RETRIEVAL AND BROWSING OF IMAGES USING IMAGE THUMBNAILS

Harvey A. Cohen

Computer Science and Computer Engineering
La Trobe University, Melbourne Victoria Australia 3083
Email: H.Cohen@latrobe.edu.au

Submission to J. of Visual Communication and Image Representation
Special Issue on Indexing, Storage, Retrieval and Browsing of Images and Video

Published with minor alterations as:
Harvey A. Cohen, *Retrieval and Browsing Image Database Using Image Thumbnails,* **Journal of Visual Computing and Image Representation**, Vol 8, No 2, 1997, pp 226-234

RETRIEVAL AND BROWSING OF IMAGES USING IMAGE THUMBNAILS

Harvey A. Cohen

Computer Science and Computer Engineering
La Trobe University, Melbourne Victoria Australia 3083
Email: H.Cohen@latrobe.edu.au
Phone +61 3 9479 2596
FAX: +61 3 9479 3060

ABSTRACT

In the emerging environment of massive image databases the display of a particular image at a workstation will be almost invariably preceded by its selection from a set of thumbnail images This paper explores the basics of image thumbnails, and the implications for efficient image storage. The advantages/disadvantages of alternate schemes for thumbnail generation are discussed., A measure of image compression is proposed to take into account the costs of multiple thumbs for each image accessed fully. Analysis suggests that to maximise the effective compression, requires the use of thumbs efficiently transferred that contain data required for whole image generation. To meet the needs of thumbnail-based retrieval, suggestions are made for the re-organisation of the code for block oriented image coding schemes leading to the specification and implementation of thumb-based vector quantization image coding, and a fast decoding thumb-fractal codec. A proposal for a thumb-oriented version of JPEG is outlined.

1.0 Thumbnail images

Thumbnail images are now commonly used as WYSIWYG directory icons for the selection of images from directories and local databases. But just what is an image thumbnail? And, seeing as there is no unique scheme for generating an image thumbnail, just what are the advantages/disadvantages of alternate formulations? We first seek to give answers to these basic questions. We then proceed to study the implications of the vital use of thumbnails in image browsing.

Notable systems for managing query of image databases, the QBIC system developed by Niblack and co-workers at IBM San Jose, [1] and the PhotoBook System developed at MIT Media Lab by Pentland et al, [2], feature arrays of small images for displaying the images most similar in content to the users query. That is, the output to an image database query, whether posed texturally or in more visual terms, takes the form of a set of images, which need to be visually perused by the user towards making final selection(s). A number of researchers, notably Picard[3][4], have stressed the important emerging role of very large image databases as in collections of compressed images stored on a CD ROM, or the far larger compilations available via the Internet. The needs for a browsable interface to these very large image albums and libraries requires the development of image compression schemes suitable and efficient for applications where the image thumbnails are more often required than full images, and full image display is invariably preceded by a data fetch of the thumbnail data presented during browsing [5]. As we discuss further in this paper, this entails that image compression schemes suitable for very large database applications should be so designed that thumbnail data is separately accessible, and that the information delivered in the thumbnail should be used with other image code for economical image synthesis or reconstruction.

In this paper we introduce a new measure of "effective compression" for image data accessed via thumbnails, to provide a direct means of evaluating image coding schemes for their application for thumbnail-based retrieval.

We then show how traditional block-oriented compressive coding such as vector quantization (VQ) and DCT-based approaches can be modified to meet the needs of thumbnail based image retrieval. A like scheme may be applied to re-organise block-oriented fractal coding, leading what is called “thumb-fractal”-- a very rapidly converging fractal codec. An outline is presented of a proposal for a thumb-based variant of JPEG, based on a partition of baseline JPEG code.

1.2 Gray-scale Thumbnails

Although image thumbnails are now commonplace, a literature search failed to locate any previous formal description of thumbnails. The following general definition of a thumbnail image is proposed:

An image thumbnail is produced after the partitioning of an image into rectangular (usually square) blocks of pixels, and constructing a thumbnail image comprising blocks, of single pixel-size or larger, of uniform pixel value. The blocks in the thumbnail are uniformly scaled with respect to those of the image.

In the simplest case of the uniform partitioning of an image into constant-sized blocks each of R rows and C columns the image block at (mR, nC) in the image may be denoted by its block row number m and block column n as B[m][n]: the corresponding block in the thumbnail contains pixels of gray-scale or colour index b[m][n].

For reasons of computational convenience b[m][n] is often computed as the block mean m_B of pixel gray-scale value q[r][c] within the image block:

$$m_B = \sum q[r][c] / RC$$

From the perspective of a thumbnail pixel being an estimation of the grayscale values present in the block, robust statistics theory (see, e.g., Staude[6]) provides two distinct estimators for thumbnail grayscale:
(a) If pixel values in the block follow a short-tail distribution, use the block mean:
(b) If pixel values in the block have a long-tailed distribution, use the block median.
It is possible for the block mean to fall in a range of gray-scale not present in the block, or surrounding blocks, whereas a (grayscale) median value has the feature of actually being present on at least one pixel in the block.
For raw images the use of the top left hand corner (TLHC) pixel of each block in the thumb is the simplest and fastest means of generating a thumb. The MIT system PhotoBook offers both mean and TLHC thumbs..

The 'true' thumbnail image containing just one pixel of gray-scale b[r][c] to represent the block B[r][c], is here simply called the image thumb. The full-scale image with blocks the same size as the corresponding blocks in the original image but with every pixel in the block the thumb value is here called the zoomed thumbnail.

In Fig 1, examples of the three varieties of thumbs so far specified are given for the Lena image.

FF image of Lena

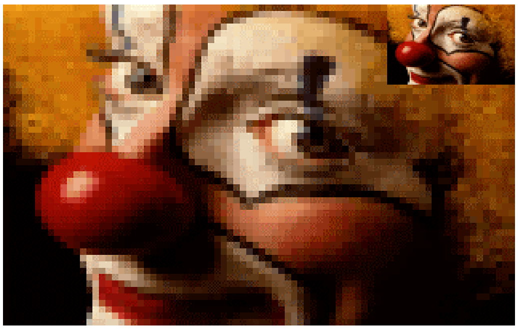
Block mean thumb, with exploded thumb

TLHC thumb with exploded thumb

Block median thumb with exploded thumb

Fig 1 256x256 8-bit grayscale Lena image with three varieties of thumbnails, based on 4x4 blocking. Exploded version of the thumbnails are supplied to highlight issues of edge jaggedness, discrepancies in local image texture, and the presence of irritating blocking artifacts.

Where an image has been decomposed into regularly scaled blocks, as in quad-tree decomposition, the thumb is clearly the thumbnail scaled so that the smallest block is of single pixel size. For irregular image blocking, such as the irregular partitions used in some recent fractal coding schemes, based on a muliples of a 4x4 block [7] , the thumb is clearly based on single pixels representing such 4x4 building blocks.



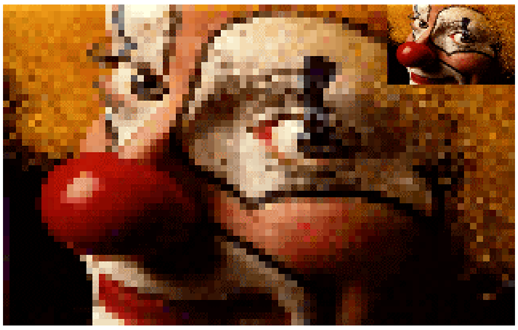


Fig 2 From top: Clown 320x200 8-bit indexed colour. Zoomed-out colour mean thumbnail with (inset) thumb, PSNR = 23.24 dB, total (un-normalised) histogram difference = 20,434. Zoomed-out corner value thumbnail with (inset) thumb, PSNR = 19.59, histogram difference = 10, 544.

1.3 Colour Thumbs

For 24-bit and 16-bit colour images, often called "true colour"images, each pixel is a vector comprising three colour co-ordinates. In classic (RGB) colour co-ordinates, each pixel in the block B has coordinates (q[r][c][RED], q[r][c][BLUE],q[r][c][BLUE]). For such images, "mean colour" thumbnails can be produced with each component of the thumb the average of corresponding component (red, green, blue) of the pixels in each block, according to the formula:

$$m[colour] = \sum q[r][c][colour] / RC$$ (where colour = RED, GREEN, or BLUE.)

Here the sum is over all the RC pixels in the block. This simple scheme yields a 24-bit (RGB) thumbnail. More compressed versions of such thumbnails can be determined using colour quantization.

For indexed colour images, the pixel value q[r][c] at location (r,c) is a pointer into a palette table, such that the RED, GREEN, and BLUE components of the pixel are (palette[q[r][c]][RED], palette[q[r][c]][GREEN], palette[q[r][c]][BLUE]), It follows that the mean RGB colour components in a block is given by

$$m[colour] = \sum palette[q[r][c]][colour] / RC$$ (where colour = RED, GREEN, or BLUE)

where the sum is over all q[r][c] pixels in the block. This particular RGB colour can be represented, using the same palette for the thumb as for the FF image, by the colour in the palette that is closest, say the colour index p such that

$$\| palette[p] - m\| \leq \| palette[q] - m \|$$ any colour index q.

For RGB coordinates, the metric is simply Euclidean,

$$\| palette[p] - m\leq \|^2 = \sum (palette[p][colour] - m[colour])^2$$ (summing over colour)

For other colour systems, such as HSI, other metrics are applicable. The use of such a “nearest match” for the mean colour in each block implies a colour error for each block, suggesting that colour thumbs should be dithered. The recent version of a well-known share-ware utility, Image Alchemy GWS (Graphic Workshop) does in fact offer dithered thumbs.

1.4 Evaluating Thumbnail Image Quality

Thumbnail images are small images derived from a larger image by subsampling or downsampling, possibly in conjunction with convolution and error diffusion. However, it is most important to stress, that unlike the sparsely sampled images that are the subject of Shannon’s Theory and most of classic sampling theory, this subsampled data is “compressed” to a smaller size. The quality of a thumbnail image is clearly related to how well it indicates the visual information present in the full-scale (FF) image. How should thumbnail images be evaluated in terms of quality?

The simplest measure of thumbnail quality would involve summing the squares of difference between the thumbnail pixel denoting each block, and all the pixels in that block, and summing over all image blocks. This quality measure equivalent to the squared difference between the original image and the exploded thumbnail as defined above. In the figures of this paper, giving examples of thumbnails, the exploded thumbnails are shown, and the Peak Signal to Noise Ratio (PSNR). However, the images in these Figures reveal immediately the extreme adhocness of this measure; the exploded thumbnail features blocking artifacts not at all present in the (smallest) thumb. Hence, apart from its simplicity, the PSNR of the exploded thumb is a well-defined objective measure but of limited validity as a quality measure for thumbnails.. However, the PSNR figure for an exploded thumb is of special interest in thumbnail based image retrieval, as the exploded thumb in the first stage in local generation of a full-sized image.

The development of a truly adequate measure of the quality of thumbnails will require some novel modelling of the human visual system. An example will make this claim clearer. Consider, a one dimensional (temporal) thumbnail -- for example a 1:4 downsampled version of Chopin’s Minute Valse (Waltz), reproduced over 15 seconds. Such a temporal acoustic thumbnail would be essentially non-recognisable, due to the higher acoustical frequencies present. However in the spatial domain, thumbnail images are recognisable, despite the gross differences in spatial frequencies in an image compared to its thumbnail. The human visual system extracts information from images in such a way that scale and frequency are not the primary issue issues. For example edges are especially significant in (human) image interpretation; the more important edges of an image, those confirmed at multiple scales in the original image in Marr’s model of perception [8], will still be present in the image thumbnail.

Recently several workers have used image histogram differences as a similarity measure. This measure can be simply applied to images of different sizes, by dealing with normalised histograms. In this paper, the need for normalisation is avoiding by computing the histogram difference between an image and the exploded thumbnail, with the same number of pixels in both.

In detail, if image has histogram image_hist[q], where q is gray-scale or colour index, and exploded thumbnail has histogram thumb_hist[q], the histogram difference has value

$$\text{Histogram difference} = \sum | \text{image\_hist}[q] - \text{thumb\_hist\_}[q] |$$

where the sum is over all pixel values. This quality measure is not sensitive to image distortions -- an image and its reflection have identical histograms -- but it certainly measures how much image and thumb are similar in a grayscale or colour sense. The histogram difference has been used in selecting images by colour in QBIC[1] and PhotoBook [2].

A further statistic that is applicable to assessing the quality of thumbnails is the number of gray-scales or colours actually present in an image.

Experimental results for the quality measures for the thumnails of Figure 1,2 are

Lena: Image has 211 distinct gray-scales. Thumb quality data is:

Thumb type	PSNR	Histogram difference	No of grays
Block mean	23.80 dB	14,076	188
TLHC	19.80 dB	10,832	185
Median	23.21 dB	13,950	192

Clown: Indexed colour image, featuring 255 distict colours

Thumb type	PSNR	Histogram difference	No of colours
Colour mean	23.24 dB	20,434	243
TLHC	19.59dB	10,544	248
Colour mean #	23.13 dB	15,400	244

Colour mean # involves use of pixel colour with each block that is closest to the colour mean.

2.0 Compression Efficiency of Image Retrieval With Thumbnails

The crudest measure of efficiency of image compression is the compression ratio, lets call it C, of the entire image, as it is stored remotely. C measures the efficiency of storage of whole image in a database. If the user selects such an image for full-scale display via a thumbnail based interface, then the cost of accessing includes the transfer cost of all the thumbnails perused. We here define an effective compression as a function of the probability p that from a set of thumbnails a whole-image selection will be made. For simplicity suppose that one is dealing with a set of images all of the same size.

Then, formally:
If p is the probability that a viewed thumbnail image will be selected for full-image display
T = data to be transferred to generate (one) image thumbnail
F = data to be transferred for full-size image display
S = savings in using thumbnail data to generate display

Note that the average number of thumbnails displayed for each selection of one image for full-scale display is $\frac{1}{p}$. Note that if p =0.1, this means that on average 10 thumbnails are fetched for each full-size image display. We use F to mean the compressed data required at the workstation when no use is made of thumb data to generate the full-size image. However, when that data is utilised, some savings S is achieved.

The data required at a workstation to display a particular image and its thumbnail, in absence of savings, is just T + F. However from the systems perspective the average amount of image data delivered for the display of each full-sized image is just:

$$\frac{1}{p} T + F - S \quad \text{where } 0 \le S \le T$$

By definition of compression, the image data in the display is just C F .

Hence that the effective compression C_{Eff} in terms of bytes in the display versus transferred bytes is

$$C_{Eff} = C \frac{\frac{pF}{T}}{1 - \frac{pS}{T} + \frac{pF}{T}}$$

The following are indicative values:

C	60	60	45	60	45
p	0.1	0.1	0.1	0.1	0.1
$\frac{F}{T}$	3	3	4	1	5
$\frac{S}{T}$	0	1.0	1.0	1	1.0
C_{Eff}	13.86	15.0	13.86	6	16.5

The table shows the dominant significance of the ratio $\frac{F}{T}$ on the effective compression, and the significant impact of the savings S. One can conclude that it is most crucial to devise efficient thumbnail compression schemes, and that if the thumbnail information is utilised for the synthesis of the displayed image there will be significant advantages, which especially ought to be incorporated into network browsers.

3.0 Thumb-based Block Code

In this section a simple way of modifying block-oriented image coding is presented, which greatly and efficiently facilitates thumbnail-based image retrieval. Such a modified code is termed a thumb-based codec. To facilitate the analysis, thumb-based VQ shall be first described and analysed, and then the general problem addressed. This analysis is preceded by a brief review of vector quantization.

3.1 Vector quantization basics

Image coding via vector quantization requires the use of a code book whose contents are best called tiles, though often, with mathematical emphasis, called vectors. The coding algorithm involves the partition of an image into blocks, and the determination for each block in the image of the block-code, namely the number of the tile in the code-book that is closest, usually in the least squares sense. Classic references are [10][11][12].

Thus VQ code of an image comprises

(a) Header information
(b) Block Code - array of tile numbers for each block
(c) The code book - an album of tiles/vectors

For regularly blocked image, for which the code is an array the header holds just the few bytes that specify image and block sizes, together with display palette specification.. For irregular sized blocks, as for quad-tree or BSP coding, the header must detail the construction tree indicating the sequence to be followed on synthesis.

High efficiency VQ coding involves compression schemes to further reduce the overall data-size, and the construction tree may be scattered through the code. However the point to be made here is that there is no separate facility for thumbnails within the code, and the full VQ code needs to be received before an image thumbnail can be constructed.

3.2 Thumb- VQ

What is proposed is that the VQ code should consist of four parts instead of the traditional three:

(a) Header information
(b) Block Code - set of tile numbers for each block
(c) Thumb code: a table of mean gray-scale or colour-mean for each tile in the code book
(d) The code book - an album of tiles/vectors

If the thumb code is available from the database as indicated, then part (d) of the image code is not required for the construction of the thumb by the browser. Using the formula for effective compression above, one can readily estimate the considerable savings for typical situations.

The thumb code is relatively small, and adds little to overall coded image size. The extra storage costs for the thumb code can be made negligible, at computational cost: for gray-scale images, knowledge of the block-mean, enables the more significant bits of one pixel to be determined from the pixel values in the block.

In sum, for regular blocked, or regularly graded blocks as in quad-tree, VQ coding is readily augmented to facilitate thumb-based retrieval.

3.3 Thumb-block oriented Coding

Thumb-block oriented coding can be applied to image codecs for which the image to be coded is partitioned into non-overlapping square blocks, which are scaled regularly with respect to a minimum size. Ordinary block-oriented code has the following contents::

(a) Header information
(b) Image specific data
(c) The block code - an array of data for the generation of each block

Where the image blocking is uniform, the image thumbnail would be simply the image for which each single pixel is the gray-scale or colour mean of the corresponding image block. Where the image blocking involves a regular scaling as for quad-tree decomposition, the thumb would be similarly blocked as the image, but with the smallest block of the thumbnail consisting of a single pixel; the pixels in each block of the thumbnail are equal to the gray-scale or colour mean of the corresponding block of the FF image.

Thus to produce the image thumbnail, what is required is the header information, perhaps not all, some or all of the image specific data, together with the value of the gray-scale or colour mean for each block. This leads to the following model for thumb-oriented block code:

(a) Header information
(b) Image specific data
(c) Thumb code: a table of mean gray-scale or colour-mean for each tile in the code book
(d) The block code - an array of data for the generation of each block

Then the image code necessary and sufficient to produce the thumb comprises T =(a) + (b) +(c); in thumb-based retrieval the FF image would be generated following the further transmission of (d).

3.4 Thumb Fractal

Block oriented fractal coding, as introduced by Jacquin, [13][14] can be viewed as a special variety of vector quantization; however, instead of block tiles being stored in a separate code-book, the tiles are determined as compressed and transformed copies of larger blocks (range blocks) within the image itself.. To follow Fisher[15] in detail the fractal code for each block comprises the location of a larger block in the image, a block transformation, and a linear transform of pixel gray-scale, involving a luminance and contrast parameter.

A thumb fractal codec can be specified by applying the basic scheme of block-oriented fractal coding to the (signed) image that is the difference between the the target image and its exploded thumbnail. Using a block mean thumbnail, this means that the blocks that are so coded have zero mean, and the luminace parameter for gray-scale linear transform is zero, so the thumb fractal code requires somewhat less data to specify than otherwise. A full discussion of this still image codec is presented by Cohen [16]. In figure 3, experimental results for the decoding of a thumb fractal Lena image are presented. What is notable is the very fast convergence. In the example presented, the image has been uniformally blocked into 4x4 blocks, and the compression is only modest; however, as discussed in [16], the approach can be extended to more elaborate image blocking schemes such as quad-tree, which are highly compressive.

Lena 256x256x8-bit image

Lena Thumb Fractal
First iteration PSNR = 31.33 dB

Lena exploded 4x4 Thumb
PSNR = 23.80 dB

Lena Thumb Fractal Decode
Second iteration PSNR = 32.68 dB

Fig 3 Demonstration of the fast convergence of the thumb fractal code developed by the author. Iteration starts from the exploded block mean thumb, and is essentially completed on the second iteration. See text, and Cohen [16]

4. JPEG coding and thumbs

JPEG coding is now one of the two most important image compression schemes for use on the Internet. The ISO standard [17] [18] admits a number of variants, but the most commonly used involves the use of regular blocking into 8x8 pixel blocks; which are separately coded, except for the "DC component" of each block, which is essentially the block mean. In JPEG, the table of means is separately stored, after compression. Thus at the logical level of description, JPEG is already a thumb-based codec. Unfortunately, image storage standards are not amenable to thumb-based retrieval along the lines presented above. It would seem that some relatively modest alteration to the standard, with this thumb material separately accessible, could be advantageous for thumb-based retrieval.

In the hierarchical mode of JPEG [17], an image is stored at several increasing resolutions. For example, an image 16x16, then 128x128, then 1024x1024 . Certainly one of the smaller images can function as a thumb. However, this mode is more of a promise than a reality. Hierarchical mode is geared to the idea of slow but progressive update of the image during display. However, the microprocessors of today can readily decode JPEG in acceptable times for single images.

The definition of a thumb-nail for JPEG colour images is complicated by the different scale of the three components: A common subsampling scheme is to use one Cb and Cr sample for each four Y samples.
JFIF JPEG [19] is the prime version of baseline sequential JPEG. In fact JPEG [17][18] is a rather catholic standard, admitting a number of variants, although only the baseline sequential variety must be supported by any decoder. This very technical note is to distinguish thumb-JPEG from these variants..

JFIF, the JPEG File Interchange Format was defined in a technical note from C-Cube Microsystems. [19]. A JFIF file has a JPEG style header, and the characteristic "JFIF" character string after the first 6 bytes of the file, which contains a JPEG image that is either grayscale or YCbCr colour. A JFIF file may also (ie optionally) **also** include a thumb-nail RGB (!) version of the image. Note that this inbuilt JFIF RGB thumbnail is additional to the ordinary code, so such (rare) JFIF code is NOT a thumb-JPEG codec - the thumbnail data is very much additional, the savings S of the formula above is zero. The developers of JFIF have been aware of the importance of thumbnails but had not analysed the full needs of thumbnail-based image retrieval

5. Discussion

Although the thumb is a small image, typically containing only 6% of the pixels in the full-size image, network traffic in thumb data is of the same order as that of the data for full-scale image display. This is for two reasons. Firstly, in thumbnail-based access, many thumbnails are accessed for each full-scale image display, Secondly, smaller images are not as readily compressed as larger images. Thus, for instance, for the Clown image, 320x200 pixels, a JPEG coding size was 13,709 bytes, while its 80x50 pixel thumb was compressed by JPEG to 2,065 bytes -- despite a 16:1 pixel ratio the ratio of bytes in JEG is only 6.64. This comparison was made for a medium quality JPEG for both, but there is arguably greater need for a higher Q for the thumb than for the full-scale image.
We have defined thumbnail images rather formally in terms of the image blocking used in most image compression schemes, where a pixel value within the thumbnail is taken to be representative of the various pixel values within the corresponding block of the whole image. For gray-scale images, this leads naturally to the (common) formulation of using a mean gray-scale, although statistical considerations would tend to favour the use of the block-median. A not-uncommon, and

computationally very cheap alternative, is to use a pixel value from a particular block position, such as the top left-hand corner of each block.
In Section 1.4 we intoduced three objective measures of thumbnail quality: the PSNR of the exploded thumb, the histogram difference, and the number of grays/colours used. These measures as applied to the Lena and Clown images indicate reasonably the poorer quality of the TLHC thumb, and suggest some subtle differences of assment between the mean and median thumbs for grayscale Lena, and varieties of colour mean thumbs for the indexed colour Clown of figure 2. It is worth contrasting these óbjective' measures with a considered assesment from viewing the thumbs on a monitor, using a 24-bit (”True-colour”) display allowing simultaneous viewing. For the gray-scale Lena, the block-mean thumb is generally the more satisfactory over generally smoothly varying features, but there is considerable blurry of highly textured regions such as the hat and hair. TLHC thumb Lena has a somewhat displeasing overall appearance, with an element of jaggedness, but textured regions are better represented. The median thumb has somewhat better rendering of texture of the hat and hair than the mean thumb. The colour-mean Clown thumbnails appear as smoothed but acceptable representatives of the image, with negligible visual differences between.variants; the TLHC thumb had the anticipated jaggedness, but does have a sharper, cleaner, colouring.

The regular grid of image blocking has artificially introduced new frequencies into the thumb-image. However, only in the "exploded" or "zoomed-out" thumbnails is this aliasing especially noticeable.

A new parameter "effective compression" is introduced that is a function of the probability p that a browsed thumbnail will be selected for full image display. We examine the relationship between the usual image compression, and the effective compression, in terms of the utilisation of the image thumb data in the generation of the display image. For credible values of the probability p, it is clear that it is well-nigh vital to make effective use of the thumb data for whole image generation..

In Section 3 it is shown how conventional VQ code can be re-organised, with almost no further overhead, to be a thumb-VQ code, where the thumb image data is purposefully used for FF image generation. It is exceedingly fruitful to consider FF image generation, with thumb data available, as providing corrections to the exploded thumbnail. This perspective leads both to the formulation of thumb-fractal codices, and the implementation of a thumb fractal codec [16]. In baseline JPEG, the DC component of each 8x8 block is essentially the thumb pixel value. In the existing code the DC data is separately treated and compressed. It follows that only a relatively minor refinement of the established standard can lead to a thumb-JPEG code, tailored to thumbnail-based access. Because of the different scale of the YCbCr components, commonly 4:2:2, the JPEG thumbnail involves combining the Y-thumb, based on 8x8 blocks, with the chromatic Cb, Cr thumbs based on different scales.

The existing standard and de-facto image compression standards were based on the needs of digital image archiving for purposes such as medicine [20]. However, we must now consider the implications for the development of the huge digital libraries via networks such as the World Wide Web. Already there are under development prototype large systems such as the Chabot Project aiming at providing thumbnail based access to 500,000 high resolution images [5]. For such libraries it is not enough to merely compress each image as much as one can - each image in a massive image data-base must be accessed via retrieval keys, and none is as basic as the image thumbnail. Thus Chabot, which puts high resolution on slow access, will store image thumbnails at fast access with other index keys.[5]. In the next generation of image storage schemes retrieval material and image data must be effectively and efficiently coded in line with the scheme of access, which will be via keys including the image thumbnails.

FOOTNOTE

The images reproduced here are available on the author's "Thumbnail Primer" at the URL:

http://www.cs.latrobe.edu.au/~image/thumb.html

References

[1] W. Niblack and M. Flickner, Find Me the Pictures That Look Like This: IBM's Image Query Project, *Advanced Imaging*, April 1993, pp. 32-35.

[2] A. Pentland, R.W. Picard and S. Sclaroff, Photobook: Tools for content-base manipulation of image databases, in *Proc Society for Optical Engineers, SPIE Storage and Retrieval of Image and Video Databases II*, San Jose, California, 1993.

[3] R. W. Picard and T. Kabir, Finding Similar Patterns in Large Image Databases,*Proc. IEEE Int'l Conf Acoustics, Speech and Signal Processing, ICASSSP-93*, Minneapolis, MI, 1993, Vol V, pp. V-161-164.

[4] R. W. Picard and Fang Liu, A New Wold Ordering for Image Similarity, *Proc. IEEE Conference on Acoustics, Speech and Signal Processing, ICASSSP-94*, Adelaide, South Australia, 1994, Vol V, pp. V-129-132.

[5] V.E. Ogle, Chabot: Retrieval from a Relational Database of Images, *IEEE Computer*,Vol 28, No 9, Sept

[6] R.G.. Staude and S.J. Sheather, *Robust Estimation and Testing*, Wiley, New York, 1990

[7] M. Tanimoto et al, A novel fractal image coding scheme employing blocks of variable sizes, *Proc IEEE Conf on Image Processing, ICIP-96*, Lausanne, Switzerland, 1996, Vol 1, pp. 137-140.

[8] D. Marr, *Vision*, W.H. Freeman, San Francisco, 1982.

[10] Y. Linde, A. Buzo and R.M. Gray, An algorithm for vector quantizer design, *IEEE Trans, Commun.*, COM-28(1), 1980, pp. 84-95.

[11] S.P.Lloyd, Least squares quantization in PCM, *Bell Lab Memo*, July 1957; also in *IEEE Trans Info Theory, 'IT-28*, 1982, pp. 129-137.

[12] J. Max, Quantizing for minimum distortion, *IRE Trans Info Theory*, IT-6(1), 1960, pp. 7-12.

[13] A.E. Jacquin, A novel fractal block-coding technique for digital images, *Proc Int'l Conf Acoustics, Speech and Signal Processing, ICASSP'90*, 1990, pp 2225-2228

[14] A.E.Jacquin, A novel fractal block-coding technique for digital images, *IEEE Trans. Image Processing*, Vol 1, No 1, 1992, pp. 18-30.

[15] Fisher, T.P. Shen and D. Rogovin, Fractal (self VQ) encoding of video sequences, in Proc SPIE: VCIP, Chicago, Illlinois, 1994. SPIE - 2304-1

[15] Y. Fisher et al, Fractal Image Compression using Iterated Transforms, in *Image and Text Compression* (J. Storer, (Ed.), pp 35-61, Kluwer Academic Publishers, Pittsburg, PA, 1989

[16] H. A. Cohen, Thumbnail-based Image Coding Using the Fractal Transform, *Proc. IEEE Conference on Image Processing, ICIP'96*, Laussanne, Switzerland, 1996. Vol 1, pp. 145-148.

[17] ISO/IEC CD 10918-1 Digital Compression and Coding of Continuous-tone Still Images, Part 11: Requirements and guidelines, New York, ANSI

[18] G. K. Wallace, The JPEG Still Picture Compression Standard, *Communications of the ACM*, Vol 34, 1991, No 4, pp 30-44

[19] E. Hamilton, JPEG File Interchange Format Version 1.01, C-Cube Microsystems, Inc, 399A West Trimble Road, San Jose, CA 95131, Aug 20, 1991.

[20] J.R. Cox et al, Some Design Considerations for Picture Archiving and Communications Systems, *IEEE Computer*, Vol 16, No 8, August 1983, pp. 39- 49.