ADAPTIVE REPRESENTATION OF JPEG 2000 IMAGES USING HEADER-BASED PROCESSING

Ramesh Neelamani[*]

Rice University
ECE Department, 6100 S. Main
Houston, TX, 77005
neelsh@rice.edu

Kathrin Berkner

Ricoh Innovations, Inc.
2882 Sand Hill Rd., Suite 115
Menlo Park, CA 94025
berkner@rii.ricoh.com

ABSTRACT

To bridge the mismatch between the sizes of images and display devices, we present an efficient and automatic algorithm to create an adaptive image representation called SmartNail. Given a digital image and rectangular display frame smaller than the image, we define the SmartNail as an appropriately cropped part of a suitably scaled-down image. We choose the SmartNail-defining parameters — down-scaling factor and cropping location — to maximize a bit-allocation-based cost function that quantifies the visual importance of the image content in the SmartNail. For JPEG 2000-encoded images, the SmartNail parameters can be determined using just the header information available in the encoded file. Hence only the wavelet coefficients required to reconstruct the SmartNail need to be decoded from the entire JPEG 2000 code stream. Consequently, the SmartNail construction requires minimal computations and memory requirements. Simulations demonstrate the effectiveness of SmartNail representations.

1. INTRODUCTION

The variety of displays used to browse and view images has created a need to adapt an image representation to the size constraints of a given display. For example, a high-resolution image might need to be displayed simultaneously on a desktop monitor and a PDA. The conventional thumbnail representation obtained by uniformly scaling the whole image to fit in the display area is often unsatisfactory; for example, the image of a text document scaled down to fit on a cell-phone display most likely does not convey much meaningful information.

Several approaches have been proposed to represent images in a smaller display. One approach is to create a representation by pasting key-words extracted from the original image using optical character recognition (OCR) onto a scaled version of the image [1]. However, such an approach is neither universally applicable due to the unavailability of text key-words in all types of images (see *Lena* image in Figure 4), nor computationally efficient due to OCR usage. Another approach is to manually associate a resolution-dependent importance value with different image regions to aid cropping and scaling at the application end [2]. The need for manual image editing makes the second approach undesirable for general image browsing applications.

In this paper, we describe a framework to construct an adaptive reduced-size image representation termed *SmartNail* (smart thumbnail) by automatically and efficiently scaling and cropping a given image. The SmartNail is characterized by an appropriately located window in a suitably down-scaled original image (also see Figure 1). We obtain the scaling and location parameters by optimizing over a bit-allocation-based cost function that measures the visual importance of image information contained in the image representation defined by the parameters.

Characterizing a SmartNail in the wavelet domain allows us to create the SmartNail using wavelet-based image coders such as the state-of-the-art JPEG 2000 (J2K) coder [3]. Thanks to the J2K code-stream syntax, the optimal scale and location parameters defining the SmartNail are calculated from the compressed file using only the header information of the J2K-encoded image; no decoding is required for this step. With knowledge of these parameters, the SmartNail can be constructed by decoding the minimum number of wavelet coefficients from the J2K code stream. Thus the SmartNail construction is very efficient both memory wise and computationally.

While a compressed domain image processing algorithm has been reported for traditional JPEG images [4], the algorithm is not truly header-based, because it needs to parse the entire code stream. Further, the algorithm is not truly multiresolution, because it does not have access to any local frequency information without decoding the image.

2. MATHEMATICAL PROBLEM FORMULATION

The SmartNail creation problem (also see Figure 1) described in Section 1 can be formally posed as the following optimization problem. Let I denote the original continuous image with normalized support $[0, 1] \times [0, 1]$, and I^N be the image obtained by sampling the low-pass filtered (for anti-aliasing) I at rate N sample/unit-length. Define the $X \times Y$-sized image representation $\mathcal{D}(N, A) := \{I^N(x, y) | (x, y) \in (A_x, A_y) + [0, X-1] \times [0, Y-1]\}$, where $I^N(x, y)$ denotes the pixel of I^N at image location (x, y), and $A := (A_x, A_y)$ denotes the anchor location for the display. Let $\Lambda(N, A)$ be a cost function that measures the visual importance of the representation $\mathcal{D}(N, A)$. Then the SmartNail solution is specified as

$$\begin{aligned} \text{SmartNail} &= \mathcal{D}(N^*, A^*), \\ \text{where } \{N^*, A^*\} &:= \arg\max_{N,A} \Lambda(N, A). \end{aligned}$$

The challenge in the above SmartNail formulation is to choose an easily computable cost function that measures the visual importance of image information.

We propose using a weighted sum of the bits spent at low bit rates on a local region $\mathcal{D}(N, A)$ by reasonable multiresolution image coders as a cost function that yields meaningful SmartNails

[*] The author performed the work at Ricoh Innovations, Inc.

with minimal processing. The number of bits spent is termed the MultiResolution Local Information (MRLI). The multiresolution aspect of the chosen cost function is clear — the sampling rate N characterizes the resolution at which any spatial region is described.

To motivate that the MRLI provides a reasonable metric to quantify the visual importance of a given display, we take a closer look at "ideal" multiresolution image coders. An "ideal" multiresolution image coder strives to provide the best visual representation of a given image within the available bits. At low bit rates, an "ideal" image coder spends relatively more bits describing the important spatial features of the image at the appropriate resolution. Hence the total bits spent by the ideal coder on $\mathcal{D}(N, A)$ can be viewed as a metric for the visual importance of all spatial features observed at the resolution determined by N in the representation $\mathcal{D}(N, A)$. Thus the MRLI provided by "ideal" multiresolution image coders will serve as an excellent cost function to determine SmartNails. Though practical multiresolution image coders cannot be considered truly "ideal", their bit allocation at low bit rates still provide useful information about the underlying image. Hence MRLI with weights chosen to offset some of the non-ideal nature of practical image coder's bit allocation provides a reasonable metric to quantify the importance of a chosen display.

3. SMARTNAILS IN THE WAVELET DOMAIN

3.1. Wavelets

The wavelet transform represents a continuous image I in terms of shifted versions of a low-pass scaling function ϕ and shifted and scaled versions of a bandpass wavelet function ψ [5]. For special choices of ϕ and ψ, the functions $\phi_{j,k}(t) := 2^{j/4}\,\phi\big(2^j t - k\big)$, and $\psi_{j,k,l}(t) := 2^{j/4}\,\psi_l\big(2^j t - k\big)$ with $t := (t_x, t_y) \in \mathbb{R} \times \mathbb{R}$, shift $k := (k_x, k_y) \in \mathbb{Z} \times \mathbb{Z}$, scale $j \in \mathbb{Z}$, and sub-bands $l \in \{1, 2, 3\}$ form an orthonormal basis. Let $N = 2^{J_1+1}$ for some $J_1 \in \mathbb{Z}$. Then, a finite-resolution, continuous approximation $\widetilde{I}^N$ to I is given by

$$\widetilde{I}^N(t) \;=\; \sum_k c_{J_0,k}\,\phi_{J_0,k}(t) \;+\sum_{l=1}^{3}\sum_{j=J_0}^{J_1}\sum_k d_{j,k,l}\,\psi_{j,k,l}(t), \quad (1)$$

with the scaling coefficients $c_{j,k} := \langle I, \phi_{j,k}\rangle$ and wavelet coefficients $d_{j,k,l} := \langle I, \psi_{j,k,l}\rangle$. The scale parameter j increases from the coarsest scale J_0 to the finest scale J_1. The $J_1 = \log_2(N) - 1$ controls the resolution of the wavelet reconstruction $\widetilde{I}^N$ of I; in fact, the L_2 norm $\|\widetilde{I}^N - I\|_2 \to 0$ as $\log_2(N) \to \infty$.

In practice, an image is typically specified in a sampled form with N samples available along each direction; these samples define I^N. Conventionally, the samples of I^N are assumed to well-approximate the scaling coefficients $c_{J_1+1,k}$ of I at scale $J_1 + 1 = \log_2 N$ [5]. A good low-resolution, sampled image $I^{2^{J+1}}$ for $J < J_1$ is defined by using the coarse scaling coefficients $c_{J+1,k}$ as samples. The coarse scaling and wavelet coefficients are easily obtained by passing I^N through a filter-bank.

3.2. Wavelet-domain mathematical reformulation

The mathematical formulation described in Section 2 can be reformulated in the wavelet domain using the notation described above. Any $X \times Y$ image representation $\mathcal{D}(2^{J+1}, A)$ obtained from an arbitrarily sub-sampled image $I^{2^{J+1}}$ for $J \le J_1$ can be redefined in the wavelet domain as $\mathcal{D}(2^{J+1}, A) := \{c_{J+1,k} | k \in (A_x, A_y) + [0, X-1] \times [0, Y-1]\}$. The set of scaling and wavelet coefficients W required to approximately reconstruct this image display $\mathcal{D}$ is given by[1]

$$\begin{aligned} W(J,A) \;:=\; & \{c_{J_0,k} \,|\, k \in \mathcal{K}(J_0)\} \;\bigcup \\ & \{d_{j,k,l} \,|\, J_0 \le j \le J,\; k \in \mathcal{K}(j),\; l \in \{1,2,3\}\}\,, \end{aligned} \quad (2)$$

with $\mathcal{K}(j) := \left\{ \left\lfloor 2^{(J_0-J-1)}\; support(\mathcal{D}(N,A)) \right\rfloor \right\} \subset \mathbb{Z}$. The $\lfloor . \rfloor$ denotes *floor to integer*. The set $W(J, A)$ corresponding to the different image display choices of Figure 1 are marked by the hashed regions in Figure 2.

The MRLI cost can be specified in the wavelet domain as

$$\Lambda(J,A) \;=\; \sum_{c_{J_0,k} \in W(J,A)} \gamma_{J_0,k}\, B(c_{J_0,k}) + \sum_{d_{j,k,l} \in W(J,A)} \delta_{j,k,l}\, B(d_{j,k,l}), \quad (3)$$

where $B(.)$ denotes the fraction of bits spent on the corresponding coefficient, and γ and the δ denote the weights chosen to offset the non-ideal nature of practically available bit allocation. Thus the SmartNail problem can be reformulated in the wavelet domain as

$$\{J^*, A^*\} \;\;:=\;\; \arg\max_{J,A}\, \Lambda(J,A). \quad (4)$$

In words, the SmartNail problem in the wavelet domain is equivalent to simultaneously choosing the appropriate wavelet scale J^* and anchor location A^*.

4. SMARTNAILS FROM J2K IMAGES

J2K is a newly standardized wavelet-based coding scheme to represent digital images in a coherent code stream and file format [3]. The J2K image coding algorithm consists of the following steps: (1) Optionally divide the image into independent rectangular tiles. (2) Take the wavelet transform of each tile. (3) Partition the wavelet domain at each level and sub-band to form local groups of typically 32×32 or 64×64 coefficients termed *code blocks*. (4) Encode each code block independently using an arithmetic coder. (5) Organize the coded coefficients into one or more *layers* to facilitate progression. (6) Assemble all the coded and organized data into smaller units called *packets*. (7) Store the organizational information required to parse the coded data in the packet headers.

4.1. MRLI from J2K headers

J2K is designed to suit many different applications. For example, a particular image may be J2K-encoded such that it can be reconstructed at a variety of bit rates. Such features are enabled because a wide variety of information is stored in the headers.

Using the J2K header information, it is possible to infer the number of bits allocated to the different code blocks in the wavelet domain at low bit rates with minimal processing — only the packet headers need to be read; no decoding of wavelet coefficients is required.

The packet headers contain information such as number of bits allocated to each code block. For some J2K files, the bit allocation at low bit rates can also be extracted from the packet headers. If the low bit rate information cannot be directly obtained from the headers, then the number of zero bit-planes and coding passes available in the packet headers for each code block could be used to estimate the bit allocation at low bit rates from the bit allocation at high bit rates. For simplicity, we assume that the low bit rate bit allocation is available. One such extracted multiresolution bit allocation for the 512×512 *Lena* image is illustrated in Figure 3.

[1] For a Haar wavelet system, the reconstruction is exact.

From the extracted bit allocation, we can estimate the MRLI cost for any image display choice. One obstacle in estimating the MRLI cost is that while the headers provide the total bits spent on each code block, we need the fraction of bits spent of individual wavelet coefficients to determine the MRLI using (4). To resolve this lack of resolution, we assume that each wavelet coefficient within a code block equally shares the total bits allocated by the J2K coder to the code block. For example, if $\mathcal{C}$ denotes some 32×32-sized code block that is allocated $B(\mathcal{C})$ bits, then we estimate $B(d_{j,k,l}) = \frac{B(\mathcal{C})}{32\times 32}$ for all $d_{j,k,l} \in \mathcal{C}$. The MRLI cost can now easily be calculated using (4) with weights γ and δ chosen appropriately.

4.2. SmartNail algorithm for J2K-encoded images

The algorithm to construct the SmartNail given a J2K-encoded image, and a specified display area of size $X \times Y$ pixels is as follows:

1. Extract the total bits allocated to individual code blocks from the J2K packet headers.
2. Compute the fraction of bits allocated to each wavelet coefficient from the total bit allocation of Step 1.
3. Scale the inferred bit allocation with weights γ and δ.
4. Search for the scale and anchor location of the SmartNail that maximizes the MRLI cost function.
5. Extract and decode only the packets containing coefficients required to reconstruct the chosen SmartNail.

Steps 1 through 3 access only the header information from the J2K image. Since the header constitutes a small fraction (less than 2 %) of the J2K file, these steps can be implemented very efficiently with little memory. Step 4 can be implemented in linear time using convolution of the weighted bit allocation with a $X \times Y$ averaging filter. Finally, Step 5 saves on expensive J2K decoding operations. Thus the SmartNail can be constructed with minimal processing from J2K headers.

5. RESULTS

We illustrate the performance of our SmartNail algorithm described in Section 4.2 using the 512×512-pixel images *Lena* and *Java* shown in Figure 4. Images encoded with the J2K encoder operating in the range 0.1–0.75 bits per pixel yield similar SmartNails. The results in this paper are obtained by encoding the images with a J2K encoder at 0.2 bits per pixel using 32×32-sized code blocks with five decomposition levels of the 5-3 reversible wavelet transform. We compute SmartNails of sizes 192×192 and 128×192 pixels from each image. The desired display areas are chosen to be multiples of the code-block size for simplicity. To incorporate human tendency to center the important objects in a photograph, the weights γ and δ in the MRLI cost function are chosen to linearly decrease from 1 to 0.77 toward the edges of the image at all scales.

As shown in Figure 4, for the 192×192 display frame, the SmartNail algorithm scaled both the *Lena* and *Java* images by a factor of two along each direction, and selected the lower and upper left regions respectively. For the 128×192 display frame, the SmartNail algorithm selected the lower-central and upper-central pixels respectively from both the twice down-scaled images. For both images, only 16.7% and 13.2% of all wavelet coefficients had to be decoded for reconstructing the 192×192 and 128×192 SmartNails respectively. In contrast to the *Lena* and *Java* thumbnails, which are obtained by uniformly down-scaling the respective images to fit the given display areas, the SmartNails provide more recognizable image representations. For example, the text in the Java SmartNail 2, is more readable compared to the Java thumbnail 2.

6. CONCLUSIONS

We have proposed a novel framework to represent images within a smaller display frame using a SmartNail — an adaptive representation created by scaling and cropping the original image. We obtained the SmartNail-characterizing parameters by maximizing a bit-allocation-based cost function that quantifies the visual importance of the information contained within the representation. For J2K-encoded images, the SmartNail is characterized using just the J2K header information, and constructed with minimal decoding of wavelet coefficients from the code stream. Though we have described and demonstrated the SmartNail problem and solution for rectangular displays and square images, the approach trivially generalizes to displays and images with arbitrary shapes.

We are currently refining the algorithm by improving the estimation of the number of bits allocated to each individual wavelet coefficient from the total bits allocated to a code block. This would improve the MRLI cost function, thereby yielding better SmartNails. Further, we are also refining the estimation of the MRLI cost function at low bit rates from a given a high-bit-rate J2K-encoded image. This will enable SmartNail construction from all types of J2K-encoded images.

With standardization, J2K is poised to become the image representation format of choice for the internet, in medical imaging, and in digital cameras. With the number of digital images ever-increasing, fast header-based algorithms such as our SmartNail algorithm will become extremely desirable. In addition to adaptive image representations, we envision that the framework introduced in this paper will also enable other desirable multiresolution image processing operations for JPEG 2000 images such as header-based segmentation and feature extraction.

ACKNOWLEDGEMENTS

We thank Edward Schwartz, Michael Gormish, and Martin Boliek for many productive discussions and help with experiments.

7. REFERENCES

[1] A. Woodruff, A. Faulring, R. Rosenholtz, J. Morrison, and P. Pirolli, “Using Thumbnails to Search the Web,” in *Proc. of SIGCHI’01*, 2001.

[2] K. Lee, H.S. Chang, H. Choi, and S. Sull, “Perception-based Image Transcoding for Universal Multimedia Access,” in *Proc. IEEE Int. Conf. Image Processing – ICIP ’01*, 2001, vol. 2, pp. 475–478.

[3] ITU-T Rec. T.800—ISO/IEC 15444-1:2000, *Information Technology – JPEG2000 Image Coding System.*

[4] R. de Queiroz and R. Eschbach, “Fast Segmentation of the JPEG Compressed Documents,” *Electronic Imaging*, vol. 7, no. 2, pp. 367–377, April 1998.

[5] C. S. Burrus, R. A. Gopinath, and H. Guo, *Introduction to Wavelets and Wavelet Transforms: A Primer*, Prentice-Hall, 1998.

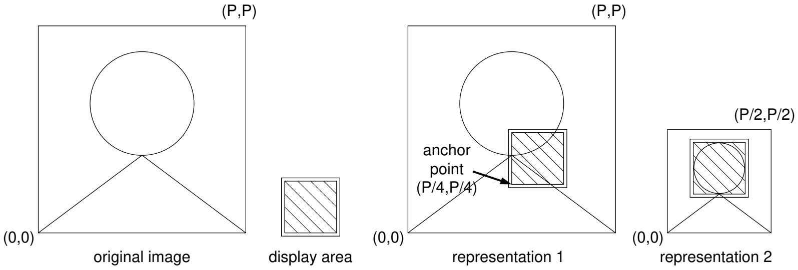

Fig. 1. *SmartNail problem. From the many possible representations, two examples with representation = hashed area are given to fit the original image within the illustrated display area. Representation 1 suggests a cropped window in the original resolution image, while representation 2 suggests a cropped window in the original image scaled down by a factor of two in each direction. The ideal representation is the one that conveys the maximum visual information.*

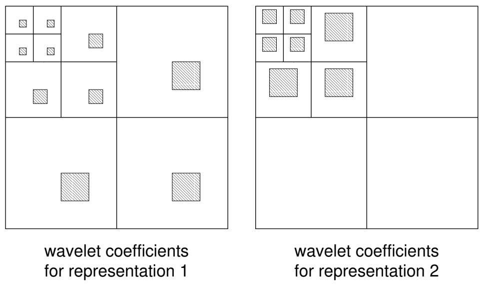

Fig. 2. *The hashed regions mark the wavelet coefficients required to reconstruct the respective representations shown in Figure 1.*

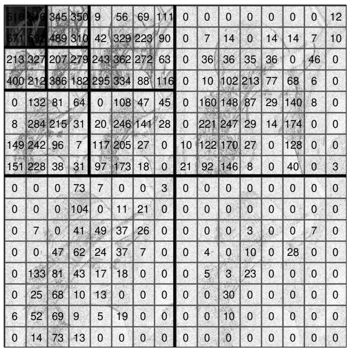

Fig. 3. *Lena's bit allocation extracted from the J2K header: The numbers shown are the bits allocated by J2K at* 0.5 *bits per pixel using three levels of decomposition to the respective code blocks (separated by thin gray lines). The gray underlying patterns denote the wavelet coefficients of Lena, with the thick black lines separating the wavelet sub-bands. The bit allocations of J2K provide a good measure for the visual importance of the different features at their relevant resolutions. We can see that J2K allocates many bits to the fine-scale features such as the feathers in the finest scale. The face region receives more bits in the intermediate scale, while the smooth, unimportant background receives few bits.*

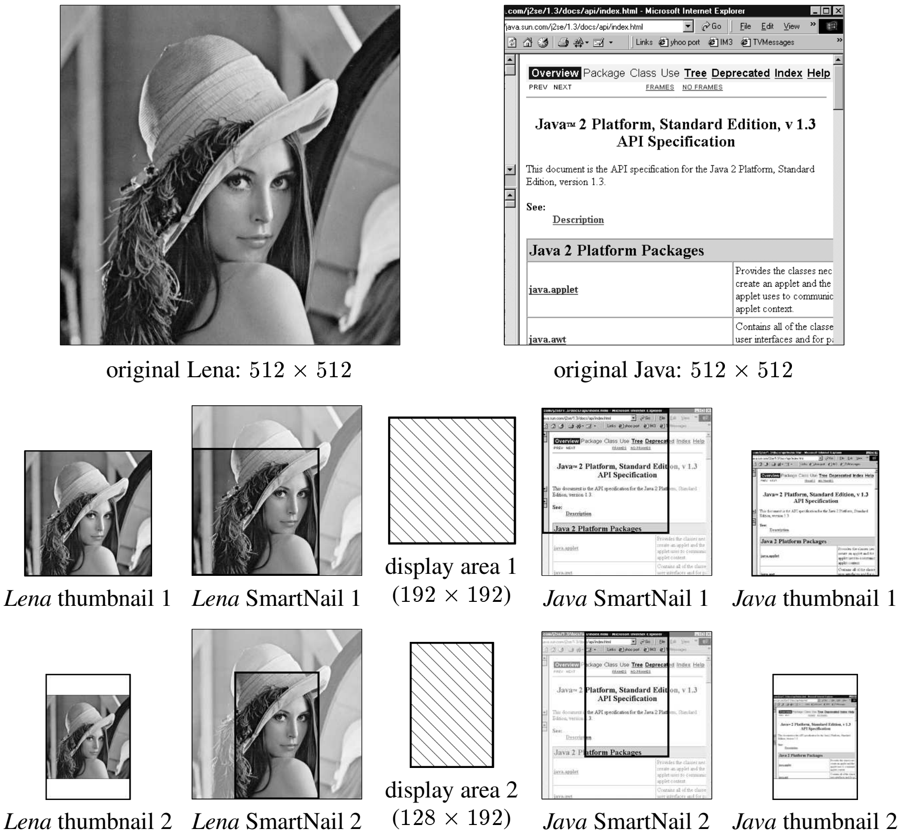

Fig. 4. *Results: The Lena and Java image (both* 512×512 *pixels) need to be represented within two different display areas of sizes* 192×192 *and* 128×192 *pixels respectively. For both images, the coarsest and finest scales are* $J_0 = 4$ *and* $J_1 = log_2(512) - 1 = 8$ *respectively. The respective SmartNails are displayed with greater contrast and are bounded by thick lines. The Lena* 192×192 *SmartNail is defined by cropping a display-sized window located at* $A^* = (0, 0)$ *in the twice down-scaled Lena image (wavelet scale* $J^* = J_1 - 1 = 7$*). The scale and anchor location defining the Lena* 128×192 *SmartNail are* $\{J^*, A^*\} = \{7, (64, 0)\}$*. On the other hand, the Java* 192×192 *and* 128×192 *SmartNails are defined by* $\{J^*, A^*\} = \{7, (0, 64)\}$ *and* $\{J^*, A^*\} = \{7, (64, 64)\}$ *respectively. In contrast to the respective thumbnails, the SmartNails provide more-recognizable representations. The relative scaling between all the displayed images is preserved.*