Refinement by Browsing: an Alternative to Thumbnails *

Joaquim Gabarró[1] and Isabel Vallejo[1] and Fatos Xhafa[1]
and Alan Stewart[2] and Maurice Clint[2]

[1] Dep. LSI, Universitat Politècnica de Catalunya,
Jordi Girona, 1-3, Barcelona 08034, Spain.
[2] School of Computer Science, The Queen's University of Belfast,
Belfast BT7 1NN, Northern Ireland.

Abstract. Thumbnails are widely used in image retrieval and browsing systems. The use of such images compacts information. However, in situations where there are a large number of images on a server site, the use of thumbnails may still result in unacceptable *response times* when a user has access to a limited bandwidth. In this paper a refinement technique is proposed as an alternative to using thumbnails. The technique allows a user to identify the contents of an image early in its transmission. An efficient means of constructing a chain of image refinements of increasing quality to a given image is presented. Approximation chains can be employed either statically or dynamically. In the first case the level of detail is driven by the user; in the second case the level of detail is driven by a bandwidth aware server. Thus the proposed approach has application in *grid and web* computing.

Keywords. *URL, thumbnails, image refinement.*

1 Introduction

It is commonplace for web sites to contain large numbers of images. One disadvantage associated with web sites with multiple images is navigation time. The *response time* – the elapsed time between the onset of a stimulus and a response to that stimulus – is critical in certain applications such as browsing image data bases. The use of *Thumbnails* is a widely used means for improving response times. Thumbnails are images of smaller size and poorer quality and they are often generated by *shrinking* the original images. Normally, the original image can be restored by clicking on its thumbnail.

One disadvantage associated with thumbnails is image quality. This problem is particularily accute for very small thumbnails. In practice lack of resolution may result in an unwanted image being downloaded in its entirety. It is desirable to have a system which combines good response time with good image approximation features. Thus, if a thumbnail can be recognised as being of no interest

* Work partially supported by Spanish CICYT under grant TIC2002-04498-C05-03 (Tracer) and IST program of the EU under contract IST-2001-33116 (Flags) and CoreGRID Network of Excellence.

a user can abort his search immediately and jump to another image or page. Essentially thumbnails do not always provide a useful means of recognising image content early during transmission. In certain situations – for example, browsing large image databases with a limited bandwidth – thumbnails may not reduce *response time* to an acceptable level.

In this paper a *refinement* technique is proposed as an alternative to thumbnails. In §2 a review of common problems associated with image transmission on low bandwidth connections is given. In §3 the use of image approximation chains of increasing quality is proposed. In §4 the use of approximate images by a bandwidth aware server is considered. We briefly outline future work on section §5 and give some conclusions in §6.

2 Bandwidth and image transmission

L. Cardelli and R. Davis [1] have proposed the *Berners–Lee Computer* as a model of Web computing. The main idea is that an URL is "a pointer with a bandwidth" – bandwidth being a crucial parameter. For example, the following table [9, 8] gives the population rates for various kinds of Spanish bandwidth connections:

Connection type	spanish population rate	Bandwidth
56k Modem	75,46%	56 kbps
ISDN	3,74%	64 kbps
DSL	11,84%	2mbps

Here the majority of connections are low bandwidth – such connections are often terminated before a complete response is available because of the download time.

Problem 1 (Maximum response time for images). It is often the case that the number of bytes used to represent an image dominates the available bandwidth capability. In situations where the transmission time is greater than 10 seconds [10] a user may have a strong psychological impression of unacceptable behavior.

The maximal image sizes, for each of the networks above, which avoid download times in excess of 10 seconds are:

Connection type	Maximal image size
56k Modem	34 kb
ISDN	150 kb
DSL	2 mb

One approach to resolving problems associated with browsing images with a low bandwidth connection is using image compression techniques to derive thumbnails. A *thumbnail* is a miniature display of a image or page. Conventionally thumbnails are used to *enhance image browsing.* Clicking on a thumbnail usually results in the the complete image being displayed on the screen [2]. There

are several ways to build thumbnails – see [3] for details. Alternatively, a theory of approximations applicable to web and grid computing has been developed – see [5, 6]. A series of approximations to an image of the Pisa tower is given below in order to motivate the discussion.

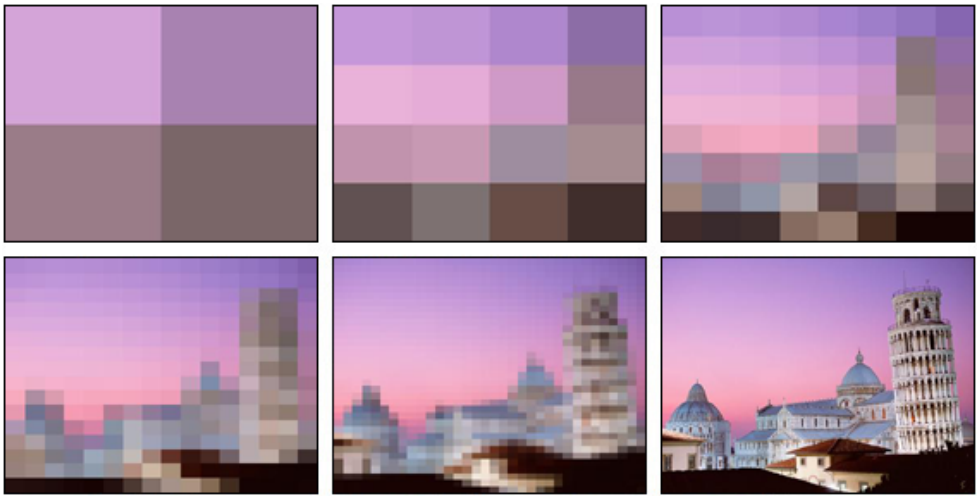


Fig. 1. Six increasing quality approximations of the Pisa tower.

In particular a thumbnail compaction of an image *img* may be considered to be an "initial approximation" – in this sense the thumbnail is viewed as bottom, $\perp_{img}$, in a partial order [4] of approximations. As $\perp_{img}$ is represented by a small number of bytes it can be transmitted rapidly. However, the problem of having acceptable response times may reoccur if a user requests a *complete* version of *img*. Here we propose replacing the conventional two-level partial order (thumbnail and complete image) with a more elaborate partial order of approximate images. For example, one possible partial order of image approximations is illustrated in Figure 1.
Thus, the sequence of approximate images in Figure 1 forms a *chain* of approximations to the image of the Pisa tower – here images are ordered according to image resolution. The use of partial orders of approximations allows for the possibility of a bandwidth aware server selecting an appropriate level of resolution for the capacity of a user process.

3 Mesh partial orders and image refinements

The usefulness of many image compaction techniques can be measured by *contour completion* rates. Poor compaction techniques may result in badly broken image contours. However, optimatisation techniques may be employed during

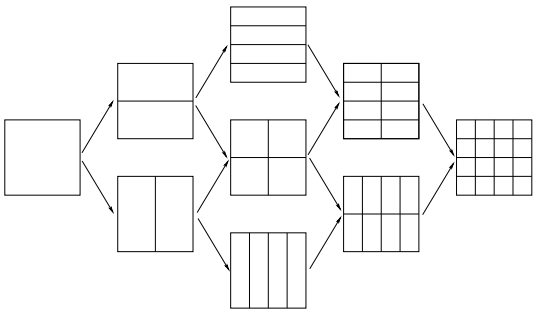


Fig. 2. The partial order $Mesh_{4,4}$.

compaction in order to enhance contour completion. In one sense optimisation algorithms for maximising contour completion can be related to the generation of a chain of improving image approximations. Here the notion of improving image approximations is captured through partial orders [4]. Consider a sequence of image approximations based on uniform block partitions. Such partitions form the basis of the proposed image refinement. Consider an image of dimensions $w \times h$. A partial order of approximations to this image can be derived; a component of the partial order, (sX, sY), denotes an approximate image generated from partitioning the original image into $\frac{w}{sX} \times \frac{h}{sY}$ blocks, each of size $sX \times sY$. The set of elements in the order (i.e. the set of all possible uniform rectangular compactions) are:

$$Mesh_{w,h} = \{(sX, sY) \mid h \mod sX = 0\ ,\ w \mod sY = 0\}.$$

Elements are ordered according to the relation $\sqsubseteq$:

$$(sX, sY) \sqsubset (sX,' sY') \quad =_{def} \quad \exists\, x, y : 0 < x, y : sX = x \cdot sX'\ ,\ sY = y \cdot sY'.$$

Thus, if $(sX, sY) \sqsubset (sX,' sY')$ then blocks of the mesh (sX, sY) partition (regularly) into blocks of the mesh (sX', sY'). Figure 2 gives the Hasse diagram of $(Mesh_{4,4}, \sqsubset)$. In general, bottom of $(Mesh_{w,h}, \sqsubseteq)$, is $\bot_{w,h} = (w, h)$ and top is $\top_{w,h} = (1, 1)$. The operations join and meet satisfy, respectively:

$$\begin{aligned}(sX, sY) \sqcup (sX', sY') &= (\gcd(sX, sX'), \gcd(sY, sY')) \\ (sX, sY) \sqcap (sX', sY') &= (\mathrm{lcm}(sX, sX'), \mathrm{lcm}(sY, sY')).\end{aligned}$$

Moreover $(Mesh_{w,h}, \sqcup, \sqcap)$ is a distributive lattice. However, it is not, in general, a boolean algebra.

The use of rectangular meshes can be employed in an image refinement system. For example, an image approximation implementation, `ImageApp`, has been developed based on the use of uniform rectangular meshes. The program can be downloaded from [7] the following URL:

`http://www.lsi.upc.es/~gabarro/RefinementByBrowsing`

In particular, the program contains a method, `getImage()`, which reads an image and stores it, using RGBA encoding, into an array of integers, `pix` – `pix` represents an image using row-wise storage. `pix` can be compacted by dividing it into `sizeX`×`sizeY` blocks (as described above). A method `getAverageImage()` can be invoked to store an image refinement into an `sizeX` × `sizeY` array, `pixSolution`. The mesh partial order above can be induced onto the set of image refinements:

$$Pixels_{w,h} = \{\texttt{pixSolution}(sX, sY) \mid (sX, sY) \in Mesh_{w,h}\},$$
$$\texttt{pixSolution}(sX, sY) \sqsubset \texttt{pixSolution}(sX', sY') \text{ iff } (sX, sY) \sqsubset (sX', sY').$$

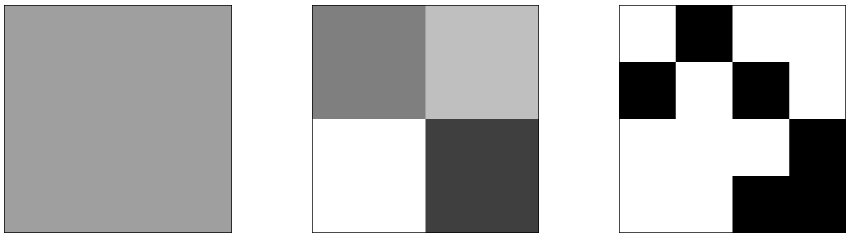
The mesh $\bot_{w,h}$ induces the least image approximations while the mesh $\top_{w,h}$ induces the top approximation:

$$\bot_{\texttt{Pixels}_{w,h}} = \texttt{pixSolution}(\bot_{w,h})$$
$$\top_{\texttt{Pixels}_{w,h}} = \texttt{pixSolution}(\top_{w,h}) = \texttt{pix}.$$

The least upper bound of two image approximations, generated using meshes $m = (sX, sY)$ and $m' = (sX', sY')$, can be found by compacting the original image using a mesh of size $m \sqcup m'$:

$$\texttt{pixSolution}(sX, sY) \sqcup \texttt{pixSolution}(sX', sY') = \texttt{pixSolution}((sX, sX') \sqcup (sY, sY'))$$

A similar result holds for $\sqcap$.



(a) $image(4,4)$. (b) $image(2,2)$. (c) $image(1,1)$.

Fig. 3. Three increasing approximations for a 4×4 pixel image img: $\bot_{img} = image(4,4) \sqsubset image(2,2) \sqsubset image(1,1) = img$.

Example 2. Consider a 4×4 pixel image, img. Figure 3 shows a possible chain of approximations to img. Suppose that the original image is:

$$\text{pix} = [\text{w}, \text{b}, \text{w}, \text{w}, \text{b}, \text{w}, \text{b}, \text{w}, \text{w}, \text{w}, \text{w}, \text{b}, \text{w}, \text{w}, \text{b}, \text{b}]$$

where, using the RGBA coding, b = black = $(0,0,0,0)$ and w = white = $(255,255,255,0)$. Corresponding meshes are ordered according to: $\bot_{4,3} = (4,4) \sqsubset (2,2) \sqsubset (1,1) = \top_{4,4}$. Hence,

$$\texttt{pixSolution(4,4)} \sqsubset \texttt{pixSolution(2,2)} \sqsubset \texttt{pixSolution(1,1)}$$

where

$$\texttt{pixSolution}(4,4) = \left[\frac{3}{8}\mathrm{w}\right] = \bot_{\texttt{pix}} \ , \ \texttt{pixSolution(2,2)} = \left[\frac{1}{2}\mathrm{w}, \frac{3}{4}\mathrm{w}, \mathrm{b}, \frac{1}{4}\mathrm{w}\right]$$
$$\texttt{pixSolution(1,1)} = [\mathrm{w}, \mathrm{b}, \mathrm{w}, \mathrm{w}, \mathrm{b}, \mathrm{w}, \mathrm{b}, \mathrm{w}, \mathrm{w}, \mathrm{w}, \mathrm{w}, \mathrm{b}, \mathrm{w}, \mathrm{w}, \mathrm{b}, \mathrm{b}] = \mathrm{pix}$$

□

4 Refinement by browsing: a first approach

The ideas in the previous section can be extended to cater for situations in which a user may browse through approximations to an image until an appropriate level of detail is revealed (within the confines of the available bandwidth). An easy web program contains (i) picture information – for instance, a text and an image – and (ii) a pointer (an `href`) to another easy program – essentially the pointer can be used to uncover a better image approximation. An *esay* `html` *program* is defined to be $\texttt{EasyProgram.html} = [D, \texttt{href}]$ where $D \sqsubseteq first.\texttt{*href}$. Here `*href` denotes the `html` program pointed by `href`, the required image and accompanying text is represented by an element of the partial order D and $first$ selects the first component of a pair. A one step browsing transition for `EasyProgram.html` is $\texttt{EasyProgram.html} \longrightarrow \texttt{*href}$. Informally, a one step browsing transition improves the available image approximation while leaving the possibility of further refinement open. Easy programs may be ordered according to:

$$\texttt{EasyProgram.html} \sqsubseteq \texttt{EasyProgram'.html} =_{def} D \sqsubseteq D' \text{ and } \texttt{*href} \sqsubseteq \texttt{*href'}$$

The refinement is strict when $D \sqsubset D'$ or $\texttt{*href} \sqsubset \texttt{*href'}$. Note that the definition above is the conventional way of ordering the Cartesian product of two underlying partial orders. One consequence of refinement in this ordering is that an image can be left unchanged while the pointer domain is refined by selecting images higher up in the partial order (i.e. information pertinent to the generation of approximations may be disgarded by jumping further up the pointer chain). Alternatively, in order to avoid the possibility above, the definition of refinement can be strengthened by conjoining the clause

$$\texttt{EasyProgram.html} \longrightarrow^* \texttt{EasyProgram'.html}$$

to the previous definition of refinement. Here $\longrightarrow^*$ denotes zero or more transition steps. Thus, progress to a better image is also accompanied by the preservation of the remainder of the pointer chain.

Example 3 (Refinement by browsing). Let $\texttt{Pisa}_i\texttt{.html}$ be a sequence of easy programs, $1 \leq i \leq 9$, each one containing an image $\texttt{pisa}_i\texttt{.jpg}$ and a comment $\texttt{pisa}_i\texttt{.txt}$ such that [7]:

$$\bot_{\texttt{pisa.jpg}} = \texttt{pisa}_1\texttt{.jpg} \sqsubset \texttt{pisa}_2\texttt{.jpg} \sqsubset \cdots \sqsubset \texttt{pisa}_9\texttt{.jpg} = \top_{\texttt{pisa.jpg}}$$
$$\bot_{\texttt{pisa.txt}} = \texttt{pisa}_1\texttt{.txt} = \cdots = \texttt{pisa}_8\texttt{.txt} \sqsubset \texttt{pisa}_9\texttt{.txt} = \top_{\texttt{pisa.txt}}$$

Further, let $\texttt{Pisa}_i\texttt{.html}$ point to $\texttt{Pisa}_{i+1}\texttt{.html}$,$1 \leq i < 9$, and $\texttt{Pisa}_9\texttt{.html}$ point to itself. Then:

$$\texttt{Pisa}_1\texttt{.html} \longrightarrow \cdots \longrightarrow \texttt{Pisa}_9\texttt{.html} \longrightarrow \texttt{Pisa}_9\texttt{.html} \longrightarrow \cdots$$

Then $\texttt{Pisa}_i\texttt{.html} \sqsubset \texttt{Pisa}_i + 1\texttt{.html}$, $1 \leq i < 9$, and $\texttt{Pisa}_9\texttt{.html} \sqsubseteq \texttt{Pisa}_9\texttt{.html}$ □

Example 4 (One step Refinement by Thumbnail). Conventional tumbnails can be viewed as two level approximations:

$$\begin{aligned} \texttt{Pisa}_\bot\texttt{.html} &= [\bot_{\texttt{pisa.jpg}}\ ,\ \bot_{\texttt{pisa.txt}}\ ,\ \text{href}_\bot] \quad \text{with} \quad *\text{href}_\bot = \texttt{Pisa}_\top\texttt{.html} \\ \texttt{Pisa}_\top\texttt{.html} &= [\top_{\texttt{pisa.jpg}}\ ,\ \top_{\texttt{pisa.txt}}\ ,\ \text{href}_\top] \quad \text{with} \quad *\text{href}_\top = \texttt{Pisa}_\top\texttt{.html} \end{aligned}$$

Thumbnails are thus an extreme case of image approximation. □

5 Future work: context aware servers

The view that a URL is "a pointer with a bandwidth" [1] is consistent with the idea of refinement through a partial order of approximations. The available bandwidth can be used by a server as a means of selecting an appropriate element within the order. It is proposed to develop context aware applications [12] which utilise bandwidth in determining response behaviour. In [7] you can find an outline of a client server architecture which includes a bandwidth aware server is sketched. The server estimates (roughly) the available bandwidth from the response time of a ping message. Subsequently the server can respond, on the basis of the bandwidth estimate, by making available an appropiate (to the bandwidth) chain of refinements. It is hoped that the use of context aware servers together with chains of approximate images may reduce the normal fustration levels associated with browsing with thumbnails using low bandwidth connections.

6 Concluding remarks

The definitions of mesh size and ordering presented in the paper are conventional mathematical concepts. However, the choice of mesh used to generate an approximate images may have significant psychological implications in the way that the resultant image is viewed. For example, an approximate image could be generated from the use of either vertical or horizontal rectangular meshes. For example, Figure 4 illustrates an alternative chain of refinements to the Pisa tower based on a different mesh configuration.
The choice of an appropriate mesh for compaction may be an art form rather than a mathematical discipline. For example, it would seem appropriate to compact an image of a traditional Mark Rothko painting using horozonal meshes – however, from a mathematical perspective, vertical meshes would be equally good.

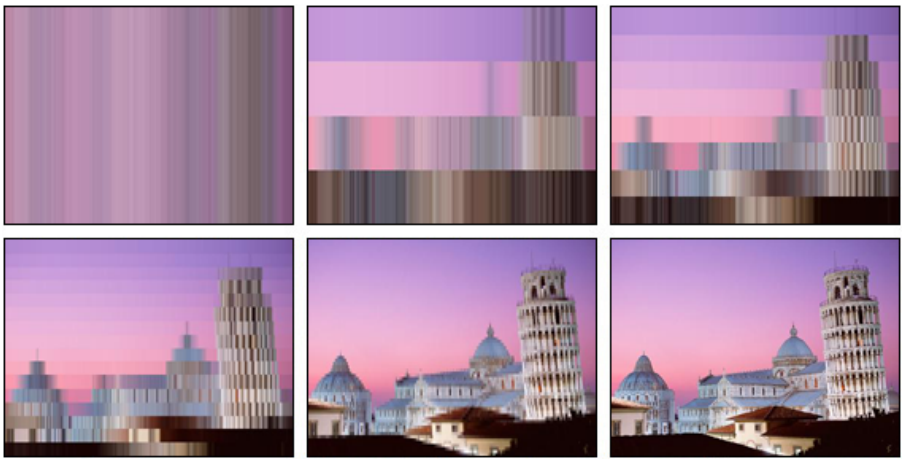


Fig. 4. Alternative approximations generated from different meshes.

The approach used to define meshes in this paper assumes uniformity. However, the definition of mesh can be easily extended so that a mesh may be identified with a non-uniform block partition of an image. The order relation for such a domain can again be defined using a form of inclusion. The use of non-uniform meshes would allow for the possibility of generating image approximations by concentrating detail in the centre of the image and allowing much greater rates of compaction around the image boundary.

References

1. Cardelli, L.; Davis, R.: Service Combinators for Web Computing. Proc. of the First Usenix Conference on Domain Specific Languages, Santa Barbara, 1997.
2. `http://www.computeruser.com/resources/dictionary`.
3. Cohen, H.: Retrieval and browsing images databases using image thumbnails, Journal of Visual Computing and Image Representation, Vol 8, No 2, 1997, 226-234.
4. Davey, B.A.; Priestley, H.A: *Introduction to lattices and order*. Second edition, Cambridge University Press, 2002.
5. Gabarró, J; Stewart, A; Clint, M: Grab and Go Systems: a CPO approach to concurrent web and grid-based computation, ENTCS 66 No. 3 (2002). `http://www.elsevier.nl/locate/entcs/volume66.html`.
6. Gabarró, J; Stewart, A; Clint, M; Boyle, E; Vallejo, I: Computational Models for Web- and Grid-Based computation. In Euro-Par2003, LNCS 2790, 640–650, 2003.
7. `http://www.lsi.upc.es/~gabarro/RefinementByBrowsing`
8. `http://www.ine.es`
9. `http://www.cnice.mecd.es/ayudas/tipo_conexion.htm`
10. Nielsen, J: *Designing Web Usability: The Practice of Simplicity*. New Riders Publishing, Indianapolis, 2000.
11. Vallejo I.: Visualización de sistemas Grab-And-Go, Bachelor thesis, UPC, 2004.
12. Schilit, B.N.; Adams, N; Want, R: Context-aware computer applications. In proceedings of the IEEE Workshop on Mobile Computing Systems and Applications, 1994.