ISSN 2712-0562
Sustainable Engineering and Innovation *Original Research*
Vol. 3, No. 2, June 2021, pp.92-101
https://doi.org/10.37868/sei.v3i1.id 144

Lossy codecs for digital image signatures

Benjamin Kommey[1*], Seth Kotey[1], Gideon Adom-Bamfi[2], Eric Tutu Tchao[1]
[1] Department of Computer Engineering, Kwame Nkrumah University of Science and Technology, KNUST-Kumasi, Ghana
[2] Department of Electrical and Electronics Engineering, Kwame Nkrumah University of Science and Technology, KNUST-Kumasi, Ghana

*Corresponding author: bkommey.coe@knust.edu.gh

Abstract
Most applications in recent times make use of images one way or the other. As physical devices for capturing images improve, the quality and sizes of images also increase. This causes a significant footprint of images on storage devices. There is ongoing research to reduce the footprint of images on storage. Since storage is a finite resource, the goal is to reduce the sizes of images while maintaining enough quality pleasant to the human eye. In this paper, the design of two lossy codecs for compressing grayscale digital signature images has been presented. The algorithms used either simple thresholding or transform coding to introduce controlled losses into the image coding chain. This was to reduce, to a great extent, the average number of bits per pixel required to represent the images. The codecs were implemented in MATLAB and experiments were conducted with test images to study the performances of the algorithms.

Keywords: Lossy compression, Lossy coding, Threshold codec, Discrete Cosine Transform

1. Introduction

Images are increasingly being employed for use in a wide variety of applications [1]. Uncompressed images usually take a large amount of storage space and can also result in slow transmission times across a communication medium. This has led to the development of several image compression techniques which reduce the footprint of images on storage devices and also reduce transmission times [2].

Shannon in his famed foundational works in image theory, relates the information provided by a source like an image (used interchangeably hereafter) to *a priori* uncertainty through probability theory by using the notion of experiments [3], [4]. Borrowing some ideas from statistical physics, he viewed a discrete source as a random system that is assumed to have a set X of possible states it can be in, and a probability distribution over those states. Assuming that such a source is memoryless, Shannon defined the measure of average information received from the source as its entropy, $H(X) = -\sum_{i\epsilon X} p(i) \log_2 p(i)$, which is quantified by the bit (fundamental axiom 1) [5]. For sources with memory, we refer to the entropy rate *H(X)*. In his noiseless source coding theorem, Shannon postulated that, as a limit case, the entropy of source X denoted the minimum number of bits per symbol, i.e., rate *L(X)* required on average to fully represent *X*.

Like any other source coding, image compression is a widely used process of capture or representing information provided by an image source with the least amount of resources. The idea of coding an image *X* is an attempt to map its alphabet, χ unto the alphabet *C* of another source X_C, which requires fewer bits. This is done in order to push *L(X)* towards the limit, *H(X)*. This process is very useful for the storage and possible

transmission of images. The compression of an image is done using an encoder whiles the reversal of the mapping procedure is attained with a decoder. Together, the two procedures or devices form a codec.

Generally, image coding involves the reduction of bits by identifying and eliminating statistical redundancy in an original image source [6]. Based on the requirements of reconstruction, compression schemes can be divided into two broad classes: lossless and lossy. In most cases, the encoding operation ($\chi \rightarrow C$) is lossless, i.e., all information from the image is retained and hence a near complete reconstruction of X is possible from the compressed data. Lossless compression is generally used for applications that cannot tolerate any differences between the original and reconstructed data. Examples of such mapping schemes are Huffman [7] and Arithmetic [8], [9] coding. On the other hand, compression of images may be lossy, meaning the exact original sequence cannot be regenerated from the compressed sequence as some information from the original sequence is lost. Lossy compression is useful in many real applications like speech coding where the presence of moderate amount of reconstruction errors can be tolerated. In other words, it may be preferable to 'sacrifice' some source integrity or quality to improve compression efficiency (i.e., pushing $L(X)$ below $H(X)$). Losses may range from rarely noticeable elements like psycho-visual redundancies (immune to lossless coding) to heavy degradation of source quality. Removal of random noise may be classified as losses as they have very high information content. The concept of a distortion measure is often used to quantify the controlled reduction of quality. It should be noted that, unlike for discrete sources, lossy compression is not an option but a necessity for continuous sources. This is because the infinite precision required for an exact representation of such sources cannot be provided by lossless coding.

In this paper, the design of two lossy coding schemes for compressing digital signature images, is presented. The algorithms introduce controlled losses into image coding chain to reduce the average number of bits per pixel required to represent the images. The codecs were implemented in MATLAB and experiments were conducted with test images to study the performances of the algorithms. The rest of the report is organized as follows: Section 2 presents a literature review and section 3 details the design and implementation of the two compression schemes. Next, the developed algorithms are tested on sample image sources and obtained results are discussed in Section 4. Section 5 concludes this paper.

2. Literature review

Goyal et al. [10] proposed a lossy image compression method based on Partition of Intervals. The compression method was based on the standard JPEG baseline [11]. Their method segments an image into various regions based on a segmentation scheme. The segmentation is done according to the importance of various zones in the image. The image is further divided into 8 x 8 non-overlapping blocks, and depending on the region in which resides, it is multiplied by a corresponding quantization matrix. For the outer regions, a higher quality factor is used and a lower quality factor is used for inner regions to preserve more information in the inner regions.

Jeromel & Zalik [12] proposed a lossy compression method for cartoon images. The proposed method divides an image into regions with similar colours, which are then transformed with Burrows-Wheeler [13] and Move-to-Front transforms [14]. The Run-Length Encoding and arithmetic coding methods are used to encode the resultant transforms.

Rahman et al. [15] proposed a histogram modification-based lossy compression method using Huffman coding. The method reduces the total number of different probabilities and pixels value by increasing the frequencies in the Huffman coding. This results in a reduction in the Average Code Length, increasing compression ratio.

Walker [16] proposed a lossy still-image compression method called adaptively scanned wavelet difference reduction (ASWDR). The method is an improvement of the wavelength difference reduction algorithm [17],

[18]. A wavelet transformation is applied to the image, yielding the transformed image. Then values of the transformed image are scanned through linearly in two passes; the significance pass and the refinement pass. During the significance pass, values of the wavelet transform of the image equal to or greater than the threshold set are recorded, and their positions are encoded in a process called difference reduction. The refinement pass refines the previous significant values to increase precision. Arithmetic coding is used to encode the values. Uvaze et al. [19] proposed a codebook-based lossy compression error which uses prediction error and vector quantization concepts. The proposed method applies a combination of artificial bee colony (ABC) algorithm and genetic algorithm (GA) to construct the codebook. Two artificial neural networks are used as predictors at the compression and decompression stages. The prediction error is then determined from the difference between the actual and predicted image pixel values. Vector quantization based on the error codebook generated by the ABC-GA algorithm is the applied to the prediction errors. The codebook indices corresponding to the closest code words are stored, instead of the original image, thereby achieving compression.

3. Method

3.1. Introduction

Two coding schemes are presented for compressing the images, a Threshold-based codec and a Discrete Cosine Transform (DCT) based codec. The requirement of the system is summarized in Fig. 1. The system is to have an encoder which compresses an image source *X* into a file XC using a coding algorithm and some user-provided inputs like quality control. The codec must also provide a decompression tool takes XC as into sole input and is capable of estimating X i.e., with some distortion. It is required that the image X is in a file format with limited 'pre-compression' such as TIF or BMP. X is also assumed to be grayscale always.

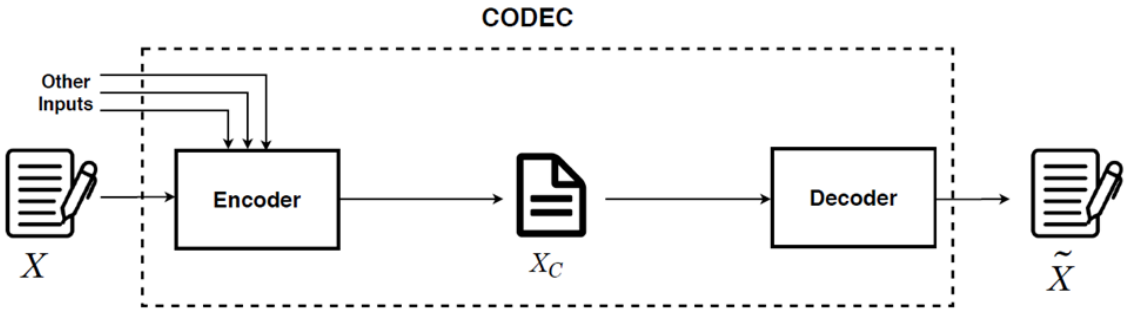


Figure 1. A schematic view of the desired codec

3.2. Thresholding-based codec

The encoding algorithm of this approach employs thresholding, which is a form of uniform scalar quantization, as a simple technique to introduce losses. Here, the image symbol set is divided into two parts that map to one of two binary regions using a threshold. That is the 8-bit integer pixel values are scaled down to a single bit ($\{0, 1\}^8 \rightarrow \{0, 1\}$). The threshold can be chosen reasonably from the range (0, 255). Pixels above the threshold are assigned a '1' otherwise a '0'. The thresholded image is then entropy coded. The thresholding step creates an image with a large '1' population as the bulk of the signature image pixels are white. Taking advantage of the context, run-length (RL) coding is applied to achieve some compression. Therefore, each row of the binary image is compressed coded with a '1'-based RL encoder. Alternatively, the binary image's pixels could be inverted and then a '0'-based RL coder used. Run lengths gathered from all rows are concatenated into a vector *V* separated by special end of row (EOB) symbols. The resulting vector is then coded with a probability coder. In this work, the (optimal) Huffman coder is used. The unique run-lengths in *V* and their probability distribution are determined and the vector is Huffman coded. It should be noted that since each pixel value of the binary image is a Bernoulli source with parameter p where $p \rightarrow 1$, a

Golomb coder could have been a suitable substitute for the Huffman coder. As a last step in the encoder, the coded vector, the Huffman dictionary and variables necessary for the reconstruction process are saved to the compressed file.

In the decoder, the various elements of the compressed file are extracted. The coded vector and the dictionary are used by Huffman decoder to retrieve the vector *V*. With the help of the EOB symbols, the elements of *V* are restructured row-wise just as they were before the RL coding. A corresponding RL decoder is applied on each row to reconstruct the binary image. This image serves as an estimate of the original image. This is because the loss incurred in the thresholding step of the encoder is irreversible as the mapping was many-to-one. The thresholding-based codec is summarized illustratively in Fig. 2.

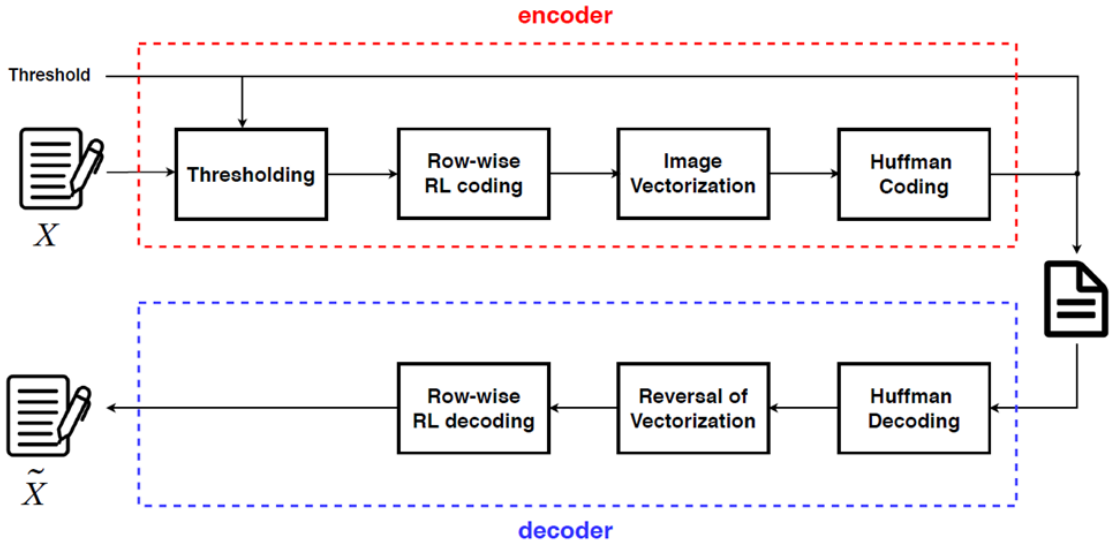


Figure 2. Framework of the thresholding-based coder

3.3. DCT-based codec

The approach adopted in this codec is akin to that which is used in the popular JPEG image coding standard. Here, the image's pixel values are first level shifted by 128, making the values symmetric around zero with average close to zero. This is done to save some bits. The image is then divided into non-overlapping blocks of 8 × 8 pixels, each of which would undergo a discrete cosine transform (DCT). For this partitioning, it is desired that the width and height of the image is a multiple of 8. Therefore, the image dimensions are first checked and padded more white background pixels if necessary. The blocks are then processed row by row, left to right, as follows: To each of the blocks, the DCT transform is applied, producing frequency coefficients. The importance of these coefficients is dependent on the human visual system, the eye is much more sensitive to low frequencies [20]. These coefficients are quantized using a scaled version of a standard luminance quantization matrix *Q*, resulting in most of the high frequency coefficients reducing to zero. This introduces losses, which can be controlled by the strictly positive scale factor, *s*. The higher *s* is the fewer coefficients will be set to zero. The quantization matrix is shown in (1).

$$Q = \frac{1}{s}\begin{pmatrix} 16 & 11 & 10 & 16 & 24 & 40 & 51 & 61 \\ 12 & 12 & 14 & 19 & 26 & 58 & 60 & 55 \\ 14 & 13 & 16 & 24 & 40 & 57 & 69 & 56 \\ 14 & 17 & 22 & 29 & 51 & 87 & 80 & 62 \\ 18 & 22 & 37 & 56 & 68 & 109 & 103 & 77 \\ 24 & 35 & 55 & 64 & 81 & 104 & 113 & 92 \\ 49 & 64 & 78 & 87 & 103 & 121 & 120 & 101 \\ 72 & 92 & 95 & 98 & 112 & 100 & 103 & 99 \end{pmatrix} \quad (1)$$

Next, each quantized block is converted into a vector by gathering its coefficients in a zig-zag fashion using the so-called zig-zag scan shown in Fig. 3. This method allows for the arrangement of coefficients in increasing frequency. The first element of the resulting vector is the DC component whiles the remaining 63 are the AC components. Due to the quantization, it is likely to obtain long substrings of zeros at the tail of the array. The final step is to apply lossless coding.

These AC and DC coefficients are treated separately in the entropy coding process. The DC coefficients are coded first. It is expected that the DC coefficients of cluster of neighbouring blocks differ only slightly from each other in value. Hence, differential coding is applied to the DC coefficients of the blocks in row-by-row, left to right fashion and the differences with the previous DC coefficients are recorded (The DC value of the first block which remains unchanged). The unique values in the array and their distribution are determined and the array of DC differences is then Huffman coded. By exploiting the presence of clusters of zeros in the array tails, the AC coefficients of each block are '0'-RL encoded.

The RL code is subsequently entropy coded with the Huffman algorithm. The two entropy coded vectors, the two Huffman dictionaries and some necessary variables (such as s and image dimensions) are stored to a file.

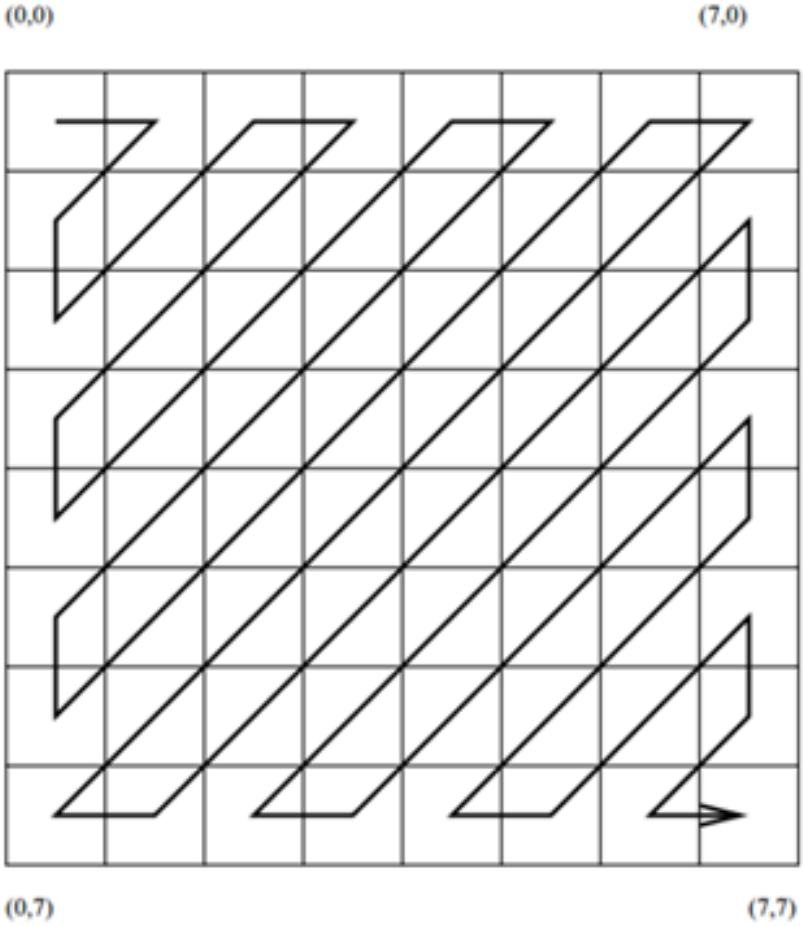


Figure 3. Zig-zag order of scanning an 8 x 8 block

At the decoder, the coded vectors, dictionaries and other elements are extracted from the compressed file. The AC and DC vectors are retrieved with the Huffman decoder. The differential coding on the DC coefficients is reversed. The AC vector is RL decoded to obtain the array of 63 frequency ordered AC components with the tail zeros. Each of the 8 x 8 blocks is then rebuilt from the AC components and their corresponding DC components by retracing the zig-zag pattern in an opposite direction.

The inverse-DCT transform is applied block-by-block to undo the transform coding and then each block is scaled by s. Finally, each pixel value of the decompressed image is shifted by +128 to obtain the estimate of the encoded signature image. An overview of the DCT-based codec is shown in Fig. 4.

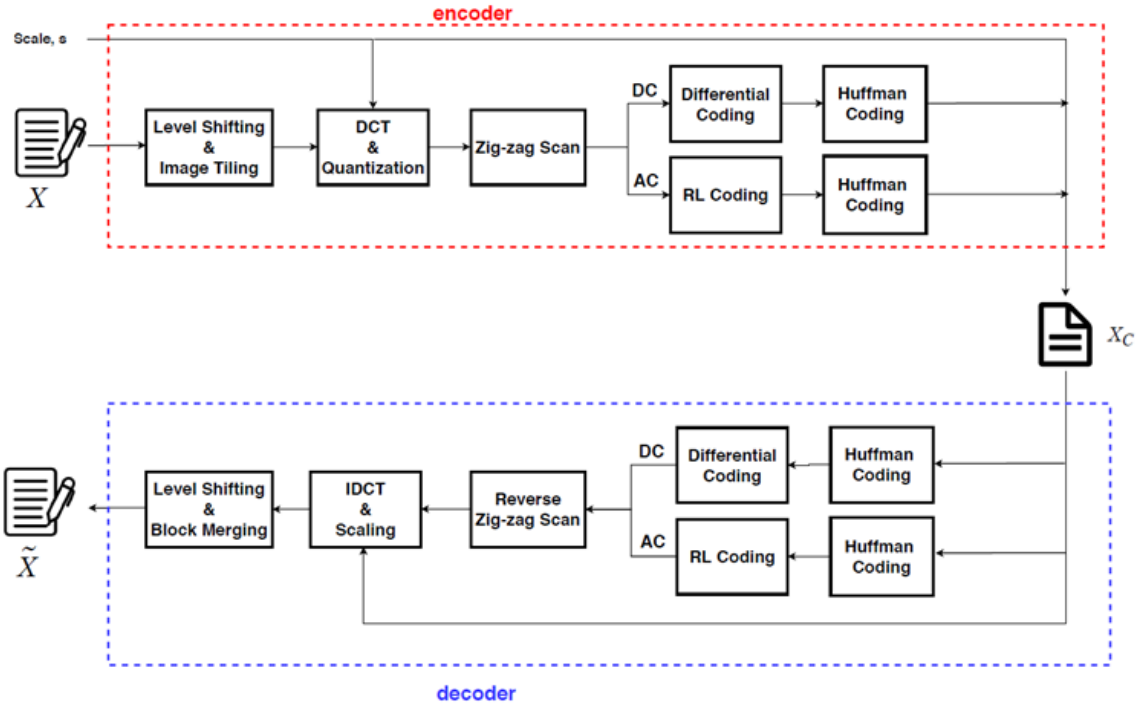


Figure 4. Framework of the DCT-based coder

4. Results and discussion

The designed codecs were implemented and tested in MATLAB®. Two images were used to test both codecs. The original images are shown in Fig. 5a and Fig. 6a. For the thresholding-based codec, values in the range [80, 150] were chosen as threshold. The reconstructed images are shown in Figs. 5b, 5c, 6b and 6c. For Fig. 5b, the threshold was set at 80 and for fig. 5c, the threshold was set at 150. The second image had the threshold set at 100 shown in Fig. 6b and 140 shown in fig. 6c.

The test images were also coded with the DCT-based algorithm for different scales s ϵ [1, 20]. Figs. 5d, 5e, 6d and 6e show the results from the transform codec. A scale value of 10 was used for fig. 5d and a value of 20 was used for fig. 5e. Fig. 6d had a scale value of 5 and fig. 6e had a scale value of 15.

The performances of the coding schemes were measured by the rate of the compressed file and the distortion incurred by attempting to reduce the rate beyond the image entropy. For an image X with dimensions a x b, if size of the compressed file is z in bytes and the reconstructed image is $\hat{X}$, the rate and distortion can be evaluated using (2) and (3).

$$Rate = \frac{z\,x\,8}{a\,x\,b} \tag{2}$$

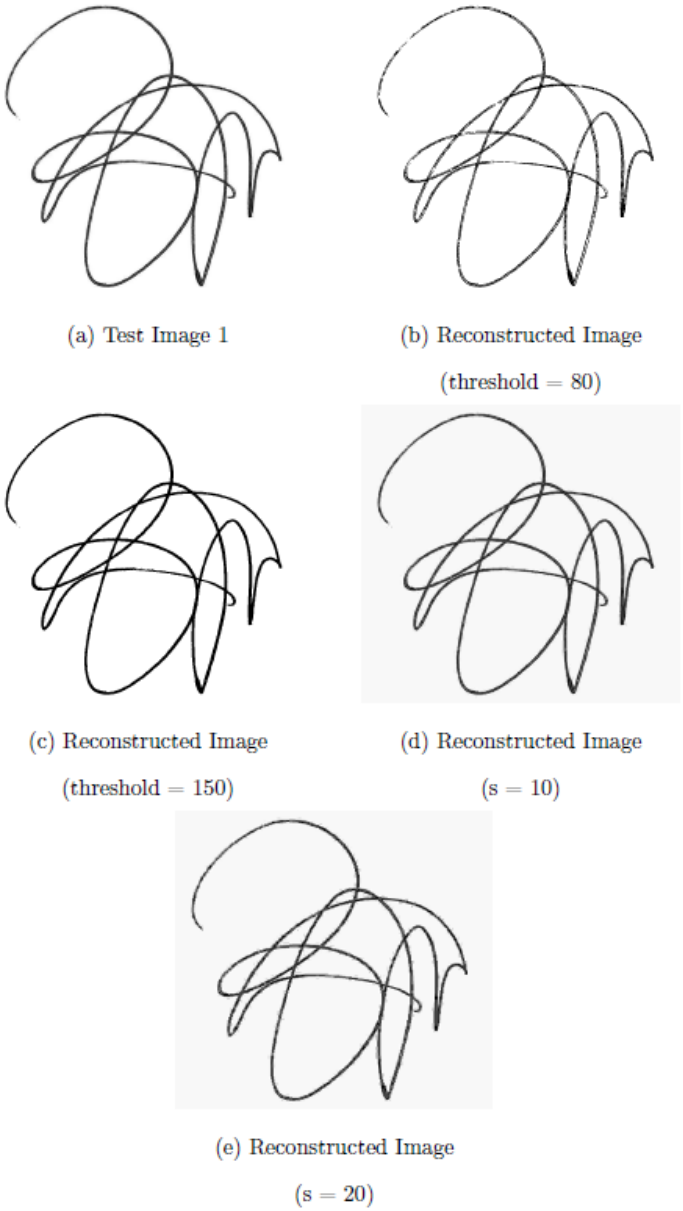
$$Distortion = \frac{1}{a\,x\,b}\sum_{i=1}^{a}\sum_{j=1}^{b}(X_{ij} - \widehat{X_{ij}})^2 \tag{3}$$

The summary of the properties and performances of both algorithms are summarized in Table 1. The rate-distortion relationships of the two coding algorithms were also examined. Figure 7 shows the rate-distortion curve for test image 1. From the results obtained (Table 1 and Figs. 5, 6, and 7), it can be seen that the transform-based coding offers better compression of images as compared to the thresholding-based approach. However, as a trade-off, using DCT with quantization to improve compression introduces larger distortion. This is evident in the greying of the originally white background of the image; an occurrence which is

pronounced for larger s values. For s ϵ [1, <10], this effect is almost absent and the algorithm produces rates comparable to the thresholding approach. It is also seen from Table 1 that, both lossy codec produced compressed files with lower rates than the entropy of the original image, thus validating lossy coding theory.

Table 1. Performance of both codecs with test images

Parameter	**Test Image 1**				**Test Image 2**			
Entropy (bits/pixel)	1.1431				0.9937			
Dimension	801 x 856				904 x 1028			
Codec	Threshold-based		DCT-based		Threshold-based		DCT-based	
Quality level	t=80	t=150	s=10	s=20	t=100	t=140	s=5	s=15
Rate (bits/pixel)	0.1735	0.1434	0.1382	0.0814	0.1200	0.1136	0.1505	0.0708
Distortion	8.90	13.71	52.16	52.87	5.93	7.62	6.75	51.62



(a) Test Image 1

(b) Reconstructed Image (threshold = 80)

(c) Reconstructed Image (threshold = 150)

(d) Reconstructed Image (s = 10)

(e) Reconstructed Image (s = 20)

Figure 5. Outputs of the two codecs for test image 1

(a) Test Image 2

(b) Reconstructed Image (threshold = 100)

(c) Reconstructed Image (threshold = 140)

(d) Reconstructed Image (scale = 5)

(e) Reconstructed Image (scale = 15)

Figure 6. Outputs of the two codecs for test image 2

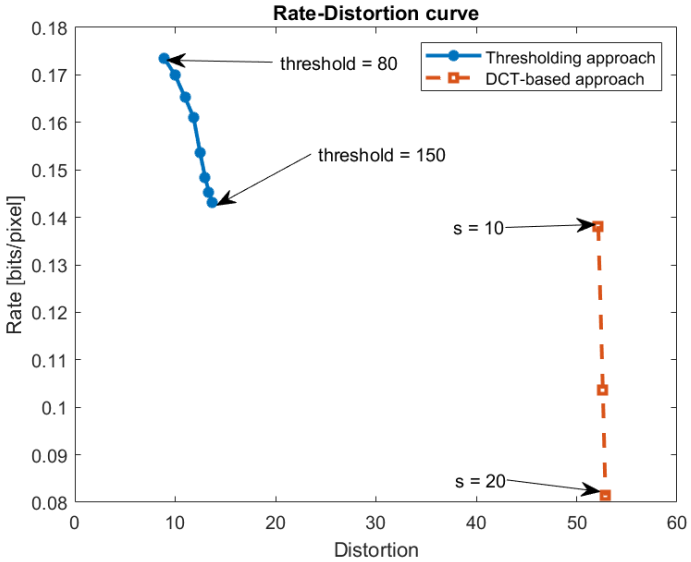


Figure 7. Rate-distortion curves for both coders

5. Conclusion

In this paper, the design of two lossy coding schemes for compressing grayscale digital signature images, has been presented. The algorithms employed either simple thresholding or transform coding to introduce controlled losses into image coding chain in order to reduce the average number of bits per pixel required to represent the images. The codecs were implemented in MATLAB and experiments were conducted on test images to study the performances of the algorithms. The results of the experiments validated the theory of lossy coding.

References

[1] C. V Castro, A. K. Sudha, I. T. Parithi, T. Arulraj, and R. Balasubramanian, "A Novel Method for Lossy Image Compression Using Optimized Side Match Vector Quantization," vol. 6, no. 3, pp. 479–488, 2019.

[2] M. Singh, S. Kumar, S. Singh, and M. Shrivastava, "Various Image Compression Techniques : Lossy and Lossless," vol. 142, no. 6, pp. 23–26, 2016, doi: 10.5120/ijca2016909829.

[3] C. E. Shannon, "A mathematical theory of communication," *The Bell System Technical Journal*, vol. 27, no. 3, pp. 379–423, 1948.

[4] C. E. Shannon, "Communication theory of secrecy systems," *The Bell System Technical Journal*, vol. 28, no. 4, pp. 656–715, 1949.

[5] P. M. Lee, "On the Axioms of Information Theory," *The Annals of Mathematical Statistics*, vol. 35, no. 1, pp. 415–418, 1964, [Online]. Available: http://www.jstor.org/stable/2238053.

[6] M. Kunt, A. Ikonomopoulos, and M. Kocher, "Second-generation image-coding techniques," *Proceedings of the IEEE*, vol. 73, no. 4, pp. 549–574, 1985, doi: 10.1109/PROC.1985.13184.

[7] D. A. Huffman, "A Method for the Construction of Minimum-Redundancy Codes," *Proceedings of the IRE*, vol. 40, no. 9, pp. 1098–1101, 1952, doi: 10.1109/JRPROC.1952.273898.

[8] J. Rissanen and G. G. Langdon, "Arithmetic Coding," *IBM Journal of Research and Development*, vol. 23, no. 2, pp. 149–162, 1979, doi: 10.1147/rd.232.0149.

[9] I. H. Witten, R. M. Neal, and J. G. Cleary, "Arithmetic coding for data compression," *Communications of the ACM*, vol. 30, no. 6, pp. 520–540, 1987, doi: 10.1145/214762.214771.

[10] D. Goyal, H. Kushwah, and A. K. Tiwari, "A NOVEL REGION BASED ADAPTIVE LOSSY IMAGE COMPRESSION," 2010, doi: 10.13140/RG.2.2.35138.09926.

[11] ISO/IEC, "Digital compression and coding of continuous-tone still images." 1994.

[12] A. Jeromel and B. Zalik, "An efficient lossy cartoon image compression method," *Multimedia Tools and Applications*, no. 79, pp. 433–451, 2020, doi: 10.1007/s11042-019-08126-7.

[13] M. Burrows and D. J. Wheeler, "A block sorting lossless data compression algorithm," 1994.

[14] J. L. Bentley, D. D. Sleator, R. E. Tarjan, and V. K. Wei, "A Locally Adaptive Data Compression Scheme," *Communications of the ACM*, vol. 29, no. 4, pp. 320–330, 1986.

[15] A. Rahman, M. M. F. Rabbi, M. Rahman, M. Islam, and R. Islam, "Histogram modification based lossy image compression scheme using Huffman coding," pp. 279–284, 2018.

[16] J. S. Walker, "Lossy image codec based on adaptively scanned wavelet difference reduction," no. May, pp. 1891–1897, 2021.

[17] J. Tian and R. O. Wells, "A lossy image codec based on index coding," in *Proceedings of Data Compression Conference - DCC '96*, 1996, p. 456, doi: 10.1109/DCC.1996.488388.

[18] J. Tian and R. O. Wells, "Embedded Image Coding Using Wavelet Difference Reduction," *The International Series in Engineering and Computer Science*, vol. 450, pp. 289–301, 2002, doi: 10.1007/0-306-47043-8_17.

[19] M. Uvaze, A. Ayoobkhan, E. Chikkannan, and K. Ramakrishnan, “Lossy image compression based on prediction error and vector quantisation,” 2017, doi: 10.1186/s13640-017-0184-3.

[20] A. Swart, “An introduction to JPEG compression using MATLAB.” 2003, [Online]. Available: https://www.researchgate.net/publication/265185146.