A Region-of-Interest Method for Texturally-Rich Document Image Coding

X-W. Yin, A. C. Downton, M. Fleury, and J. He
Department of Electronic Systems Engineering,
University of Essex, United Kingdom
Tel. +44 (0)1206 872817
Fax. +44 (0)1206 872900
fleum@essex.ac.uk

***Abstract* – Region-of-Interest (ROI) techniques are often utilized to improve coding for detailed regions in natural still-image coding standards such as JPEG2000 [1], but no specific method is stated for determining the ROI map. In this paper, an ROI-based method, in which rectangular regions are extracted using document image analysis (DIA), is proposed specifically for document image coding. These rectangular regions can be efficiently coded using wavelets, and DIA may also be used to distinguish between important and unwanted foreground regions, allowing further coding gains (as illustrated in one of the example documents in the paper). Compared to multi-layer methods currently used for document image coding [2], the method is simpler and scalable, while improving visual quality and the Peak-Signal-to-Noise Ratio (PSNR).**

***EDICS category* – 2.IMMD--Image and Multidimensional Signal Processing**

I. INTRODUCTION

Historically, the most common format for document images has been binary for reasons of efficient storage, leading to the development of binary document image coding standards such as JBIG1 and JBIG2 [3], based upon run-length coding techniques. However, as demand for higher image quality has grown and the range of digitized documents increased, gray-scale and color document image representations have become common, although these increase storage space and/or transmission time. Hence, it is now essential to design document image coding algorithms that can compactly represent texturally-rich document images, which are increasingly being made accessible through online document archives [4] (see [5] for an example used in this paper).

The main contribution of this paper is to point out the advantages of using rectangular ROI-based compression on document images, in terms of algorithmic simplicity and accuracy of representation. By way of illustration, a case study showing extraction of ROIs for differing types of textual regions is demonstrated, using a low-complexity wavelet codec [6] adapted for ROI extraction. The paper makes no special claims for the DIA techniques applied, which would typically be adapted to the type of document archive being coded. However, the paper *does* demonstrate that the rectangular-ROI technique, combined with bit-plane shifted wavelet coding, is very competitive in both objective and subjective visual quality with current multi-layer document coding techniques [2]; it is also considerably more computationally efficient.

The texturally-rich examples in this paper include documents with regions of printed, typed or handwritten text, and line and ink drawings in gray-scale or color. High resolution is always required to display this foreground information. The residual can be regarded as background, which, while still important to show context (such as ageing, discoloring, texture or other attributes of the paper surface), can be displayed later and at lower resolution than the foreground. Current color document image compression algorithms use segmentation-based multi-layer methods to meet the differing requirements of foreground and background document image coding. One example, DjVu [2], defines three layers: mask layer, foreground layer, and background layer. The mask layer specifies the shape of text and lines, distinguishing which pixels should be coded using the foreground or background coding algorithms. The DjVu foreground layer defines the color detail within the mask layer, while background texture outside the mask is coded using wavelet-coding techniques. The independent coding of foreground and background layers allows high subjective document image quality to be maintained by prioritizing the bit-budget towards the foreground layer. However, multi-layer methods require a segmentation algorithm as a preprocessing stage. Furthermore, different coding methods are subsequently applied to each of the three layers, effectively coding each image three times. This increases the total complexity and coding delay, and constrains potential application areas. It also introduces redundant data, leading to sub-optimal objective performance. Our experimental work [7] also shows that the mask layer typically occupies at least half of the total file size, limiting scalability.

In this paper, an alternative ROI-based method is proposed that aims to overcome these disadvantages. The method's three stages are: (1) Detect ROIs in the original document (2) After wavelet transform, map the ROIs from the spatial to the wavelet transform domain; and (3) Compress using a wavelet coefficient encoder, in the course of which ROI bit-plane shifting takes place to prioritize the bit budget towards the ROI. A rectangular text block detection process is used in stage 1 to segment the ROI, instead of the foreground segmentation algorithm used in multi-layer methods. As a result, the proposed coding method is simple, can include both `lossy' and near lossless representations in a single stream, and supports both Signal-to-Noise Ratio (SNR) and resolution scalability with comparable visual quality at equivalent scale, all by virtue of using JPEG2000 or an equivalent wavelet image coding.

The main objective of the paper is to demonstrate the overall advantages of using rectangular ROIs for document compression rather than introduce algorithmic novelties, though a number of consequential simplifications of existing algorithms are given in Section IV. For simplicity, the Set Partitioning in Hierarchical Trees (SPIHT) [6] wavelet encoder was chosen to illustrate stage (3) of the method. However, the results in Section V compare several bit-plane shifted wavelet encoders with DjVu as an example of the multi-layer method, and show a general advantage for the rectangular ROI-based wavelet coders.

II. TEXT REGION DETECTION

Archive documents are a typical example where the richer representation of a color document image is required to convey not only the textual content of the document (which could be satisfactorily represented in binary or even as encoded characters) but also its feel and context. Figure 1 shows an example document

image from an index card archive of Lepidoptera (butterflies and moths) at the UK Natural History Museum, with superimposed rectangular textual ROI's. The full Lepidoptera archive consists of several hundred thousand cards, now searchable over the Internet [5].

The format of archive index cards consists of several independent blocks of text, and each block contains one or more logically related text fields. Blocks retain a fairly consistent mutual layout over a complete archive, but the layout of text fields within each block is not strictly fixed. Nor are there any tabular guidelines defining fixed block boundaries. The X-Y cuts algorithm [8] is therefore an appropriate segmentation algorithm for this class of document image structure. Pixel smearing [9], with a threshold sufficient to join adjacent text characters but not adjacent horizontal words or vertical lines, is first applied as a pre-processing non-linear low-pass filter to each archive card image. The X-Y cuts algorithm then extracts and stores the contents of each index card into a hierarchical tree structure (the so-called X-Y tree), consisting of text blocks, lines and words. The result of first level block segmentation is shown in Figure 1. Alternative techniques are widely reported in the document analysis literature for segmenting document images with different characteristics, but in general the rectangular structure of text-based documents (even for complex designs such as newspapers, and vertically-written scripts) supports straightforward extraction of rectangular bounding boxes.

In addition to segmentation, DIA typically labels each segmented region (as shown in Figure 1), in this case based upon a template layout pre-registered during system configuration for the specific archive format being processed. This makes it possible to automatically remove any redundant field from the document image if desired (replacing it with average background pixels from the surrounding region). In fact, the only redundant foreground in Figure 1 is the "Original Spelling…" stamp, which is a special case, as it may appear anywhere on the image with any orientation, but with a fixed overall format. As part of our archive processing system [10], a special tool has been developed to remove such stamps, based upon fuzzy matching of global features of the stamp. The features used are relative corner angles, distances and font sizes of the outer boundary of the stamp, and Figure 1(b) shows the rotated block identified for the stamp in Figure 1. In Section V, Table 1 we show the effect on PSNR both of removing and retaining this unwanted foreground region, while Figures 6 (c) and (d) show the subjective results of coding with the unwanted region removed.

III. GENERATING THE REGION OF INTEREST MAP

This Section points out how, because each text ROI is rectangular in shape, the calculation of the ROI map in Stage 2 of the method can be dramatically reduced, adapting an existing algorithm [11] for that purpose. As in a conventional wavelet-decomposition, once the ROI map is generated at the image scale, the identification process is recursively repeated at each lower subband, until the predefined maximum level depth is reached. The exact mechanism for generating the ROI map is related to the wavelet algorithm chosen. The well-known 9/7-tap filter serves as an example to explain how to generate the interest map.

Suppose there is a one dimensional (1-D) ROI (a linear strip of pixels), the sequence $\{X(n) \mid n_0 \le n \le n_1\}$, where n_0 and n_1 identify the first and last pixels of the ROI. After the 9/7 tap wavelet transform, the low- and high-frequency wavelet coefficient sets, denoted respectively L_n and H_n, that are responsible for the reconstruction of $\{X(n)\}$ are:
if n is even, with $m = n/2$

$$L_n = \{L(m-1), L(m), L(m+1)\}$$
$$H_n = \{H(m-2), H(m-1), H(m), H(m+1)\}$$

else if n is odd, let $m = (n-1)/2$

$$L_n = \{L(m-1), L(m), L(m+1), L(m+2)\}$$
$$H_n = \{H(m-2), H(m-1), H(m), H(m+1), H(m+2)\}$$

Because the version of the 2-D wavelet transform used in this paper is separable, a 2-D mapping can be identified from 1-D mappings in the horizontal and vertical direction, which correspond to L_n and H_n, depending on subband level . For any one pixel in the ROI, let x_{L_n} represent the sets of Cartesian coordinates from the coefficient set L_n (and similarly for x_{H_n}). Then, four displaced rectangular regions can be identified as their direct product coefficient coordinate sets:

$$x_{L_n} \otimes x_{L_n}, x_{L_n} \otimes x_{H_n}, x_{H_n} \otimes x_{L_n}, \text{ and } x_{H_n} \otimes x_{H_n}. \quad (1)$$

Each pixel in an ROI generates four similar sets of wavelet coordinate coefficient sets, and the four contributing subband regions are the union of the corresponding coordinate coefficient sets of all pixels.
Rather than applying (1) to each pixel in the ROI, by determining the wavelet coefficients corresponding to just the top left and bottom right corner pixels of each rectangular ROI [11], the complete set of displaced rectangular regions can be generated in a simplified manner. Let $(x_m, y_n), (x_k, y_l)$ denote respectively the top left and bottom right corners of the text region. Apply (1) to each of these two corners, with $(m^i_{\min}, n^i_{\min})$, i= 0,1,2,3, representing in turn the coordinate of the top-most left coefficient in each of the four subband regions generated by (x_m, y_n), and $(k^i_{\max}, l^i_{\max})$ similarly corresponding in turn to the bottom-most right coefficient of each of the four subband regions generated by (x_k, y_l). Then, for each of the four subband regions, indexed, i= 0,1,2,3, the coordinates correspond to the set

$$\{(x^i_h, y^i_p) \mid m^i_{\min} \le h < k^i_{\max}; n^i_{\min} \le p < l^i_{\max}\} \quad (2)$$

with, in general, $(m^i_{\min}, n^i_{\min})$, and $(k^i_{\max}, l^i_{\max})$ being different for each subband. Figure 2 is an example showing a 3×3 rectangular coefficient region, with (1) performed on the two corner coefficients to generate two groups of four sets in each subband region for one level of subband decomposition. The extremes of these sets identify the desired subband regions, corresponding to the differing values of $(m^i_{\min}, n^i_{\min})$ and

($k^i_{\max}, l^i_{\max}$), i= 0,1,2,3. For the next level of subband decomposition, the ROI in the top left-hand quadrant of Figure 2(d), now in the wavelet domain, is used to generate four sets in the top left-hand quadrant, as in the level-one decomposition. The process continues recursively for any succeeding levels.

IV. REGION OF INTEREST TECHNIQUE

Once the ROI map is generated, in Stage 3 of the method, all the wavelet coefficients within that region need to be included in the compressed bitstream. Several bit-plane shifting algorithms are available [11] for making these coefficients appear as early as possible. ROI algorithm performance is encoder dependent. The algorithm in [12] for the SPIHT encoder [6] avoids bit-plane shifting. If bit-plane shifting were to be used then ROI and non-ROI coefficients would be mixed in the same bit-plane. As a result some correlations within the ROI could not be used to reduce the bit-rate. Therefore, bit-plane shifting is not used in [12]. The research in [12] utilizes: (i) a parent of ROI (PROI) mask; and (ii) the use of per-bit-plane multi-lists, which retain information on tested coefficients that lie outside the ROI, thus preventing information wastage in later encoding rounds.

The work herein uses a similar method to [12], but with some further enhancements, and adaptations for text. For document coding, neither an ROI, nor PROI map is needed, because each ROI is a rectangular box, and can be represented as two corner coordinates (Section III). Instead, a dynamically applied function determines (using the stored corner coordinates) whether the current coefficient or set of coefficients is or contains an ROI coefficient. This modification considerably reduces the memory that would otherwise be needed if ROI and PROI maps were to be used. To avoid excessive modification of the SPIHT algorithm, a single list was used rather than the multi-lists of [12]. This is achieved by simply adding a coefficient bit-plane level indicator, in addition to the coefficient co-ordinates normally held in SPIHT's significance lists. The indicator records the bit-plane level at which a coefficient could become significant during later processing rounds. Neither of these two changes affects the performance of the SPIHT algorithm.

Given a wavelet-transformed image, X, let $n_{\max} = \lfloor \log_2(\max(\{c_{i,j}\})) \rfloor$, $c_{i,j} \in X$. In the same manner as SPIHT, define sets $D(b)$ as all descendants for coefficient b, and $E(b)$ as all descendants except the direct descendants. Also, define three ordered auxiliary lists: (i) LIP to contain insignificant pixels; (ii) LIS to contain insignificant sets; and (iii) LSP to contain significant pixels/coefficients. Entries in LIS can be of type A or B (corresponding to D or E type descendants).

Two parameters, which avoid shifting, are user definable; these are p and r, as shown, together with matching bit-plane thresholds n_{max} to $n_{0,}$ in Figure 3. The first p bitplanes, indexed 0 to p-1, are encoded using SPIHT over the whole image. Subsequently, the ROI only is encoded for r bit-planes, indexed p to p+r-1, and the resulting bits are transmitted. Then, the complete transform image, apart from the ROI, is encoded for the same r bit-planes, reusing significance information from the ROI-only encoding rounds. Finally, the remaining bit planes, indexed p+r to max, are encoded, using SPIHT over the whole image.

For bitplane *q,* when $p \le q < p + r$, a modified SPIHT is applied, in a similar manner to Figure 7 in [12]. However, unlike [12], all decisions about membership of an ROI or PROI are replaced with a `judgment' function, Figure 4. The judgment is made either on a single coefficient in a LIP or LSP, or on coefficient sets of type A or B in a LIS. Initially (in round 0), all coefficient bit-plane indicators are set to zero. In each bitplane round, once a coefficient's significance has been tested, if it is significant then its indicator value will be increased. For example in bitplane *q,* suppose $D(i, j)$ is found to be significant and $Judge\_RoI((i, j), A)$ is true. Then, for each of $c_{2i,2j}, c_{2i+1,2j}, c_{2i,2j+1}, c_{2i+1,2j+1}$, and $E(i,j)$, if it belongs to the ROI map, test its significance, and if significant, increment its significance indicator to $q + 1$. If a coefficient is not in the ROI map there is no need to test significance, and the significance indicator remains as q. Thus, the algorithm can precisely record in which bitplane round each coefficient becomes significant. In fact, significance counting is not confined to the ROI bitplane rounds, but is used for all bitplane rounds. For the remaining full bitplanes after the ROI rounds, special treatment should be accorded to the coefficients or sets of coefficients that indicate possible significance. In bitplane q, only those coefficients having a significance indicator equal to q are tested, because it is already known that other coefficients are not significant in this cycle.

V. EXPERIMENTAL RESULTS

DjVu [2] was compared with the proposed ROI method. ROI's were implemented using IW44, SPIHT [6], and JPEG2000 [1] algorithms. JPEG2000 testing was based on the source code of JJ2000, which is recommended on the JPEG official webpage. IW44 is the wavelet encoder used for DjVu background images. Because the parameter r, used to control the relative importance between foreground and background as explained in Section IV, can range from 0 to r_{max}, it was impractical to test the impact of all possible values. Thus, a middle value of 4, and a high value of 8 were respectively chosen. For SPIHT, a max-shift method [11] was also implemented. The main advantage of the max-shift method is that the interest map doesn't need to be sent to the decoder, but, in this case, the information about ROIs is just a few coordinates. Therefore, no obvious advantages for the max-shift method are shown by the test results. The Peak Signal-to-Noise (PSNR) figures are compared in two ways, first for the whole image, and secondly for the ROI areas only. The PSNR values are an accurate guide to what is otherwise only revealed by closer visual inspection.

The image used as an example of our results was randomly chosen from the card archives held at the Natural History Museum in London. As shown in Figure 1, five separate ROI areas were detected excluding the stamp. Two representative rate points are illustrated in the results, a low rate (at 0.075 bpp) and a relatively high rate (at 0.28 bpp). Test results are given in Table 1. From those data, it can be seen that the SPIHT ROI generally gives better objective performance than DjVu. ROI image quality is improved when r is increased, trading off against the background image quality. This can be seen by the comparison between SPIHT_roi(4) and SPIHT_roi(8). The subjective image quality of the SPIHT ROI is comparable with DjVu,

as shown by Figure 5. Image *c* in Figure 5 is more readable than image *a*, especially if blown up so that detail is visible. Image *b* seems as readable as image *d*, but the proposed method is closer to the original image, as is clear from the detailed text region (zoomed 300% in the left top corner of the test image). The JPEG2000 Arbitrary ROI 'Maxshift' method [13] resulted in a drastic trade-off between ROI and background, evidenced by the very low whole image PSNR's in Table 1.

Another test image was chosen from the Thomas Jefferson papers at the Library of Congress; test data in Table 1 shows similar results, and subjective performance is compared in Figure 6, confirming the results from the archive card image. The trend of the IW44_roi ROI-only data differs from that of the card image, because, for the Jefferson letter, simple bit-plane shifting over more bit planes upsets correlations exploited by IW44's binary context adaptive arithmetic coding, degrading the relative PSNR. For reasons of space, Figure 6 shows a zoomed-in extract from a Jefferson letter, rather than the complete letter to which Table 1 refers.

VI. CONCLUSIONS

In this paper, a simple block-based document segmentation algorithm, combined with ROI methods applied to wavelet coding, is proposed as an alternative to current multi-layer document coding methods that apply different coding techniques to each layer. The proposed approach can be applied to a number of wavelet coding algorithms: the paper compares objective PSNR and subjective visual performance of IW44, SPIHT and JPEG2000 implementations with a well-known commercial multi-layer coding algorithm (DjVu). By taking advantage of the simple representation of ROIs using corner coefficients, further optimization of SPIHT is possible, resulting in a computationally efficient scalable coder with favorable performance at very low bit rates.

REFERENCES

[1] C. Christopoulos, A. Skodras, and T. Ebrahimi, "The JPEG 2000 still image coding system: An overview," *IEEE Transactions On Consumer Electronics*, 46(4):1103-1127, Nov. 2000.

[2] L. Bottou, P. Haffner, P. G. Howard, P. Simard, Y. Bengio, and Y. LeCun. High quality document image compression with DjVu. *Journal of Electronic Imageing*, 7(3):410-425, 1998.

[3] P. G. Howard, Text image compression using soft pattern matching. The Computer Journal, 40(2/3):146-156, 1997.

[4] I. H. Witten, A. Moffat and T. C. Bell, "Managing gigabytes : compressing and indexing documents and images" (2nd ed), Morgan-Kaufmann, San Francisco, 1999.

[5] The Global Lepidoptera Names Index, G. Beccaloni, M. Scoble, G. Robinson and B. Pitkin, available at http://www.nhm.ac.uk/entomology/lepindex/. (last accessed xi 14 03)

[6] A. Said and W. A. Pearlman, "A new, fast, and efficient image codec based on set partitioning in hierarchical trees," *IEEE Transactions. On Circuits and Systems for Video Technology,* 6:243-250, June 1996.

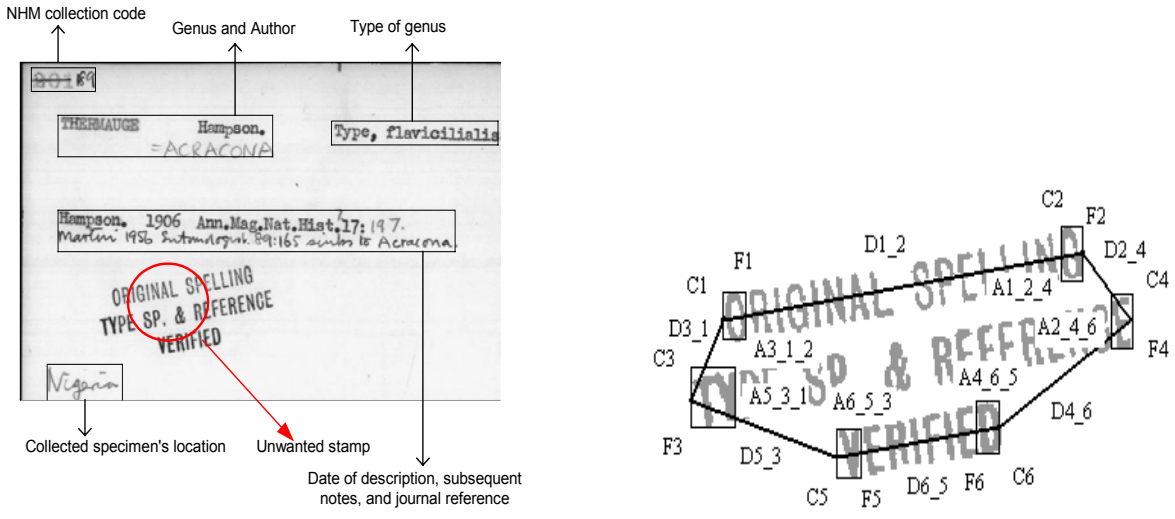
[7] X-W. Yin, M. Fleury, and A. C. Downton, "Archive Image Communication with Improved Compression", Proceedings of 7th Int. Conf. on Document Analysis and Recognition, ICDAR 2003, Vol. I, pp. 92-96, August 2003.

[8] J. Ha, R. M.Haralick and I. T. Phillips, 'Recursive X-Y Cut using Bounding Boxes of Connected Components', Proc. 3rd Int. Conf. on Document Analysis and Recognition (ICDAR), Vol. II, pp. 952-955 August 1995.

[9] K. Y. Wong, R. G. Casey, and F. M. Wahl, Document Analysis System, IBM Journal of Research and Development, 26(2):647-656, 1982.
[10] A. C. Downton, S. M. Lucas, G. Patoulas, G. W. Beccaloni, M. J. Scoble and G. S. Robinson, 'Computerising natural history card archives', 7th Int. Conf. on Document Analysis and Recognition ICDAR 2003, Vol. 1, pp. 354-358, August 2003.
[11] C. Christopoulos, J. Askelof, and M. Larsson, "Efficient methods for encoding region of interest in the upcoming JPEG2000 still image coding standard," *IEEE Signal Processing Letters*, 7(9):247-249, Sept. 2000.
[12] K. Park and H. W. Park, "Region-of-interest Coding Based on Set Partitioning in Hierarchical Trees", *IEEE Transactions On Circuits and Systems for Video Technology*, 12(2):106-113, Feb. 2002.
[13] JPEG 2000 Part I Final Committee Draft Version 1.0, ISO/IEC FCD15444-1:2000, Annex H, 2000.

Rate \ RoI methods			DjVu	IW44	IW44_ roi(4)	IW44_ roi(8)	SPIHT	SPIHT_ roi(4)	SPIHT _roi(8)	SPIHT_ roi(max)	JPEG2 000_roi
Arch-ive Card (Figure 5)	.075 bpp	Whole	21.97	26.98	22.58	20.85	27.11 *29.83*	27.10 *29.83*	22.47 *25.59*	22.47 *25.59*	14.14 (dB)
		ROI only	18.07	22.35	23.40	23.40	22.30 *23.87*	22.31 *23.86*	24.56 *24.79*	24.56 *24.79*	25.06 (dB)
	.28 bpp	Whole	31.08	34.91	26.69	21.38	35.19 *38.77*	35.33 *38.77*	30.16 *38.72*	23.09 *26.86*	14.21 (dB)
		ROI only	26.97	30.52	33.45	34.31	30.79 *33.85*	31.00 *33.85*	34.40 *34.41*	34.99 *35.09*	35.95 (dB)
Jeffer-son Letter (Figure 6)	.039 bpp	Whole	30.20	35.30	35.30	34.29	34.65	34.65	30.53	30.53	13.12 (dB)
		ROI only	26.15	32.01	32.01	31.48	31.25	31.25	32.20	32.20	33.25 (dB)
	.139 bpp	Whole	40.80	41.03	41.03	37.26	40.76	40.76	40.75	31.52	13.14 (dB)
		ROI only	38.82	39.24	39.24	36.16	39.21	39.21	39.34	40.57	41.15 (dB)

Table 1: PSNR comparisons of each ROI method for Archive Card and `Jefferson' letter examples, with plain font SPIHT archive card results being with stamp treated as background, and italic font being with unwanted stamp removed.



(a) (b)

Figure 1: (a) Example Lepidotera archive index card showing text region coding; (b) Detection of the unwanted stamp ROI, prior to removal.

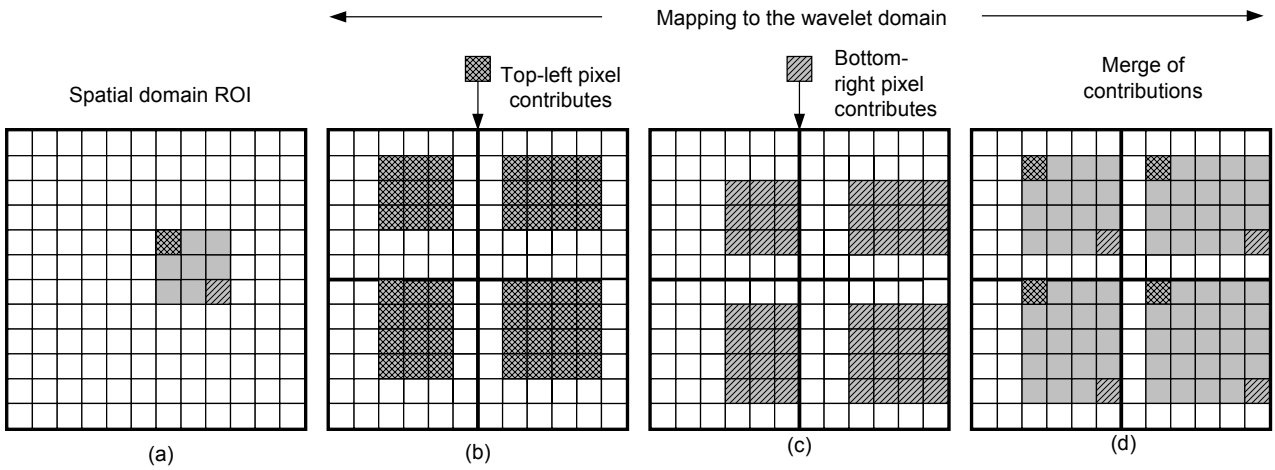


Figure 2: This figure shows how the top-left and bottom-right pixels of an ROI in the spatial domain used to generate the ROI mapping in a wavelet image decomposed into subbands after one level: (a) represents the original ROI, (b) shows the ROI mapping generated simply by (a)'s top-left pixel, (c) shows the ROI map generated within the same subband by the bottom-right pixel, and (d) shows that the whole ROI map can be easily obtained by merging (using the scheme of equation (2)) the contributions of (b) and (c).

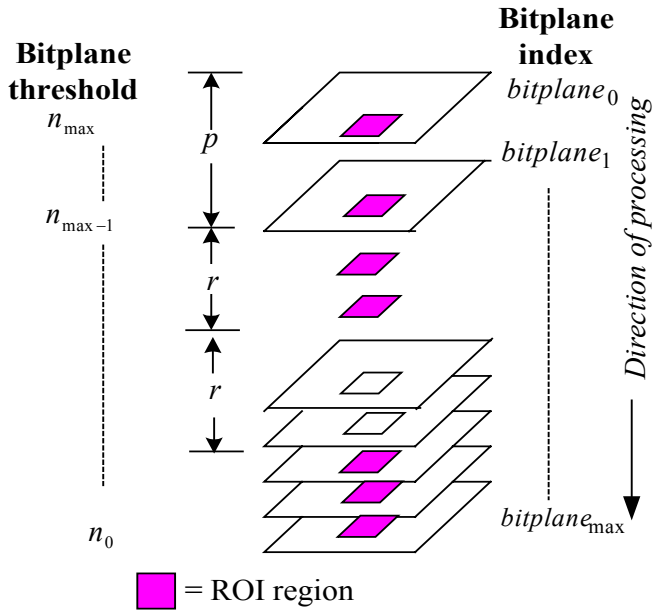
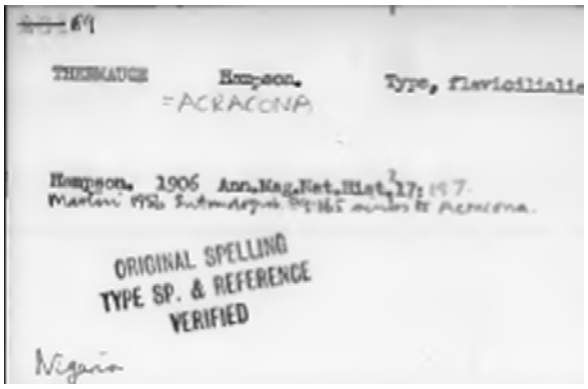


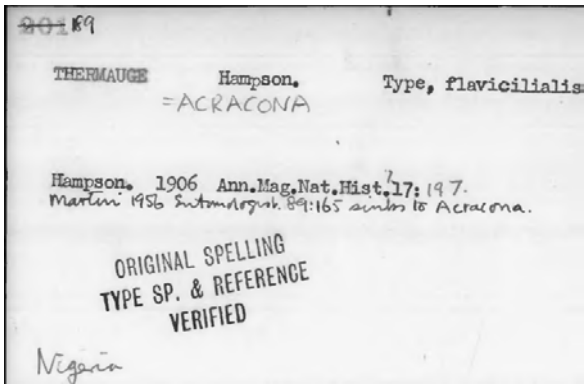
Figure 3: After the *p*th bitplane encoding, the ROI will be encoded *r* bitplanes earlier than the rest of image.

```
Judge_ROI((i,j), type) {
  case type:
      "coefficient": Check whether c_{i,j}
                     belongs to the ROI;
                     if so return TRUE;
      "A": Check whether any memberof D(c_{i,j})
           belongs to the ROI;
           if so return TRUE;
      "B": Check whether any member of L(c_{i,j})
           belongs to the ROI;
           if so return TRUE;
    }
```

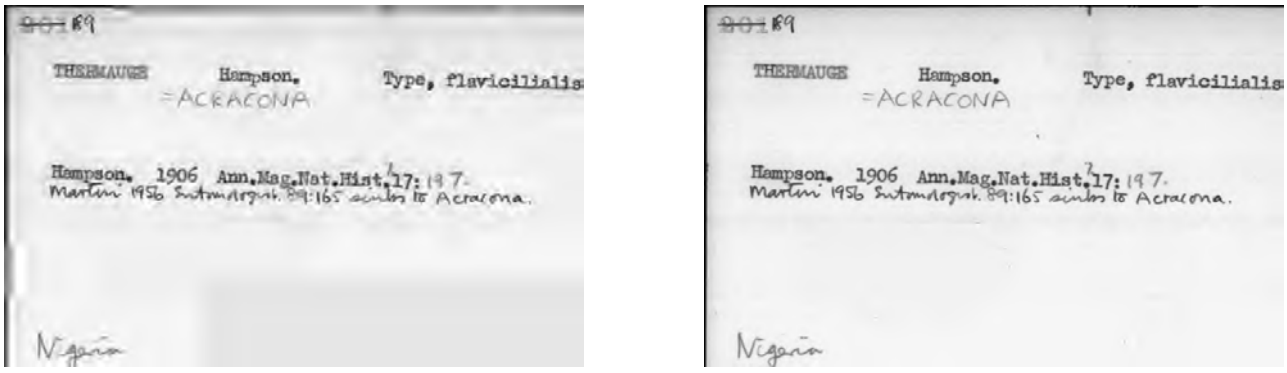
Figure 4: Judgment function used to replace PROI in [12]



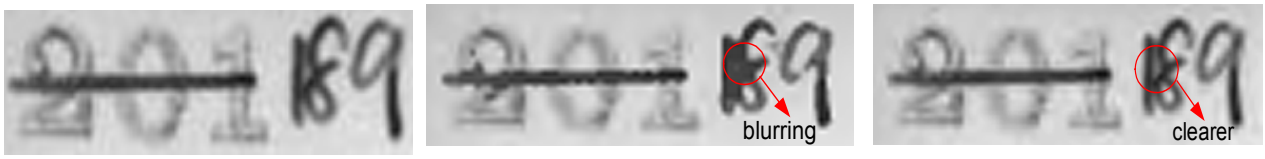
(a)



(b)

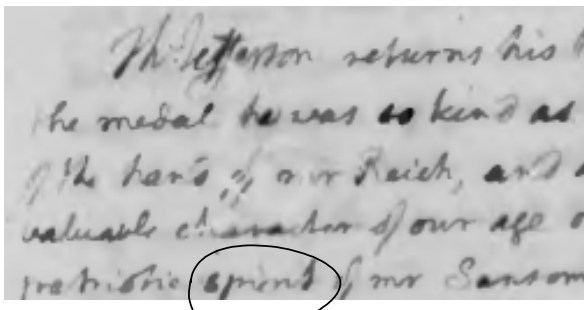


(c) (d)

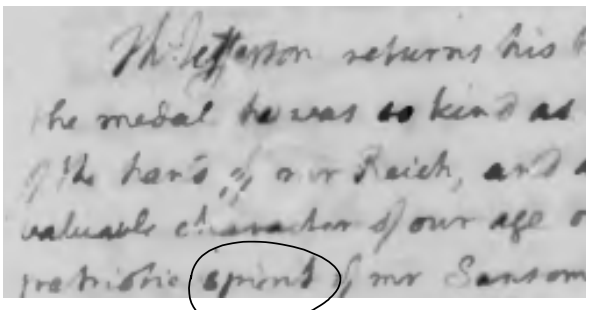


(e) (f) (g)

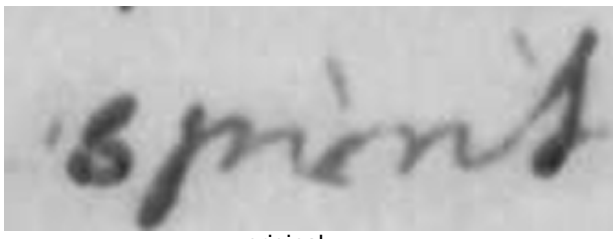
Figure 5: Visual image quality comparison:(a) DjVu at 0.075 bpp, (b) DjVu at 0.28 bpp (c) SPIHT_roi_max at 0.075 bpp, (d) SPIHT_roi_max at 0.28 bpp, (e) 300% zoom of the original card (f) 300% zoom of the top left corner of (b) showing blurring, (g) 300% zoom of the top left corner of (d) showing two clearer figures.



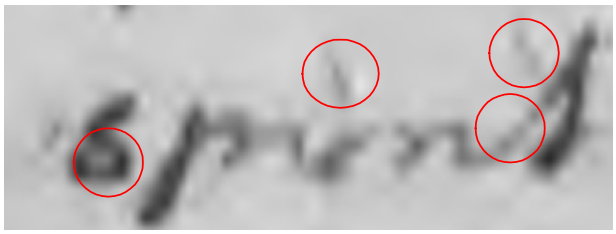
(a)



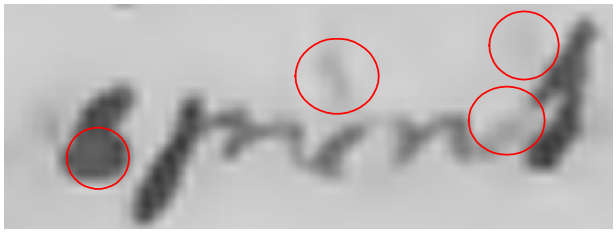
(b)



original



ROI method



DjVu method

(c)

Figure 6: Thomas Jefferson Papers Series 1: Extract from letter: (a) generated by DjVu (b) generated by the ROI method (c) zoomed in to show relative text blurring in multi-layer (DjVu) method compared to ROI method.