IMPROVED RESOLUTION SCALABILITY FOR BI-LEVEL IMAGE DATA IN JPEG2000

Rahul Raguram, Michael W. Marcellin, and Ali Bilgin
Department of Electrical and Computer Engineering, The University of Arizona
Tucson, Arizona 85721
Email: {rraguram,marcellin,bilgin}@ece.arizona.edu

ABSTRACT

In this paper, we address issues regarding bi-level image compression using JPEG2000. While JPEG2000 is designed to compress both bi-level and continuous tone imagery using a single unified framework, there exist significant limitations with respect to its use in the lossless compression of bi-level imagery. In particular, substantial degradation in image quality at low resolutions severely limits the resolution scalable features of the JPEG2000 code-stream. We analyze these effects and present two efficient methods to improve resolution scalability for bi-level imagery in JPEG2000. It may be noted that both proposed methods are compliant with Part-I of the JPEG2000 standard.

I. INTRODUCTION

Bi-level (or binary) images are often encountered in applications such as document archiving and retrieval, as well as digital libraries and facsimile, where they provide a compact means of representing black-and-white documents containing text and drawings. There exist a number of formats that specifically target the bi-level image compression task, such as the CCITT G3 and G4 fax standards, and the more recent JBIG and JBIG2 standards. The JPEG2000 standard for still image compression is also capable of bi-level image compression; in fact, one of the desired features of the standard was the efficient compression of both bi-level and continuous tone imagery, using a single unified compression architecture. To this end, the compression performance offered by JPEG2000 is very similar to the CCITT G4 standard [1]. It must be noted, however, that there exist certain limitations with regard to using JPEG2000 for the compression of bi-level imagery. These limitations severely restrict the use of the resolution scalable features of the JPEG2000 code-stream.

Scalability is one of the central concepts of the JPEG2000 paradigm [1][2]. The JPEG2000 codec is transform based, and resolution scalability is a direct consequence of the multi-resolution properties of the Discrete Wavelet Transform (DWT). A code-stream is said to be resolution scalable if it contains identifiable subsets that represent successively lower resolution versions of the original image. Since bi-level images are invariably digitized at high resolutions, this property of the code-stream is potentially very useful. Consider the case where high resolution images are being viewed by a user over a network. Typically, the image at full resolution will be too large to display on the user's monitor. By making use of the inherent scalability of the JPEG2000 code-stream, it is possible to stream only the relevant portions of the image to the client. This allows JPEG2000 content to be delivered in a manner which matches the user's display resolution.

2007 Data Compression Conference (DCC'07)
0-7695-2791-4/07 $20.00 © 2007 IEEE

However, for bi-level imagery, the visual quality at lower resolutions can be too poor to be of any practical use. In the following sections, we analyze the issues concerning bi-level image compression in JPEG2000 and identify two methods which may be used to improve image quality at low resolutions, thereby enabling efficient resolution scalable delivery of compressed bi-level imagery. It may be noted that both of these methods maintain JPEG2000 Part-I compliance [3]. To our knowledge, these are the first schemes that seek to optimize the JPEG2000 codec for bi-level imagery, while doing so in a Part-I compliant fashion.

II. Bi-Level Image Compression in JPEG2000

JPEG2000 can be used to efficiently code bi-level imagery, subject to suitable choices of the coding parameters. One commonly used rule of thumb while compressing bi-level imagery is the use of zero levels of DWT, in order to maximize raw coding efficiency. In this case, the JPEG2000 block coder codes the binary valued image data using a single coding pass. However, while this results in good coding performance, resolution scalability is sacrificed since there is no multi-resolution hierarchy.

In order to introduce resolution scalability, one or more levels of wavelet transform may be applied. Part-I of the JPEG2000 standard allows two wavelet transforms, the 5/3 and 9/7, corresponding to lossless and lossy image compression, respectively. Due to practical and cultural considerations, lossless compression of bi-level imagery is of interest to the library and archive communities. Thus, we consider only the 5/3 transform here. Introduction of the 5/3 transform for bi-level imagery degrades lossless compression performance, since it is designed primarily for the efficient compression of continuous tone imagery. Table I reports the degradation in compression efficiency as resolution scalability is introduced.

TABLE I

JPEG2000 compression ratios for bi-level images with varying number of transform levels.

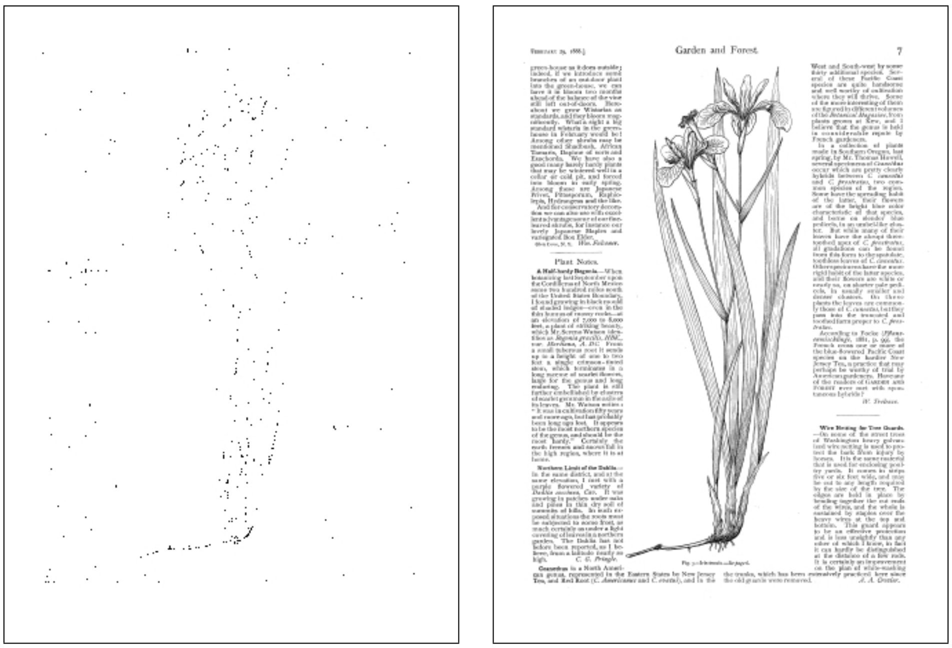
Image	Number of resolution levels				
	0	1	2	3	4
garden2 (5088x7216)	15.2	9.1	7.6	7.2	7.1
garden3 (5088x7216)	4.0	2.7	2.6	2.5	2.5
000012 (7344x5388)	28.4	19.5	17.4	16.8	16.7
000014 (5728x7500)	7.1	5.0	4.6	4.5	4.5
000015 (11056x7492)	6.5	4.5	4.0	4.0	4.0

In a remote browsing application, this loss in compression performance may be offset by the resolution scalable properties of the code-stream. In particular, although the compressed file size is larger, the client can now directly access only the intermediate resolution desired, which may effectively result in less data being transferred. Consider a high resolution bi-level image compressed with zero levels of transform. Even if the client desires only a low resolution version of this image, there are no lower resolutions directly available; thus, the image at full resolution must be transmitted to the client, where it may then be downsampled. In contrast, when the same high resolution image is compressed using multiple levels of transform, only the data for the resolution required by the client needs to be transferred, leading to more efficient transmission.

While the above discussion might indicate the desirability of resolution scalable delivery of bi-level images, there exists a significant obstacle to the use of JPEG2000

2007 Data Compression Conference (DCC'07)
0-7695-2791-4/07 $20.00 © 2007 IEEE

for this purpose. Specifically, use of the 5/3 transform results in rapid degradation of image quality at decreasing resolutions. This effect is shown in Figure 1(a). The high resolution ‘garden2’ image (5088x7216) was compressed using 4 levels of transform, and the lowest resolution level, corresponding to 1/16th the original resolution, was obtained by decompressing the relevant portions of the code-stream. For comparison, Figure 1(b) shows an image at similar resolution, obtained by applying a low-pass averaging filter to the original high resolution image, followed by downsampling. It can be seen that the JPEG2000 image has lost all detail and is unrecognizable. This drastic loss in visual quality poses a serious obstacle to the resolution scalable transmission of bi-level imagery.



(a) Lowest resolution level from JPEG2000. (b) Subsampled version of original image.

Fig. 1. Poor visual quality of JPEG2000 compressed bi-level imagery.

Resolution reduction schemes in standards such as JBIG are carefully matched to the bi-level image compression task. For instance, JBIG uses a template-based resolution reduction scheme, involving tables defining exception rules that aim to preserve edges and lines, as well as periodic and dither patterns [4]. Wavelet transforms can be inherently ill-suited for this task, since requirements such as smoothness and vanishing moments, which are considered to be desirable in a constructed wavelet basis, may not be relevant when applied to bi-level imagery. Furthermore, the rounding steps that are introduced in the 5/3 transform to ensure reversibility can cause significant damage to bi-level images during the encoding process.

In the following section, we present two schemes that seek to overcome the above drawbacks in a JPEG2000 Part-I compliant fashion.

III. Improving Resolution Scalability

1) Method 1: Observations Based on Rounding: In JPEG2000, both the reversible and irreversible transforms can be implemented using a common lifting framework [5]. In a

2007 Data Compression Conference (DCC'07)
0-7695-2791-4/07 $20.00 © 2007 IEEE

broad sense, lifting provides a means to generate invertible mappings between sequences of numbers, and the invertibility is unaffected even when arbitrary operators, which may be linear or non-linear, are introduced in the lifting steps. This flexibility allows the use of non-linear rounding operations in the lifting steps, in order to ensure that the transform coefficients are integers. The analysis equations for the reversible 5/3 transform, corresponding to the lifting realization, are presented below. We denote the input signal, low-pass subband signal, and high-pass subband signal by $x[n]$, $s[n]$ and $d[n]$, respectively. We also define $x_0[n]$=$x[2n]$ and $x_1[n]$=$x[2n+1]$, to represent the even and odd indexed samples of the input signal, respectively. We then have

$$d[n] \quad = \quad x_1[n] - \left\lfloor \frac{1}{2}(x_0[n] + x_0[n+1]) \right\rfloor \tag{1}$$

and

$$s[n] \quad = \quad x_0[n] + \left\lfloor \frac{1}{4}(d[n] + d[n-1]) + \frac{1}{2} \right\rfloor . \tag{2}$$

Lifting may be viewed as comprising three basic stages - split, predict, and update. In the split step, the input sequence is decomposed into its even and odd components, $x_0[n]$ and $x_1[n]$. In the next stage, the odd indexed coefficients are predicted using a combination of the neighboring even indexed coefficients. If $x[n]$ is smooth, then the predicted values will be close to the actual values; thus, a more compact representation may be obtained by replacing $x_1[n]$ by the prediction residual, $d[n]$. This sequence may be thought of as representing the extent to which the original signal fails to be linear. In terms of frequency content, these coefficients capture the high frequencies present in the original signal. In the update step, the even indexed coefficients are transformed into a low pass sequence $s[n]$, by updating $x_0[n]$ with a combination of the prediction residuals.

In view of the above interpretation, we note from equation (1) that $x_1[n]$ is predicted to be the average of its two neighboring samples. While the rounding operation in this step does not drastically affect grayscale imagery, it plays a much more pivotal role in the case of bi-level imagery, since it now involves making decisions between two extremes - black and white - as opposed to decisions between two neighboring gray levels.

It may be noted that that while pixels in a bi-level image take on one of two values, 0 or 1, the interpretation of these values as black or white is left to the application. One common interpretation, in analogy with grayscale imagery, is to assign black the lowest value, or 0, and white the highest, or 1. Alternatively, black foreground pixels could be assigned a value of 1 and white background pixels a value of 0. While examples of either convention can be found in the literature, we present an argument for using the latter interpretation when coding bi-level imagery using JPEG2000.

From equation (1), it may be seen that when one of the two even indexed samples $x_0[n]$ or $x_0[n+1]$ is 0 and the other 1, there exists an ambiguous case. Since the lifting steps in the JPEG2000 standard employ a floor operator, we observe that for the ambiguous case, the predicted pixel value is always 0. For many instances of commonly occurring bi-level imagery, white pixels occur far more frequently than black pixels. Consequently, in the case where white pixels are assigned a value of 1, the value predicted in the ambiguous case will often be incorrect. One such instance is illustrated in Figure 2, which shows a lifting step applied in the vertical direction along the third column. The two odd indexed samples $x_1[n-1]$ and $x_1[n]$ are predicted to be the average of their neighboring even

2007 Data Compression Conference (DCC'07)
0-7695-2791-4/07 $20.00 © 2007 IEEE

indexed samples. It may be observed that both these pixels are incorrectly predicted to be 0, or black. Furthermore, since bi-level images invariably have a large number of edges, the ambiguous case is encountered often. When there are two successive errors in prediction, the prediction residuals $d[n-1]$ and $d[n]$ in equation (2) cause single pixel wide black lines to be 'washed out' at lower resolutions.

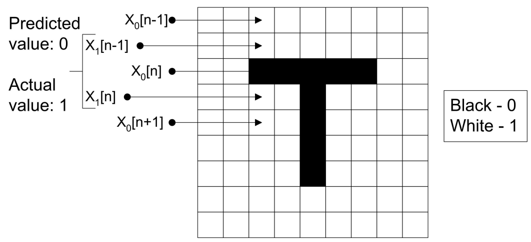


Fig. 2. Incorrectly predicted pixels in the lifting step.

One solution to this problem would be to use the alternate interpretation, with 0 for white and 1 for black. This serves to bias the prediction in favor of the more commonly occurring pixel value, which is now 0. It may be noted that for the former interpretation, the identical effect could be achieved by replacing the floor operator in equation (1) with a ceiling operator. Since modifying the nature of the lifting steps would make the encoder non-compliant with the standard, use of the alternate interpretation is preferable since it maintains JPEG2000 Part-I compliance.

The results of this strategy are shown in Figure 3. It may be seen that much more detail is retained, particularly in the textual regions. As noted in [1] (Section 16.3), the use of 0

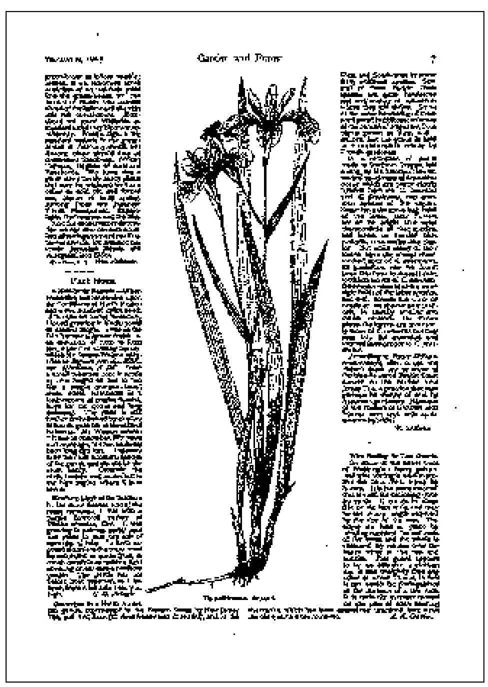


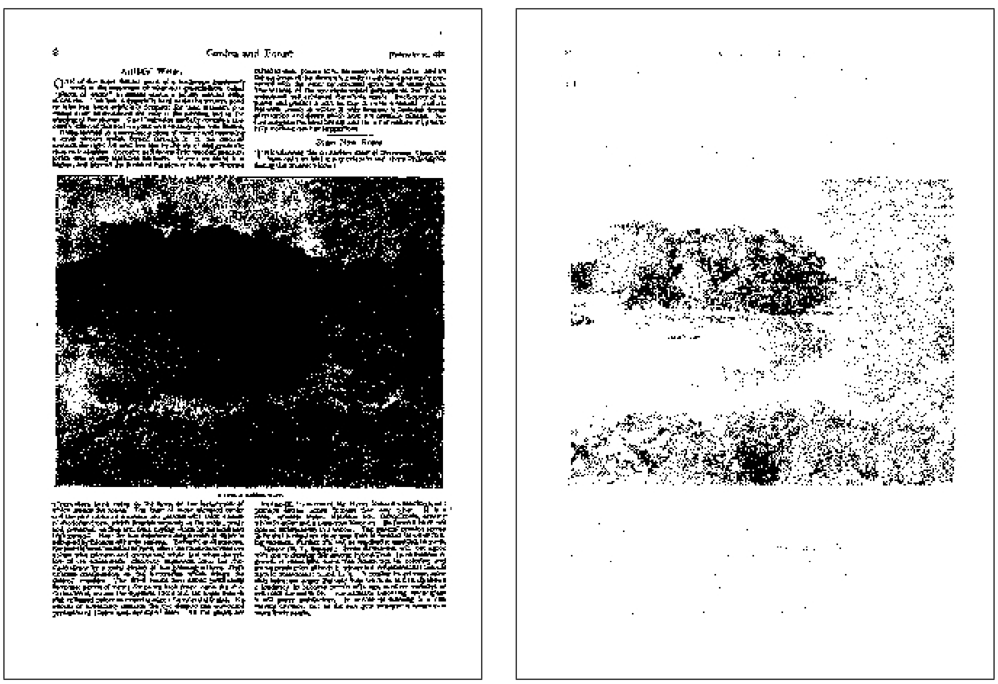
Fig. 3. Improved visual quality by assigning white - 0, black - 1.

for white pixels and 1 for black pixels results in a loss in compression performance. This is because the procedure tends to decrease the length and frequency of insignificance

2007 Data Compression Conference (DCC'07)
0-7695-2791-4/07 $20.00 © 2007 IEEE

runs and consequently, the efficiency of the significance coding primitive's run mode. Even though lossless compression rates are decreased, the visual quality of the bi-level image improves considerably.

It must also be noted that the above discussion makes the assumption that bi-level images are predominantly white, with fewer black foreground pixels. This is indeed true for a large class of bi-level imagery. In cases where this assumption is not true, it may be preferable to use 1 for white pixels, and 0 for black. This may be observed in Figure 4, which compares the two cases. Figure 4(a) shows an image compressed using 0 for white and 1 for black, while Figure 4(b) uses the opposite assignment. While Figure 4(a) retains more detail in the textual region, the halftone region appears 'blacked out'. In contrast, it may be seen from Figure 4(b) that the halftone region retains slightly more detail. Thus, a more effective strategy would be to design a scheme that adapts to the local nature of the bi-level image and assigns pixel values of 0 or 1 accordingly. We note that JPEG2000 Part 6 may be of use in this regard [6].



(a) White - 0, Black - 1.

(b) White - 1, Black - 0.

Fig. 4. Comparison of the two assignment policies.

2) Method 2: The JPIP Protocol: While the JPEG2000 standard offers many features that support interactive access of compressed imagery, Part-I of the standard describes only the core coding system and syntax for the code-stream. While it is indeed possible for a client to interact remotely with image content by intelligently accessing appropriate byte ranges from the compressed file, the JPIP protocol seeks to standardize client/server interaction in an efficient and intelligent manner.

Using the JPIP protocol, the interaction between client and server is carried out through requests made by the client, which identify the current focus window of the client-side application. Rather than describing the focus window in terms of low-level code-stream

2007 Data Compression Conference (DCC'07)
0-7695-2791-4/07 $20.00 © 2007 IEEE

constructs, these requests contain information regarding the client's spatial region of interest, resolution and image components of interest. The description of the focus window in terms of its geometric attributes is much more intuitive, and the server receives a representation of the end-user's ultimate interests, rather than a client's translation of those interests into JPEG2000 code-stream elements [7].

JPIP requests are composed of a sequence of 'name=value' pairs. A basic JPIP request typically contains the name of the target file, together with a description of the focus window. When transmitted over a text-based transport protocol such as HTTP, the name and value fields are ASCII strings, with requests being separated by the '&' character. For instance, the following request refers to a file called 'images/garden2.jp2', at a resolution whose full size ($fsiz$) is 2544x3608 (columns x rows), which corresponds to half the resolution of the original image. The request is for a square region of 600x600 pixels ($rsiz$), which is located at an offset of 1000 pixels from the left and 1200 pixels from the top of the image ($roff$).

```
target=images/garden2.jp2%fsiz=2544,3608&rsiz=600,600&roff=1000,1200
```

An important feature of the JPIP protocol is that image decompression and rendering are separated from client-server communication. JPIP specifies a means of interacting with JPEG2000 data, and mechanisms for communicating compressed image data and metadata between a client and server. However, JPIP does not specify how the client application should process/display this transmitted data. This enables us to build a client-side application that intelligently generates requests for appropriate windows of interest and then processes the data received in order to improve the quality of the low resolution bi-level imagery. Details of the post-processing operations thus are abstracted from the actual client-server communication process.

For our experiments, we used Kakadu v5.1 [8], which includes a JPIP client/server implementation compliant with the ISO JPIP final committee draft. The Kakadu image viewer initially attempts to render an image at a resolution that matches the display resolution of the user's terminal. The user can then pan and zoom as desired, and the image is re-rendered based on the data available in the cache along with further data received from the server, based on the new focus window parameters. It may be noted that in the case where the image is compressed with zero levels of transform, the image is displayed at full resolution. Even if the user desires only a low resolution image, the image at full resolution must be transferred to the client where it may then be downsampled for display purposes. This is a significant drawback for the interactive access of compressed bi-level imagery.

In order to design an improved scheme, we note the following points:

- If an image is compressed using D levels of transform, it is possible to request the image at any of the $D+1$ available resolutions. We refer to the low-pass subband LL_{D-r} as the r^{th} resolution of the image, where $r=0$ corresponds to the lowest available resolution and $r=D$ corresponds to the original image resolution. As seen in Section II, for bi-level imagery, the quality of the low resolution decompressed images can be far too poor to be of any practical use. Our experiments indicate that image quality is usually acceptable for $D-2 \leq r \leq D$, but deteriorates rapidly when lower resolutions are viewed.
- Research on human perception has shown that if text is generated on a display with grayscale capability, then visual clarity may be improved by using shades of gray in

rendered text [4]. The use of grayscale values forms an additional cue to the human visual system, which uses this extra information to compensate for the inaccurate visual data due to low resolution. In other words, the visual quality of a subsampled bi-level image may be enhanced by applying a suitable low-pass filter, retaining the intermediate gray-levels produced, and then downsampling by the required amount. This procedure is sometimes referred to as *scaling to gray*.

- Using the resolution scalable features of the JPEG2000 code-stream, we do not need to transfer the image at full resolution in order to scale down. For instance, to view the image at the r^{th} resolution, we can request the server to send data corresponding to any resolution r', where $r < r' \leq D$, and then filter and downsample the image to the required resolution. Our experiments show that in order to obtain good image quality when viewing resolution r, where $0 \leq r < D - 2$, we can request data corresponding to resolution $D-2$, and then scale-to-gray by the appropriate amount.
- Compressing bi-level imagery using the wavelet transform results in an increase in file size over the zero level case (Refer to Table I). However, since we can access intermediate resolution levels, we may transfer smaller subsets of the compressed file. In order for the proposed scheme to be attractive, it must improve the quality of the bi-level image substantially while at the same time, the amount of data transferred must be significantly less compared to transferring the image at full resolution.

The proposed algorithm is formalized below. The block diagram of the scheme is shown in Figure 6.

```
While session is active
    Determine the client's window of interest in terms of focus box parameters. These
    parameters define the resolution of interest r, and spatial region of interest.
    if r ≥ D − 2
            Do not modify focus window parameters
    else if r < D − 2
            Modify focus window parameters: (Refer to Figure 7)
                Scale the dimensions of the spatial region of interest
                    rsiz' = rsiz * 2^(D−2−r)
                    roff' = roff * 2^(D−2−r)
                Modify the resolution level parameters
                    fsiz' = fsiz * 2^(D−2−r)
            Filter the received data using appropriate low-pass filters.
            Downsample by a factor of 2^(D−2−r)
    end if
    Display
End while
```

Fig. 5. Algorithm for Method 2.

It may be seen from Figure 6 that the JPIP client is unaware of the post-processing operations performed. In particular, the JPIP client knows only that data for resolution $D-2$ is being served. The task of modifying focus window parameters and downsampling as required is carried out by the application.

The results obtained using the proposed method are shown in Figure 8. As can be seen, there is a marked improvement in the visual quality of the low-resolution decompressed

2007 Data Compression Conference (DCC'07)
0-7695-2791-4/07 $20.00 © 2007 IEEE

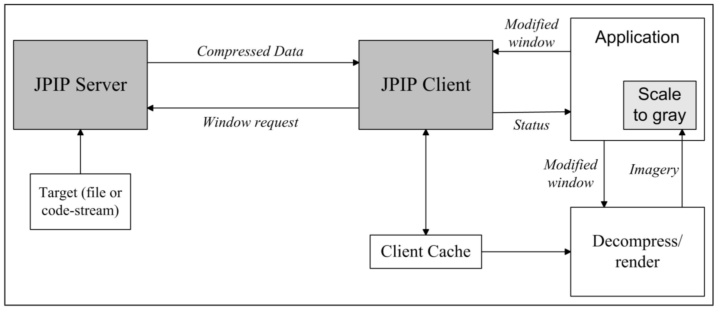


Fig. 6. Block diagram for Method 2.

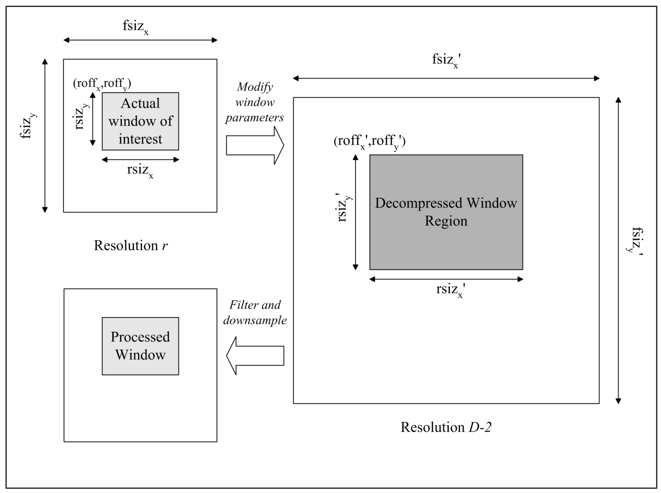


Fig. 7. Modifying the parameters of the window of interest.

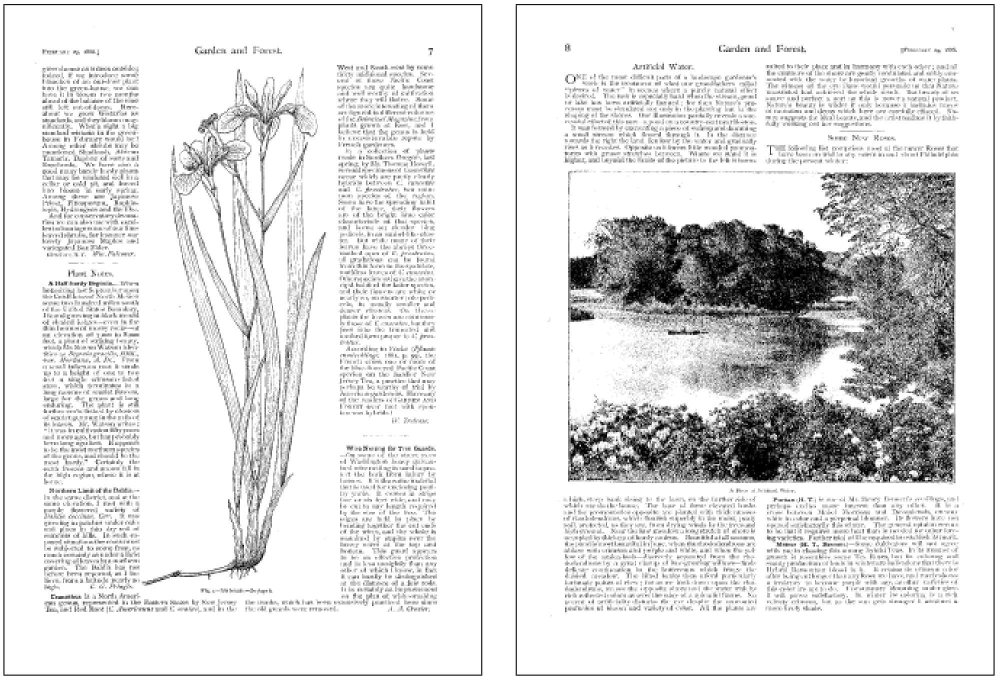
images, as compared to Figures 1(a) and 3. It may also be noted that since this method does not make any assumptions regarding the nature of the bi-level image, it produces excellent quality for both the text and image regions of a compound bi-level image. Our experiments indicate that a simple averaging filter produces results of good quality. If desired, more sophisticated low-pass filters that possess good properties for downsampling, may be used. One such filter, commonly used in the graphics community, is the Lanczos filter [9], which produces slightly smoother images.

Table II shows the data savings for the above method. Bi-level images were compressed using 4 levels of wavelet transform, and the amount of data transfer required for the proposed scheme is listed in the third column. For comparison, the second column lists the amount of data required to transfer the image at full resolution D, and then downsample. It may be observed that the proposed scheme needs, on average, 70% less data in order to achieve comparable image quality. Thus, the method achieves excellent visual quality, and does so in a highly data-efficient manner.

IV. Conclusion

This paper introduces two efficient schemes in order to improve resolution scalability for bi-level imagery in JPEG2000. The first method suggests the use of a particular black/white assignment policy in order to improve the quality of the low-resolution image, and works well for certain commonly occurring types of bi-level imagery. The second

2007 Data Compression Conference (DCC'07)
0-7695-2791-4/07 $20.00 © 2007 IEEE



7 Garden and Forest.

8 Garden and Forest.

Artificial Water.

Fig. 8. Improved low-resolution images obtained using Method 2.

TABLE II

Relative savings in data transferred for the proposed scheme.

Image	Amount of data transferred for resolution D (KB)	Amount of data transferred for the proposed scheme (KB)	Percentage savings
garden2 (5088x7216)	294.17	100.07	65.98
garden3 (5088x7216)	1115.3	181.05	83.77
000012 (7344x5388)	125.41	35.04	72.05
000014 (5728x7500)	411.91	105.31	74.43
000015 (11056x7492)	830.04	212.56	74.39

approach employs the JPIP protocol, and produces images that are comparable to those that result from downsampling the full resolution image, but requires only 30% of the data. Both schemes can be implemented in a fully JPEG2000 Part-I compliant fashion.

REFERENCES

[1] D. S. Taubman and M. W. Marcellin, *JPEG2000: Image Compression Fundamentals, Standards, and Practice*, Boston: Kluwer Academic Publishers, 2002.

[2] D. S. Taubman and M. W. Marcellin, "JPEG2000: Standard for Interactive Imaging," *Proceedings of the IEEE*, 90(8), pp. 1336-1357, 2002.

[3] *Information Technology-JPEG2000-Image Coding System-Part 1: Core Coding System*, ISO/IEC 15444-1, 2000.

[4] I. H. Witten, A. Moffat and T. C. Bell, *Managing Gigabytes*, Morgan Kaufmann, 1999.

[5] W. Sweldens, "The lifting scheme: A new philosophy in biorthogonal wavelet constructions," *Wavelet Applications in Signal and Image Processing III*, A. F. Laine and M. Unser, eds. Proc. SPIE 2569, pp. 68-79, 1995.

[6] *Information Technology-JPEG2000-Image Coding System-Part 6: Compound Image File Format*, ISO/IEC 15444-6, 2003.

[7] D. Taubman, and R. Prandolini, "Architecture, Philosophy and Performance of JPIP: Internet Protocol Standard for JPEG2000," *International Symposium on Visual Communications and Image Processing*, 2003

[8] Available online: www.kakadusoftware.com.

[9] C. E. Duchon, "Lanczos Filtering in One and Two Dimensions," *Journal of Applied Meteorology*, Vol. 18, pp. 1016-1022, Aug. 1979.