AN EVALUATION OF SAR IMAGE COMPRESSION TECHNIQUES

Fatma A. Sakarya[1,2] *Dong Wei*[3] *Serkan Emek*[1]

[1]Department of Electronics and Communication Engineering, Yildiz Technical University, Istanbul, Turkey
[2]Space Technologies Department, MRC of the Scientific and Technical Research Council of Turkey, Gebze, Kocaeli, Turkey
[3]Department of Electrical and Computer Engineering, The University of Texas, Austin, TX 78712, USA

ABSTRACT

Transform coding based on the Karhunen-Loève Transform (KLT), the Discrete Cosine Transform (DCT), and the Discrete Wavelet Transform (DWT) is well-understood for optical images. Transform coding applied to synthetic aperture radar (SAR) data, however, has not been well-studied. This paper compares the results of compressing SAR images using KLT, DCT, and DWT coders. We compare the compression results based on six performance criteria— mean-squared error, mean absolute error, peak signal-to-noise ratio, energy compaction, transform gain, and compression ratio.

1. INTRODUCTION

Modern spaceborne synthetic aperture radar (SAR) systems have onboard hardware that consists of transmitting and receiving units, an analog-to-digital converter, real-time data downlink, and a storage facility. One of the primary constraints in the design and operation of spaceborne SAR systems is the unavailability of a downlink with a high enough data rate. The data rate of each channel is proportional to the pulse repetition frequency, the number of sampled values in each received echo, and the number of quantization bits in each sample. Reducing the data rate deteriorates the system performance. Therefore, data compression offers a practical solution [1].

Data compression algorithms can be classified as lossless and lossy. Lossless compression algorithms such as Huffman coding, arithmetic coding, run-length encoding, and Lempel-Ziv coding [2] are used when exact reconstruction of the original data set is necessary. Lossy compression algorithms such as predictive coding, transform/subband coding, vector quantization, and fractal coding are used for applications in which some degree of degradation of the data is tolerable and/or high compression ratio is preferred.

In this paper, several algorithms are applied to SAR images to form the compression system shown in Figure 1. The compression system consists of a cascade of a prefilter, image transformer, quantizer, and encoder. The prefilter is a sigma filter which removes speckle noise from SAR images [3]. The transform is the Karhunen Loève Transform [4, 5], the Discrete Cosine Transform [4, 5], or the Discrete Wavelet Transform [4, 6, 7]. The quantizer is a scalar quantizer, and the encoder is the Lempel-Ziv algorithm [2], which is applied for further data reduction. In order to reconstruct the compressed image, we reverse the order of the compression steps. We evaluate the performance of the three transform techniques for compressing SAR images by comparing the original and compressed images according to six performance criteria— mean-squared error, mean absolute error, peak signal-to-noise ratio, energy compaction, transform gain, and compression ratio.

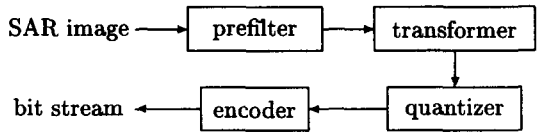

Figure 1. Block Diagram for Transform Coding

2. STRUCTURE OF SAR IMAGES

Synthetic aperture radar (SAR) systems, which are typically mounted on aircraft and satellites, transmit and receive microwaves. The microwaves travel to the earth and scatter when they bounce off of objects and the ground. Different objects and different types of terrain scatter microwaves differently. The amount of scattering (a.k.a. backscattering) determines the grey-levels (tones) of objects in the SAR image. For example, water produces a relatively dark tone since it produces little backscatter towards the radar, trees give medium tones, and buildings correspond to light tones. By analyzing the backscattering, it is possible to characterize surface properties of the earth, such as chemical composition, vegetation, and ambient temperatures. The resolution of these features is dependent on the spatial sampling in the SAR image. The target resolution area (called the resolution cell) must be sampled by at least 2×2 pixels.

A SAR image can be modeled as a sum of four components. The *micro-texture*, which is the speckle noise, appears as randomly placed bright spots the same size or a little larger than the resolution cell. The *meso-texture*, also called the scene texture, is the variation of backscatter due to the material and geometry of the objects. This grainy texture has an elementary unit covering several resolution cells and is very important for image interpretation. The *macro-texture* refers to variations in radar brightness that are larger than many resolution cells. It arises from objects such as roads, geological lineaments, field boundaries, and forest shadows. *Homogeneous regions* are characterized by the mean backscattering of homogeneous areas.

Speckle noise is formed during the processing of the radar returns into a SAR image. Speckle noise masks fine details in the SAR image. In some applications, it is desirable to reduce the inherent speckle noise. There are essentially two categories of speckle reduction techniques: multi-looking, which is performed on raw data, and filtering, which is performed on image data. In multi-looking, we average neighboring pixels incoherently to remove speckle. The number of pixels averaged is called the number of looks N. If we average N_a pixels in azimuth direction and N_r pixels in range direction, then the total number of looks becomes $N = N_a N_r$. As an alternative, filtering can be used, which acts as a first step in segmentation of an image for later classification. Speckle can be removed using filtering techniques such as mean filtering, median filtering, highpass filtering, edge preserving smoothing filtering, local statistics filtering, or sigma filtering. We tried both mean and sigma filtering and concluded that sigma filtering produced better results. So, our prefiltering step is sigma filtering.

3. TRANSFORM CODING TECHNIQUES

Transform coding has been the most popular technique for image compression [5]. For a typical transform encoder, a block of image data is transformed using a unitary transform so that a large fraction of the data's total energy is packed in relatively few transform coefficients. The transform coefficients are quantized and encoded independently. For simplicity and to reduce computational complexity, separable transforms are used. In particular, we use three widely used separable unitary transforms: the *Karhunen-Loève transform*, the *Discrete Cosine Transform*, and the *Discrete Wavelet Transform*.

The two-dimensional (2-D) separable unitary transform of an $N \times N$ image $\boldsymbol{X} = \{x(m,n)\}_{m,n}$ can be written as

$$y(k,l) = \sum_{m=0}^{N-1} \sum_{n=0}^{N-1} u(k,m)\, x(m,n)\, v(l,n)$$

for $0 \le k,l \le N-1$, where $\boldsymbol{Y} = \{y(k,l)\}_{k,l}$ is called the transformed image, and

$$\{\boldsymbol{u}_k = [u(k,0), u(k,1), \ldots, u(k,N-1)]^T\}_k$$

and

$$\{\boldsymbol{v}_l = [v(l,0), v(l,1), \ldots, v(l,N-1)]^T\}_l$$

are one-dimensional (1-D) complete orthonormal sets of basis vectors. The inverse transform is then

$$x(m,n) = \sum_{k=0}^{N-1} \sum_{l=0}^{N-1} u^*(k,m)\, y(k,l)\, v^*(l,n)$$

for $0 \le m,n \le N-1$. In matrix notation, the transform pair becomes

$$\boldsymbol{Y} = \boldsymbol{U}\boldsymbol{X}\boldsymbol{V}^T \qquad \text{and} \qquad \boldsymbol{X} = \boldsymbol{U}^H \boldsymbol{Y} \boldsymbol{V}^*.$$

3.1. The Karhunen-Loève Transform

If an $N \times N$ image $x(m,n)$ is represented by a random field whose autocorrelation function is separable and given by

$$E[x(m,n)x(m',n')] = r_1(m,m')r_2(n,n')$$

for $0 \le m,n,m',n' \le N-1$, then the basis vectors of the Karhunen-Loève transform (KLT) are the orthonormal eigenfunctions $\phi_1(m,k)$ and $\phi_2(n,l)$ obtained by solving

$$\sum_{m'=0}^{N-1} r_1(m,m')\phi_1(m',k) = \lambda_{1,k}\phi_1(m,k)$$

and

$$\sum_{n'=0}^{N-1} r_2(n,n')\phi_2(n',l) = \lambda_{2,l}\phi_2(n,l).$$

Thus, the 2-D separable KLT is determined by

$$\boldsymbol{U} = \boldsymbol{\Phi}_1^H \qquad \text{and} \qquad \boldsymbol{V} = \boldsymbol{\Phi}_2^*$$

where $\boldsymbol{\Phi}_1 = \{\phi_1(m,k)\}_{m,k}$ and $\boldsymbol{\Phi}_2 = \{\phi_2(n,l)\}_{n,l}$.

The KLT is signal-dependent and unique. It both decorrelates the input perfectly and optimizes the repacking of signal energy.

3.2. The Discrete Cosine Transform

The basis vectors of the 2-D discrete cosine transform (DCT) are given by [5]

$$u(k,m) = v(k,m) = a(k)\cos\frac{(2m+1)k\pi}{2N}$$

where

$$a(0) = \sqrt{\frac{1}{N}}, \qquad a(k) = \sqrt{\frac{2}{N}},\ 1 \le k \le N-1.$$

For highly correlated signals, the DCT closely approximates the KLT and has excellent energy compaction properties.

3.3. The Discrete Wavelet Transform

Figure 2 illustrates a one-level decomposition of an image $\boldsymbol{X}$ using a 2-D separable *discrete wavelet transform* (DWT). The filters H_0, H_1, G_0, and G_1 are 1-D L-tap *perfect reconstruction quadrature mirror filters* (PR-QMFs) [4] satisfying

$$G_0(z) = -H_1(-z), \qquad G_1(z) = H_0(-z),$$

$$H_1(z) = z^{1-L}H_0(-z^{-1}),$$

and

$$|H_0(e^{j\omega})|^2 + |H_1(e^{j\omega})|^2 = 1, \qquad \forall\omega.$$

They are combined with downsampling in both the horizontal (row-wise) and vertical (column-wise) dimensions to compute an approximate image $\boldsymbol{X}_{\mathrm{LL}}$ and three detailed images, i.e. horizontal sub-image $\boldsymbol{X}_{\mathrm{LH}}$, vertical sub-image $\boldsymbol{X}_{\mathrm{HL}}$, and diagonal sub-image $\boldsymbol{X}_{\mathrm{HH}}$ at each level. Here, the indices "L" and "H" refer to low and high frequency subband components. The decomposition procedure is as follows: the input image $\boldsymbol{X}$ is first filtered row-wise with the filter H_0 to produce a lowpass component and with the filter H_1 to produce a highpass component. The filtered outputs are then downsampled by a factor of two to resample each filtered signal component at its Nyquist rate. This does not cause any loss of information since the filtered images have half of the bandwidth of the original one and the aliasing

parts of the two subband components will be eliminated after reconstruction due to the PR property of the filter bank. Both two outputs are then filtered column-wise to produce four sub-images: X_{LL}, X_{LH}, X_{HL}, and X_{HH}. Figure 3 is the system used to reconstruct the decomposed image. It is just the dual of the system in Figure 2 and after reconstruction, $\widehat{X} = X$, provided that no quantization is performed on those subband images. Wavelet-based multiresolution decompositions have been demonstrated as excellent tools for image coding [6].

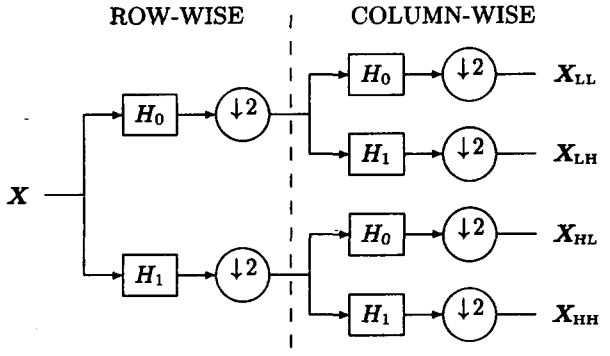

Figure 2. One Level of Forward DWT

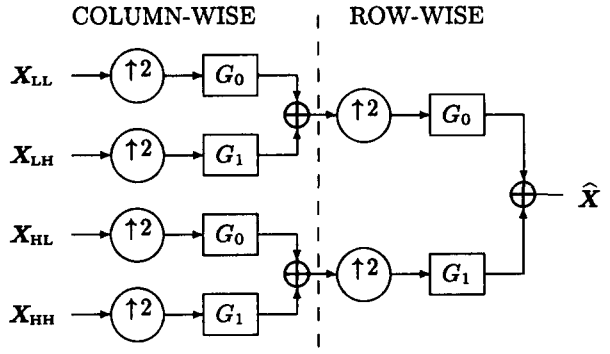

Figure 3. One Level of Inverse DWT

4. LEMPEL-ZIV CODING

We describe the Lempel-Ziv algorithm [2, 8]. for universal data compression. This algorithm is simple to implement and and has an asymptotic rate approaching the entropy of the source. The algorithm is particularly simple and has become popular as the standard algorithm for file compression on computers because of its speed and efficiency.

In the Lempel-Ziv algorithm, the source sequence is sequentially parsed into strings that have not appeared so far. For example, if the string is 1011010100010..., we parse it as 1, 0, 11, 01, 010, 00, 10,... After every comma, we look along the input sequence until we come to the shortest string that has not been marked off before. Since this is the shortest such string, all its prefixes must have occurred earlier. In particular, the string consisting of all but the last bit of this string must have occurred earlier. We code this phrase by giving the location of the prefix and the value of the last bit.

Let $c(n)$ be the number of phrases in the parsing of the input n-sequence. We need $\log c(n)$ bits to describe the location of the prefix to the phrase and 1 bit to describe the last bit. For example, the code for the above sequence is (000,1) (000,0) (001,1) (010,1) (100,0) (010,0) (001,0), where the first number of each pair gives index of the prefix and the second number gives the last bit of the phrase. Decoding the coded sequence is straightforward and we can recover the source sequence without error.

The algorithm performs two passes. The first pass parses the string and calculates $c(n)$, the number of phrases in the parsed string. We allocate $\log c(n)$ bits for the pointers in the algorithm. The second pass calculates the pointers and produces the coded string, as given above. The algorithm allots an equal number of bits to all of the pointers. This is not necessary, since the range of pointers is smaller at the initial portion of the string. The algorithm can be modified so that it requires only one pass over the string and uses fewer bits for the initial pointers. These modifications do not affect the asymptotic efficiency of the algorithm.

5. PERFORMANCE CRITERIA

In many image processing applications, mean-squared error is usually used as a performance criterion to compare the quality of a compressed image to its original version. Unfortunately, MSE does not sufficiently reflect those perceptually significant features of images which are often local rather than global in nature. A similar situation holds for SAR data because SAR data will ultimately be interpreted by a human or machines. Since MSE alone is insufficient for measuring and comparing the performance of SAR data compression algorithms, we evaluate various compression techniques based on a total of six criteria:

- *mean-squared error* (MSE)

$$\text{MSE} = \frac{1}{N^2}\sum_{m=0}^{N-1}\sum_{n=0}^{N-1}(x(m,n) - \hat{x}(m,n))^2$$

  where $x(m,n)$ and $\hat{x}(m,n)$ are respectively the original image and the compressed image of size $N \times N$. It is well-known that MSE globally measures the average energy of the error image.

- *maximum absolute error* (MAE)

$$\text{MAE} = \max_{0\le m,n\le N-1} |x(m,n) - \hat{x}(m,n)|$$

  MAE shows the worst-case error occurring in the compressed image.

- *peak signal-to-noise ratio* (PSNR)

$$\text{PSNR} = 10\log_{10}\frac{\left(\max\limits_{0\le m,n\le N-1} x(m,n) - \min\limits_{0\le m,n\le N-1} x(m,n)\right)^2}{\text{MSE}}.$$

$\lvert\mathcal{C}\rvert$	MSE	MAE	PSNR (dB)	$E_C(\mathcal{C})$ (%)	$G_T(\mathcal{C})$ (%)	CR
64	.0000	.0000	∞	100.00	.00	1.17
49	.0013	.1282	53.24	80.39	23.45	1.52
36	.0042	.1627	41.07	66.73	43.75	2.08
25	.0078	.2526	35.21	57.48	57.07	2.71
16	.0256	.3178	28.25	50.63	75.00	4.57
4	.0389	.7001	18.65	32.95	93.75	16.61

Table 1. Simulation Results for KLT

$\lvert\mathcal{C}\rvert$	MSE	MAE	PSNR (dB)	$E_C(\mathcal{C})$ (%)	$G_T(\mathcal{C})$ (%)	CR
64	.0000	.0000	∞	100.00	.00	1.17
49	.0013	.1290	52.22	80.27	23.45	1.52
36	.0042	.1667	40.68	66.95	43.75	2.05
25	.0080	.2510	34.31	57.93	57.07	2.67
16	.0254	.3170	29.60	49.28	75.00	4.57
4	.0406	.6948	18.09	29.58	93.75	18.25

Table 2. Simulation Results for DCT

filter taps	MSE	MAE	PSNR (dB)	$E_C(\mathcal{C})$ (%)	$G_T(\mathcal{C})$ (%)	CR
4	.0445	.3548	24.35	58.58	75.00	4.56
6	.0436	.3342	26.54	59.72	75.00	4.57
8	.0362	.3218	26.90	60.93	75.00	4.57
12	.0315	.3286	27.34	61.02	75.00	4.58
18	.0289	.3152	28.35	62.20	75.00	4.59
24	.0174	.3002	29.87	63.45	75.00	4.59

Table 3. Simulation Results for DWT

- *energy compaction* (E_C)

$$E_C(\mathcal{C}) = \frac{\sum_{(m,n)\in\mathcal{C}} (y(m,n))^2}{\sum_{m=0}^{N-1}\sum_{n=0}^{N-1}(y(m,n))^2}$$

where $y(m,n)$ is the transformed image and $\mathcal{C}$ denotes the set of indices of preserved transform coefficients. $E_C(\mathcal{C})$ measures the amount of energy conserved by the transform coefficients given by $\mathcal{C}$, and shows the efficiency of the image transform.

- *transform gain* (G_T)

$$G_T(\mathcal{C}) = \frac{N^2 - |\mathcal{C}|}{N^2} \times 100\%$$

where $|\mathcal{C}|$ denotes the cardinality of the set $\mathcal{C}$. $G_T(\mathcal{C})$ provides the percentage of the discarded transform coefficients determined by $\mathcal{C}$.

- *compression ratio* (CR)

$$\text{CR} = \frac{\text{number of bits for } \hat{x}(m,n)}{\text{number of bits for } x(m,n)}$$

CR reflects the achieved compression after coding the transform coefficients.

6. SIMULATIONS AND CONCLUSION

We have used 8×8 blocks in the simulations for the KLT and the DCT. The autocorrelation function of the 1-D KLT is estimated using an AR(1) model with a correlation coefficient 0.95 [4]. For the DWT, we use Coiflet filters of 6, 12, 18, and 24 taps [4, pp. 340, Table 5.2], and minimum-phase binomial QMF of taps 4 and 8 [4, pp. 246, Table 4.2]. The SAR image data is normalized so that the minimum and maximum pixel values are 0 and 1, respectively. Table 1, Table 2, and Table 3 list the simulation results. Based on these tables, the DWT coder gives the best performance and achieves compression in both time and frequency. The subimages which are kept since they conserve at least 10% of the entire image energy after the decomposition are in the size of one-fourth of the original image and preserve 60% of the entire image energy. As the number of the preserved transform coefficients of KLT and DCT decreases, energy conservation and transform gain decrease. Based on all these observations, the DWT is the most promising technique for SAR data compression. Since the goal of this paper is to evaluate three typical transform coders, we do not intend to design sophisticated codecs, with which the compression performance can be further improved.

7. ACKNOWLEDGMENT

The authors would like to thank Prof. Brian L. Evans at UT Austin for his help, and the Remote Sensing Group at MRC for providing SAR data.

REFERENCES

[1] U. Benz, K. Strodl, and A. Moreira, "A comparison of several algorithms for SAR raw data compression", *IEEE Trans. on Geoscience and Remote Sensing*, vol. 33, pp. 1266–1276, Sept. 1995.

[2] J. Ziv and A. Lempel, "A universal algorithm for sequential data compression", *IEEE Trans. on Information Theory*, vol. 23, no. 3, pp. 337–343, May 1977.

[3] J. S. Lee and I. Jurkevich, "Speckle filtering of synthetic aperture radar images: A review", *Remote Sensing Reviews*, vol. 8, pp. 313–340, 1994.

[4] A. N. Akansu and R. A. Haddad, *Multiresolution Signal Decomposition: Transforms, Subbands, and Wavelets*, Academic Press, New York, 1992.

[5] A. K. Jain, *Fundamentals of Digital Image Processing*, Prentice-Hall, Englewood Cliffs, NJ, 1989.

[6] M. Antonini, M. Barland, P. Mathieu, and I. Daubechies, "Image coding using wavelet transform", *IEEE Trans. on Image Proc.*, vol. 1, pp. 205–220, Apr. 1992.

[7] S. H. Chang, T. Y. Lee, and W. H. Fang, "High-resolution bearing estimation via unitary decomposition artificial neural network (unidann)", in *Proc. IEEE Int. Conf. on Acoustics, Speech, and Signal Proc.*, May 1995, vol. 5, pp. 3607–3610.

[8] T. M. Cover and J. A. Thomas, *Elements of Information Theory*, John Wiley and Sons, New York, 1991.