the case where the compressed image is already in the cache. The experiment consists of directing the browsers (i.e. clients) to talk to the proxy server, which is waiting infinitely for clients' connections. The server dispatches a thread to handle each request to minimize loss of any client request. Now each request consist of multiple HTTP messages being sent to server. From the HTTP request format (e.g. using the content-type and content-length), the proxy makes decisions whether to compress the image or let it go as it is. The compression ratio in our case is determined by the available bandwidth at the client side. So the users can select the browsing speed at which they want to browse a particular web page. Based on the browsing speed selected, a compression ratio ranging from 0.0 to 1.0 (lower quality image implies more compression or equivalently more compression ratio) is determined by the proxy using JPEG image compression algorithms.

3.1   Experiments and Observations

We conducted an extensive set of experiments on popular websites especially sites related to e-commerce to evaluate our proposed model. The sites were downloaded at different browsing speeds and hence at different image compression levels. Compressing the image results in reduction of the sizes, this finally leads to reduction in time because of transporting smaller files across the network. There have been many other researchers in this direction [1,10]. We computed the savings in download time with different network connection speed. The sample data presented in Table 1 has been collected at two different speeds – 28.8 kbps and 56.6 kbps. Note that this can be easily extended to any other types of connectivity and speed.

The leftmost column represents all the files present on a single web page. The next column represents the uncompressed sizes of the files. Measurements for two different levels of compression were taken. The figure 'COMPR 1' represents maximum compression without losing the sense of images while the next compression level 'COMPR 2' is the compression level which the end user will gladly accept. The idea was to prove that even at the 2nd compression level, there are substantial savings in the download time. So the end user can carry out e-business or any other applications without substantial loss of quality in a much lesser time. The next column shows the download time that each file takes assuming 28.8 kbps connection speed. There are three time measurements under that category. Time 0

represents the download time for uncompressed image. Time 1 is the time for maximally compressed images (hence the smallest time) while Time 2 represents time for lesser compression level, a better quality level. Exactly similar calculations were made out assuming 56.6 kbps connection. Finally the sum totals were calculated.

Table 2 shows a sample data and the corresponding graph depicting the reduction in size with varying quality of image. The image is one of the pictures of the planet Mars posted on the NASA website. The same experiment was carried on large variety of images and on different kinds of web sites. We took in consideration images varying from very detailed fine quality to the ones having courser appearance. Also the sample images were chosen to be representative of largely varying sizes. The resulting graph was basically of the same nature confirming our experiments.

Table 1. Measurements for the embedded images and text files

FILES	SIZE (KB)	COMPR1	SIZE1 (KB)	COMPR2	SIZE2 (KB)	28.8 kbps Time0	Time1	Time2	56.6 kbps Time0	Time1	Time2
00_120x60.gif	5	68%	1.6	56%	2.2	1.39	0.44	0.61	0.71	0.23	0.31
00_AOLSeal.gif	2	50%	1	44%	1.12	0.56	0.28	0.31	0.28	0.14	0.16
00_BBLLogo.gif	5	73%	1.35	66%	1.72	1.39	0.38	0.48	0.71	0.19	0.24
00_ClickTabs2.gif	2	20%	1.6	20%	1.61	0.56	0.45	0.45	0.28	0.23	0.23
00_Desktops.gif	2	55%	0.9	42%	1.17	0.56	0.25	0.32	0.28	0.13	0.16
00_diamond.gif	2	35%	1.3	26%	1.48	0.56	0.36	0.41	0.28	0.18	0.21
00_logo2.gif	5	63%	1.85	56%	2.2	1.39	0.51	0.61	0.71	0.26	0.31
00_Notebooks.gif	2	45%	1.1	31%	1.38	0.56	0.31	0.38	0.28	0.16	0.20
00_pcmall.gif	2	53%	0.94	43%	1.14	0.56	0.26	0.32	0.28	0.13	0.16
00_quote.gif	2	49%	1.02	27%	1.46	0.56	0.28	0.41	0.28	0.14	0.21
00_SecureIcon.gif	2	79%	0.43	74%	0.52	0.56	0.12	0.14	0.28	0.06	0.07
00_tabs.gif	4	39%	2.44	24%	3.04	1.11	0.68	0.84	0.57	0.34	0.43
00_ThisHour.gif	1	21%	0.79	21%	0.79	0.28	0.22	0.22	0.14	0.11	0.11
00_Top10.gif	1	23%	0.77	23%	0.77	0.28	0.21	0.21	0.14	0.11	0.11
00_blank.gif	1	12%	0.88	12%	0.88	0.28	0.24	0.24	0.14	0.12	0.12
00_dss.gif	3	55%	1.35	36%	1.92	0.83	0.38	0.53	0.42	0.19	0.27
00_GoldenGavel.gif	6	47%	3.18	40%	3.6	1.67	0.88	1.00	0.85	0.45	0.51
00_July4thPromo.gif	2	58%	0.84	46%	1.08	0.56	0.23	0.30	0.28	0.12	0.15
00_SummerPromo.gif	3	51%	1.47	33%	2.01	0.83	0.41	0.56	0.42	0.21	0.28
40411.jpg	4	50%	2	16%	3.36	1.11	0.56	0.93	0.57	0.28	0.47
40725.jpg	3	47%	1.59	16%	2.52	0.83	0.44	0.70	0.42	0.22	0.36
47616.jpg	3	62%	1.14	60%	1.2	0.83	0.32	0.33	0.42	0.16	0.17
52228.jpg	8	60%	3.2	58%	3.36	2.22	0.89	0.93	1.13	0.45	0.47
52431.jpg	6	76%	1.44	72%	1.68	1.67	0.40	0.47	0.85	0.20	0.24
53224.jpg	9	58%	3.78	54%	4.14	2.50	1.05	1.15	1.27	0.53	0.59
advert1.gif	2	25%	1.5	7%	1.86	0.56	0.42	0.52	0.28	0.21	0.26
advert2.gif	14	11%	12.5	1%	13.9	3.89	3.46	3.85	1.98	1.76	1.96
index.html	27	0%	27	0%	27	7.50	7.50	7.50	3.82	3.82	3.82
TOTALS	128		78.92		89.07	35.56	21.9	24.74	18.09	11.16	12.59

File Name: UBID Main Page
URL: http://www.ubid.com

Table 3 and 4 shows the effect of trying to compress the original image as well as image conceived when we extrapolate to better quality size. We found out that many images are not put at their best quality but are at reduced quality due to the same underlying assumption as ours i.e. sacrificing quality a bit does not lead to appreciable

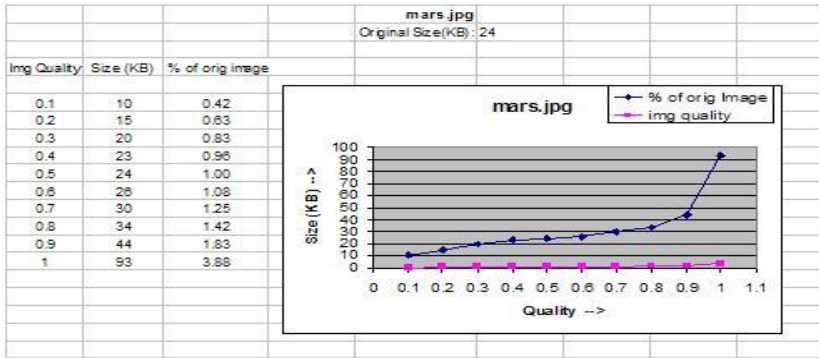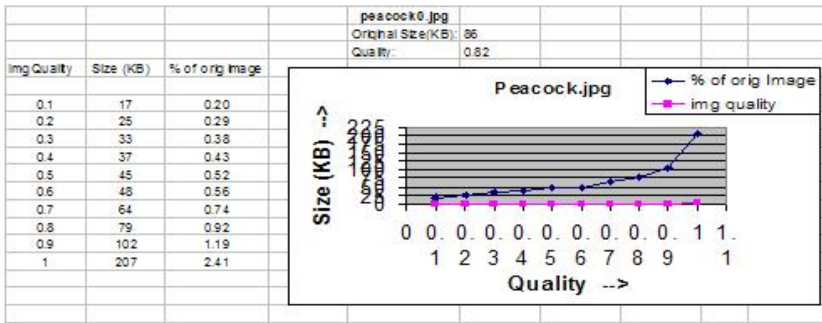**Table 2.** Size of the image as a function of the quality

Img Quality	Size (KB)	% of orig image
0.1	10	0.42
0.2	15	0.63
0.3	20	0.83
0.4	23	0.96
0.5	24	1.00
0.6	26	1.08
0.7	30	1.25
0.8	34	1.42
0.9	44	1.83
1	93	3.88

mars.jpg
Original Size(KB): 24

Table 3. Compression at original size and quality of image

Img Quality	Size (KB)	% of orig image
0.1	17	0.20
0.2	25	0.29
0.3	33	0.38
0.4	37	0.43
0.5	45	0.52
0.6	48	0.56
0.7	64	0.74
0.8	79	0.92
0.9	102	1.19
1	207	2.41

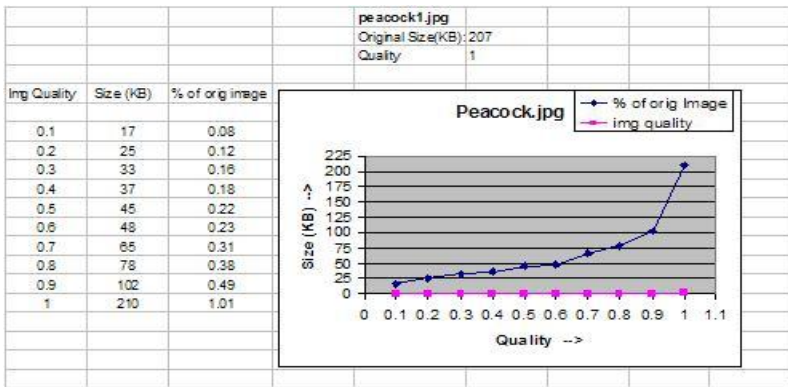peacock0.jpg
Original Size(KB): 86
Quality: 0.82

distortion of the image. The original image (peacock.jpg), for example, here was between quality 0.8 and 0.9. We tried to extrapolate the image to a better quality (and hence larger size) and then apply our techniques on that trying to compress it at different levels. We observe the same trends in the resulting graph. This was repeated for different original starting quality and hence different initial sizes.

We make the following observations from the sample data. Definitely, compressing the images result in savings of download time as well as the amount of storage required to cache the inline images. The obvious drawback is the loss in visual quality of the images [2,3,6,12]. But this is immaterial here, to an extent, because most of the times (especially in e-commerce web sites), the user is more concerned with the contents of the web page rather than the quality of images e.g. the quality of the various icons or the quality of the images that external advertiser embed in the web page. This is more pertinent in cases of mobile devices supporting browsers. A more subtle question is the tradeoff between the overhead associated with the compression and the final savings in download time. We argue on the basis of the table 2, 3, 4 and the corresponding graphs that, we get substantial savings in the size of image with a sacrifice in the quality of image. These savings in turn reflect in the savings in the download time. Compressing every image even to correspond to just 1 lower level of quality can benefit us with an appreciable savings in download time. We justify that

this kind of application is primarily aimed for users doing lots of e-business and also assuming that they have a slower Internet connectivity, even though they have top-of-line fast computers (in terms of processing power). Even with faster network connectivity, the problem envisioned is the rate at which network bandwidth is being hogged. Since the available bandwidth is, more or less, constant but the applications consuming them are burgeoning, it might result that even with a faster link, people have to suffer considerable delay because of network congestion. Hence it makes sense to bear the overhead of image compression. Also with now more and more support being built in programming languages for images, this can be done on the fly. In other words, this adaptability makes sense if the client is being supported on mobile devices or it has a low bandwidth connection to the network. Also it is most effective for sites where we can achieve significance compression without major sacrifice in visual quality. Another adaptable browser based on multi-resolution [8], has been studied and it also deduces similar conclusions based on different set of experiments and setup.

Table 4. Compression at extrapolated size and quality of image

Img Quality	Size (KB)	% of orig image
0.1	17	0.08
0.2	25	0.12
0.3	33	0.16
0.4	37	0.18
0.5	45	0.22
0.6	48	0.23
0.7	65	0.31
0.8	78	0.38
0.9	102	0.49
1	210	1.01

4 Conclusions and Future Research

We reported some experiments and observations for mobile application requirements such as Web browsing. The experimental results show the huge savings in time that can be achieved by compressing high resolution and big inline images. This approach is a step in the direction of making browsing more user-friendly and easing the constraints on system resources. We also discuss the tradeoff associated with it and the favorable factors under which proposed solution works best. For future work, we are using finer criterion for compressing the images. The compressed images should also be effectively cached at the proxy servers to avoid duplication of work in compression of same images. This work can be extended to support all the image types and even the text portion of the pages as they form a major chunk in many of the web-sites. This would result in further savings of the download time.

References

[1] Swarup Acharya, Henry Korth & Viswanath Poosala. Systematic Multiresolution and its Application to the World Wide Web, ICDE, 1999, pp. 40-49.

[2] Bharat Bhargava, Shunghe Li et al. Impact of Codec Schemes on Multimedia Communications, In Proceedings of the IETE International Conference on Multi-media Information Systems (MULTIMEDIA 96)}, Feb., 96, New Delhi, India, Published by McGraw Hill, pp. 94-105..

[3] Bharat Bhargava, Shunghe Li et al. Performance Studies for an Adaptive Video Conferencing System. In Proceedings of the IETE International Conference on Multi-media Information Systems MULTIMEDIA 96)}, Feb., 96, New Delhi, India, Published by McGraw Hill, pp 106-116.

[4] Bharat K. Bhargava, Changgui Shi, Sheng-Yih Wang: MPEG Video Encryption Algorithms. Multimedia Tools Appl. 24(1): 57-79 (2004).

[5] Bharat K. Bhargava, Changgui Shi, Sheng-Yih Wang: MPEG Video Encryption Algorithms. Multimedia Tools Appl. 24(1): 57-79 (2004)

[6] Ron Frederick. Experiences with Real-time Software Video Compression. In proceedings of the packet video workshop, Portland, Oregon, 1994.

[7] A. Fox, S. D. Gribble, E. A. Brewer, and E. Amir. Adapting to Network and Client Variability via On-demand Dynamic Distillation. ASPLOS Proceedings, Oct. 1996.

[8] Martin Gaedke, Michael Beigl, Hans-Werner Gellersen, Christian Segor: Web Content Delivery to Heterogeneous Mobile Platforms. ER Workshops 1998: 205-217

[9] Mohamed Hefeeda, Bharat K. Bhargava, David K. Y. Yau: A hybrid architecture for cost-effective on-demand media streaming. Computer Networks 44(3): 353-382 (2004)

[10] Van Jacobsen. Congestion Avoidance and Control. In proceedings of Symposium on Communications Architectures and Protocols (SIGCOMM'88)

[11] Atul Kumar, Adaptable Web Browsing, Project Report, Purdue University, 2000.

[12] Overview of the MPEG-4 Standard. International Organization for Standardization for Coding of Moving Pictures and Audio, March 1999.

[13] Standards for Mixed Media Communications. Lucent Technologies, June 1997.

[14] Sanjay Madria, Mukesh Mohania, Sourav Bhowmick, Bharat Bhargava. Mobile Data and Transaction Management, Information Science Journal, 2002.

[15] Sheng-Yih Wang, Bharat Bhargava. An adaptable network architecture for multimedia traffic management and control, IEEE International Conference on Multimedia and Expo (III) 2000: 1615-1618.

[16] Web Clipping. URL: http://cnn.com/TECH/computing