



WELCOME TO THE GATHERING PLACE

High resolution colorimetric image browsing on the Web

Kirk Martinez^{a,*}, John Cupitt^b, Steve Perry^a

^a *Multimedia Research Group, Department of Electronics and Computer Science, University of Southampton, Southampton SO17 1BJ, UK*

^b *The National Gallery, Trafalgar Square, London, UK*

Abstract

A system is being developed to allow WWW browsing of very high resolution (up to 20,000 by 20,000 pixels), colorimetric images. These come from the new generation of image acquisition systems in art galleries — the images are equivalent in resolution to a large-format transparency, but have accurate and consistent colour. Because colour is consistent, images from different sites may be compared on screen: differences in appearance between the images are guaranteed to be due to real differences between the objects, a very valuable property for art and conservation applications. The images are held as tiled pyramidal JPEG TIFF file at a server, and are transmitted via standard HTTP requests to a Java client. Only the portions of the image needed are requested and transmitted.

The research is part of the Euro-Canadian Viseum project on making secure image archive access to museums. It has set up an ATM network from Vancouver to Berlin, Paris (the Louvre) and London (The National Gallery). © 1998 Published by Elsevier Science B.V. All rights reserved.

Keywords: Java; High resolution images; Colorimetry

1. Introduction

Previous work on high resolution cameras [1,2] has led to the availability of very accurate images of fine art. These images can be up to 20,000 by 20,000 pels, or around 1.6 Gbytes each (uncompressed) from the MARC camera [3], and have higher colour accuracy than conventional images or film (an average colour error as low as 1 delta E Lab unit [4]). Because of their accuracy, these images have found applications in archiving, printing, multimedia development and museum conservation [5]. Conservation applications include studies of colour changes, damage and simulated restoration. The colorimetric cali-

bration also allows images to be compared properly, even between sites.

The Viseum project [6], due to finish in March 1998, aims to allow network access to these images. There are three key components: a small, easy-to-use colorimetric network image viewer (an X11 GUI has been used for some time [7] inside the National Gallery but this is not suitable for general Intranet/Internet use), a central indexing system to allow searches for images across all Viseum sites, and a security and billing server to control access. Access control is important, since the sale of images of this quality to publishers currently generates considerable income for galleries.

Each site in the project has a CD-ROM jukebox (from NSM [8]), a database, and a Web server. Sites

* Corresponding author.

in London, Paris, Berlin and Vancouver are connected to each other and to the central index and security servers via an ATM test-bed TCP/IP network. This provides a private channel of up to 6Mbit/s which will be used to tune the client–server design.

This paper will concentrate on the high resolution image server and client.

1.1. Related work

Kodak, Hewlett Packard and others have developed the FlashPix/OpenPix [9] storage format and the Internet imaging protocol, IIP, which will probably become a defacto standard in the future. The FlashPix image format is a pyramidal scheme very similar to ours. It is ideal for multiresolution browsing and use of SRGB [10] allows calibration of the colour, although no IIP clients currently support this.

SRGB does not allow the representation of colours outside the standard monitor gamut, a problem for many paintings. IIP provides a set of protocols for querying the image server and obtaining images, possibly processed at the server. At the moment a server exists for NT and clients can be based on ActiveX or Java, and we have made rather different design choices:

- We are serving images from a jukebox, so we do not expect many simultaneous readers. We have therefore put much more load on the server than IIP, and made the client simpler. We started work on our system before IIP began, and back then it was not clear that Java would be fast enough to allow much client-side processing.
- We have put much more emphasis on colorimetry and security than the current IIP implementations.
- Our image format is based on TIFF rather than FlashPix, and the images are therefore readable by standard packages (provided they support JPEG-in-TIFF and tiling).
- Our system is available free, with full source code.

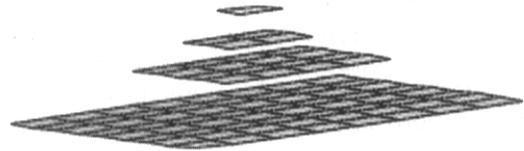
2. The Viseum TIFF file format

Previously our large images were stored in uncompressed form in a conventional raster format file. This is not efficient for region of interest browsing and also needs compression to reduce network

bandwidth. An efficient file format was needed to reduce transmission time, disk seeking and zoom/pan processing.

Instead, we are using the freely-available **IJG JPEG**¹ library, combined with a standard libtiff package. Tiled JPEG-in-TIFF is already a standard file format: we just add extra sub-images for the layers in the pyramid, and save pixels as LAB rather than RGB. The resulting images can be read by many standard TIFF viewers (the viewers which come on SGI machines, for example), although unfortunately not with current versions of Photoshop, since it does not yet support JPEG TIFF.

Low compression factors (less than 10:1) provide visually lossless images suitable for browsing. Here we typically use a compression of around 6 times. Each tile is decodable singly and is ideal for network transmission. There is no inter-resolution compression as in Photo-CD, as this makes decoding much slower. However this means that to boost performance our format stores a third as much information as a compressed large file.



Representation of pyramidal image.

Each pel is an 8:8:8 bit Lab value. Images from the National Gallery are originally 10:11:11 bits for a fine quantisation of colour space but 8 bits per channel is sufficient for display purposes. The final TIFF file contains all the levels of the pyramid, stored sequentially.

For comparison the SCOPYR format [11] used in the Louvre has a fixed, large tile size of 800 × 600 and stores each tile in a separate file with a three letter name extension related to the tile position.

3. The server

Each site has a database of its collection, with a Web interface. One of the database fields records

¹ <ftp://ftp.uu.net/graphics/jpeg>

the volume name and file name of the high-resolution image of the painting: pages generated by the database in response to queries contain links with the appropriate embedded arguments to fire up a Java

applet linked to the correct image on the jukebox (see, e.g., Figs. 1 and 2).

The image server is linked to a standard Web server with FastCGI. A typical request might look like:

```
http://machine/getimage.fcgi?file=pic.tif&session=EKJHFEF&
x=0&y=0&w=128&h=128&sub=1&ht=2&vt=2
```

where arguments have the following meanings:

file

The name of the file from which tiles should be fetched.

session

The session cookie for this client. The cookie is used for authentication, and to retrieve the user's preferences, such as their display type.

x, y, w, h

The size and position of the tile the client needs. This does not have to match the tile sizes and boundaries used in the actual image.

sub

The amount of sub-sampling to apply to the original image before transmission. This has to be a power of two.

ht, vt

Horizontal and vertical repeat. The server repeats the *x, y, w, h*.

Various alternatives were investigated: initially the server decompressed the JPEG to Lab and used the client's colour profile (ICC) to send an RGB JPEG image back to the client. This clearly puts a heavy load onto the server, but was simple to implement and allowed slow clients to be used. Another version of the server simply reads tiles and transmits them to the client, which then does all of the colour processing.

As each tile request is a separate CGI call to the server, FastCGI is used for efficiency and to maintain state information, particularly a cache. Recently requested tiles are cached, which is especially useful if the server does all of the processing. The CGI is written in C, as is the VIPS [7] image processing library used for any colour processing. This has the added advantage of automatically using parallel CPUs on multiprocessor systems.

The CD Jukebox from NSM stores 150 CDs and presents them as directories under Unix. It has four 12× speed CD drives and its driver uses a hard disk

cache. The CD swap time is only 4 seconds, but this combined with the 4 drives means the hard disk cache is important for serving more than 4 clients. Eventually a large cache would build up the typical areas of popular images such as faces, hands etc. If the complete National Gallery collection was stored with an average file size of 100 MBytes, the approximately 2000 works would fill over 300 CDs and two jukeboxes. In contrast the Louvre scans at lower resolution but more images per work and has already filled two jukeboxes. In the next two years the jukeboxes will be upgraded to DVD media (3.6 GB) and one jukebox would store the National Gallery collection. DVD would also allow uncompressed images to be stored, which can be too large to fit onto one CD-ROM.

4. The client

It was decided that a Java [12] client would offer the most machine-independent software, although initially there were doubts about its performance handling images. A test client was also written in C for comparison. The client displays an image in a scrolling area by requesting only those tiles it needs. A local cache prevents retransmission of previously visited tiles and the size of this determines browser performance in typical situations. Using the host browser's cache (e.g. Netscape) would flood it so each tile has a `pragma nocache`. Initially the **HotJava**² browser was used due to the problems of running Java 1.1 under standard browsers. However the recent Activator plug-in from JavaSoft has made it possible to use standard browsers.

An existing monitor profiling system made by Colorific [13] is used to generate the ICC for the user's monitor. This is a Web site which provides

²<http://java.sun.com/products/hotjava/1.1/>

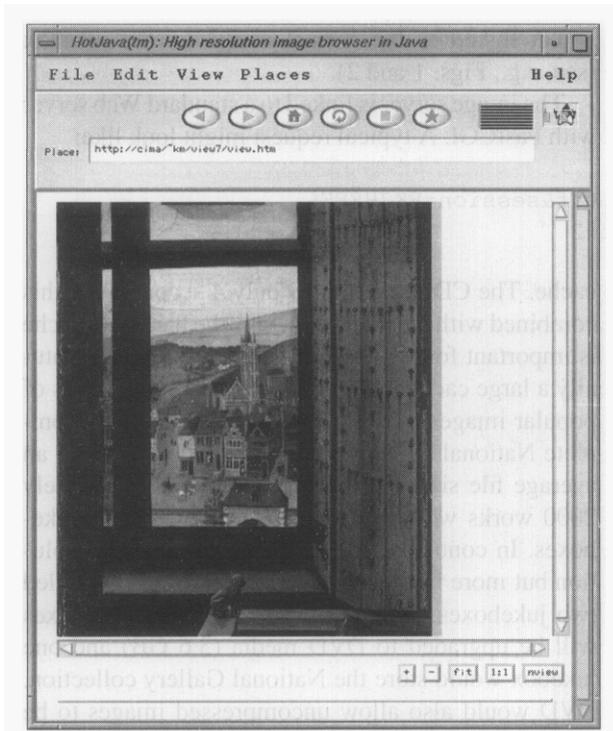


Fig. 1. Applet viewer in HotJava. Full image, from the National Gallery in London, to Southampton.

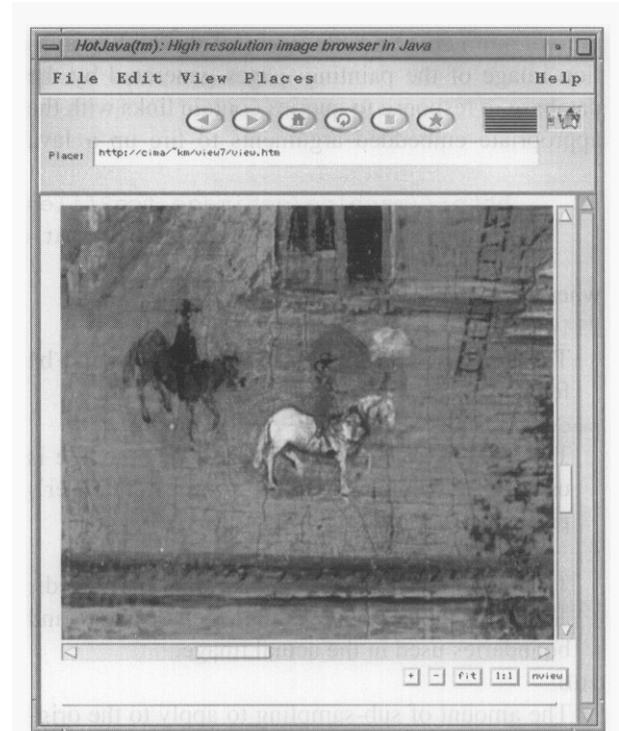


Fig. 2. Browsing highest resolution image.

a procedure of selecting colour patches on-screen and using a small blue, plastic test-pattern as a mask. The resulting colour profile (ICC [14]) is used to convert from colour spaces to display RGB. More accurate techniques require a measuring instrument, which most users would not have. Results across different monitors are promising.

We have put quite a lot of effort into tuning the display parts of the client: image display is fully threaded, so the interface stays "live", even during heavy network access; and is interruptable, so it is not possible to build up a large queue of pending requests. The just-in-time compilers for Java seem to be fast enough to do image handling at the client, with a short but noticeable slow-down on start-up.

The overall client-server communication can be summarized as in Table 1.

This is for heavy server-side processing. When the client does the colour processing the sequence looks like in Table 2.

A future step will be to move the JPEG decompression to the client, which was not done because decompressing Lab JPEG is not straightforward un-

Table 1

Client	Server
request image info	get image info and send
request area of image	read tiles decompress JPEG Lab to RGB compose into one image if needed compress to JPEG send data
	decompress JPEG and display

Table 2

Client	Server
request image info	get image info and send
request area of image	read tile decompress JPEG send Lab data uncompressed
Lab to RGB display	

der Java as the standard image loader can not be used. This is one advantage of using sRGB, where normal JPEG decompression works and the conver-

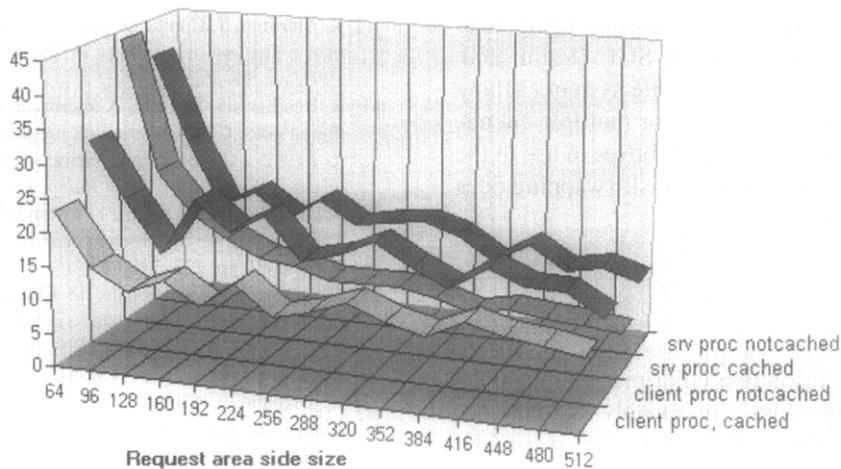


Fig. 3. Time to repaint a 1500×1500 screen in seconds vs request image size, for server processing and mainly client processing colour. 100 Mbit/s network. 128×128 tiled image served from SGI Origin 200. displayed on SUN³ UltraSPARC.

sion to display RGB is fairly simple. This may be included in a future system.

5. Trials/results

Clearly the overall system performance depends on many factors: tile size, network bandwidth, CD-ROM speeds, colour conversion technique etc. The throughput available from the image server could be measured using a C client which simply made requests and discarded incoming data. This showed that the servers were capable of providing many tiles per second and should not be a major bottleneck. An SGI Origin 200⁴ can serve around 12 tiles (128×128) per second, 7 per second with full colour processing. A fast hard disk (we use the Seagate Cheetah⁵) makes a difference due to the seeking involved. Over a LAN the client screen area refreshes in around 3.5 seconds for new areas, 2.5 seconds for server-cached areas and instantly for client-cached tiles (on a Pentium II client). This suggests that the system is client limited.

The size of the requests made by the client affect throughput because of the overhead of setting up

HTTP requests for more, smaller tiles for example. Requests which are larger than a file tile require processing at the server to compose them into whole images, but in practice this is so fast as to be unnoticeable. However large tiles mean scrolling has long infrequent delays rather than short frequent delays.

The graph in Fig. 3 shows the effect of requesting various sizes of image from the server. The area repainted is equivalent to around 9 normal screens so that the times were more easily measured. When the server does the colour processing (srv proc), requesting images smaller than the file's 128×128 tiles leads to inefficiency and slow repaints. Larger requests seem to have lower overheads and hence faster repaints. The line for cached is where the server probably has the tiles in cache already, so it does little processing. With a fast client which simply has to de-JPEG this is the fastest set-up. The blue lines are for colour processing on the Java client and show that requesting the file's tile size of 128 is very efficient and that Java processing leads to little slow-down. The Java colour processing is a faster LUT-based operation, unlike the server-side arithmetic computation, so this may balance the slower speed of Java code.

With low bandwidth or busy network connections the bottleneck is clear and the time spent computing in the browser becomes less important. In contrast tests were carried out over a 100 Mbit/s LAN where the client performance becomes critical.

³ <http://www.sun.com>

⁴ <http://www.sgi.com/Products/hardware/servers/products/Origin200.html>

⁵ <http://www.seagate.com/disc/cheetah/cheetah.shtml>

Tests were carried out with images being read from a 12X CD drive on an SGI Origin 200. This provided fast browsing and file system caching helped keep response times low for multiple clients. This suggests that the cached Jukeboxes in use in the project will cope well as long as disk-swapping does not become an issue.

6. Conclusions

Browsing high resolution colorimetric images on the Web is possible with a reasonably fast network link (from say 1Mbit/s). Using the tiled pyramidal approach provides the user with good multi-resolution viewing with a reasonably low load on the image server. IIP undoubtedly offers a more standard image server communication protocol so our software will migrate towards towards that protocol. This will mean that OpenPix browsers will be able to look at our image server, provided security issues are solved. TIFF has been found to be a flexible enough format to avoid the use of our own VIPS format or a new FlashPix one. Future work will give the interface of the Java code more features, the server a more intelligent cache, possibly with look-ahead to predict where the user is going.

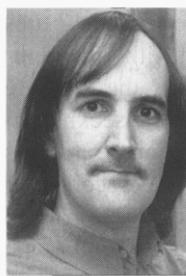
Acknowledgements

Viseum is part funded by the European Community ACTS scheme. Newbridge Networks (Canada and Europe) loaned ATM equipment. TeleGlobe Canada provided the transatlantic link. The other partners in Viseum are: LRMF, NSM Jukebox, ACT (Vancouver), Deutsche Telekom Berkomm GmbH (Berlin). Thanks to all the group for their help and contributions.

References

- [1] J. Cupitt, K. Martinez, and D. Saunders, A methodology for art reproduction in colour: the MARC project, *Computers and the History of Art Journal*, 6(2): 1–20, 1996.

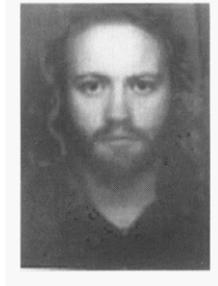
- [2] K. Martinez, J. Cupitt, D. Saunders, High resolution colorimetric imaging of paintings, in: *Proc. SPIE*, Vol. 1901, Jan. 1993, pp. 25–36.
- [3] R. Beuthauser, R. Lenz, A microscan/macroscan 3×12 bit digital color CCD camera with programmable resolution up to 20992×20480 picture elements, in: *Proc. of the Commission V Symposium Close Range Techniques and Machine Vision*, Melbourne, Austr., 1–4 March 1994. *Intl. Archive of Photogrammetry and Remote Sensing*, Vol. 30, Part 5, ISSN 0256-1840, pp. 225–230.
- [4] Commission Internationale de l'Eclairage, Recommendations on uniform colour spaces, colour difference equations, psychometric colour terms, Supp. No. 2 to CIE Publication No. 15 (E-2.3.1), 1971 (TC-1.3), 1978.
- [5] H. Chahine, J. Cupitt, D. Saunders and K. Martinez, Investigation and modelling of colour change in paintings during conservation treatment, in: *Proc. of Imaging the Past*, British Museum Occasional Paper No. 114, London 1996.
- [6] Viseum project: <http://www.InfoWin.org/ACTS/RUS/PROJECTS/ac238.htm> and <http://www.ecs.soton.ac.uk/~km/proj/s/viseum>
- [7] J. Cupitt and K. Martinez, VIPS: an image processing system for large images, in: *Proc. SPIE Conference on Imaging Science and Technology*, San Jose, Vol. 1663, 1996, pp. 19–28; also: <http://www.ecs.soton.ac.uk/~km/vips/>
- [8] NSM, <http://www.nsmjukebox.com>
- [9] FlashPix, <http://www.kodak.com/drgHome/productsTechnologies/FPX.shtml>
- [10] sRGB, <http://www.color.org/contrib/sRGB.html>
- [11] SCOPYR/JTIP, <http://www.argyro.net/demo/avelem/jtip.htm>
- [12] J. Gosling, B. Joy, G. Steele, *The Java Language Specification*, Addison-Wesley, Reading, MA, ISBN 0-201-63451-1, August 1996, http://www.javasoft.com/doc/language_specification.html
- [13] Colorific, <http://www.colorific.com>
- [14] International Colour Consortium, <http://www.color.org/>



Kirk Martinez gained a B.Sc. in Physics from the University of Reading and a Ph.D. in Image Processing at the University of Essex. He has worked on European projects such as VASARI, MARC and Viseum. His research interests include high quality colorimetric imaging, parallel image processing, multimedia databases. He is also Director of the Digital Libraries Research Centre at Southampton University.



John Cupitt. Since completing his Ph.D. in Theoretical Computer Science at the University of Kent, he has worked in the Scientific Department of the National Gallery London on the European Community-funded VASARI, MARC and VISEUM projects. He has published papers on camera calibration, image processing I/O systems, user-interface design, the measurement of colour change in paintings and infrared imaging of paintings.



Stephen Perry gained a B.Sc. in Computer Science from the University of Southampton in 1994, before beginning work on interactive segmentation and content based retrieval in the Multimedia Research Group. He has since worked on the VISEUM project developing advanced Internet applications for museums, and is currently part of the ACOHIR project, working on tools to acquire images from 3D objects and software to view

them as high resolution images with accurate colour.