ADAPTIVE-INTERPOLATIVE SUBBAND DECOMPOSITION FOR LOSSLESS AND LOSSY IMAGE COMPRESSION

By

Jeerasuda Kesorn

Dissertation
Submitted to the Faculty of the
Graduate School of Vanderbilt University
in partial fulfillment of the requirements
for the degree of

DOCTOR OF PHILOSOPHY

in

Electrical Engineering

May, 2003

Nashville, Tennessee

Approved by:

Professor James A. Cadzow
Professor D. Mitch Wilkes
Professor Richard Alan Peter II
Professor Richard G. Shiavi
Professor Douglas P. Hardin

ACKNOWLEDGEMENTS

With his warm and kindly welcome since the first day I started the Master's Degree program at Vanderbilt University, I would like to take this opportunity to express my gratitude to my advisor, Centennial Professor James A. Cadzow. He has been consistently supporting and encouraging me throughout the years I began my studies at this school. His intellectual knowledge, time, effort, and willingness conducting me to finally reach this achievement also cannot be failed to mention. In addition, I am indebted to other members of my Ph.D. committee: Prof. D. Mitchell Wilkes, Prof. Richard Alan Peters II, Prof. Richard G. Shiavi, and Prof. Douglas P. Hardin for providing valuable advices and comments concerning my dissertation.

I wish to gratefully acknowledge the Royal Thai Government for granting financial support through my M.S. and Ph.D. studies. I could not have come and fulfilled my dream without this precious patronage. I am extremely thankful to Linda Roth at English Language Center of Vanderbilt University, Kittiporn Phanvijitsiri, and Panchit Chindakul for reading and editing my dissertation during the writing stage. So many people, friends, and colleagues have helped me either in academic or moral support. Without them, living and studying far away from home would be difficult for me. I am extremely grateful to Poolsak Koseeyaporn, my academic colleague, for his valuable support. He is always a friend in need. I also offer my particular gratitude to the Sirikhandha family for their kindness and warm hospitality during my stay in Nashville. I impossibly fail to thank Wipa Sangpisit for her honestly assistance. Additionally, for those who I have not mentioned their names; however, I am sincerely appreciated their friendships, wisdom, and experiences in life they have somehow taught me. I am also deeply indebted to my guardians, Wiwat and Suree Kiranon, whom I always respect them as my parents for their motivation and for sparking my life to this day.

But, as always, the greatest contribution came from my family. My parents-Yom and Suprasri-taught me to cherish and love learning. My younger brother- Parinya always convey his warm regards and concern to me. All those efforts and credits are for my father and mother whom themselves could not have a good opportunity but always perceive the importance of education. Their unconditional love, patience, and dedication for my brother and I are endless and invaluable; nothing in my life can be compared and remunerated to them.

As many of the people I describe if this work is to be fulfilling, it must also be their integral part and efforts and I dedicate this work to these people.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

NOMENCLATURE

2D-HVDD	Two-dimensional-Horizontal Vertical Diagonal Decomposition
2D-HVDD(l_1)	l_1 Norm Based Two-dimensional-Horizontal Vertical Diagonal Decomposition
2D-HVDD(l_2)	l_2 Norm Based Two-dimensional-Horizontal Vertical Diagonal Decomposition
DWT	Discrete Wavelet Transform
LDT	Linear Decomposition Transform
LDT-HVDD	Linear Decomposition Transform-Horizontal Vertical Diagonal Decomposition
LDT-HVDD(l_1)	l_1 Norm Based Linear Decomposition Transform-Horizontal Vertical Diagonal Decomposition
LDT-HVDD(l_2)	l_2 Norm Based Linear Decomposition Transform-Horizontal Vertical Diagonal Decomposition
OS-HVD	Optimum Scalar-Horizontal Vertical Decomposition
OS-HVD(l_1)	l_1 Norm Based Optimum Scalar-Horizontal Vertical Decomposition
OS-HVD(l_2)	l_2 Norm Based Optimum Scalar-Horizontal Vertical Decomposition
OS-HVVD	Optimum Scalar-Horizontal Vertical Vertical Decomposition
OS-HVVD(l_1)	l_1 Norm Based Optimum Scalar-Horizontal Vertical Vertical Decomposition
OS-HVVD(l_2)	l_2 Norm Based Optimum Scalar-Horizontal Vertical Vertical Decomposition
OS-HVDD	Optimum Scalar-Horizontal Vertical Diagonal Decomposition
OS-HVDD(l_1)	l_1 Norm Based Optimum Scalar-Horizontal Vertical Diagonal Decomposition
OS-HVDD(l_2)	l_2 Norm Based Optimum Scalar-Horizontal Vertical Diagonal Decomposition

SPIHT	Set Partitioning in Hierarchical Trees

CHAPTER I

LITERATURE REVIEW AND BACKGROUND

Digital Signal Processing (DSP) is the science of using computers to process many data types such as images, seismic vibration, and radar/sonar echoes. It is not only one of the most important research areas in electrical engineering but also one of the most powerful technologies and has many useful applications in various disciplines. It is employed such applications as signal filtering, speech recognition, neural networks, image compression and much more. Among these applications, image compression is one of the most active areas in electrical engineering research. Nowadays, the sending and reviewing of data between computer users are very important. The World Wide Web has also received widespread interest and has a major influence on computer users. Images transmitted over the Internet require large bandwidth and provide a practical basis for the importance of image compression. In addition, data storage may be costly to organizations, for example, collecting fingerprint information of the Federal Bureau of Investigation (FBI). The FBI has been collecting fingerprint cards since 1924 [1]. As a result, this collection over the past decades increased to around 200 million cards in 1996 with this collection occupying an acre of filing cabinets in the FBI building. Furthermore, the fingerprint data continuously accumulate at a rate of 30,000-50,000 new cards per day, which causes the more critical problem of collecting and searching data. Although these cards are digitized at 500 dots per inch with 8 bits of grayscale resolution, each digitized image requires about 10 megabytes of data storage. One can imagine how large the memory storage space is to maintain the entire digitized fingerprint data in computer storage. Therefore, the FBI required a data storage method that considerably reduces this database. The given examples imply the importance of data/image compression and motivate researchers in this field to develop algorithms and theories for this research area.

Image compression algorithms can be classified as either *lossless* or *lossy*. A lossless technique has no loss of information in that the restored and original data are identical. This is important for many types of data (i.e. executable code, satellite or medical images) in which it is critical not to discard any information that may be useful later. On the other hand, discarding information is possible for some types of data. However, one has to compromise between the

degradation of data quality and the high compression ratio which is the ratio between the number of bits required to represent the original image divided by the number of bits required to represent the compressed image [2]. In other words, generally, the higher the compression ratio of an image, the more degradation of signal quality then is. A technique that causes this type of degradation is referred to as a lossy method. Nevertheless, at the acceptable quality of the restored image, the lossy techniques are still more effective than lossless methods due to its superior performance of compression ratio. Several processes are generally involved in image compression. The first process is to reduce information redundancy, which is called data decorrelation. Secondly, the decorrelated information is then transformed into integers by a quantization process. It is interesting to note that if the decorrelated coefficients are non-integer (floating point) then this process causes some degree of distortion to the reconstructed image coefficients. These integer coefficients are finally encoded; hence, the original data has been transformed into a compact representation.

Over the years, many researchers have endeavored to accomplish better performance in image compression. Their research can be classified into three research areas: the decorrelation technique, the quantized algorithm and the encoding method. This dissertation concentrates mainly on the decorrelation phase; the literature review will focus only on this area. Generally, the decorrelation process in image compression is based on prediction, which can be made by a sequential model (i.e. the CALIC [3]) or a multiresolution model (i.e. the S-transform [4][5], the S+P-transform [6][7]). In lossless application, the state-of-the-art techniques use an adaptive predictor to predict the grey level of the image pixel and employ a context-based entropy coder to encode the prediction error. Among the proposed lossless techniques in the past decade, the LOCO [8] and the CALIC [3][9] have achieved good performance of compression ratio and computational complexity. However, the adaptive predictors used in the LOCO and the CALIC are simple, resulting in the limitation of their performances [10]. The results found in [11] and [12] have shown that the combinations of subpredictors provide similar or better results compared to those of the CALIC. Recently, Deng, Ye, and Cahill [10] have presented an adaptive predictor combination (APC) technique for lossless image compression. Their scheme is based on the estimation of localized variance of prediction errors, which is closely related to the Bayesian model averaging. Even though the compression performance of using this method was comparable to, or better than, that of other published algorithms, it was not as good as that

of the APC scheme using a group of LS (least squares) -based subpredictors. However, the APC scheme using a group of LS-based subpredictors has the problem of demanding computation: the better the compression performance, the more complex the computation.

The multiresolution model is another decorrelation technique which has been received wide-spread interest among researchers. Multiresolution or subband decomposition was first applied to images by Woods, and O'Neil [13]. In the multiresolution model, a multiresolution representation of the image is generated. The characteristic of the multiresolution representation is suitable for progressive transmission, which is a technique of transmitting a low resolution of the image to a user, afterward; more information is then transmitted to construct a finer resolution of the image. The relative performances and effects of subband decomposition based on different prediction schemes (fixed or adaptive) have been compared in [14]. The results have indicated that the performance of the tested predictors degrades as the resolution of the image is reduced due to the decrease of pixel correlation. Furthermore, a number of researchers [15][16][17][18][19][20] have applied the concept of adaptive filtering to subband decomposition, and recently subband decompostion for lossless image compression based on an adaptive filter bank has been studied by [21][22]. The linear multiresolution decomposition has been extensively investigated in [20]. Additionally, some studies in nonlinear decomposition have been found in [23][24][25][26]. Due to the increased flexibility of both non-separable and non-linear decompositions, there has been growing interested in these topics. Cardoso Jr. and Silva [27] have studied a heuristic to design nonseparable nonlinear multiresolution decompositions. However, this proposed method provided many degrees of freedom of both number of decomposition stages and weighting parameters. The results implied that some criteria were required for finding the best solutions. Nevertheless, this topic is still an open problem and more studies may be needed to find more effective design methods.

Transform based decomposition is another technique for the decorrelation process and has received widespread interest in the past decades. The main concept of transformation is to decorrelate the image pixel producing a sparse transformed image in which energy is compacted in a small number of its large magnitude elements. Furthermore, the orthogonal transform plays a major role in image decorrelation due to the fact that this type of transform decomposes the image into the uncorrelated components projected on the orthogonal basis of the transform. As a result, setting to zero the small orthogonal components of the transformed image will have little

effect on the others. In the literature, many researchers have studied image compression based on the Karhunen-Loeve transform (KLT) [28][29][30][31], the discrete cosine transform (DCT) [32][33][34][35], and the discrete wavelet transform (DWT) [1][36][37][38][39][40][41][42][43] [44][45]. However, many studies based on wavelet transform in the literature over the past few decades have implied the superior performance of the wavelet transform over the others. Wavelet transform has been successfully applied for lossy image compression [1][6][36][37][38] [39][40][41][43][44][45][46]. Contrarily, the S+P-transform, which is an integer transformation, is one of among the best lossless compression algorithms in the literature.

In this dissertation, two new decorrelation techniques are proposed. The first method simply bi-directional subsamples by two the image into four subband images. One of the subband images is retained while each of other subband is approximated by a scalar multiple of the retained subband. The approximation errors are then substituted for these subbands. Another proposed decorrelation technique is motivated by a method called Linear Decomposition Transform (LDT) [47]. However, to have better prediction performance, this proposed technique employs a two-dimensional interpolative filter rather than a one-dimensional row-wise and column-wise processing as in the LDT. Reviews of the wavelet, the S+P-transform, and the LDT method are given as follows.

Wavelet Transform

The basic concept of the wavelet transform is to represent any arbitrary function f as a superposition of wavelets [36]. Wavelets have been introduced by Grossman and Morlet [48] generated by scaling and translation of a single wavelet $\psi(t)$ as given by [49]

$$\psi_{j,k}(t) = 2^{j/2}\psi(2^j t - k) \qquad j,k \in Z \tag{1.1}$$

where Z designates the set of all integers. However, the concept of resolution seems to be the best manner to provide interpretation of the wavelets from mathematical and practical viewpoints [50][51][52][53]. As a result, the wavelet analysis is started by defining the scaling function

$\varphi(t) \in L^2(R)$* and the wavelet will be derived in terms of this function. Let us consider the set of all integer translations of the basic scaling function as designated by

$$\varphi_k^0(t) = \varphi(t-k) \qquad k \in Z\,. \tag{1.2}$$

It is required that this set is an orthonormal set of functions and the closed subspace of $L^2(R)$ spanned by these functions, that is

$$V_0 = \overline{\underset{k}{Span}\left\{\varphi_0^k(t)\right\}}\,. \tag{1.3}$$

By changing the time scale of the scaling function, a two-dimensional set of functions can be generated by scaling and translating the basic scaling function as

$$\varphi_j^k(t) = 2^{j/2}\varphi(2^j t - k)\,. \tag{1.4}$$

Furthermore, for a fixed integer *k*, the $\varphi_j^k(t)$ functions, for $k \in Z$, are required to be orthonormal span at index *k* is

$$V_j = \overline{\underset{k}{Span}\left\{\varphi_j^k(t)\right\}}\,. \tag{1.5}$$

To satisfy the basic requirements of multiresolution analysis, it is further assumed that the closed subspaces generated in this fashion are nested, that is $V_j \subset V_{j+1}$ for $j \in Z$ with $V_{-\infty} = \{0\}$ and $V_\infty = L^2(R)$. According to the nesting property, the elements in space V_j are simply scaled ver

---

* The $L^2(R)$ is the space of all functions *f*(*t*) with a well defined integral of the square of the modulus of the function [49].

-sions of the elements in space V_{j+1},

$$f(t) \in V_j \;\Leftrightarrow\; f(2t) \in V_{j+1}. \tag{1.6}$$

Therefore, the basic scaling function $\varphi(t) \in V_0$ must also be contained in the subspace V_1 and can be represented as a linear combination of the basis functions $\varphi_1^k(t)$ by

$$\varphi(t) = \sum_{k=-\infty}^{\infty} h(k)\sqrt{2}\varphi(2t-k) \qquad k \in Z \tag{1.7}$$

where $h(k)$ are scaling coefficients. This equation is called the refinement equation, the multiresolution analysis equation, or the dilation equation [49]. However, the important features of $f(t) \in L^2(R)$ can be deeply discovered not only by using the scaling functions $\varphi_j^k(t)$ but also by a set of functions $\psi_j^k(t)$ generating the subspace W_j that is the orthogonal complement of V_j contained in V_{j+1}. In other words,

$$\left\langle \varphi_j^k(t), \psi_j^l(t) \right\rangle = \int \varphi_j^k(t)\psi_j^l(t)dt = 0, \qquad j,k,l \in Z \tag{1.8}$$

leading to the direct sum decomposition of the subspace V_{j+1} as designated by

$$V_{j+1} = V_j \oplus W_j \;. \tag{1.9}$$

Hence, upon iterating on the direction sum expression for the decreasing set of integers $j = J-1, J-2, ..., j_0+1, j_0$ of the subspace V_J, it follows that

$$V_J = V_{j_0} \oplus W_{j_0} \oplus W_{j_0+1} \oplus ... \oplus W_{J-2} \oplus W_{J-1}, \qquad j_0 < J \,. \tag{1.10}$$

Due to the fact that the wavelets W_j is contained in V_{j+1}, the wavelet functions $\psi(t) \in W_0$ can be represented by a weighted sum of the shifted scaling function $\varphi(2t) \in V_1$ as designated by

$$\psi(t) = \sum_{k=-\infty}^{\infty} g(k)\sqrt{2}\varphi(2t-k), \qquad k \in Z\,. \tag{1.11}$$

Up to this point, a set of functions $\varphi_1^k(t)$ and $\psi_j^k(t)$ are constructed to span all of $L^2(R)$; hence, any function $f(t) \in V_J$ as given by

$$f(t) = \sum_{k=-\infty}^{\infty} c_J(k)2^{J/2}\varphi(2^J t-k) \tag{1.12}$$

can be written as a linear combination of the scaling functions and wavelets by using the fact that $V_J = V_{J-1} \oplus W_{J-1}$ as

$$f(t) = \sum_{k=-\infty}^{\infty} c_{J-1}(k)2^{(J-1)/2}\varphi(2^{J-1}t-k) + \sum_{k=-\infty}^{\infty} d_{J-1}(k)2^{(J-1)/2}\psi(2^{J-1}t-k)\,. \tag{1.13}$$

Iterating the direction sum expression over the deceasing set of integers $j = J-1, J-2, ..., j_0+1, j_0$, the function $f(t)$ can be finally written as

$$f(t) = \sum_{k=-\infty}^{\infty} c_{j_0}(k)\varphi(2^{j_0/2}t-k) + \sum_{j=j_0}^{J-1}\sum_{k=-\infty}^{\infty} d_j(k)2^{j/2}\psi(2^j t-k)\,^{*}. \tag{1.14}$$

The first term of expression (1.14) illustrates the low resolution (coarse approximation) of $f(t)$ whereas the second term gives the finer resolution for each increasing index j. Since the scaling

---

* The starting scale $j = j_0$ can be any level in general.

functions $\varphi_j^k(t)$ and the wavelets $\psi_j^k(t)$ are orthonormal, the coefficients $c_{j_0}(k)$ can be calculated by

$$c_{j_0}(k) = \left\langle f(t), \varphi_{j_0}^k(t) \right\rangle = \int f(t)\overline{\varphi_{j_0}^k}(t)dt$$

$$= \int f(t)2^{j_0/2}\underbrace{\left[\sum_{m=-\infty}^{\infty}\overline{h}(m)\sqrt{2}\overline{\varphi}(2(2^{j_0}t-k)-m)\right]}_{\overline{\varphi}(2^{j_0}t-k)}dt, \qquad V_{j_0} \subset V_{j_0+1}$$

$$= \sum_{m=-\infty}^{\infty}\overline{h}(m)\int f(t)2^{(j_0+1)/2}\overline{\varphi}(2^{(j_0+1)}t-(2k+m))dt$$

$$= \sum_{m=-\infty}^{\infty}\overline{h}(m)c_{j_0+1}(m+2k)\,. \tag{1.15}$$

The similar derivation is repeated for the wavelet coefficients as given by

$$d_j(k) = \left\langle f(t), \psi_j^k(t) \right\rangle = \int f(t)\overline{\psi_j^k}(t)dt$$

$$= \int f(t)2^{j/2}\underbrace{\left[\sum_{m=-\infty}^{\infty}\overline{g}(m)\sqrt{2}\overline{\varphi}(2(2^{j}t-k)-m)\right]}_{\overline{\psi}(2^{j}t-k)}dt, \qquad W_j \subset V_{j+1}$$

$$= \sum_{m=-\infty}^{\infty}\overline{g}(m)\int f(t)2^{(j+1)/2}\overline{\varphi}(2^{(j+1)}t-(2k+m))dt$$

$$= \sum_{m=-\infty}^{\infty}\overline{g}(m)c_{j+1}(m+2k)\,. \tag{1.16}$$

This process is repeated over the set of decreasing integers $j = J-1, J-2, ..., j_0+1, j_0$ producing the required set of coefficients $c_{j_0}(k), d_{j_0}(k), d_{j_0+1}(k), ..., d_{J-1}(k)$ for expression (1.14). It is interesting to note that the highest resolution J is chosen large enough so that the scaling coefficients $c_J(k)$ are equal to the uniformly sampled values of the function $f(t)$ whose DWT is being generated.

Conversely, the highest resolution coefficients $c_J(k)$ can be recovered for the given set of lower resolution coefficients $c_{j_0}(k)$ and the detail coefficients $d_{j_0}(k), d_{j_0+1}(k), ..., d_{J-1}(k)$. To obtain an insight into this explanation, let expression (1.12) be taken the inner product of each side by the scaling function $\varphi_J^l(t)$ for $l \in Z$. By the orthonormality of the scaling functions, it then follows that

$$\left\langle f(t), \varphi_J^l(t) \right\rangle = \left\langle \sum_{k=-\infty}^{\infty} c_J(k) \varphi_J^k(t), \varphi_J^l(t) \right\rangle = c_J(k). \tag{1.17}$$

Additionally, the same procedure is repeated for expression (1.13) in which expressions (1.7) and (1.11) are respectively substituted for $\varphi_{J-1}^k(t)$ and $\psi_{J-1}^k(t)$ yielding

$$\begin{aligned} \left\langle f(t), \varphi_J^l(t) \right\rangle &= \left\langle \sum_{k=-\infty}^{\infty} c_{J-1}(k) \sum_{m=-\infty}^{\infty} h(m) \varphi_J^{2k+m}(t) + \sum_{k=-\infty}^{\infty} d_{J-1}(k) \sum_{m=-\infty}^{\infty} g(m) \varphi_J^{2k+m}(t), \varphi_J^l(t) \right\rangle \\ &= \sum_{m=-\infty}^{\infty} c_{J-1}(m) h(k-2m) + \sum_{m=-\infty}^{\infty} d_{J-1}(m) g(k-2m). \end{aligned} \tag{1.18}$$

This implies that $c_J(k) = \sum_{m=-\infty}^{\infty} c_{J-1}(m) h(k-2m) + \sum_{m=-\infty}^{\infty} d_{J-1}(m) g(k-2m)$. (1.19)

Particularly, the process of determining the coefficients $c_J(k)$ and $d_j(k)$ respectively shown in expressions (1.15) and (1.16) is referred to as the *analysis* phase of the DWT whereas the process of reconstructing the highest resolution coefficients shown in expression (1.19) is called the *synthesis* process of the DWT. Generally, the DWT analysis and synthesis processes can be summarized as follows,

DWT analysis : $c_j(k) = \sum_{m=-\infty}^{\infty} \overline{h}(m) c_{j+1}(m+2k)$ and $d_j(k) = \sum_{m=-\infty}^{\infty} \overline{g}(m) c_{j+1}(m+2k)$

DWT synthesis : $c_{j+1}(k) = \sum_{m=-\infty}^{\infty} c_j(m) h(k-2m) + \sum_{m=-\infty}^{\infty} d_j(m) g(k-2m)$.

To implement the DWT analysis, the scaling coefficients $c_j(k)$ and $d_j(k)$ are obtained by first filtering the scaling coefficients $c_{j+1}(k)$ by the linear shift-invariant filters whose impulse responses are $\bar{h}(-k)$ and $\bar{g}(-k)$, respectively, then down sampling by two* the filter output responses. Contrarily, the synthesis process is accomplished by first up sampling by two** the coefficients $c_j(k)$ and $d_j(k)$ then filtering these up-sampled sequences by the linear shift-invariant filters whose impulse responses are $h(k)$ and $g(k)$, respectively. Finally, the filter responses are added to construct the higher resolution coefficients $c_{j+1}(k)$. The following black diagram shown in Figure 1.1 depicts the DWT analysis and synthesis implementation.

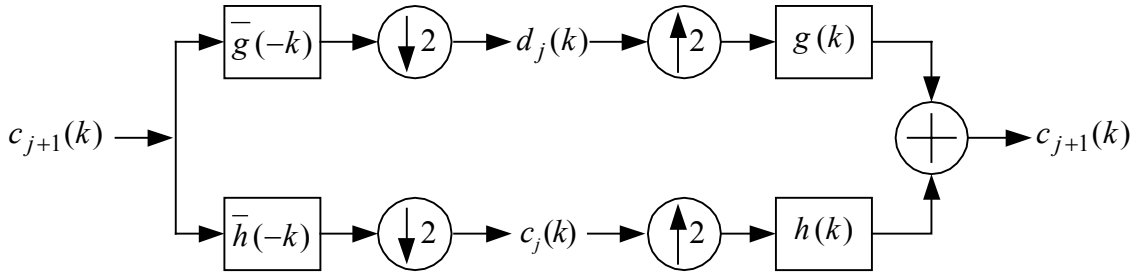

Figure 1.1 One-level of the one-dimensional DWT analysis and synthesis implementation.

For a two-dimensional DWT, there exist various extensions of the one-dimensional DWT to higher dimensions [36]; however, based on Mallat [52], the horizontal and vertical orientations are considered preferential for the two-dimensional DWT implementation. Similarly, in the two-dimensional wavelet analysis, a separable function $\varphi(x, y)$ is introduced such that

$$\varphi(x, y) = \varphi(x)\varphi(y) \tag{1.20}$$

where $\varphi(x)$ is a one-dimensional scaling function.

---

* i.e. retaining only the even or odd indexed elements.

** i.e. inserting zero between each element.

If $\psi(x)$ is the one-dimensional wavelet associated with the scaling function $\varphi(x)$, then the three two-dimensional wavelets are defined as

$$\begin{aligned} \psi^H(x,y) &= \varphi(x)\psi(y) \\ \psi^V(x,y) &= \psi(x)\varphi(y) \\ \psi^D(x,y) &= \psi(x)\psi(y) \ . \end{aligned} \tag{1.21}$$

The one-level of the two-dimensional DWT analysis and synthesis implementation is depicted in Figure 1.2.

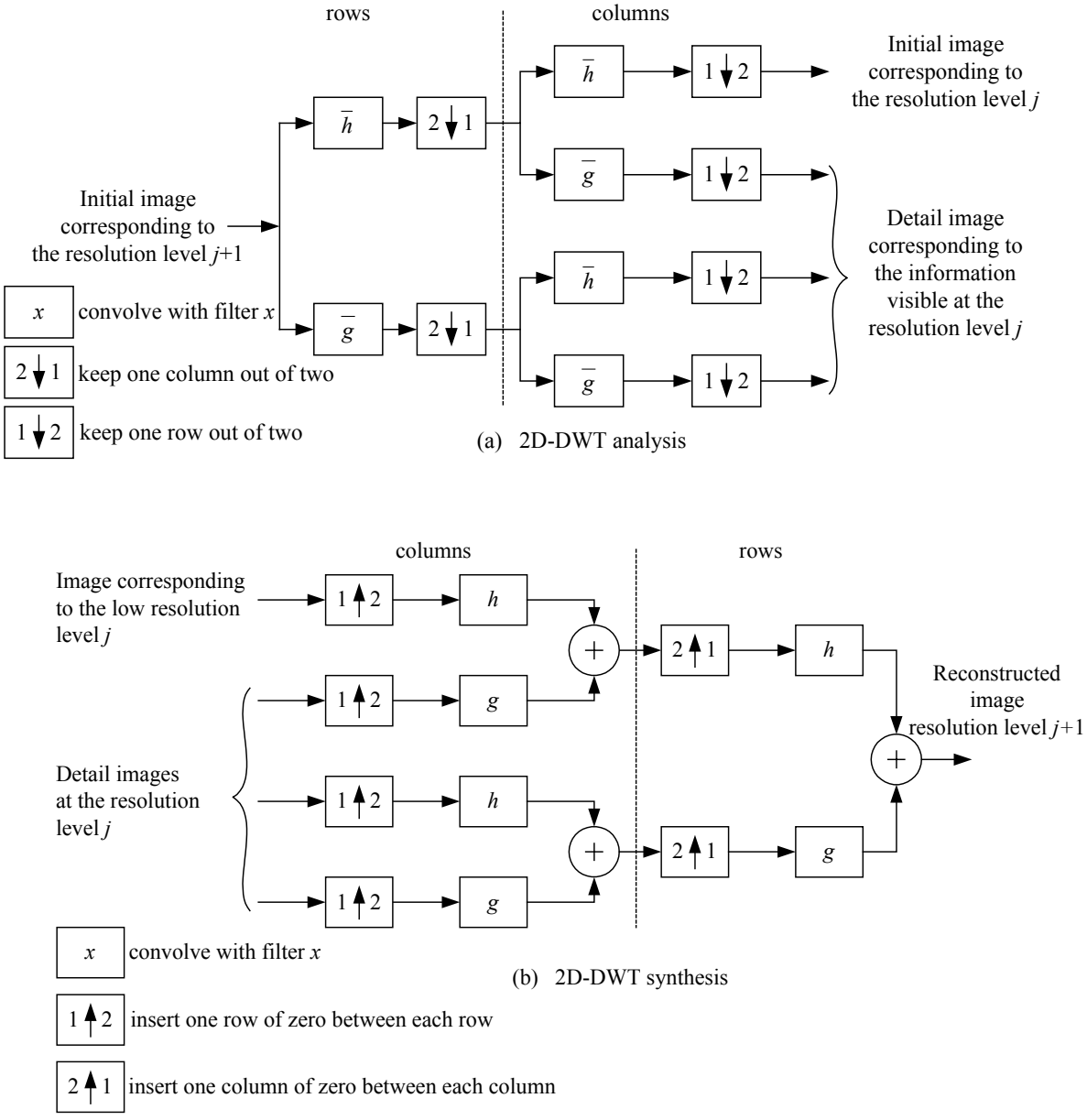

Figure 1.2 One-level of the two-dimensional DWT implementation.

(a) 2D-DWT analysis

(b) 2D-DWT synthesis

S+P-Transform

The S+P-transform was introduced by Said and Pearlman [6] in which is a modification of the S-transform. For a given even sequence $x(n), n = 0,1,...,2^N - 1$, it can be represented by

$$h(n) = x(2n) - x(2n+1) \quad , n = 0,1,...,2^N - 1 \tag{1.22}$$

$$l(n) = \left\lfloor \frac{x(2n) + x(2n+1)}{2} \right\rfloor, n = 0,1,...,2^N - 1 \tag{1.23}$$

where the symbol $\lfloor \cdot \rfloor$ designates downward truncation which $\lfloor x \rfloor$ is the largest integer less than or equal to x.

The $l(n)$ and $h(n)$ sequences generate the S-transform of $x(n)$ and the inverse transformation is given by

$$x(2n) = l(n) + \left\lfloor \frac{h(n)+1}{2} \right\rfloor \tag{1.24}$$

and

$$x(2n+1) = x(2n) - h(n) \ . \tag{1.25}$$

The S-transform is very simple and can be very efficiently calculated. Furthermore, it significantly decreases the first-order entropy. However, due to aliasing from the low frequency components of the original sequence, there is a residual correlation in the highpass components [6]. The S+P-transform was introduced to solve this problem. Based on the fact that the predictive coding need not be linear for perfect reconstruction, the S-transform can then be improved by predictive coding. In the S+P-transform, the S-transform sequences $l(n)$ and $h(n)$ are employed to estimate a new sequence $\hat{h}(n)$ which is given by

$$\hat{h}(n) = \sum_{i=-L_0}^{L_1} \alpha_i \Delta l(n+i) - \sum_{j=-1}^{H} \beta_j h(n+j) \tag{1.26}$$

where $\Delta l(n) = l(n-1) - l(n)$. This $\hat{h}(n)$ sequence is further needed to calculate the different sequence $h_d(n)$ in which

$$h_d(n) = h(n) - \left\lfloor \hat{h}(n) + \frac{1}{2} \right\rfloor, n = 0,1,...2^N - 1. \tag{1.27}$$

The sequence $h(n)$ is replaced by the different sequence $h_d(n)$ forming a new transformed image with smaller first-order entropy. For the inverse transformation, the prediction can be accomplished in a reverse order as follows

$$h(n) = h_d(n) + \left\lfloor \hat{h}(n) + \frac{1}{2} \right\rfloor, n = 2^N - 1, 2^N - 2,...,0. \tag{1.28}$$

The two-dimension S+P-transform can be readily accomplished by applying the described one-dimension S+P-transform sequentially to the columns and rows of the image. However, truncation is a nonlinear operation; this fact makes the order of transformation important. In other words, the inverse transformation must be a backward operation of the forward process. Additionally, the authors have studied three different methods to determine the predictor coefficients α_i and β_j based on minimum entropy, minimum variance, and frequency domain design. Based on the extensive tests with different types of images, the universal predictors, which are effective for a broad class of images, are designed. In this dissertation, the predictor B shown in [6] is employed for this scheme.

Linear Decomposition Transform

The basic concept of the linear decomposition transform (LDT) is to employ the even indexed sequence and the associated l_p norm optimal interpolation even-length filter to approximate the odd indexed sequence. For a given data sequence $x(n)$ for $0 \le n \le 2^N - 1$, the LDT-analysis equations are defined by

$$c(n) = x(2n) \quad \text{and} \quad d(n) = x(2n+1) - \sum_{q=-Q}^{Q-1} h(q)x(2n-2q) \tag{1.29}$$

for $0 \le n \le 2^{N-1} - 1$ and Q is a positive number. The optimum filter coefficients $h(q)$ appearing in expression (1.29) are designed, based on l_p norm error minimization, to cause many of detail coefficients $d(n)$ small in magnitude. Inversely, the original data sequence $x(n)$ is reconstructed by using the following synthesis equations

$$x(2n) = c(n) \quad \text{and} \quad x(2n+1) = d(n) + \sum_{q=-Q}^{Q-1} h(q)c(n-q), \ 0 \le n \le 2^{N-1} - 1. \tag{1.30}$$

This one-dimensional signal decomposition can be generalized to higher dimensions in a straightforward manner. In this dissertation, one of the decorrelation techniques for two-dimensional data proposed in [47], which is called horizontal-vertical-diagonal decomposition (LDT-HVDD), is selected to be a representative of the LDT method. An image X is decomposed into four sub-images defined by X^{EE}, X^{EO}, X^{OE}, and X^{OO}. The even-even indexed image, X^{EE}, is employed to generate the interpolation errors in even-odd, odd-even, and odd-odd subbands as respectively designated by the following analysis equations

$$X^{EE}(m,n) = X(2m,2n) \tag{1.31}$$

$$D^{EO}(m,n) = \overbrace{X(2m,2n+1)}^{X^{EO}(m,n)} - \overbrace{\sum_{q=-Q}^{Q-1} h^{EO}(q)X(2m,2n-2q)}^{\hat{X}^{EO}(m,n)} \tag{1.32}$$

$$D^{OE}(m,n) = \overbrace{X(2m+1,2n)}^{X^{OE}(m,n)} - \overbrace{\sum_{q=-Q}^{Q-1} h^{OE}(q) X(2m-2q,2n)}^{\hat{X}^{OE}(m,n)} \tag{1.33}$$

$$D^{OO}(m,n) = \overbrace{X(2m+1,2n+1)}^{X^{OO}(m,n)} - \overbrace{\sum_{q=-Q}^{Q-1} h^{OO}(q) X(2m-2q,2n-2q)}^{\hat{X}^{OO}(m,n)} \tag{1.34}$$

for $0 \le m \le 2^{M-1}-1, 0 \le n \le 2^{N-1}-1$. In addition, the following synthesis relationships, are employed to reconstruct the even-even, even-odd, odd-even and odd-odd indexed elements of the original image *X*, respectively

$$X(2m,2n) = X^{EE}(m,n) \tag{1.35}$$

$$X(2m,2n+1) = D^{EO}(m,n) + \overbrace{\sum_{q=-Q}^{Q-1} h^{EO}(q) X(2m,2n-2q)}^{\hat{X}^{EO}(m,n)} \tag{1.36}$$

$$X(2m+1,2n) = D^{OE}(m,n) + \overbrace{\sum_{q=-Q}^{Q-1} h^{OE}(q) X(2m-2q,2n)}^{\hat{X}^{OE}(m,n)} \tag{1.37}$$

$$X(2m+1,2n+1) = D^{OO}(m,n) + \overbrace{\sum_{q=-Q}^{Q-1} h^{OO}(q) X(2m-2q,2n-2q)}^{\hat{X}^{OO}(m,n)} \tag{1.38}$$

for $0 \le m \le 2^{M-1}-1, 0 \le n \le 2^{N-1}-1$. Based on this one-level image decompositon and reconstruction, a multilevel image decomposition and reconstruction of the LDT transform is simply constructed. Further detail of the LDT can be found in [47].

Recently, Deng [7] has proposed interpolation based subband decomposition where the median FIR filter is used for the interpolation filter. The illustration of the interpolative filter based subband decomposition is shown in Figure 1.3. In this decomposition technique, the down-sampled image is used to interpolate the missing part of the image in which the down-sampled image is considered as the lowpass subband, whereas the interpolation error is the highpass subband. The interpolation technique does not only provide successful results but it is also an active research area.

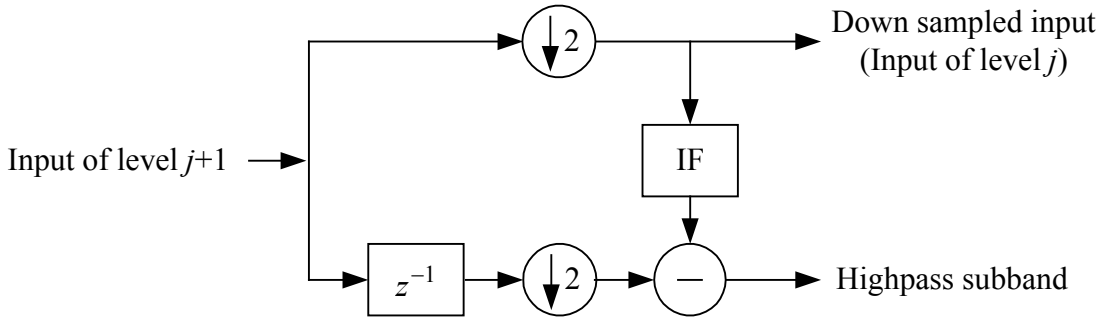

Figure 1.3 Interpolative subband decomposition.

A basic concept of the interpolative subband decomposition is to employ the down-sampled image (treated as the low-pass subband) to interpolate other subbands of image. The proposed image decomposition in this dissertation is also based on this technique. To reduce statistical dependence between pixels of the prediction error, the interpolation filter is optimally designed to minimize error based on l_p norm criterion. The application of the proposed methods is mainly for lossless and lossy image compression. For lossless compression, it is necessary that the prediction error must be integers. This requirement can be directly solved by truncating the output of the interpolation filter to its nearest integer. Contrarily, this procedure is not required for lossy compression. In addition, the proposed methods generate a multiresolution representation of the transform image in which a spatial self-similarity exists between subbands [54]. This observation has been taken advantage of in the set partitioning in hierarchical trees (SPIHT) coding proposed by Said and Pearlman [54]. This coding algorithm is selected to encode the transform coefficients and to measure the compression ratio performance of tested decomposition methods. The organization of this dissertation is given as following.

Dissertation Organization

The analysis of data correlation is first discussed in Chapter II. The decorrelation of the one-dimensional data based on the proposed one-dimensional decomposition technique is then introduced. Later, this technique is applied to the two-dimensional data where three different techniques: 1) Horizontal Vertical Decomposition (HVD) 2) Horizontal Vertical Vertical Decomposition (HVVD) 3) Horizontal Vertical Diagonal Decomposition (HVDD) are proposed. To improve the decorrelation performance for the two-dimensional data, an alternative two-

dimensional decorrelation technique is therefore introduced in Chapter III. Generally, the optimum scalars and interpolation filter coefficients of the proposed techniques discussed in Chapter II and Chapter III, respectively, are designed base on finding of the minimum l_p norm solution, where the indices $p = 1$ and $p = 2$ are primarily examined. Theorems associated with such algorithms are therefore addressed in Chapter IV. The proposed techniques, the S+P-transform, and the wavelet transform are implemented on the MATLABR12 program [55] with a set of example grayscale images in which the experimental results and discussion are provided in Chapter V. Finally, Chapter VI is devoted to the conclusion of this research.

CHAPTER II

OPTIMUM SCALARS DECOMPOSITION

One-Dimensional Data Decorrelation

Pixel correlation of the image plays an important role in image decomposition. As a result, several techniques have been proposed over the years [3][6][56][57][58][59]. The main concept of these methods is generally based on taking advantage of correlation among neighboring pixels of the image. To gain insight into the importance of pixel correlation, let consider a one-dimensional real data sequence $\{x(n)\}$ defined on the interval $1 \le n \le 2^N$ where N a positive integer, that is

$$\{x(1), x(2), x(3), ..., x(2^N - 2), x(2^N - 1), x(2^N)\}. \tag{2.1}$$

The data sequence $\{x(n)\}$ depicted in Figure 2.1 can be generated either by an inherently discrete time process or by sampling a continuous time function.

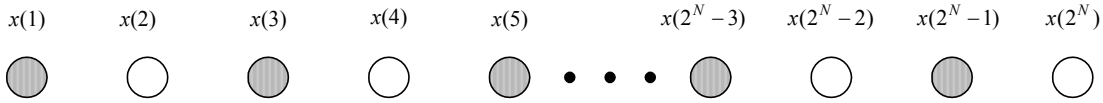

Figure 2.1 The illustration of a one-dimensional data sequence $\{x(n)\}$ where the shading and non-shading circles represent odd and even indexed components, respectively.

It is assumed that neighboring sequence elements of $\{x(n)\}$ are highly correlated. In many applications, it is desired to represent the information contained in $\{x(n)\}$ with fewer samples than its prototype. As can be seen in Figure 2.1, $x(2)$ is somehow associated with its adjacent elements $x(1)$ and $x(3)$, $x(4)$ is associated with its adjacent elements $x(3)$ and $x(5)$, and similarly for other odd indexed elements. Based on this concept, the following decorrelation technique is introduced. The original data sequence $\{x(n)\}$ is preliminarily decomposed into two

disjoint half-length sequences $\{x_o(n)\}$ and $\{x_e(n)\}$ whose their components are odd and even indexed elements of the original sequence, respectively, as designated by

$$\{\underline{x}_o, \underline{x}_e\} = T(\underline{x}) \tag{2.2}$$

where the odd indexed elements are specified by

$$x_o(n) = x(2n-1) \quad for \quad 1 \le n \le 2^{N-1} \tag{2.3}$$

and the even indexed sequence elements are given by

$$x_e(n) = x(2n) \quad for \quad 1 \le n \le 2^{N-1}. \tag{2.4}$$

The decomposition operator T is linear since it is readily verified that $T(a\underline{x}_1 + b\underline{x}_2) = T(a\underline{x}_1) + T(b\underline{x}_2)$ holds for all real scalars a, b and all real vectors $\underline{x}_1$, $\underline{x}_2$. This transformation is clearly linear and also invertible. The invertibility is due to the fact that, for given information of the odd and even indexed half-length sequences, the original full-length data sequence can be recovered. The inverse procedure can be written as

$$\underline{x} = T^{-1}\{\underline{x}_o, \underline{x}_e\}. \tag{2.5}$$

Additionally, the described decomposition and reconstruction procedure can be simply operated by using an invertible structure called a *Lazy Wavelet transform* [49] as shown in Figure 2.2.

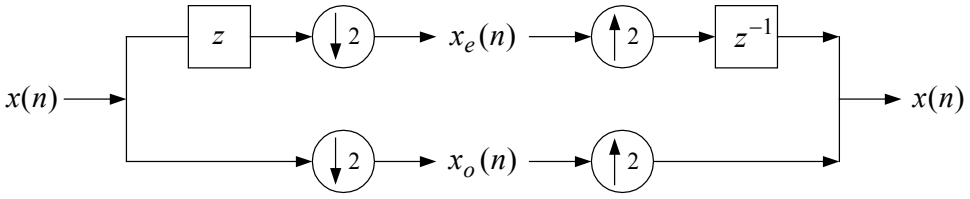

Figure 2.2 The Lazy wavelet transform.

Figure 2.2 illustrates an overview of splitting and merging data processes. The odd indexed sequence is obtained by down-sampling by two the input sequence depicted as a circle enclosing a downward pointing arrow and the number two adjacent to the arrow. Likewise, the even indexed sequence is obtained by employing a similar procedure, except a standard unit left-shift operator symbolized as a letter z enclosed with a square is used before the down-sampling process. As a result up to this point, the Lazy wavelet transform maps the input data sequence into two half-rate sequences $\{x_o(n)\}$ and $\{x_e(n)\}$. Furthermore, Figure 2.2 implies that the Lazy wavelet transform is always invertible in which the original full-length data sequence can be perfectly recovered given knowledge of the two half-length sequences $\{x_o(n)\}$ and $\{x_e(n)\}$. As depicted in Figure 2.2, the up-sampling by two operation represented by a circle enclosing an upward pointing arrow with the number two adjacent to it maps two half-length data sequences into the full-length data sequence. The reconstructed odd and even indexed elements are respectively given by

$$y_o(n) = \begin{cases} x_o\big((n+1)/2\big) & \textit{for } n \textit{ is odd} \\ 0 & \textit{for } n \textit{ is even} \end{cases} \tag{2.6}$$

and

$$y_e(n) = \begin{cases} 0 & \textit{for } n \textit{ is odd} \\ x_e\big(n/2\big) & \textit{for } n \textit{ is even}. \end{cases} \tag{2.7}$$

In the process of reconstruction, a unit right-shift operator symbolized by a letter z^{-1} enclosed with a square is employed after up-sampling process for the even indexed sequence. By substituting equations (2.3) and (2.4) into equations (2.6) and (2.7), respectively, this then yields

$$y_o(n) = \begin{cases} x(n) & \textit{for } n \textit{ is odd} \\ 0 & \textit{for } n \textit{ is even} \end{cases} \tag{2.8}$$

and

$$y_e(n) = \begin{cases} 0 & \textit{for } n \textit{ is odd} \\ x(n) & \textit{for } n \textit{ is even} \end{cases} \tag{2.9}$$

where the summation of $\{y_o(n)\}$ and $\{y_e(n)\}$ sequences reconstructs the original full-length data sequence $\{x(n)\}$. With no loss in generality, let $\underline{x}_o$ and $\underline{x}_e$ be row vectors* of odd and even indexed sequences, respectively, whose components are $x_o(n)$ and $x_e(n) \in R$ for $1 \le n \le 2^{N-1}$. By setting $\underline{x}_o$ to be a reference vector and using relationship between odd and even indexed sequences, it then follows that $\underline{x}_e$ can be approximated by a scalar multiple of $\underline{x}_o$ as designated by

$$\hat{\underline{x}}_e = \alpha \underline{x}_o \tag{2.10}$$

where $\hat{\underline{x}}_e$ is an approximated version of $\underline{x}_e$. The scalar α is unrestricted and is selected so as to minimize the l_p norm of the function $\left\| \underline{x}_e - \alpha \underline{x}_o \right\|_p$ for $1 \le p \le \infty$.** This generates an error vector of approximation given by

$$\underline{d} = \underline{x}_e - \alpha \underline{x}_o . \tag{2.11}$$

Moreover, the even indexed vector can be recovered by given the knowledge of $\underline{x}_o$ and $\underline{d}$ as designated by

$$\underline{x}_e = \underline{d} + \alpha \underline{x}_o . \tag{2.12}$$

As can be seen in expressions (2.10) – (2.12), there exists an equivalency between the following vector pairs

$$\{\underline{x}_o, \underline{x}_e\} \Leftrightarrow \{\alpha, \underline{x}_o, \underline{d}\} \tag{2.13}$$

---

* The column vector form is also a possible choice.

** The role between $\underline{x}_o$ and $\underline{x}_e$ is interchangeable without loss of generality, namely $\hat{\underline{x}}_o = \alpha \underline{x}_e$.

which implies that given one set, the other set can be obtained irrespectively of how a scalar α is chosen. The decomposition and reconstruction of this transform can be defined as follows.

Definition 2.1 : Optimum Scalar Transform

Let T_{os} be an operator of the l_p-norm optimum scalar transform decomposing an even-length input data sequence $\{x(n)\}$ into two half-length sequences $\{c(n)\}$ and $\{d(n)\}$. The optimum scalar transform based decomposition is designated by

$$(\{c(n)\},\{d(n)\}) = T_{os}(\{x(n)\}). \tag{2.14}$$

The consequent transformed data sequences, $\{c(n)\}$ and $\{d(n)\}$, then form the *analysis* equations as given by

$$c(n) = x(2n-1) \quad for \quad 1 \le n \le 2^{N-1} \tag{2.15}$$

and

$$d(n) = x(2n) - \alpha^o x(2n-1) \quad for \quad 1 \le n \le 2^{N-1}. \tag{2.16}$$

The one-level optimum scalar transform based decomposition is illustrated in Figure 2.3.

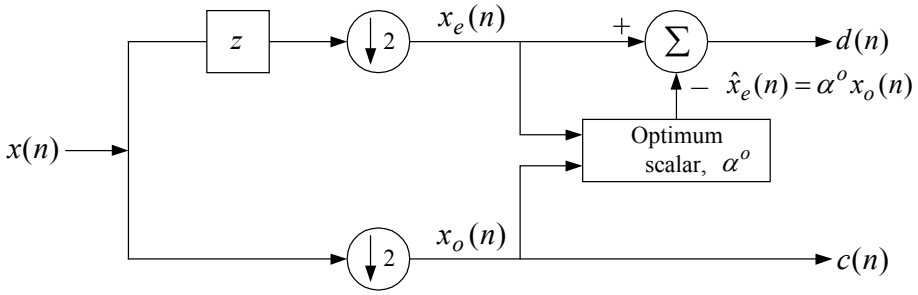

Figure 2.3 One-level optimum scalar transform based decomposition.

The optimum scalar transform is a linear operation, which complies with a principle of superposition. It is readily proved that the optimum scalar transform of a linear combination of any two arbitrary input data sequences is equal to a linear combination of the optimum scalar transform associated with two data sequences. Mathematically, let $\{x_1(n)\}$ and $\{x_2(n)\}$ be any two arbitrary input data sequences. A linear combination of two sequences is therefore given by

$$x(n) = ax_1(n) + bx_2(n) \tag{2.17}$$

where *a* and *b* are arbitrary constants. The responses of the optimum scalar transform associated with the input data sequences $\{x_1(n)\}$ and $\{x_2(n)\}$ are $(\{c_1(n)\},\{d_1(n)\})$ and $(\{c_2(n)\},\{d_2(n)\})$, respectively. Hence, expressions (2.15) and (2.16) can be respectively rewritten as

$$\begin{aligned} c(n) &= x(2n-1) \\ &= ax_1(2n-1) + bx_2(2n-1) \\ &= ac_1(n) + bc_2(n) \end{aligned} \tag{2.18}$$

and

$$\begin{aligned} d(n) &= x(2n) - \alpha^o x(2n-1) \\ &= \left(ax_1(2n) + bx_2(2n)\right) - \alpha^o\left(ax_1(2n-1) + bx_2(2n-1)\right) \\ &= a\left(x_1(2n) - \alpha^o x_1(2n-1)\right) + b\left(x_2(2n) - \alpha^o x_2(2n-1)\right) \\ &= ad_1(n) + bd_2(n)\,. \end{aligned} \tag{2.19}^*$$

As shown in expressions (2.18) and (2.19), the linearity of the optimum scalar transform is thus established. However, this process is not shift invariant due to the existence of the down-sampling by two operator.

---

* where $\alpha^o = \dfrac{\underline{x}_e \underline{x}_o^T}{\underline{x}_o^T \underline{x}_o^T} = \dfrac{(a\underline{x}_{1e} + b\underline{x}_{2e})(a\underline{x}_{1o} + b\underline{x}_{2o})^T}{(a\underline{x}_{1o} + b\underline{x}_{2o})(a\underline{x}_{1o} + b\underline{x}_{2o})^T}$.

Definition 2.2 : The Inverse of Optimum Scalar Transform

Let T_{os}^{-1} be the operator of the inverse optimum scalar transform mapping two half-length data sequences $\{c(n)\}$ and $\{d(n)\}$ into a double-length data sequence $\{x(n)\}$. The optimum scalar transform based reconstruction is designated by

$$\left(\{x(n)\}\right) = T_{os}^{-1}\left(\{c(n)\},\{d(n)\}\right) \tag{2.20}$$

which is readily shown to be linear. The original odd and even indexed data sequences are given by the following *synthesis* equations

$$x(2n-1) = c(n) \quad for \quad 1 \le n \le 2^{N-1} \tag{2.21}$$

and

$$x(2n) = d(n) + \alpha^{o} c(n) \quad for \quad 1 \le n \le 2^{N-1}. \tag{2.22}$$

The diagram of one-level optimum scalar transform based reconstruction is shown in Figure 2.4. This procedure can be generalized to multilevel decomposition and reconstruction in a straightforward manner; however, such detail is excluded in this research. The following section discusses the application of this technique to decorrelate two-dimensional data.

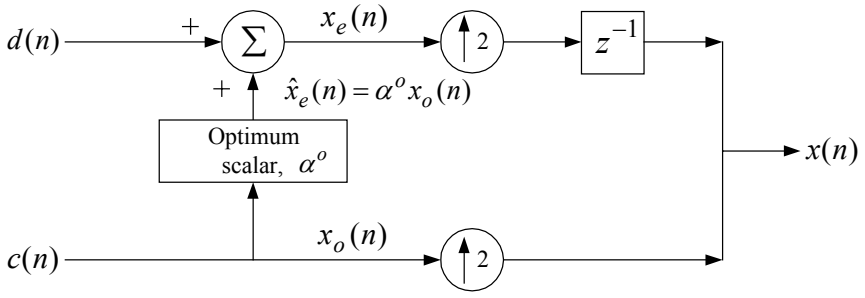

Figure 2.4 One-level optimum scalar transform based reconstruction.

Two-Dimensional Data Decorrelation

With the given notion of one-dimensional data correlation in mind, the similar concept can be generalized in a straightforward manner to two-dimensional data such as an image. Let there be given a digital image X derived from the sampling process (digitization) of a two-dimensional continuous space. The image takes a form of a $2^M \times 2^N$ matrix whose elements or pixels $X(m,n)$ are defined on a two-dimensional discrete space, in which $1 \le m \le 2^M$, $1 \le n \le 2^N$ and M,N are even integers. The value assigned on each pixel is the image intensity (gray level), which is the average brightness in the pixel rounded to the nearest integer value. An example of the digitization effect is illustrated in Figure 2.5. The image is therefore decorrelated by the following techniques.

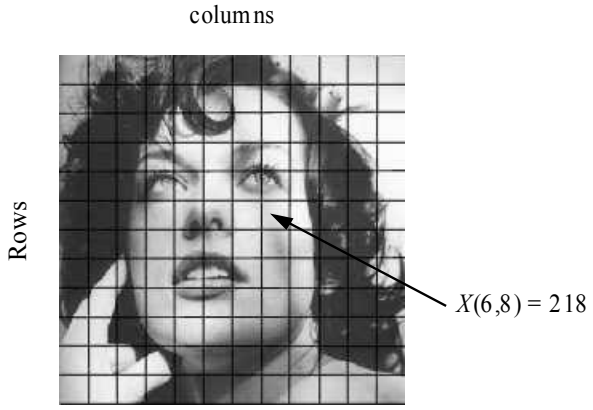

Figure 2.5 An example of digitization of a continuous image where the pixel at coordinates (m = 6, n = 8) has the integer brightness value 218.

Optimum Scalar-Horizontal Vertical Decomposition (OS-HVD) Method

An image is first decomposed by taken advantage of column pixels correlation (horizontal correlation) and then row pixels correlation (vertical correlation).* In other words, the image X is decomposed into two sub-images X^{AO} and X^{AE} whose their 2^{N-1} column

* The role between horizontal and vertical correlation is interchangeable without loss of generality.

vectors are specified by $X\underline{e}_{2n-1}$ and $X\underline{e}_{2n}$, respectively, for $1 \le n \le 2^{N-1}$.* The decomposition procedure is designated by

$$\left\{X^{AO}, X^{AE}\right\} = T_c(X). \tag{2.23}$$

This relationship implies that the even column indexed image X^{AE} can be approximated by a scalar multiple of its adjacent odd column indexed image X^{AO} and the associated error matrix is given by

$$D^{AE} = X^{AE} - \alpha X^{AO}. \tag{2.24}$$

The choice for scalar α in this analysis is unrestricted; however, it is subsequently chosen so that the matrix norm $\left\|D^{AE}\right\| = \left\|X^{AE} - \alpha X^{AO}\right\|$ is minimized. Any norm matrix can be used in the above matrix approximation problem; nevertheless, the norm employed is here restricted to the l_p induced norm matrix as specified by

$$\left\|D^{AE}\right\|_p = \left[\sum_{m=1}^{M}\sum_{n=1}^{N}\left|D^{AE}(m,n)\right|^p\right]^{1/p} \tag{2.25}$$

in which the norm index *p* can be any number satisfying $p \ge 1$. Among these norms, the choices for *p* = 1 and *p* = 2 have received widespread interested [60][61][62][63][64][65][66] and will be mainly employed in this research. To solve the optimized problem in expression (2.25), it is useful to represent a matrix by its equivalent column vector concatenated form. Let *X* be any $P \times Q$ matrix. Its equivalent column vector concatenated form $\underline{x} \in R^{PQ\times 1}$ is then designated by

$$\underline{x} = C_{P,Q}(X) = \left[\underline{x}_1\ \underline{x}_2 \cdots \underline{x}_Q\right]^T \tag{2.26}$$

---

* $\underline{e}_n$ is the standard basis vector whose components are all zero except for its n^{th} element which is one.

where $\underline{x}_q = X\underline{e}_q$ for $1 \le q \le Q$ are the q^{th} column vector of the matrix X. The concatenated column transformation $C_{P,Q} : R^{P\times Q} \to R^{PQ\times 1}$ is linear as well as invertible. The inverse transformation, which produces the original matrix $X \in R^{P\times Q}$, is given by

$$C_{P,Q}^{-1}(\underline{x}) = X . \tag{2.27}^{*}$$

Upon employing this concatenated column transformation, the original optimized problem can be likewise expressed as

$$\begin{aligned} \left\| D^{AE} \right\|_p &= \left\| X^{AE} - \alpha^o X^{AO} \right\|_p \\ &= \left\| \underline{x}^{AE} - \alpha^o \underline{x}^{AO} \right\|_p \\ &= \min_{\alpha\in R} \left\| \underline{x}^{AE} - \alpha \underline{x}^{AO} \right\|_p \end{aligned} \tag{2.28}$$

in which $\underline{x}^{AO} = C_{2^M,2^{N-1}}\left(X^{AO}\right)$ and $\underline{x}^{AE} = C_{2^M,2^{N-1}}\left(X^{AE}\right)$ are column vector concatenated transforms of the matrices X^{AO} and X^{AE}, respectively. Using this relationship, the matrix X^{AE} can be recovered by

$$X^{AE} = D^{AE} + \alpha X^{AO} . \tag{2.29}$$

Thus, regardless to the choice of the scalar α, the equivalency between expressions (2.24) and (2.29) is expressed as the following matrix pairs

$$\left\{X^{AO}, X^{AE}\right\} \Leftrightarrow \left\{\alpha, X^{AO}, D^{AE}\right\} \tag{2.30}$$

---

* It is necessary to explicitly employ the subscript order information since the vector $\underline{x} \in R^{PQ\times 1}$ can also produce the *2P×Q/2* matrix as $C_{2P,Q/2}^{-1}(\underline{x})$.

which implies that given one set, the other set can be generated. Additionally, the inverse transformation is thus given by

$$X = T_c^{-1}\left\{X^{AO}, X^{AE}\right\}. \tag{2.31}$$

In the same manner, the existing $2^M \times 2^{N-1}$ matrix X^{AO} is subsequently decomposed into two matrices X^{OO} and X^{EO} where 2^{M-1} row vectors are specified by $\underline{e}_{2m-1}^T X^{AO}$ and $\underline{e}_{2m}^T X^{AO}$, respectively, for $1 \le m \le 2^{M-1}$. The vertical decomposition procedure is defined by

$$\left\{X^{OO}, X^{EO}\right\} = T_r(X^{AO}). \tag{2.32}$$

The vertical decomposition based on correlation between adjacent rows of the image can be accomplished by approximating row even indexed image X^{EO} as a scalar multiple of adjacent row odd indexed image X^{OO} which is expressed by

$$X^{EO} = D^{EO} + \beta X^{OO}. \tag{2.33}$$

Without regard to the choice of the scalar β, it is unrestricted, but subsequently selected to minimize the norm induced function $\left\|\underline{x}^{EO} - \beta \underline{x}^{OO}\right\|_p$ where $\underline{x}^{EO}$ and $\underline{x}^{OO}$ are the equivalent column vector concatenated form of X^{EO} and X^{OO} matrices, respectively*. As a result, it provides the equivalency between the following matrix pairs

$$\left\{X^{OO}, X^{EO}\right\} \Leftrightarrow \left\{\beta X^{OO}, D^{EO}\right\}. \tag{2.34}$$

By considering expressions (2.32) and (2.33), the reconstruction of the image X^{AO} is given by

---

* where $\underline{x} = C_{P,Q}(X) = \left[\underline{x}_1^T\ \underline{x}_2^T \cdots \underline{x}_P^T\right]^T$ for $\underline{x}_p = \underline{e}_p^T X$ for $1 \le p \le P$ are the p^{th} row vector of the matrix X.

$$
\begin{aligned}
X^{AO} &= T_r^{-1}\left(X^{OO}, D^{EO} + \beta X^{OO}\right) \\
&= T_r^{-1}\left(X^{OO}, \beta X^{OO}\right) + T_r^{-1}\left(O, D^{EO}\right)
\end{aligned} \tag{2.35}
$$

where the $2^{M-1} \times 2^{N-1}$ matrix O is assigned to be zero matrix. In addition, the even column indexed image X^{AE} of the original image X can be reconstructed by

$$
\begin{aligned}
X^{AE} &= D^{AE} + \alpha X^{AO} \\
&= T_r^{-1}\left(D^{OE}, D^{EE}\right) + \alpha\left(T_r^{-1}\left(X^{OO}, \beta X^{OO}\right) + T_r^{-1}\left(O, D^{EO}\right)\right) \\
&= \alpha\left(T_r^{-1}\left(X^{OO}, \beta X^{OO}\right)\right) + T_r^{-1}\left(D^{OE}, \alpha D^{EO} + D^{EE}\right)
\end{aligned} \tag{2.36}
$$

where $\left\{D^{OE}, D^{EE}\right\} = T_r\left(D^{AE}\right)$. In accordance with expressions (2.35) and (2.36), the original image X is perfectly recovered from the given information of the entities $\alpha, \beta, X^{OO}, D^{OE}, D^{EE}, D^{EO}$ as symbolically expressed by

$$
X \Leftrightarrow \left\{\alpha, \beta, X^{OO}, D^{OE}, D^{EE}, D^{EO}\right\} \tag{2.37}
$$

where $X^{OO}, D^{OE}, D^{EE}, D^{EO}$ are $2^{M-1} \times 2^{N-1}$ matrices. This relationship is clearly independent on the choice of scalars α and β.

The one-level OS-HVD method can be generalized to the multilevel process. At end stage of the one-level HVD method, an $2^M \times 2^N$ image X is decomposed into four $2^{M-1} \times 2^{N-1}$ sub-images $X_1^{OO}, D_1^{OE}, D_1^{EE}$ and D_1^{EO} where the subscript is used to explicitly recognize the level of image decomposition. For the second level of decomposition, the sub-image X_1^{OO} is decomposed into four $2^{M-2} \times 2^{N-2}$ sub-images $X_2^{OO}, D_2^{OE}, D_2^{EE}$ and D_2^{EO}. The $2^{M-2} \times 2^{N-2}$ sub-image X_2^{OO} is further decomposed into four $2^{M-3} \times 2^{N-3}$ sub-matrices $X_3^{OO}, D_3^{OE}, D_3^{EE}$ and D_3^{EO} at the third level of the process. The three-level OS-HVD method is demonstrated in Figure 2.6. The decomposition procedure can be continued in a similar fashion to the L^{th} level, for which $2^L \le \max\left\{2^M, 2^N\right\}$, generating a set of $3L+1$ sub-images as follows

$$X \Leftrightarrow \left\{X_L^{OO}, D_L^{OE}, D_L^{EE}, D_L^{EO}, D_{L-1}^{OE}, D_{L-1}^{EE}, D_{L-1}^{EO}, ..., D_1^{OE}, D_1^{EE}, D_1^{EO}\right\}. \tag{2.38}$$

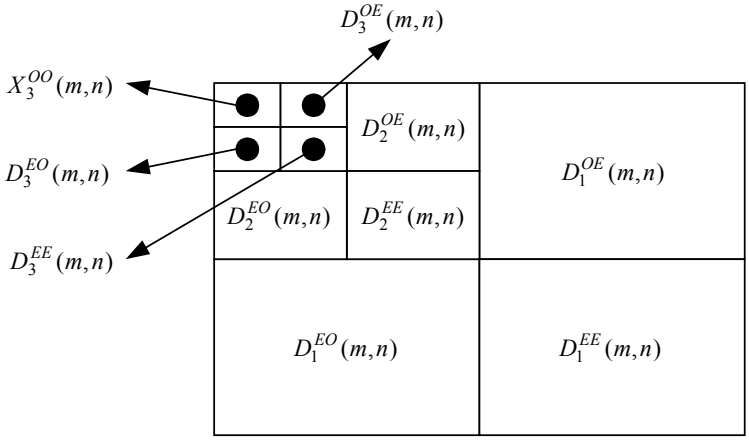

Figure 2.6 Three-level of OS-HVD method.

Optimum Scalar-Horizontal Vertical Vertical Decomposition (OS-HVVD) Method

This method is an extended version of the OS-HVD method. Upon examination at each level of HVD method, one can apply the row-based decomposition to the detail matrix D^{AE}. Consequently, it follows that the matrix D^{EE} can be approximated by a scalar multiple of D^{EE} as γD^{OE} where the scalar γ is made to minimize the norm induced function $\left\|\tilde{D}^{EE}\right\| = \left\|D^{EE} - \gamma D^{OE}\right\|$. There also exists the equivalency of the following two sets

$$\left\{D^{OE}, D^{EE}\right\} \Leftrightarrow \left\{\gamma, D^{OE}, \tilde{D}^{EE}\right\} \tag{2.39}$$

where the matrix D^{AE} can be perfectly recovered by

$$D^{AE} = T_r^{-1}\left(D^{OE}, D^{EE}\right) = T_r^{-1}\left(D^{OE}, \tilde{D}^{EE} + \gamma D^{OE}\right) \tag{2.40}$$

for given knowledge of the scalar γ, the detail matrix D^{OE} and $\tilde{D}^{EE}$.

With this additional decomposition performed on the detail matrix D^{AE}, the even column indexed image X^{AE} of the original image X can thus be reconstructed by

$$\begin{aligned} X^{AE} &= D^{AE} + \alpha X^{AO} \\ &= T_r^{-1}\left(D^{OE}, \tilde{D}^{EE} + \gamma D^{OE}\right) + \alpha\left(T_r^{-1}\left(X^{OO}, \beta X^{OO} + D^{EO}\right)\right) \\ &= \alpha\left(T_r^{-1}\left(X^{OO}, \beta X^{OO}\right)\right) + T_r^{-1}\left(D^{OE}, \alpha D^{EO} + \gamma D^{OE} + \tilde{D}^{EE}\right). \end{aligned} \tag{2.41}$$

Expression (2.41) indicates that the matrix X^{AE} has been decomposed into the sum of two matrices. One depends only on the matrix X^{OO} whereas another one depends only on the detail matrices D^{OE}, D^{EO}, and $\tilde{D}^{EE}$. By employing equations (2.40) and (2.41) and giving the knowledge of the entities $\alpha, \beta, \gamma, X^{OO}, D^{OE}, D^{EO}$, and $\tilde{D}^{EE}$, the original image X can be perfectly recovered as symbolically expressed by

$$X \Leftrightarrow \left\{\alpha, \beta, \gamma, X^{OO}, D^{OE}, D^{EO}, \tilde{D}^{EE}\right\} \tag{2.42}$$

where $X^{OO}, D^{OE}, D^{EO}, \tilde{D}^{EE}$ are $2^{M-1} \times 2^{N-1}$ matrices. This relationship implies the independency of the choice of real scalars α, β and γ.

A multilevel OS-HVVD method can be achieved by repeating the similar decomposition on the $2^{M-1} \times 2^{N-1}$ sub-image X_j^{OO} (for any current level *j*) which produces four $2^{M-2} \times 2^{N-2}$ sub-images $X_{j+1}^{OO}, D_{j+1}^{OE}, D_{j+1}^{EO}$, and $\tilde{D}_{j+1}^{EE}$. This procedure may continue to the L^{th} level, for which $2^L \le \max\left\{2^M, 2^N\right\}$, generating a set of 3*L*+1 sub-images as given by

$$X \Leftrightarrow \left\{X_L^{OO}, D_L^{OE}, D_L^{EO}, \tilde{D}_L^{EE}, D_{L-1}^{OE}, D_{L-1}^{EO}, \tilde{D}_{L-1}^{EE}, ..., D_1^{OE}, D_1^{EO}, \tilde{D}_1^{EE}\right\}. \tag{2.43}$$

With no loss in generality, let the level of decomposition *L* be 3, then three-level OS-HVVD method can be depicted in Figure 2.7.

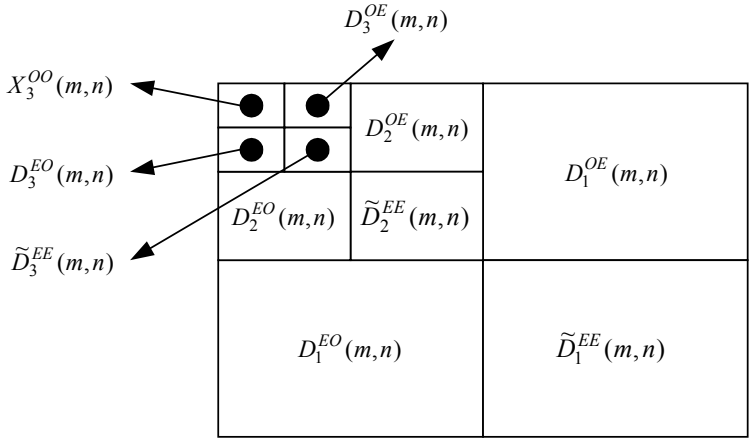

Figure 2.7 Three-level OS-HVVD method.

Optimum Scalar-Horizontal Vertical Diagonal Decomposition (OS-HVDD) Method

This method is slightly different from two previous methods in that the image is not only decorrelated in the horizontal and vertical directions, but also in the diagonal direction. Let X be any $2^M \times 2^N$ image and be decomposed as

$$\left\{X^{OA}, X^{EA}\right\} = T_r\left(X\right) \tag{2.44}$$

where X^{OA} and X^{EA} have their 2^{M-1} row vectors specified by $\underline{e}_{2m-1}^T X$ and $\underline{e}_{2m}^T X$, respectively, for $1 \le m \le 2^{M-1}$. Thereafter, the matrices X^{OA} and X^{EA} are decomposed in the horizontal direction as follows

$$\left\{X^{OO}, X^{OE}\right\} = T_c\left(X^{OA}\right) \tag{2.45}$$

where the $2^{M-1} \times 2^{N-1}$ matrices X^{OO} and X^{OE} have their 2^{N-1} column vectors specified by $X^{OA}\underline{e}_{2n-1}$ and $X^{OA}\underline{e}_{2n}$, respectively, for $1 \le n \le 2^{N-1}$

and $$\left\{X^{EO}, X^{EE}\right\} = T_c\left(X^{EA}\right) \tag{2.46}$$

where the $2^{M-1} \times 2^{N-1}$ matrices X^{EO} and X^{EE} have their 2^{N-1} column vectors specified by $X^{EA}\underline{e}_{2n-1}$ and $X^{EA}\underline{e}_{2n}$, respectively, for $1 \le n \le 2^{N-1}$. Since X^{OA} and X^{EA} have their row elements adjacent to each other, it implies that they are highly correlated to one another as well as their associated matrices, X^{OO}, X^{OE}, X^{EO}, and X^{EE}. By taking the matrix X^{OO} as a reference matrix, this matrix is correlated with the matrices X^{OE}, X^{EO}, and X^{EE} in horizontal, vertical, and diagonal directions, respectively. Let us employ this advantage to approximate the matrices X^{OE}, X^{EO}, and X^{EE} by scalar multiple of X^{OO}. Hence, the error of the approximated matrices and the inverse operations associated with the matrices X^{OE}, X^{EO}, and X^{EE} can be respectively expressed by

$$D^{OE} = X^{OE} - \alpha X^{OO} \quad \text{and} \quad X^{OE} = D^{OE} + \alpha X^{OO} \tag{2.47}$$

$$D^{EO} = X^{EO} - \beta X^{OO} \quad \text{and} \quad X^{EO} = D^{EO} + \beta X^{OO} \tag{2.48}$$

and

$$D^{EE} = X^{EE} - \gamma X^{OO} \quad \text{and} \quad X^{EE} = D^{EE} + \gamma X^{OO} \tag{2.49}$$

where D^{OE}, D^{EO}, D^{EE} are $2^{M-1} \times 2^{N-1}$ matrices and all real scalars α, β, γ are selected so that the associated norm induced functions $\left\|D^{OE}\right\| = \left\|X^{OE} - \alpha X^{OO}\right\|_p$, $\left\|D^{EO}\right\| = \left\|X^{EO} - \beta X^{OO}\right\|_p$, and $\left\|D^{EE}\right\| = \left\|X^{EE} - \gamma X^{OO}\right\|_p$ are minimized, respectively.

Likewise, there exists the equivalent relationship between the following matrix pairs as designated by

$$\left\{X^{OO}, X^{OE}\right\} \Leftrightarrow \left\{\alpha, X^{OO}, D^{OE}\right\}, \tag{2.50}$$

$$\left\{X^{OO}, X^{EO}\right\} \Leftrightarrow \left\{\beta, X^{OO}, D^{EO}\right\}, \tag{2.51}$$

and

$$\left\{X^{OO}, X^{EE}\right\} \Leftrightarrow \left\{\gamma, X^{OO}, D^{EE}\right\}. \tag{2.52}$$

Additionally, the matrices X^{OA} and X^{EA} are recoverable by the following relationships

$$\begin{aligned} X^{OA} &= T_c^{-1}\left(X^{OO}, X^{OE}\right) \\ &= T_c^{-1}\left(X^{OO}, \alpha X^{OO} + D^{OE}\right) \\ &= T_c^{-1}\left(X^{OO}, \alpha X^{OO}\right) + T_c^{-1}\left(O, D^{OE}\right) \end{aligned} \tag{2.53}$$

where the $2^{M-1} \times 2^{N-1}$ matrix O is defined to be a zero matrix, and

$$\begin{aligned} X^{EA} &= T_c^{-1}\left(X^{EO}, X^{EE}\right) \\ &= T_c^{-1}\left(\beta X^{OO} + D^{EO}, \gamma X^{OO} + D^{EE}\right) \\ &= T_c^{-1}\left(\beta X^{OO}, \gamma X^{OO}\right) + T_c^{-1}\left(D^{EO}, D^{EE}\right). \end{aligned} \tag{2.54}$$

Equation (2.53) indicates that the matrix X^{OA} can be decomposed into the sum of a matrix that depends only on the matrix X^{OO} and a matrix that depends only on the detail (error) matrix D^{OE}. In the same fashion, as shown in expression (2.54), the matrix X^{EA} can be decomposed into the sum of a matrix that depends only on the matrix X^{OO} and a matrix that depends only on the detail (error) matrices D^{EO} and D^{EE}. Hence, with the knowledge of the following entities $\alpha, \beta, \gamma, X^{OO}, D^{OE}, D^{EO}, D^{EE}$, the original image X can be reconstructed independently to the choice of the real scalars α, β, γ as symbolically expressed by

$$X \Leftrightarrow \left\{\alpha, \beta, \gamma, X^{OO}, D^{OE}, D^{EO}, D^{EE}\right\} \tag{2.55}$$

where $X^{OO}, D^{OE}, D^{EO}, D^{EE}$ are $2^{M-1} \times 2^{N-1}$ matrices.

A multilevel based OS-HVDD method is a simple generalization of the one level process in which at the final of the first level of OS-HVDD method. Let L be the total level of decomposition, for which $2^L \le \max\left\{2^M, 2^N\right\}$, it then follows that a set of $3L+1$ sub-images is generated as given by

$$X \Leftrightarrow \left\{X_L^{OO}, D_L^{OE}, D_L^{EO}, D_L^{EE}, D_{L-1}^{OE}, D_{L-1}^{EO}, D_{L-1}^{EE}, ..., D_1^{OE}, D_1^{EO}, D_1^{EE}\right\}. \tag{2.56}$$

The optimum scalar decomposition has been discussed in this chapter. The proposed methods based on this decomposition technique for two-dimensional data case will be processed to the images. Another proposed technique motivated by the LDT-HVD method is described in the following chapter.

CHAPTER III

TWO-DIMENSIONAL-HORIZONTAL VERTICAL DIAGONAL DECOMPOSITION

The optimum scalar decomposition described in the previous chapter and the linear decomposition transform employ one-dimensional decorrelation technique to decorrelate two-dimensional data. This technique decomposes an image by simply applying the one-dimensional decorrelation procedure in a separable manner to the image. In other words, each column of the image is decorrelated then the image is decorrelated row-wise and/or diagonally. Such technique can be referred to as a separable two-dimensional decorrelation. However, by using this scheme, correlation in all directions of the image has not been simultaneously taken advantage of. Consequently, the one-dimension decorrelation technique may not be an efficient method to decorrelate two-dimensional data. Alternatively, a better decorrelation performance can be achieved by using a non-separable two-dimensional decorrelation rather than consecutive one-dimensional column-wise, row-wise and/or diagonally processing. In the literature, many techniques based on two-dimensional decorrelation have been proposed which can be categorized either as separable/non-separable, linear/non-linear, adaptive/non-adaptive, or hybrid of these types [6][7][10][21][27][67]. However, in this dissertation, a linear, non-separable, two-dimensional decomposition technique is proposed in this chapter and the underlying concept is described as follows.

Two-Dimensional-Horizontal Vertical Diagonal Decomposition (2D-HVDD) Method

Let a $2^M \times 2^N$ image X, where M, N are integers, be decomposed into four poly-phases, odd-odd, odd-even, even-odd and odd-odd components, as respectively specified by

$$X^{OO}(m,n) = X(2m-1,2n-1) \quad for \quad 1 \le m \le 2^{M-1}, 1 \le n \le 2^{N-1}. \tag{3.1}$$

$$X^{OE}(m,n) = X(2m-1,2n) \quad for \quad 1 \le m \le 2^{M-1}, 1 \le n \le 2^{N-1}. \tag{3.2}$$

$$X^{EO}(m,n) = X(2m,2n-1) \quad for \quad 1 \le m \le 2^{M-1}, 1 \le n \le 2^{N-1}. \tag{3.3}$$

$$X^{EE}(m,n) = X(2m,2n) \quad for \quad 1 \le m \le 2^{M-1}, 1 \le n \le 2^{N-1}. \tag{3.4}$$

These sub-images contain the essential features of the original image at a coarser scale with different orientations (horizontal, vertical, and diagonal directions). The example of poly-phase components of the image is given in Figure 3.1. Based on the concept described in the previous chapter that the adjacent pixels tend to be highly correlated, the pixels in horizontal, vertical and diagonal directions can be formulated their correlation in a two-dimensional structure.

Given the poly-phase image configuration shown in Figure 3.1 and neighborhood pixels correlation, an odd-even pixel component $X^{OE}(m,n)$ can be approximated by using four pixels located above, below, left and right of the current pixel $X^{OE}(m,n)$. In other words, those pixel elements are $X^{OO}(m,n)$, $X^{OO}(m,n+1)$, $X^{EE}(m-1,n)$, and $X^{EE}(m,n)$, respectively, and the associated relationship is given by

$$\hat{X}^{OE}(m,n) = \alpha^{OE} X^{OO}(m,n) + \beta^{OE} X^{OO}(m,n+1) + \lambda^{OE} X^{EE}(m-1,n) + \gamma^{OE} X^{EE}(m,n) \tag{3.5}$$

for $1 \le m \le 2^{M-1}, 1 \le n \le 2^{N-1}$. Additionally, the associated detail elements are designated by

$$D^{OE}(m,n) = X^{OE}(m,n) - \hat{X}^{OE}(m,n) \quad for \ 1 \le m \le 2^{M-1}, 1 \le n \le 2^{N-1}. \tag{3.6}$$

The real valued scalars $\alpha^{OE}, \beta^{OE}, \lambda^{OE}$ and γ^{OE} are selected so that the error of approximation is in some sense minimized. The two-dimensional decorrelation structure for the sub-image X^{OE} is depicted as a diamond shape in Figure 3.2.

In an analogous manner, an even-odd pixel $X^{EO}(m,n)$ can be approximated by employing a similar structure where the four neighborhood pixels used in the approximation are $X^{EE}(m,n-1)$, $X^{EE}(m,n)$, $X^{OO}(m,n)$, and $X^{OO}(m+1,n)$. According to the following relationship

$$\hat{X}^{EO}(m,n) = \alpha^{EO} X^{EE}(m,n-1) + \beta^{EO} X^{EE}(m,n) + \lambda^{EO} X^{OO}(m,n) + \gamma^{EO} X^{OO}(m+1,n) \tag{3.7}$$

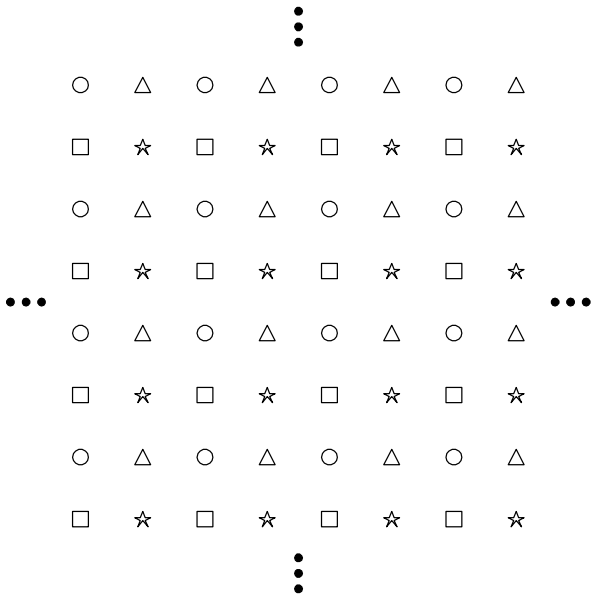

Figure 3.1 The example of poly-phase components of an image where ○, △, □ and ☆ respectively represent $X^{OO}(m,n)$, $X^{OE}(m,n)$, $X^{EO}(m,n)$ and $X^{EE}(m,n)$ for $1 \le m \le 2^{M-1}, 1 \le n \le 2^{N-1}$.

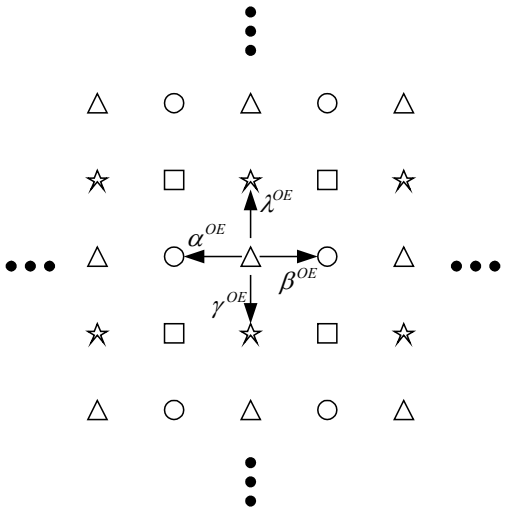

Figure 3.2 Two-dimensional decorrelation structure for the sub-image X^{OE}.

for $1 \le m \le 2^{M-1}, 1 \le n \le 2^{N-1}$ and all real valued scalars $\alpha^{EO}, \beta^{EO}, \lambda^{EO}$ and γ^{EO} are selected so as to in some sense minimize the error of approximation defined by

$$D^{EO}(m,n) = X^{EO}(m,n) - \hat{X}^{EO}(m,n) \quad for \ 1 \le m \le 2^{M-1}, 1 \le n \le 2^{N-1}. \tag{3.8}$$

The two-dimensional decorrelation structure for the sub-image X^{EO} is shown in Figure 3.3. Finally, an even-even pixel component $X^{EE}(m,n)$ can be approximated by using the two-dimensional decorrelation structure as a square shape illustrated in Figure 3.4.

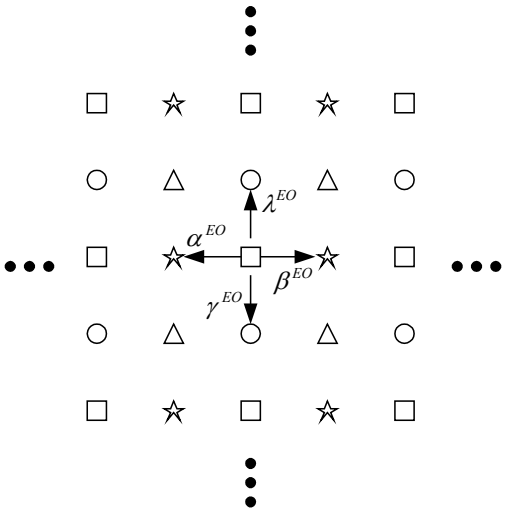

Figure 3.3 Two-dimensional decorrelation structure for the sub-image X^{EO}.

As shown in Figure 3.4, an even-even pixel $X^{EE}(m,n)$ is approximated by the four pixels $X^{OO}(m,n), X^{OO}(m+1,n), X^{OO}(m,n+1),$ and $X^{OO}(m+1,n+1)$ in which their relationship is given by

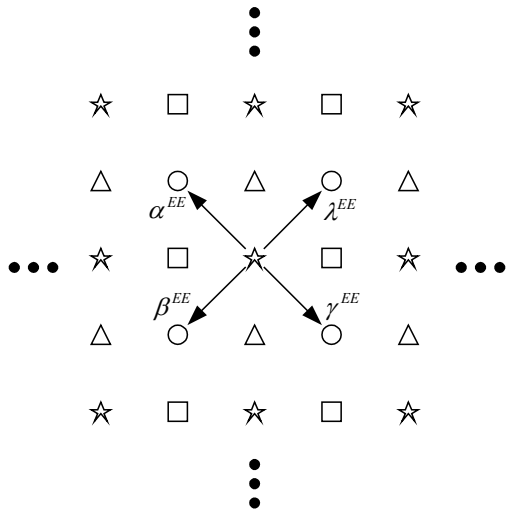

Figure 3.4 Two-dimensional decorrelation structure for the sub-image X^{EE}.

$$\hat{X}^{EE}(m,n) = \alpha^{EE} X^{OO}(m,n) + \beta^{EE} X^{OO}(m+1,n) + \lambda^{EE} X^{OO}(m,n+1) + \gamma^{EE} X^{OO}(m+1,n+1) \quad (3.9)^*$$

for $1 \le m \le 2^{M-1}, 1 \le n \le 2^{N-1}$. Furthermore, the associated error (detail) elements are designated by

$$D^{EE}(m,n) = X^{EE}(m,n) - \hat{X}^{EE}(m,n) \quad for \ \ 1 \le m \le 2^{M-1}, 1 \le n \le 2^{N-1}. \tag{3.10}$$

Based on the two-dimensional decorrelation procedure previously described, the boundary problem is inevitable for this case. On one hand, it can be assumed that the elements that lie outside the interval $1 \le m \le 2^{M-1}, 1 \le n \le 2^{N-1}$ are taken to be zero. On the other hand, the symmetric extension is an alternative method to alleviate the boundary problem. In many applications, the scalars shown in expressions (3.5), (3.7), and (3.9) are selected so as to minimize the magnitude of their associated error matrices. The details of searching for such

---

* The real scalars $\alpha^{EE}, \beta^{EE}, \lambda^{EE}$ and γ^{EE} are unrestricted but selected so that the detail matrices are as small as possible in magnitude in l_p norm sense.

scalars will be addressed in the next chapter. The procedure described at this stage is a one-level decomposition, which is illustrated in Figure 3.5.

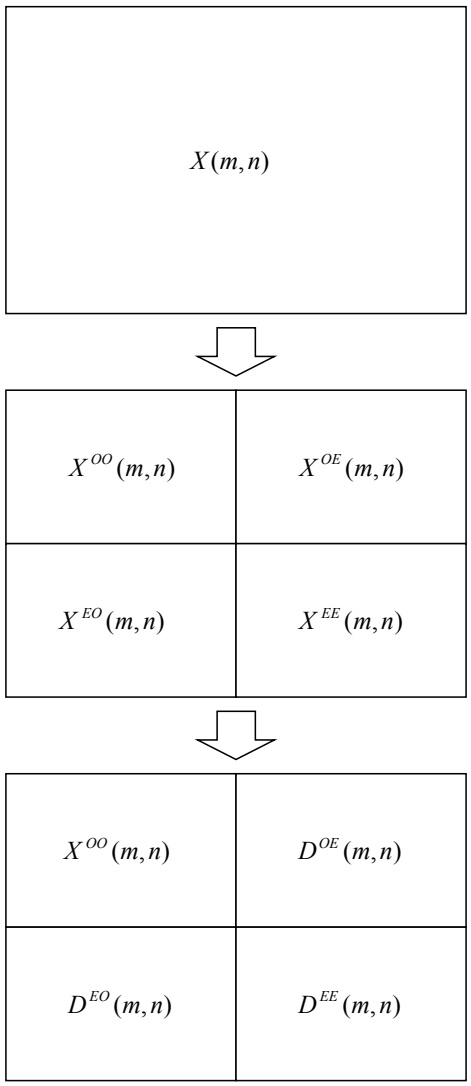

Figure 3.5 One-level two-dimensional HVD method.

Based on the described one-level decomposition, the original image can be achieved by arranging expressions (3.6), (3.8), and (3.10) to generate the original sub-images X^{OO}, X^{OE}, X^{EO} and X^{EE}. In fact, the original odd-odd indexed pixels are directly obtained by equation (3.1), that is

$$X(2m-1,2n-1)=X^{OO}(m,n) \quad for \quad 1\le m\le 2^{M-1},1\le n\le 2^{N-1}. \tag{3.11}$$

By rearranging expression (3.6) and employing expression (3.5), the odd-even indexed elements are specified as

$$X(2m-1,2n)=\hat{X}^{OE}(m,n)+D^{OE}(m,n) \quad for \quad 1\le m\le 2^{M-1},1\le n\le 2^{N-1}. \tag{3.12}$$

In a similar manner, the even-odd indexed elements are obtained by using equation (3.7) and rearranging equation (3.8), which yields

$$X(2m,2n-1)=\hat{X}^{EO}(m,n)+D^{EO}(m,n) \quad for \quad 1\le m\le 2^{M-1},1\le n\le 2^{N-1}. \tag{3.13}$$

Additionally, the even-even indexed pixels are regenerated from the rearrangement of expression (3.10), in which expression (3.9) is also used, that is

$$X(2m,2n)=\hat{X}^{EE}(m,n)+D^{EE}(m,n) \quad for \quad 1\le m\le 2^{M-1},1\le n\le 2^{N-1}. \tag{3.14}$$

It is possible to generalize of the one-level decomposition procedure to a multilevel case. At end stage of the one-level decomposition, the $2^M\times 2^N$ image X is decomposed into four $2^{M-1}\times 2^{N-1}$ sub-images, $X_1^{OO}, D_1^{OE}, D_1^{EO}$ and D_1^{EE}.* The $2^{M-1}\times 2^{N-1}$ coarser sub-image X_1^{OO} is decomposed into four $2^{M-2}\times 2^{N-2}$ sub-images, $X_2^{OO}, D_2^{OE}, D_2^{EO}$ and D_2^{EE}, at the second level. At the third level, the $2^{M-2}\times 2^{N-2}$ even-even sub-image X_2^{OO} is further decomposed into four $2^{M-3}\times 2^{N-3}$ sub-images, $X_3^{OO}, D_3^{OE}, D_3^{EO}$ and D_3^{EE}. The three-level two-dimensional HVDD method is depicted in Figure 2.6. A similar decomposition procedure can be repeated with any L levels for which $2^L\le \max\{2^M,2^N\}$ and the following set of $3L+1$ sub-images are generated

* The subscript is used to explicitly indicate the level of decomposition.

$$\left\{X_L^{OO}, D_L^{OE}, D_L^{EO}, D_L^{EE}, D_{L-1}^{OE}, D_{L-1}^{EO}, D_{L-1}^{EE}, ..., D_1^{OE}, D_1^{EO}, D_1^{EE}\right\}. \tag{3.15}$$

At each level j, the decomposition process is obtained by the following relationships,

$$X_j^{OO}(m,n) = X_{j-1}^{OO}(2m-1, 2n-1) \tag{3.16}$$

$$X_j^{OE}(m,n) = X_{j-1}^{OO}(2m-1, 2n) \tag{3.17}$$

$$X_j^{EO}(m,n) = X_{j-1}^{OO}(2m, 2n-1) \tag{3.18}$$

$$X_j^{EE}(m,n) = X_{j-1}^{OO}(2m, 2n) \tag{3. 19}$$

and the poly-phase detail elements are therefore given by

$$D_j^{OE}(m,n) = X_j^{OE}(m,n) - \overbrace{\left(\alpha_j^{OE} X_j^{OO}(m,n) + \beta_j^{OE} X_j^{OO}(m,n+1) + \lambda_j^{OE} X_j^{EE}(m-1,n) + \gamma_j^{OE} X_j^{EE}(m,n)\right)}^{\hat{X}_j^{OE}(m,n)} \tag{3.20}$$

$$D_j^{EO}(m,n) = X_j^{EO}(m,n) - \overbrace{\left(\alpha_j^{EO} X_j^{EE}(m,n-1) + \beta_j^{EO} X_j^{EE}(m,n) + \lambda_j^{EO} X_j^{OO}(m,n) + \gamma_j^{EO} X_j^{OO}(m+1,n)\right)}^{\hat{X}_j^{EO}(m,n)} \tag{3.21}$$

$$D_j^{EE}(m,n) = X_j^{EE}(m,n) - \overbrace{\left(\alpha_j^{EE} X_j^{OO}(m,n) + \beta_j^{EE} X_j^{OO}(m+1,n) + \lambda_j^{EE} X_j^{OO}(m,n+1) + \gamma_j^{EE} X_j^{OO}(m+1,n+1)\right)}^{\hat{X}_j^{EE}(m,n)} \tag{3.22}$$

where the index intervals for expressions (3.16)-(3.22) are $1 \le m \le 2^{M-j}$ and $1 \le n \le 2^{N-j}$, $1 \le j \le L$.

In an obvious manner, the inverse procedures for each level j of the multilevel reconstruction are readily achieved by rearranging equations (3.16)-(3.22), which respectively yields

$$X_{j-1}^{OO}(2m-1,2n-1) = X_j^{OO}(m,n) \tag{3.23}$$

$$X_{j-1}^{OO}(2m,2n) = D_j^{EE}(m,n) + \overbrace{\left(\alpha_j^{EE} X_j^{OO}(m,n) + \beta_j^{EE} X_j^{OO}(m+1,n) + \lambda_j^{EE} X_j^{OO}(m,n+1) + \gamma_j^{EE} X_j^{OO}(m+1,n+1)\right)}^{\hat{X}_j^{EE}(m,n)} \tag{3.24}$$

$$X_{j-1}^{OO}(2m-1,2n) = D_j^{OE}(m,n) + \overbrace{\left(\alpha_j^{OE} X_j^{OO}(m,n) + \beta_j^{OE} X_j^{OO}(m,n+1) + \lambda_j^{OE} X_j^{EE}(m-1,n) + \gamma_j^{OE} X_j^{EE}(m,n)\right)}^{\hat{X}_j^{OE}(m,n)} \tag{3.25}$$

$$X_{j-1}^{OO}(2m,2n-1) = D_j^{EO}(m,n) + \overbrace{\left(\alpha_j^{EO} X_j^{EE}(m,n-1) + \beta_j^{EO} X_j^{EE}(m,n) + \lambda_j^{EO} X_j^{OO}(m,n) + \gamma_j^{EO} X_j^{OO}(m+1,n)\right)}^{\hat{X}_j^{EO}(m,n)} \tag{3.26}$$

for $j = L, L-1,...,3,2,1$ and $1 \le m \le 2^{M-j}, 1 \le n \le 2^{N-j}$ *.

The decomposition procedure described in this chapter employs the two-dimensional structure. At each level of decomposition, once the optimum weighting coefficients associated with three sub-images are established, the related two-dimensional decorrelation structures are constructed, and the two-dimension convolution is therefore performed to generate the associated error matrices. This approach not only provides an efficient alternative method for image decomposition but also gives an adaptive performance in which the optimum scalars used in

* It is noted that the even-even subband at level j must be first reconstructed, since the pixels of this subband are employed to reconstruct the odd-even and even-odd subbands.

approximation can be updated at each level of the decomposition process. To achieve a good performance of compression, many proposed methods have endeavored to give the detailed matrices as small in magnitude as possible. In addition to the proposed method in this chapter, all optimum scalars used in approximation are selected to achieve the same purpose. The algorithms of finding such scalars are described in the next chapter.

CHAPTER IV

OPTIMUM PARAMETERS SELECTION

According to the one-dimensional and two-dimensional decorrelational techniques discussed in Chapter II and Chapter III, respectively, it is seen that the optimum parameters play an important role in the decomposition process. To achieve a good compression performance, these real valued scalars must be selected so that the error vector or matrix of approximation has as many of its components small in magnitude as possible. In other words, these scalars must be chosen to minimize error of approximation in some sense. Generally, the topic in this chapter is related to a linear fit modeling problem, which has long a history of its development [68][69]. This chapter mainly focuses developing an algorithm for determining the minimum l_p norm approximate solution to a linear system of *M* equations in one unknown and *N* unknowns. Cases in which $p = 1$ and $p = 2$ are particular interest in this dissertation.

Overview of Modeling Problem

Let there be given an empirical data set $\{y(1), y(2), y(3),..., y(M)\}$ such that $\underline{y} \in R^{M\times 1}$ and a parameter vector $\underline{x} \in R^{N\times 1}$ whose components are unknown real value as given by $\{x(1), x(2), x(3),..., x(N)\}$. The modeling relationship of a system of *M* equations in *N* unknowns is governed by

$$\underline{y} = A\underline{x} + \underline{e} \tag{4.1}$$

where $\underline{e} \in R^{M\times 1}$ is the error vector and A is the $M\times N$ system matrix whose components $A(m,n) \in R$ for $1 \le m \le M$ and $1 \le n \le N$. The error vector appearing in equation (4.1) can be represented either as additive measurement noise or as an error incurred when this linear model is invoked and the process is nonlinear. Upon rearranging expression (4.1), we have

$$\underline{e} = \underline{y} - A\underline{x}. \tag{4.2}$$

The goodness of approximation is measured by the l_p norm sense as specified by

$$\left\|\underline{e}\right\|_p = \left[\sum_{m=1}^{M} |e(m)|^p\right]^{1/p} \tag{4.3}$$

where the parameter p takes on any value greater than or equal to one. For $p = 1$ and $p = 2$, it corresponds to sum of error magnitudes and sum of squared error criterion, respectively. For general $1 \le p < \infty$, this l_p norm of the error vector can also be expressed as

$$f_p(\underline{x}) = \left\|\underline{y} - A\underline{x}\right\|_p . \tag{4.4}$$

This norm-induced function possesses some very important and well-known properties useful for characterizing its minimum value. The key properties are therefore now briefly summarized.

Definition (4.1) : *Norm* [70] *A norm defined on a vector space X over the field F, which maps every vector* $\underline{x}$ *in X into a non-negative real number in R, is denoted by* $\|\underline{x}\|$ *(which is called the norm of* $\underline{x}$*). The norm functional must satisfy the following axioms.*

(*i*) $\|\underline{x}\| \ge 0$ *for all* $\underline{x} \in X$ *and* $\|\underline{x}\| = 0$ *if and only if* $\underline{x} = \underline{0}$

(*ii*) $\|\alpha\underline{x}\| = |\alpha|\|\underline{x}\|$ *for all* $\underline{x} \in X$ *and* $\alpha \in F$

(*iii*) $\|\underline{x}_1 + \underline{x}_2\| \le \|\underline{x}_1\| + \|\underline{x}_2\|$ *for all* $\underline{x}_1, \underline{x}_2 \in X$ (*triangular inequality*)

A norm is actually a special case of a more general function called a metric, which is a function that measures the distance between two points in a set. The definitions of a norm and a metric are quite similar and closely related. Additionally, a vector space X taken together with an appropriately defined norm is called a *normed vector space*. For further detail of this topic, the content can be found in [70] or other textbooks.

Definition (4.2) : *Convex Functional* [71] *A real-valued functional f defined on a convex subset C of a linear vector space is said to be convex if*

$$f\left(\alpha \underline{x}_1 + (1-\alpha)\underline{x}_2\right) \le \alpha f\left(\underline{x}_1\right) + (1-\alpha) f\left(\underline{x}_2\right) \text{ for all } \underline{x}_1, \underline{x}_2 \in C \text{ and } 0 \le \alpha \le 1. \tag{4.5}$$

In the theory of optimization, convex functional plays a very special role due to the fact that a local minimum of *f* is also a global minimum of *f*, when the convex functional is applied.

Proposition(4.1) [71] *Let* $f\left(\underline{x}\right)$ *be a convex functional defined on a convex subset C of a normed space X. Let* $\mu = \inf_{\underline{x} \in C} f\left(\underline{x}\right)$, *then*

1. *The subset* Ω *of C where* $f\left(\underline{x}\right) = \mu$ *is convex.*
2. *If* $\underline{x}_o$ *is a local minimum of* $f\left(\underline{x}\right)$ *for* $\underline{x} \in C$, *hence* $\underline{x}_o$ *is also a global minimum.*
3. *Let* Ψ *be a neighborhood about* $\underline{x}_o$ *in which* $\underline{x}_o$ *minimizes* $f\left(\underline{x}\right)$. *For any* $\underline{x}_1 \in C$, *there is* $\underline{x} \in \Psi$ *such that* $\underline{x} = \alpha \underline{x}_o + (1-\alpha)\underline{x}_1$ *for some* α, $0 < \alpha < 1$.
   Therefore $f\left(\underline{x}_o\right) \le f\left(\underline{x}\right) \le \alpha f\left(\underline{x}_o\right) + (1-\alpha) f\left(\underline{x}_1\right)$ *or* $f\left(\underline{x}_o\right) \le f\left(\underline{x}_1\right)$.

Proof : 1. Let $\underline{x}_1, \underline{x}_2 \in \Omega$ and for $\underline{x} = \alpha \underline{x}_1 + (1-\alpha)\underline{x}_2$, $0 < \alpha < 1$, it then follows that

$$f\left(\underline{x}\right) = f\left(\alpha \underline{x}_1 + (1-\alpha)\underline{x}_2\right) \le \alpha f\left(\underline{x}_1\right) + (1-\alpha) f\left(\underline{x}_2\right) = \mu. \tag{4.6}$$

But it is necessary for any $\underline{x} \in C$ that $f\left(\underline{x}\right) \ge \mu$, therefore $f\left(\underline{x}\right) = \mu$.

Definition (4.3) : *Continuity* [70] *Let X and Y be normed vector spaces and let* $f : X \to Y$. *Then for each* $\underline{x} \in X$, *the functional f is continuous at x* $\Leftrightarrow$ *for each* $\varepsilon > 0$ *there exists a* $\delta > 0$ *such that*

$$\left\|\underline{x}_1 - \underline{x}_2\right\|_X < \delta \Rightarrow \left\|f\left(\underline{x}_1\right) - f\left(\underline{x}_2\right)\right\|_Y < \varepsilon. \tag{4.7}$$

Theorem (4.1) [72] The l_p norm-induced function $f_p(\underline{x}) = \left\| \underline{y} - A\underline{x} \right\|_p$, for which $\underline{y} \in R^{M\times 1}$ and $A \in R^{M\times N}$ are given, is a convex and continuous function of vector $\underline{x}$.

Proof : The functional $f_p(\underline{x})$ is a norm-induced function which satisfies the properties of a normed vector space. For any arbitrary vectors $\underline{x}_1, \underline{x}_2 \in R^N$ and a scalar $\alpha \in R$, $0 \le \alpha \le 1$, follows from the norm triangle inequality that

$$\begin{aligned} f_p(\alpha\underline{x}_1 + (1-\alpha)\underline{x}_2) &= \left\| \underline{y} - A(\alpha\underline{x}_1 + (1-\alpha)\underline{x}_2) \right\|_p \\ &= \left\| \alpha(\underline{y} - A\underline{x}_1) + (1-\alpha)(\underline{y} - A\underline{x}_2) \right\|_p \\ &\le \left\| \alpha(\underline{y} - A\underline{x}_1) \right\|_p + \left\| (1-\alpha)(\underline{y} - A\underline{x}_2) \right\|_p \\ &= |\alpha| \left\| (\underline{y} - A\underline{x}_1) \right\|_p + |1-\alpha| \left\| (\underline{y} - A\underline{x}_2) \right\|_p \\ &= \alpha f_p(\underline{x}_1) + (1-\alpha) f_p(\underline{x}_2). \end{aligned} \tag{4.8}$$

The convexity of a functional $f_p(\underline{x})$ is then established. For a proof of the continuity, use of the triangular inequality yields

$$\begin{aligned} f_p(\underline{x} + \underline{\Delta}_x) &= \left\| \underline{y} - A(\underline{x} + \underline{\Delta}_x) \right\|_p \\ &\le \left\| \underline{y} - A\underline{x} \right\|_p + \left\| A\underline{\Delta}_x \right\|_p \\ &\le \left\| \underline{y} - A\underline{x} \right\|_p + \left\| A \right\|_p \cdot \left\| \underline{\Delta}_x \right\|_p = f_p(\underline{x}) + \left\| A \right\|_p \cdot \left\| \underline{\Delta}_x \right\|_p. \end{aligned} \tag{4.9}$$

It can be shown in a similar fashion that $f_p(\underline{x}) \le f_p(\underline{x} + \underline{\Delta}_x) + \left\| A \right\|_p \cdot \left\| \underline{\Delta}_x \right\|_p$. It then follows that for any positive scalar ε, a positive scalar δ can be selected such that

$$\left\| \underline{\Delta}_x \right\|_p \le \delta = \frac{\varepsilon}{\left\| A \right\|_p} \;\Rightarrow\; \left| f_p(\underline{x} + \underline{\Delta}_x) - f_p(\underline{x}) \right| \le \varepsilon. \tag{4.10}$$

Thus, the requirement for continuity of a functional $f_p(\underline{x})$ is satisfied.

Theorem (4.2) [72] *Let the M×N real matrix A characterizing the convex functional* $f_p(\underline{x}) = \left\| \underline{y} - A\underline{x} \right\|_p$ *have full rank (linearly independent columns) N. Then there exists a unique optimum model parameter vector* $\underline{x}$ *for the least squares error solution corresponding to p = 2 as given by*

$$\underline{x}_2^o = \left[A^T A \right]^{-1} A^T \underline{y} \,. \tag{4.11}$$

Furthermore, the associated least squares error vector is orthogonal to the column vector of matrix A so that $A^T \underline{e}_2^o = A^T [\underline{y} - A\underline{x}_2^o] = \underline{0}$.

If the M×N matrix A satisfies the additional Haar property which requires that any selection of N rows of A are linearly independent, it then follows that

(i) For p = 1, a minimum l_1 *norm (least absolute deviation (LAD) solution) exists for which at least N components of its associated error vector* $\underline{e}_1^o = \underline{y} - A\underline{x}_1^o$ *are equal to zero.*

(ii) For p = ∞*, a Chebyshev solution exists for which at least N+1 components of an error vector* $\underline{e}_\infty^o = \underline{y} - A\underline{x}_\infty^o$ *have equal maximum magnitudes.*

The proof of this theorem for $p = 1$ and $p = 2$ will be given in a subsequent section. For $p = \infty$ case, its proof can be found in [72]. It was established in Proposition (4.1) that for a convex functional, any local minimum point is also global minimum point. This implies that the minimum l_p norm method certainly provides a minimum solution to a functional $f_p(\underline{x})$. By the convexity property of a functional $f_p(\underline{x})$ for $1 < p < \infty$, there is guarantee for a unique solution. Unfortunately, this is not always be the case for $p = 1$, which a minimum l_1 norm problem may have many solutions. The sum of error magnitude function $f_1(\underline{x})$ is of use in many applications when the data vector $\underline{y}$ contains a small number of data outliers (i.e. unrepresentative or bad

data points). On the other hand, the sum of squared error criterion $f_2(\underline{x})$ is unduly influenced by the data outliers often leading to a poor selection of the coefficient vector [73]. Algorithms for obtaining minimum l_1 norm and l_2 norm solutions are discussed in the following sections.

A Linear System of *M* Equations in One Unknown

Minimum l_1 Norm Solution to a Linear System of *M* Equations in One Unknown

To determine a minimum l_1 norm solution for a linear system of *M* equations in one unknown, the special case *N* = 1 shall be employed in finding a solution to a general linear system of *M* equations in *N* unknowns case. The error vector can be expressed as

$$\underline{e} = \underline{y} - x\underline{a} \tag{4.12}$$

where $\underline{y}, \underline{a}$ and $\underline{e} \in R^{M\times 1}$ and $\underline{x} \in R$. It is now desired to select a real scalar *x* so as to minimize the error of approximation in the l_1 norm sense, as given by

$$f_1(x) = \left\| \underline{y} - x\underline{a} \right\|_1 = \sum_{m=1}^{M} \left| y(m) - xa(m) \right|. \tag{4.13}$$

An efficient algorithm to solve the least absolute deviation problem was introduced by Bloomfield and Steiger [69]. Their solution procedure is used extensively for solving the minimum l_1 approximate solution of the general linear system of *M* equations in *N* unknowns [72]. To obtain some insight into an interpretation of $f_1(x)$, the summation of expression (4.13) is separated into a summand term for which $a(m) = 0$ and another term for which $a(m) \neq 0$, which yields

$$\begin{aligned} f_1(x) &= \sum_{a(m)=0} \left| y(m) \right| + \sum_{a(m)\neq 0} \left| y(m) - xa(m) \right| \\ &= \sum_{a(m)=0} \left| y(m) \right| + \sum_{\substack{a(m)\neq 0 \\ x(m)\geq x}} \left| a(m) \right| \left[x(m) - x \right] + \sum_{\substack{a(m)\neq 0 \\ x(m)< x}} \left| a(m) \right| \left[x - x(m) \right]. \end{aligned} \tag{4.14}$$

Examination of expression (4.14) indicates that $f_1(x)$ is a piecewise linear function of a variable x whose slope changes at the points where $x(m) - x = 0$. When deriving of expression (4.14) from the first line to the second line, the value of $y(m)$ is substituted by $x(m)a(m)$ where the entities $x(m)$ identify the value of x for which the m^{th} component of error vector is equal to zero, that is

$$x(m) = \frac{y(m)}{a(m)} \quad for \;\; a(m) \neq 0 . \tag{4.15}$$

Let us rearrange the scalars $x(m)$ as described in expression (4.15) in the monotonically non-decreasing order

$$x(m_1) \;\leq\; x(m_2) \;\leq\; x(m_3) \;\leq \ldots \leq\; x(m_{Q-1}) \;\leq\; x(m_Q)$$

where $Q \leq M$ represents the number of nonzero component of $\underline{a}$. With this in mind, the derivative of $f_1(x)$ with respective to x exists for all $x \neq x(m_q), 1 \leq q \leq Q$. This function can then be expressed as

$$\begin{aligned} \frac{\partial f_1(x)}{\partial x} &= -\sum_{x(m_q)>x} \left|a(m_q)\right| + \sum_{x(m_q)<x} \left|a(m_q)\right| \\ &= -\sum_{m=1}^{M} \left|a(m)\right| + 2 \sum_{x(m_q)<x} \left|a(m_q)\right| \quad for \quad x \neq x(m_q) . \end{aligned} \tag{4.16}$$

This derivative result is a piecewise constant function of x that changes by the amount $\left|a(m_q)\right|$ to the immediate right of the $x(m_q)$ zero points. Moreover, this derivative monotonically increases from the value $-\sum_{m=1}^{M} \left|a(m)\right|$ assumed at $x = -\infty$ to the value $\sum_{m=1}^{M} \left|a(m)\right|$ assumed at $x = \infty$. A minimum solution for a functional $f_1(x)$ is therefore indicated whenever the derivative has been changed from a negative value to either a zero or positive value to the immediate right of a

$x(m_q)$ zero point. If the derivative sign is changed from negative to positive then the zero point $x^o(m_q)$ is the unique optimum choice for the scalar x. On the other hand, the non-unique minimum solution exists whenever the derivative sign is changed from negative to zero at the transition point $x^o(m_q)$. In this case, any selection of the scalar x in the closed interval $x^o(m_q) \le x \le x^o(m_{q+1})$ corresponds to the optimum l_1 selection. This analysis indicates that an efficient algorithm entails sequentially evaluating the right hand derivatives at the $x(m_q)$ zero points as specified by

$$\frac{\partial f_1\left(x(m_q)^+\right)}{\partial x} = -\sum_{m=1}^{M}\left|a(m)\right| + 2\sum_{x(m_q)<x}\left|a(m_q)\right| \quad for \quad 1 \le q \le Q. \tag{4.17}$$

The smallest value of q for which the derivative first transforms from a negative to non-negative value indicates that $x^o(m_q)$ point is a minimum l_1 selection. If $\partial f_1\left(x^o(m_q)^+\right)/\partial x$ is greater than zero then $x^o(m_q)$ is the unique choice for a minimum l_1 selection while a zero value of $\partial f_1\left(x^o(m_q)^+\right)/\partial x$ indicates that any selection for a scalar x in the interval $x^o(m_q) \le x \le x^o(m_{q+1})$ is an optimum l_1 selection.

From this algorithm, the computer requirement is seen to entail at most M division operations using for determination of the zero points $x(m) = y(m)/a(m)$, a reordering of these zero points into a monotonically non-decreasing set and at most $2M$ summation operations giving a total computational complexity on the order of M.

Minimum l_2 Norm Solution to a Linear System of *M* Equations in One Unknown

The minimum sum of squared error is the most commonly used measurement in finding an approximate solution to a linear system of equations. In an analogous manner, the special case for $N = 1$ is useful in solving the general linear system of M equations in N unknowns. As previously shown in expression (4.11), the problem of finding a scalar x, which minimizes the l_2 norm of the error vector, has a closed form solution. With this in mind, let the analysis begin with the squared l_2 norm of the error vector as specified by

$$f_2(x) = \left\| \underline{y} - x\underline{a} \right\|_2$$
$$= \sqrt{\left[\underline{y} - x\underline{a}\right]^T \left[\underline{y} - x\underline{a}\right]}$$
$$= \sqrt{\underline{y}^T \underline{y} - 2x\underline{a}^T \underline{y} + x^2 \underline{a}^T \underline{a}} \,. \tag{4.18}$$

A necessary condition for an optimum selection of real variable x is obtained by setting the derivative of this function with respect to x as given by

$$\frac{df_2(x)}{dx} = \frac{-\underline{a}^T \underline{y} + x\underline{a}^T \underline{a}}{\left\| \underline{y} - x\underline{a} \right\|_2} \tag{4.19}$$

equal to zero. The unique minimum sum of squared error solution is therefore given by

$$x^o = \frac{\underline{a}^T \underline{y}}{\underline{a}^T \underline{a}} \,. \tag{4.20}$$

A Linear System of *M* Equations in *N* Unknowns

According to the decorrelation approach described in Chapter III, the property of a two-dimensional decorrelation is completely governed by the weighting coefficients (see expressions (3.5), (3.7) and (3.9)). It is obvious that without regard to how the scalars are selected, this decomposition transform is always invertible. However, these coefficients are normally chosen so that the detail matrices have as many small magnitude elements as possible. Consequently, only the steady-state components are employed in the procedure of finding the optimum weighting coefficients to achieve this objective. Upon observation the poly-phase components of the image shown in Figure 3.1, it is found that the steady-state intervals at level j for both the odd-even indexed elements are $2 \le m \le 2^{M-j}$ and $1 \le n \le 2^{N-j} - 1$. For the even-odd indexed elements, the steady-state intervals at level j are $1 \le m \le 2^{M-j} - 1$ and $2 \le n \le 2^{N-j}$. Finally, the steady-state interval at level j for the even-even indexed elements are $1 \le m \le 2^{M-j} - 1$ and $1 \le n \le 2^{N-j} - 1$. With this in mind, let start with the set of steady-state odd-even detail elements which are governed by

$$D_j^{OE}(m,n) = X_j^{OE}(m,n)$$
$$-\left(\alpha_j^{OE} X_j^{EE}(m-1,n) + \beta_j^{OE} X_j^{EE}(m,n) + \lambda_j^{OE} X_j^{OO}(m,n) + \gamma_j^{OE} X_j^{OO}(m,n+1)\right) \quad (4.21)$$

for $2 \le m \le 2^{M-j}, 1 \le n \le 2^{N-j} - 1$. This set of steady-state odd-even detail elements can be column concatenated and expressed in a convenient vector form as

$$\underline{d}_j^{OE} = \underline{x}_j^{OE} - A_j^{OE} \underline{h}_j^{OE} \quad (4.22)$$

where

$$\underline{d}_j^{OE} = \begin{bmatrix} D_j^{OE}(2,1) \\ D_j^{OE}(3,1) \\ \vdots \\ D_j^{OE}(2^{M-j}, 2^{N-j} - 1) \end{bmatrix} \quad (4.23)$$

$$\underline{x}_j^{OE} = \begin{bmatrix} X_j^{OE}(2,1) \\ X_j^{OE}(3,1) \\ \vdots \\ X_j^{OE}(2^{M-j}, 2^{N-j} - 1) \end{bmatrix} \quad (4.24)$$

$$A_j^{OE} = \begin{bmatrix} X_j^{EE}(1,1) & X_j^{EE}(2,1) & X_j^{OO}(2,1) & X_j^{OO}(2,2) \\ X_j^{EE}(2,1) & X_j^{EE}(3,1) & X_j^{OO}(3,1) & X_j^{OO}(3,2) \\ \vdots & \vdots & \vdots & \vdots \\ X_j^{EE}(2^{M-j} - 1, 2^{N-j} - 1) & X_j^{EE}(2^{M-j}, 2^{N-j} - 1) & X_j^{OO}(2^{M-j}, 2^{N-j} - 1) & X_j^{OO}(2^{M-j}, 2^{N-j}) \end{bmatrix} \quad (4.25)$$

and

$$\underline{h}_j^{OE} = \begin{bmatrix} \alpha_j^{OE} \\ \beta_j^{OE} \\ \lambda_j^{OE} \\ \gamma_j^{OE} \end{bmatrix}. \quad (4.26)$$

As has been mentioned that the main contribution of this section is to select the weighting coefficient vector so as to cause the steady-state detail elements to have as many small magnitudes as possible. To accomplish this objective, the optimum selection of the weighting coefficients vector at level j, $\underline{\overset{o}{h}}_j^{OE}$, must be the one which minimizes the following norm-induced functional

$$f\left(\underline{h}_j^{OE}\right) = \left\| \underline{y}_j^{OE} - A_j^{OE}\,\underline{h}_j^{OE} \right\|_p \quad for \;\; 1 \le j \le L\,. \tag{4.27}$$

On the other hand, it is also possible to use the same weighting coefficients at each level of decomposition. In this case, one can seek for the optimum weighting coefficients vector $\underline{\overset{o}{h}}$ which satisfies the criterion

$$\min_{\underline{h}\in R^{4x1}} f(\underline{h}) = \min_{\underline{h}\in R^{4x1}} \sum_{j=1}^{L} \left\| \underline{y}_j^{OE} - A_j^{OE}\,\underline{h} \right\|_p = \min_{\underline{h}\in R^{4x1}} \left\| \underline{y}^{c} - A^{c}\,\underline{h} \right\|_p \tag{4.28}$$

in which $\underline{y}^c$ and A^c are column concatenations of $\underline{y}_j^{OE}$ and A_j^{OE} for $1 \le j \le L$ as respectively specified by $\underline{y}^c = \left[\left(\underline{y}_1^{OE}\right)^T \left(\underline{y}_2^{OE}\right)^T \cdots \left(\underline{y}_L^{OE}\right)^T\right]^T$, $A^c = \left[\left(A_1^{OE}\right)^T \left(A_2^{OE}\right)^T \cdots \left(A_L^{OE}\right)^T\right]^T$.

Similarly, it is obvious that one can construct such systems of linear equations shown in expressions (4.27)-(4.28) for the steady-state even-odd and even-even elements. The set of steady-state even-odd detail elements is given by

$$\begin{aligned} D_j^{EO}(m,n) &= X_j^{EO}(m,n) \\ &\quad - \left(\alpha_j^{EO} X_j^{OO}(m,n) + \beta_j^{EO} X_j^{OO}(m+1,n) + \lambda_j^{EO} X_j^{EE}(m,n-1) + \gamma_j^{EO} X_j^{EE}(m,n)\right) \end{aligned} \tag{4.29}$$

for $1 \le m \le 2^{M-j} - 1, 2 \le n \le 2^{N-j}$. Additionally, a compact vector form for the steady-state even-odd elements is expressed as

$$\underline{d}_j^{EO} = \underline{x}_j^{EO} - A_j^{EO}\,\underline{h}_j^{EO} \tag{4.30}$$

where

$$\underline{d}_j^{EO} = \begin{bmatrix} D_j^{EO}(1,2) \\ D_j^{EO}(2,2) \\ \vdots \\ D_j^{EO}(2^{M-j}-1,2^{N-j}) \end{bmatrix} \tag{4.31}$$

$$\underline{x}_j^{EO} = \begin{bmatrix} X_j^{EO}(1,2) \\ X_j^{EO}(2,2) \\ \vdots \\ X_j^{EO}(2^{M-j}-1,2^{N-j}) \end{bmatrix} \tag{4.32}$$

$$A_j^{EO} = \begin{bmatrix} X_j^{OO}(1,2) & X_j^{OO}(2,2) & X_j^{EE}(1,1) & X_j^{EE}(1,2) \\ X_j^{OO}(2,2) & X_j^{OO}(3,2) & X_j^{EE}(2,1) & X_j^{EE}(2,2) \\ \vdots & \vdots & \vdots & \vdots \\ X_j^{OO}(2^{M-j}-1,2^{N-j}) & X_j^{OO}(2^{M-j},2^{N-j}) & X_j^{EE}(2^{M-j}-1,2^{N-j}-1) & X_j^{EE}(2^{M-j}-1,2^{N-j}) \end{bmatrix} \tag{4.33}$$

and

$$\underline{h}_j^{EO} = \begin{bmatrix} \alpha_j^{EO} \\ \beta_j^{EO} \\ \lambda_j^{EO} \\ \gamma_j^{EO} \end{bmatrix}. \tag{4.34}$$

For the even-even detail elements, the steady-state set is given by

$$\begin{aligned} D_j^{EE}(m,n) &= X_j^{EE}(m,n) \\ &\quad - \left(\alpha_j^{EE} X_j^{OO}(m,n) + \beta_j^{EE} X_j^{OO}(m,n+1) + \lambda_j^{EE} X_j^{OO}(m+1,n) + \gamma_j^{EE} X_j^{OO}(m+1,n+1)\right) \end{aligned} \tag{4.35}$$

for $1 \le m \le 2^{M-j} - 1, 1 \le n \le 2^{N-j} - 1$ and a compact vector form for the steady-state even-odd elements can be written as

$$\underline{d}_j^{EE} = \underline{x}_j^{EE} - A_j^{EE} \underline{h}_j^{EE} \tag{4.36}$$

where

$$\underline{d}_j^{EE} = \begin{bmatrix} D_j^{EE}(1,1) \\ D_j^{EE}(2,1) \\ \vdots \\ D_j^{EE}(2^{M-j} - 1, 2^{N-j} - 1) \end{bmatrix} \tag{4.37}$$

$$\underline{x}_j^{EE} = \begin{bmatrix} X_j^{EE}(1,1) \\ X_j^{EE}(2,1) \\ \vdots \\ X_j^{EE}(2^{M-j} - 1, 2^{N-j} - 1) \end{bmatrix} \tag{4.38}$$

$$A_j^{EE} = \begin{bmatrix} X_j^{OO}(1,1) & X_j^{OO}(1,2) & X_j^{OO}(2,1) & X_j^{OO}(2,2) \\ X_j^{OO}(2,1) & X_j^{OO}(2,2) & X_j^{OO}(3,1) & X_j^{OO}(3,2) \\ \vdots & \vdots & \vdots & \vdots \\ X_j^{OO}(2^{M-j} - 1, 2^{N-j} - 1) & X_j^{OO}(2^{M-j} - 1, 2^{N-j}) & X_j^{OO}(2^{M-j}, 2^{N-j} - 1) & X_j^{EE}(2^{M-j}, 2^{N-j}) \end{bmatrix} \tag{4.39}$$

and

$$\underline{h}_j^{EE} = \begin{bmatrix} \alpha_j^{EE} \\ \beta_j^{EE} \\ \lambda_j^{EE} \\ \gamma_j^{EE} \end{bmatrix}. \tag{4.40}$$

It is noted that the primary interest of l_p norm is confined to the case for $p = 1$ and $p = 2$ in which the iterative algorithm for $p = 1$ case developed by Cadzow [72] will be reviewed in the following section.

Minimum l_1 Norm Solution to a Linear System of *M* Equations in *N* Unknowns

The general system of *M* equations in *N* unknowns can be expressed as

$$r(\underline{x}) = \underline{y} - A\underline{x}. \tag{4.41}$$

The vector $\underline{y} \in R^{M\times 1}$ and matrix $A \in R^{M\times N}$ are given whereas the parameter vector $\underline{x} \in R^{N\times 1}$ is selected so as to minimize the following norm-induced function

$$f_1(\underline{x}) = \left\|r(\underline{x})\right\|_1 = \left\|\underline{y} - A\underline{x}\right\|_1 = \sum_{m=1}^{M}\left|y(m) - \underline{e}_m^T A\underline{x}\right|. \tag{4.42}$$

In this expression, $\underline{e}_m$ is the $M \times 1$ standard basis vector. It then follows that the term $\underline{e}_m^T A\underline{x}$ represents the m^{th} component of the $M \times 1$ vector $A\underline{x}$. The residue error components for $1 \le m \le M$ can be decomposed as three disjoint sets in which the residue error components $r_m(\underline{x}) = y(m) - \underline{e}_m^T A\underline{x}$ are positive, negative and zero. Based on this decomposition, expression (4.42) can be rewritten as

$$\begin{aligned} f_1(\underline{x}) &= \sum_{r_m(\underline{x})>0}[y(m) - \underline{e}_m^T A\underline{x}] - \sum_{r_m(\underline{x})<0}[y(m) - \underline{e}_m^T A\underline{x}] + \sum_{r_m(\underline{x})=0}[r_m(\underline{x})] \\ &= \sum_{r_m(\underline{x})>0}[y(m) - \underline{e}_m^T A\underline{x}] - \sum_{r_m(\underline{x})<0}[y(m) - \underline{e}_m^T A\underline{x}]. \end{aligned} \tag{4.43}$$

It is now shown that there exists a real $N \times 1$ parameter vector $\underline{x}^o$ which minimizes the sum of the residue error magnitude function $f_1(\underline{x})$ such that the associated residual error vector

$$r(\underline{x}^o) = \underline{y} - A\underline{x}^o \tag{4.44}$$

has at least *N* zero components. Suppose that $\underline{x}^o$ is the optimum solution for which only $0 \le N_0 \le N-1$ components of the residue error vector $r(\underline{x}^o)$ are zero. Now let us perturb $\underline{x}^o$ to

$\underline{x}^o + \varepsilon\underline{\Delta}$ in which ε is a real valued scalar and $\underline{\Delta}$ is a perturbation direction vector. The perturbation direction vector is chosen so that the original N-1 zero components in the unperturbed residual error vector $r(\underline{x}^o) = \underline{y} - A\underline{x}^o$ are maintained in the perturbed residual error vector $r(\underline{x}^o + \varepsilon\underline{\Delta}) = \underline{y} - A\underline{x}^o - \varepsilon A\underline{\Delta}$. This restriction is always possible since the perturbation direction vector Δ can always be selected to be orthogonal to each of the $0 \le N_0 \le N-1$ row vectors of the matrix A associated with the zero components of $r(\underline{x}^o) = \underline{y} - A\underline{x}^o$. Furthermore, the scalar ε is chosen to be small enough in magnitude so that the signs of all the nonzero components in the perturbed residue error vector $r(\underline{x}^o + \varepsilon\underline{\Delta}) = \underline{y} - A\underline{x}^o - \varepsilon A\underline{\Delta}$ are maintained. With this constraint, the corresponding sum of the residue error magnitude criterion at the perturbed vector $\underline{x}^o + \varepsilon\underline{\Delta}$ is given by

$$f_1(\underline{x}^o + \varepsilon\underline{\Delta}) = \sum_{r_m(\underline{x})>0}[y(m) - \underline{e}_m^T A(\underline{x}^o + \varepsilon\underline{\Delta})] - \sum_{r_m(\underline{x})<0}[y(m) - \underline{e}_m^T A(\underline{x}^o + \varepsilon\underline{\Delta})]$$

$$= f_1(\underline{x}^o) + \varepsilon\left[\sum_{r_m(\underline{x})<0}\underline{e}_m^T - \sum_{r_m(\underline{x})>0}\underline{e}_m^T\right]A\underline{\Delta}. \qquad (4.45)$$

The vector term shown in the bracket of equation (4.45) is seen to have components exclusively plus one, minus one and zero. It is also observed that if this bracketed vector is not orthogonal to $A\underline{\Delta}$ then a scalar ε can be chosen so that $f_1(\underline{x}^o + \varepsilon\underline{\Delta}) < f_1(\underline{x}^o)$ which contradicts to the fact that $\underline{x}^o$ is the optimum solution. On the other hand, if this bracketed vector is orthogonal to $A\underline{\Delta}$ then a scalar ε may be gradually increased or decreased from zero until a formally nonzero element of the unperturbed residual error vector $r(\underline{x}^o) = \underline{y} - A\underline{x}^o$ is first driven to zero while maintaining the condition $f_1(\underline{x}^o + \varepsilon\underline{\Delta}) = f_1(\underline{x}^o)$. This procedure is continued until eventually N components of the perturbed error vector $r(\underline{x}^o + \varepsilon\underline{\Delta}) = \underline{y} - A\underline{x}^o - \varepsilon A\underline{\Delta}$ are zero and therefore results in the following theorem.

Theorem (4.3) [72] For any given vector $\underline{y} \in R^{M\times 1}$ and matrix $A \in R^{M\times N}$ of rank *N*, there exists a real valued vector $\underline{x}^o \in R^{N\times 1}$ which minimizes the l_1 norm-induced function $f_1(\underline{x}^o)$ appearing in expression (4.42) such that the associated residue error vector $r(\underline{x}^o) = \underline{y} - A\underline{x}^o$ has at least *N* zero components*.

Since a convenient closed form for a real valued vector $\underline{x}^o \in R^{N\times 1}$, which minimizes the l_1 norm-induced function $f_1(\underline{x}^o)$, does not exist it is therefore necessary to develop an iterative algorithmic procedure to numerically accomplish such solutions. Many algorithms to determine a minimum l_1 norm solution employ the previously developed property stated in Theorem (4.3). Additionally, some are based on the exchange principal [68] [74] [75] in which one of the equations in the present set of *N* equations having zero residual is exchanged for another equation in the remaining set of *M*-*N* equations in such a manner that the new set of *N* equations results in a better l_1 residual error vector norm. In this dissertation, the perturbation algorithm developed by Cadzow [72] is employed. This technique provides an effective algorithm, which converges to a solution at a faster rate than the class of exchange algorithms. The procedure of this technique begins with a perturbation analysis of the functional $f_1(\underline{x})$ as given by

$$\begin{aligned} f_1(\underline{x}) &= \sum_{r_m(\underline{x})>0} [y(m) - \underline{e}_m^T A\underline{x}] - \sum_{r_m(\underline{x})<0} [y(m) - \underline{e}_m^T A\underline{x}] \\ &= \sum_{r_m(\underline{x})>0} [y(m) - \underline{a}_m^T \underline{x}] - \sum_{r_m(\underline{x})<0} [y(m) - \underline{a}_m^T \underline{x}] \end{aligned} \tag{4.46}$$

in which $\underline{a}_m^T = \underline{e}_m^T A$ is 1 × *N* vector corresponding to the m^{th} row vector of the *M* × *N* system matrix *A* whereas $r_m(\underline{x}) = y(m) - \underline{e}_m^T A\underline{x}$ is the m^{th} component of the residue error vector $r(\underline{x})$.

---

* The vector $\underline{x}$ is said to be an extreme, degenerate or non-degenerate point of the system of linear equations if the associated error vector $r(\underline{x}) = \underline{y} - A\underline{x}$ has at least, more than or exactly *N* zero components, respectively [72]. Furthermore, by this Theorem, there always exist a minimum l_1 norm solution which is an extreme point. Contrarily, an extreme point need not be a minimum l_1 norm solution.

Let N_x for $0 \le N_x \le M$ be the number of zero residue error vector components (i.e., $r_m(\underline{x}) = \underline{0}$) and let the unperturbed $N \times 1$ vector $\underline{x}$ now be the perturbed vector $\underline{x} + \varepsilon \underline{\Delta}$ in which $\underline{\Delta}$ is a perturbation direction vector whose length is controlled by a real valued step size scalar ε. The magnitude of ε is restricted to be sufficiently small so that the signs of the nonzero components of the residue error vector $r(\underline{x})$ are the same as those of the residue error vector $r(\underline{x} + \varepsilon \underline{\Delta})$. Under the sign preservation, it follows that the perturbed l_1 functional can be expressed as

$$\begin{aligned} f_1(\underline{x} + \varepsilon \underline{\Delta}) &= \sum_{r_m(\underline{x})>0} [y(m) - \underline{a}_m^T(\underline{x} + \varepsilon \underline{\Delta})] - \sum_{r_m(\underline{x})<0} [y(m) - \underline{a}_m^T(\underline{x} + \varepsilon \underline{\Delta})] + \sum_{r_m(\underline{x})=0} \left| - \underline{a}_m^T(\varepsilon \underline{\Delta}) \right| \\ &= f_1(\underline{x}) + \sum_{r_m(\underline{x})>0} [-\underline{a}_m^T(\varepsilon \underline{\Delta})] - \sum_{r_m(\underline{x})<0} [-\underline{a}_m^T(\varepsilon \underline{\Delta})] + |\varepsilon| \sum_{r_m(\underline{x})=0} \left| \underline{a}_m^T \underline{\Delta} \right| \\ &= f_1(\underline{x}) - \varepsilon \left[\sum_{r_m(\underline{x})>0} \underline{a}_m^T - \sum_{r_m(\underline{x})<0} \underline{a}_m^T \right] \underline{\Delta} + |\varepsilon| \sum_{r_m(\underline{x})=0} |A_x \underline{\Delta}| \\ &= f_1(\underline{x}) - \varepsilon \underline{b}_x^T \underline{\Delta} + |\varepsilon| \left\| A_x \underline{\Delta} \right\|_1 . \end{aligned} \tag{4.47}$$

The A_x shown in the third line of expression (4.47) is the $N_x \times N$ sub-matrix of the $M \times N$ system matrix A whose row vectors correspond to the row vectors of A associated with the zero components of the unperturbed residue error vector. The $N \times 1$ vector $\underline{b}_x$ shown in the last line of expression is specified by

$$\underline{b}_x = \sum_{r_m(\underline{x})>0} \underline{a}_m - \sum_{r_m(\underline{x})<0} \underline{a}_m . \tag{4.48}$$

This perturbation analysis leads to the following perturbation theorem.

Theorem (4.4) [72] *Let the $M \times N$ system matrix $A \in R^{M \times N}$ have a full column rank N. Furthermore, for a given $N \times 1$ vector $\underline{x} \in R^{N \times 1}$, let the associated residue error vector $r(\underline{x}) = \underline{y} - A\underline{x}$ have N_x zeros with $0 \le N_x < N$. It is always possible to perturb the parameter vector $\underline{x}$ to $\underline{x} + \varepsilon \underline{\Delta}$ in a manner such that $f_1(\underline{x} + \varepsilon \underline{\Delta}) \le f_1(\underline{x})$ while simultaneously maintaining*

the original N_x *zero elements in the perturbed residue error vector* $r(\underline{x} + \varepsilon \underline{\Delta})$ *and causing at least one previously nonzero element of* $r(\underline{x})$ *to be zero*.

Proof : Let the perturbation direction vector $\underline{\Delta}$ be selected to be any nonzero vector contained in $R^{N\times 1}$ for which $A_x \underline{\Delta} = 0$. This possibility is always valid since the dimension of the null space of the $N_x \times N$ submatrix A_x is greater than or equal to $N - N_x$. By using the constraint $A_x \underline{\Delta} = 0$, it follows that the perturbation l_1 norm functional indicated in expression (4.47) is simplified to be

$$f_1(\underline{x} + \varepsilon \underline{\Delta}) = f_1(\underline{x}) - \varepsilon \underline{b}_x^T \underline{\Delta}. \tag{4.49}$$

Upon observation expression (4.49), three possibilities arise depending on whether the term $\underline{b}_x^T \underline{\Delta}$ is positive, negative or zero. If $\underline{b}_x^T \underline{\Delta}$ is greater than zero, the scalar ε gradually increases from zero, thereby causing $f_1(\underline{x} + \varepsilon \underline{\Delta}) < f_1(\underline{x})$, until a value is eventually reached whereby one of the previously nonzero residue error elements is first driven to zero. Such a positive scalar selection is assured because otherwise the nonnegative functional $f_1(\underline{x} + \varepsilon \underline{\Delta})$ could be driven negative. Similarly, if $\underline{b}_x^T \underline{\Delta}$ is negative, then the scalar ε gradually decreases from zero to cause $f_1(\underline{x} + \varepsilon \underline{\Delta}) < f_1(\underline{x})$ until a value eventually arrives in which one of the previously nonzero residue error elements is first driven to zero. Such a negative scalar selection is also guaranteed because otherwise the nonnegative functional $f_1(\underline{x} + \varepsilon \underline{\Delta})$ could be driven negative. Finally, if $\underline{b}_x^T \underline{\Delta}$ is zero, the scalar ε gradually increases or decreases from zero until at least one of the previously nonzero residue error elements is driven to zero while maintaining the functional value $f_1(\underline{x} + \varepsilon \underline{\Delta}) = f_1(\underline{x})$.

Theorem (4.4) basically provides a mechanism for perturbing a non-optimum vector $\underline{x}$ into an extreme point whose the associated residual error vector $r(\underline{x}) = \underline{y} - A\underline{x}$ has at least N zero components while simultaneously decreasing or maintaining the value of the functional $f_1(\underline{x})$. It is also noted that if the vector $\underline{x}$ is not the minimum l_1 norm solution with $N_x < N$, it

has been empirically found that this hypothesized procedure for selecting perturbation direction vector $\underline{\Delta}$ has always led to an improvement such that $f_1(\underline{x}+\varepsilon\underline{\Delta}) < f_1(\underline{x})$. By continuing this procedure, an extreme point $\underline{x}$ is eventually reached in which the associated residual error vector $r(\underline{x}) = \underline{y} - A\underline{x}$ has at least N zero components while the l_1 norm of the residue error vector is either decreased or at worst maintained. Once this extreme point has been obtained, it is necessary to verify whether it is a required minimum l_1 norm solution. The following theorem provides a mechanism for making the decision.

Theorem (4.5) [69] *Let the $M \times N$ system matrix $A \in R^{M\times N}$ have full column rank N and let the $N \times 1$ vector $\underline{x} \in R^{N\times 1}$ be a non-degenerate extreme point so that its associated residue error vector $r(\underline{x}) = \underline{y} - A\underline{x}$ has exactly N zero elements. Furthermore, let A_x be the $N_x \times N$ whose row vectors correspond to the row vectors of the matrix A associated with the N zero elements of the residue error vector and let $\underline{b}_x$ be specified by equation (4.48). If the matrix A is invertible then the non-degenerate extreme point $\underline{x}$ is a minimum l_1 norm solution if and only if all the components of the vector*

$$\underline{c}_x = \left[A_x^T\right]^{-1}\underline{b}_x \tag{4.50}$$

have magnitudes less than or equal to one. Moreover, this solution is a unique minimum l_1 norm solution if and only if all the components of $\underline{c}_x$ have magnitudes strictly less than one.

Proof [*] : Let use the fact that the column vectors of $\left[A_x\right]^{-1}$ form a basis of $R^{N\times 1}$ so that any perturbation vector can be uniquely represented as $\varepsilon\underline{\Delta} = \left[A_x\right]^{-1}\underline{\delta}$ for $\underline{\delta} \in R^{N\times 1}$. Upon this expression has been substituted into expression (4.47) for the perturbation vector and $\underline{b}_x = A_x^T\underline{c}_x$ has been used, it then follows that

---

[*] A proof of this theorem can be found in [69], however, an alternative proof given by Cadzow [72] is herein presented.

$$
\begin{aligned}
f_1(\underline{x}+\varepsilon\,\underline{\Delta}) &= f_1(\underline{x}) - \left[A_x^T\,\underline{c}_x\right]^T A_x^{-1}\,\underline{\delta} + \left\|A_x A_x^{-1}\,\underline{\delta}\right\|_1 \\
&= f_1(\underline{x}) - \underline{c}_x^T A_x A_x^{-1}\,\underline{\delta} + \left\|\underline{\delta}\right\|_1 \\
&= f_1(\underline{x}) - \underline{c}_x^T\,\underline{\delta} + \left\|\underline{\delta}\right\|_1 \\
&= f_1(\underline{x}) - \sum_{n=1}^{N}\left[c_x(n)\delta(n)\right] + \sum_{n=1}^{N}\left|\delta(n)\right| \\
&= f_1(\underline{x}) - \sum_{n=1}^{N}\left[c_x(n)\left|\delta(n)\right|\operatorname{sgn}\left(\delta(n)\right)\right] + \sum_{n=1}^{N}\left|\delta(n)\right| \\
&= f_1(\underline{x}) + \sum_{n=1}^{N}\left|\delta(n)\right|\left[1 - \operatorname{sgn}\left(\delta(n)\right)c_x(n)\right]. \qquad (4.51)
\end{aligned}
$$

For a given set of $c_x(n)$ coefficients, the signs of the $\delta(n)$ coefficients should be selected so that $\operatorname{sgn}\left(\delta(n)\right) = \operatorname{sgn}\left(c_x(n)\right)$ in order to cause the summand terms $-\left|\delta(n)\right|\operatorname{sgn}\left(\delta(n)\right)c_x(n)$ appearing in expression (4.51) maximally negative which leads to

$$
f_1(\underline{x}+\varepsilon\,\underline{\Delta}) = f_1(\underline{x}) - \sum_{n=1}^{N}\left|\delta(n)\right|\left[\left|c_x(n)\right| - 1\right]. \qquad (4.52)
$$

It is obvious that if each element $c_x(n)$ has its magnitude less than or equal to one then this summation is always non-positive and the improvement of the l_1 norm functional cannot be made, that is $f_1(\underline{x}+\varepsilon\,\underline{\Delta}) \geq f_1(\underline{x})$. On the other hand, if $c_x(n_1)$ is designated to be any element of $\underline{c}_x$ whose magnitude is greater than one then the choice $\delta(n) = 0$ for all n except for $\delta(n_1) = \beta$ therefore yields

$$
f_1(\underline{x}+\varepsilon\,\underline{\Delta}) = f_1(\underline{x}) - \left|\beta\right|\left[\left|c_x(n_1)\right| - 1\right] \qquad (4.53)
$$

which results in the desired improvement $f_1(\underline{x}+\varepsilon\,\underline{\Delta}) < f_1(\underline{x})$. However, the scalar β must be chosen to be small enough to achieve the aforementioned sign preservation of the residue error

vector. It is further noted that if more than one component of $\underline{c}_x$ have a magnitude greater than one, a variety of different improving perturbation directions can be developed. Upon arriving at an extreme vector $\underline{x}$ whose associated residue error vector has N zero elements, Theorem (4.5) is then employed to examine whether or not this vector is a minimum l_1 norm solution. If this extreme vector is not an optimum, the following theorem is used as a mechanism to determine an improving perturbation so that $f_1(\underline{x} + \varepsilon \underline{\Delta}) < f_1(\underline{x})$.

Theorem (4.6) [72] Let the M × N system matrix $A \in R^{M \times N}$ have full column rank N and let the N × 1 vector $\underline{x} \in R^{N \times 1}$ be a non-degenerate extreme point so that its associated residue error vector $r(\underline{x}) = \underline{y} - A\underline{x}$ has exactly N zero elements. Furthermore, let the $N \times N$ matrix A_x whose row vectors correspond to the row vectors of the matrix A associated with the N zero elements of the residue error vector be invertible and let $\underline{c}_x$ be specified by expression (4.50). If $c_x(n_1)$ denotes any component whose magnitude is larger than one, it then follows that a perturbation direction vector $\underline{\Delta}$ which renders the improvement $f_1(\underline{x} + \varepsilon \underline{\Delta}) < f_1(\underline{x})$ is given by

$$\underline{\Delta} = \alpha^o \left[A_x\right]^{-1} \underline{e}_{n_1} \tag{4.54}$$

where α^o is a unique value of the scalar α for which the vector $\alpha\left[A_x\right]^{-1}\underline{e}_{n_1}$ best approximates the residue error vector $r(\underline{x})$ in the l_1 norm sense. This new improving perturbed vector causes the associated perturbed residue error vector to have at least one zero element.

This theorem is readily proven by equation (4.53) which indicates that the improving vector perturbation lies in the one-dimensional space spanned by $\left[A_x\right]^{-1}\underline{e}_{n_1}$. The algorithm for obtaining a minimum l_1 norm solution to a linear system of *M* equations in one unknown discussed in the previous section is used to find an optimum scalar α^o which causes $\alpha^o\left[A_x\right]^{-1}\underline{e}_{n_1}$ to best approximate the residue error vector $r(\underline{x})$ in l_1 norm sense. It is recalled that the optimum scalar α^o shown in expression (4.54) of Theorem (4.6) has been predicated on the assumption that an extreme point is non-degenerate where its associated residue error vector

$r(\underline{x})$ has exactly N zero elements. However, when a degenerate extreme point is encountered, the following theorem gives a method to determine whether this point is an optimum.

Theorem (4.7) [72] *Let the full column rank matrix A satisfy the Haar condition* and let* $\underline{x} \in R^{N\times 1}$ *be a degenerate point so that its associated residue error vector* $r(\underline{x}) = \underline{y} - A\underline{x}$ *has* $Q > N$ *zero elements. Furthermore, let the* $Q \times N$ *matrix* A_x *whose row vectors correspond to the row vectors of matrix A associated with these Q zero error elements and let* $\underline{c}_x$ *be specified by expression (4.50). This degenerate extreme point is a minimum* l_1 *norm solution if and only of none of the vectors*

$$\underline{c}_x^{(k)} = \left[\left[A_x^{(k)}\right]^T\right]^{-1}\underline{b}_x \quad for \quad 1 \le k \le \frac{Q!}{N!(Q-N)!} \tag{4.55}$$

has $\left\|\underline{c}_x^{(k)}\right\|_\infty > 1$ *in which the* $\left\{A_x^{(k)}\right\}$ *designates a set of all* $N \times N$ *sub-matrices of matrix A.*

A summary of perturbation algorithm for finding a minimum l_1 approximate solution is now given.

Algorithmic Solution for Finding a Minimum l_1 Norm Problem

The algorithmic approach now described is predicated on systemically proceeding from a present $N\times 1$ vector $\underline{x}$ (i.e. $\underline{x}_i$) to an updated $N\times 1$ vector $\underline{x}$ (i.e. $\underline{x}_j$) in such a manner so that the l_1 norm functional $f_1(\underline{x})$ is decreased at each transition (i.e. $f_1(\underline{x}_j) \le f_1(\underline{x}_i)$). This algorithmic provides more efficient procedure, in which the number of iterations needed until the optimum solution is reached, is significantly smaller than the brute force and row-exchange algorithms. The algorithm is composed of sequential steps as following.

---

* The set of vectors $\{\underline{x}_1, \underline{x}_2, ..., \underline{x}_N\}$ contained in vector space $R^{M\times 1}$ is said to satisfy the Haar condition [73] if every selection of $M \le N$ of its elements forms a linearly independent set.

Step 1. For an arbitrary initial selection of $\underline{x}$ (e.g., the zero vector or the minimum l_2 approximate solution), generate the associated residue error vector $r(\underline{x}) = \underline{y} - A\underline{x}$.

Step 2. If the residue error vector $r(\underline{x})$ has fewer than N zero elements then go to Step 3. Otherwise go to Step 5.

Step 3. Using the procedure as described in Theorem (4.4), perturb the vector $\underline{x}$ to $\underline{x} + \varepsilon\underline{\Delta}$ in such a manner that $f_1(\underline{x} + \varepsilon\underline{\Delta}) \le f_1(\underline{x})$ while causing the associated perturbed error vector to have at least one additional zero element.

Step 4. If the residue error vector associated with the new vector $\underline{x}$ has fewer than N zero elements then repeat Step 3. Otherwise go to Step 5.

Step 5. Determine whether the vector $\underline{c}_x = \left[A_x^T\right]^{-1}\underline{b}_x$ has at least one component whose magnitude is greater than one. If such a component exists then make the improvement in the l_1 norm functional $f_1(\underline{x})$ in accordance with Theorem (4.6) and then go to Step 2. If such a component does not exist, the algorithm has converged to the optimal solution.

At the end of Step 5., the vector $\underline{x}$ has its associated residue error vector with N zero elements. It is now a candidate for the optimum solution in accordance with Theorem (4.3). Furthermore, if the improvement condition of l_1 norm function cannot be made, it implies that the vector $\underline{x}$ must be the optimum solution due to the property of convexity of functional $f_1(\underline{x})$.

Minimum l_2 Norm Solution to a Linear System of *M* Equations in *N* Unknowns

It is now desired to select a parameter vector $\underline{x} \in R^{N\times 1}$ which minimizes the l_2 norm of the residue error vector $r(\underline{x})$ shown in expression (4.41). This corresponds to find the optimum solution [73] which minimizes the l_2 norm-induced function as designated by

$$f_2(\underline{x}) = \left\|r(\underline{x})\right\|_2 = \left\|\underline{y} - A\underline{x}\right\|_2$$

$$= \sqrt{\left[\underline{y} - A\underline{x}\right]^T \left[\underline{y} - A\underline{x}\right]}$$

$$= \sqrt{\underline{y}^T \underline{y} - 2\underline{y}^T A\underline{x} + \underline{x}^T A^T A\underline{x}}\,. \tag{4.56}$$

It is recalled that a necessary condition for a vector $\underline{x}^o$ to minimize function $f_2(\underline{x})$ is that the gradient of this function when evaluated at $\underline{x}^o$ is equal to the zero vector. Using standard differentiation rules it is found that this gradient vector is designated by

$$\nabla_{\underline{x}} f_2(\underline{x}) = \frac{1}{\left\|\underline{y} - A\underline{x}\right\|_2}\left(A^T A\underline{x} - A^T \underline{y}\right). \tag{4.57}$$

Upon setting this gradient equal to zero vector, it is seen that a necessary condition for a minimum l_2 norm solution is one which satisfies the so-called *normal system equat*ion [71] as specified by $A^T A\underline{x}^o = A^T \underline{y}$. Thus the following fundamental theorem has been proven.

Theorem (4.8) [73] *For any vector* $\underline{y} \in R^{M\times 1}$ *and matrix* $A \in R^{M\times N}$, *the set of vectors* $\underline{x}_2^o$ *which minimizes the sum of squared residue error function* $f_2(\underline{x}) = \left\|\underline{y} - A\underline{x}\right\|_2$ *is equal to all solutions of the consistent linear system of normal equation as designated by*

$$A^T A\underline{x}^o = A^T \underline{y}\,. \tag{4.58}$$

Furthermore, all solutions to the normal equation result in the same associated residue error vector $r(\underline{x}^o) = \underline{y} - A\underline{x}^o$ *which is orthogonal to the row vectors of matrix A, that is*

$$A^T r(\underline{x}^o) = \underline{0}\,. \tag{4.59}$$

If the matrix A has full column rank N (the matrix A is invertible) then there exists a unique solution to the normal equation as given by

$$\underline{x}^{o} = \left(A^{T} A\right)^{-1} A^{T} \underline{y} \,. \tag{4.60}$$

Otherwise,

$$\underline{x}^{o} = A^{\dagger} \underline{y} \tag{4.61}$$

in which $A^{\dagger}$ *designates the pseudo inverse of the* $M \times N$ *matrix A.*

Proof : The fact that all solutions which minimize the function $f_2(\underline{x}) = \left\| \underline{y} - A\underline{x} \right\|_2$ and satisfy the normal equation has already been established. To verify the orthogonal condition $A^{T} r(\underline{x}^{o}) = \underline{0}$, one simply left multiplies the residue error vector $r(\underline{x}^{o}) = \underline{y} - A\underline{x}^{o}$ by the matrix A^{T} to obtain $A^{T} r(\underline{x}^{o}) = A^{T} \underline{y} - A^{T} A\underline{x}^{o}$ which is equal to the zero vector since $\underline{x}^{o}$ satisfies the normal equation. Finally, if matrix A has rank N then the $N \times N$ matrix product $A^{T} A$ is invertible leading to the unique solution shown in expression (4.60).

According to the given optimum scalars selection algorithms in this chapter, the proposed decompositions based on one-dimensional and two-dimensional decorrelation given in Chapter II and III, respectively, are therefore processed in an efficient manner. It is noted that the applications of the proposed method will focus on the lossless and lossy image compression. The coding method employed in this research is the SPIHT developed from the underlying principles of the embedded zerotree wavelet (EZW) technique [76] into an alternative exposition providing improved results [77]. The detail of this coding algorithm can be found in [54] and [78]. The lossless and lossy image compression results are represent in the next chapter.

CHAPTER V

RESULTS AND DISCUSSIONS

In this chapter, the experiments based on the proposed image decomposition algorithms are conducted for decorrelation performance evaluation, lossless and lossy image compressions. The experimental results obtained from the proposed schemes using the l_1 and l_2 norm optimum interpolation filters are compared with those obtained by the LDT-HVDD method employing the l_1 and l_2 norm optimum interpolation length-4 filters, the S+P-transform* and the wavelet transform using the Daubechies length-4 filters. These image decompositions are applied to a set of gray scale images shown in Appendix A.

Decorrelaion Derformance Evaluation

To measure the decorrelation performance of the tested methods, the first-order entropy (bits/pixel) results of the interpolation errors are examined. In this study, such results obtained with one-level lossless image decomposition of the S+P-transform, the l_1 and l_2 norm based modified S+P-transform**, the l_1 and l_2 norm based 2D-HVDD method, the l_1 and l_2 norm based optimum scalar method and the l_1 and l_2 norm based LDT-HVDD scheme are compared and shown in Table 5.1. Furthermore, the results of two-level lossless image decomposition tabulated in Table 5.2 illustrate the tendency of decorrelation performance of the tested schemes.

As shown in Table 5.1 for the decorrelation performance of 1-level image decomposition based on different methods, the l_1 norm based 2D-HVDD method (2D-HVDD(l_1)) gives the smallest average bits per pixel of the first-order entropy of the interpolation errors. However, this numerical result is not significantly smaller than the result of the S+P-transform. For the 2-level decomposition case shown in Table 5.2, the S+P-transform is superior to the other methods.

---

* Since the S+P-transform is an integer image transformation, it is only compared with the proposed techniques for lossless image compression.

** It is noted that the modified S+P-transform has the same basic concept as the conventional S+P method; however, the l_p norm minimization method is employed to determine the parameters used in the calculation of the $\hat{h}(n)$ sequence.

It is noted that the decorrelation performance of tested methods in 1-level decomposition is a litter bit better than that of the 2-level case. This is because image pixels tend to be less correlated at the second level of image decomposition. Therefore, the decorrelation performance in the second level decomposition is not as good as in the first level.

Table 5.1 Comparative evaluation of the first-order entropy (bits/pixel) of detail coefficients obtained with different tested methods for 1-level image decomposition.

Comparative evaluation of the first-order entropy (bits/pixel) of detail coefficients obtained with different image transformations (1-level image decomposotion)													
Methods/Images	Airplane	Camera	Barbara	Face	Lena	Peppers	Milkdrop	Tiffany	Zelda	CT	X-rays	Fingerprint	Average (bpp)
S+P	4.9198	4.8289	5.1098	4.2188	4.6807	4.5674	4.2122	4.7101	4.2675	3.0228	3.3135	5.7015	4.4628
Modified S+P (l_1)	5.0663	4.9363	5.1292	4.3159	4.7608	4.7584	4.3936	4.7848	4.5055	3.1307	3.7663	5.8984	4.6205
Modified S+P (l_2)	5.0968	4.9491	5.3071	4.3187	4.8050	4.8099	4.3782	4.8243	4.5446	3.1420	3.8796	5.9027	4.6632
2D-HVDD (l_1)	4.9357	4.8018	4.9916	4.1319	4.6244	4.6518	4.2973	4.5950	4.2986	3.0000	3.4992	5.7190	4.4622
2D-HVDD (l_2)	4.9394	4.8027	4.9971	4.1368	4.6361	4.6606	4.2994	4.6025	4.2989	3.1410	3.5077	6.0061	4.5024
OS-HVD (l_1)	5.3013	5.0232	6.1316	4.7387	5.3993	5.2387	4.6752	5.1291	5.0817	3.2555	4.2926	5.8427	5.0091
OS-HVD (l_2)	5.2996	5.0645	6.1550	4.7697	5.4042	5.2472	4.6860	5.1291	5.0807	3.4470	4.3418	6.1272	5.0627
OS-HVVD (l_1)	5.2314	4.9979	5.8954	4.6659	5.2316	5.0797	4.5435	5.0622	4.9066	3.1897	4.1229	5.8485	4.8979
OS-HVVD (l_2)	5.2422	5.0307	5.9117	4.6874	5.2547	5.0909	4.5637	5.0626	4.9063	3.3888	4.1445	6.3589	4.9702
OS-HVDD (l_1)	5.4657	5.1491	6.2132	4.8597	5.4754	5.3750	4.8272	5.1911	5.1871	3.3172	4.4347	5.9098	5.1171
OS-HVDD (l_2)	5.4635	5.2073	6.2408	4.8999	5.4804	5.3824	4.8354	5.1911	5.1862	3.5282	4.4961	6.4272	5.1949
LDT-HVDD (l_1)	5.1422	4.9306	5.7411	4.3291	4.9783	4.8425	4.4578	4.8448	4.5714	3.0913	3.7342	6.0776	4.7284
LDT-HVDD (l_2)	5.1459	4.9463	5.8041	4.3459	4.9766	4.8502	4.5156	4.8857	4.5688	3.1854	3.7398	6.3211	4.7738

Table 5.2 Comparative evaluation of the first-order entropy (bits/pixel) of detail coefficients obtained with different tested methods for 2-level image decomposition.

Comparative evaluation of the first-order entropy (bits/pixel) of detail coefficients obtained with different image transformations (2-level image decomposotion)													
Methods/Images	Airplane	Camera	Barbara	Face	Lena	Peppers	Milkdrop	Tiffany	Zelda	CT	X-rays	Fingerprint	Average (bpp)
S+P	4.9893	4.8584	5.1998	4.2564	4.7443	4.6474	4.2702	4.6964	4.3291	3.1914	3.4724	5.8034	4.5382
Modified S+P (l_1)	5.2133	4.9671	5.2489	4.3813	4.8277	4.8328	4.4589	4.7972	4.5621	3.2844	3.8796	5.9681	4.7018
Modified S+P (l_2)	5.2655	4.9996	5.4149	4.4035	4.8799	4.8930	4.4548	4.8518	4.6136	3.3136	3.9687	5.9668	4.7521
2D-HVDD (l_1)	5.0927	4.9286	5.2239	4.2971	4.7880	4.8367	4.4651	4.7068	4.4627	3.1475	3.7062	5.8417	4.6248
2D-HVDD (l_2)	5.0955	4.9289	5.2394	4.2995	4.8018	4.8472	4.4625	4.7161	4.4639	3.2882	3.7102	6.0998	4.6628
OS-HVD (l_1)	5.4218	5.1083	6.2472	4.8638	5.5296	5.3945	4.8379	5.2368	5.2320	3.3477	4.4355	5.9491	5.1337
OS-HVD (l_2)	5.4233	5.2033	6.2712	4.9121	5.5417	5.4100	4.8554	5.2368	5.2317	3.5568	4.5046	6.4889	5.2197
OS-HVVD (l_1)	5.3724	5.0928	6.0501	4.7967	5.3815	5.2531	4.7207	5.1764	5.0671	3.3035	4.2857	5.9645	5.0387
OS-HVVD (l_2)	5.3883	5.1668	6.0700	4.8286	5.4051	5.2664	4.7475	5.1767	5.0695	3.5086	4.3259	6.7073	5.1384
OS-HVDD (l_1)	5.5869	5.2340	6.3090	4.9829	5.6035	5.5276	4.9876	5.2978	5.3387	3.4041	4.5717	5.9949	5.2366
OS-HVDD (l_2)	5.5908	5.3491	6.3334	5.0388	5.6143	5.5414	5.0009	5.2983	5.3381	3.6301	4.6492	6.7013	5.3405
LDT-HVDD (l_1)	5.3000	5.0489	5.8799	4.4933	5.1497	5.0308	4.6134	4.9655	4.7289	3.2344	3.9371	6.2168	4.8832
LDT-HVDD (l_2)	5.2962	5.0669	5.9575	4.5070	5.1498	5.0385	4.6643	5.0027	4.7252	3.3005	3.9485	6.5337	4.9326

Lossless Image Compression Results

For lossless image compression, the Huffman coding is first employed to encode the transformed images obtained from different tested methods. The average Huffman coded bits per pixel obtained by testing methods for 3-level and 5-level lossless image decomposition are shown in Table 5.3 and 5.4, respectively.

Table 5.3 Comparative Huffman coding (bits/pixel) of 3-level lossless image decomposition.

Comparative Huffman coding (bits/pixel) of 3-level image decomposition obtained with different image transformations													
Methods/Images	Airplane	Camera	Barbara	Face	Lena	Peppers	Milkdrop	Tiffany	Zelda	CT	X-rays	Fingerprint	Average (bpp)
S+P	5.0724	5.0163	5.3678	4.4605	4.9468	4.8651	4.4728	4.8728	4.5524	3.3441	3.7677	5.9429	4.7235
Modified S+P (l_1)	5.3076	5.1208	5.4285	4.5858	5.0253	5.0498	4.6511	4.9829	4.7859	3.4399	4.1222	6.1135	4.8844
Modified S+P (l_2)	5.3638	5.1573	5.5801	4.6165	5.0832	5.1134	4.6391	5.0384	4.8412	3.4731	4.1885	6.1016	4.9330
2D-HVDD (l_1)	5.3148	5.1186	5.4667	4.5308	5.0234	5.0884	4.6963	4.9296	4.7121	3.3016	3.9989	6.0065	4.8490
2D-HVDD (l_2)	5.3232	5.1203	5.4849	4.5309	5.0399	5.1021	4.7030	4.9408	4.7188	3.4306	4.0013	6.2231	4.8849
OS-HVD (l_1)	5.5358	5.2751	6.4061	5.0621	5.7456	5.6074	5.0510	5.4535	5.4524	3.4668	4.6662	6.0777	5.3166
OS-HVD (l_2)	5.5480	5.3950	6.4291	5.1046	5.7555	5.6249	5.0793	5.4534	5.4535	3.6696	4.7441	6.6548	5.4093
OS-HVVD (l_1)	5.4935	5.2648	6.2215	4.9914	5.6056	5.4803	4.9355	5.3974	5.3034	3.4269	4.5328	6.0993	5.2294
OS-HVVD (l_2)	5.5148	5.3613	6.2444	5.0281	5.6311	5.4979	4.9754	5.3978	5.3047	3.6257	4.5798	6.8593	5.3350
OS-HVDD (l_1)	5.7035	5.3984	6.4682	5.1776	5.8140	5.7302	5.1961	5.5133	5.5577	3.5211	4.5798	6.1287	5.3991
OS-HVDD (l_2)	5.7218	5.5340	6.4917	5.2285	5.8201	5.7466	5.2160	5.5146	5.5572	3.7372	4.7930	6.8690	5.5191
LDT-HVDD (l_1)	5.5192	5.2324	6.0558	4.7098	5.3859	5.2758	4.8389	5.1879	4.9698	3.3826	4.2121	6.3705	5.0951
LDT-HVDD (l_2)	5.5079	5.2554	6.1445	4.7199	5.3803	5.2855	4.8705	5.2333	4.9658	3.4587	4.2173	6.6804	5.1433

Table 5.4 Comparative Huffman coding (bits/pixel) of 5-level lossless image decomposition.

Comparative Huffman coding (bits/pixel) of 5-level image decomposition obtained with different image transformations													
Methods/Images	Airplane	Camera	Barbara	Face	Lena	Peppers	Milkdrop	Tiffany	Zelda	CT	X-rays	Fingerprint	Average (bpp)
S+P	5.0724	4.9198	5.2677	4.3839	4.8431	4.7713	4.3569	4.7582	4.4502	3.3352	3.6676	5.8675	4.6412
Modified S+P (l_1)	5.3076	5.0293	5.3348	4.5225	4.9245	4.9666	4.5471	4.8784	4.6961	3.4321	4.0195	6.0462	4.8087
Modified S+P (l_2)	5.3638	5.0671	5.4842	4.5606	4.9853	5.0283	4.5363	4.9312	4.7536	3.4690	4.0589	6.0360	4.8562
2D-HVDD (l_1)	5.2310	5.0367	5.3884	4.4730	4.9353	5.0147	4.6111	4.8289	4.6284	3.2942	3.9157	5.9487	4.7755
2D-HVDD (l_2)	5.2361	5.0364	5.4102	4.4728	4.9535	5.0296	4.6155	4.8418	4.6344	3.4282	3.9157	6.1713	4.8121
OS-HVD (l_1)	5.5358	5.1985	6.3383	5.0090	5.6681	5.5452	4.9699	5.3577	5.3758	3.4492	4.5865	6.0205	5.2545
OS-HVD (l_2)	5.5480	5.3353	6.3631	5.0627	5.6820	5.5652	5.0030	5.3590	5.3761	3.6574	4.6735	6.6234	5.3541
OS-HVVD (l_1)	5.4935	5.1864	6.1513	4.9374	5.5273	5.4155	4.8497	5.3007	5.2246	3.4125	4.4536	6.0414	5.1662
OS-HVVD (l_2)	5.5148	5.2984	6.1753	4.9832	5.5550	5.4345	4.8956	5.3025	5.2261	3.6157	4.5090	6.8279	5.2782
OS-HVDD (l_1)	5.7035	5.3252	6.4019	5.1313	5.7397	5.6715	5.1202	5.4192	5.4858	3.5057	4.7152	6.0715	5.3576
OS-HVDD (l_2)	5.7218	5.4782	6.4264	5.1905	5.7502	5.6896	5.1445	5.4205	5.4857	3.7263	4.8112	6.8408	5.4738
LDT-HVDD (l_1)	5.4408	5.1584	5.9849	4.6617	5.3120	5.2109	4.7579	5.0996	4.8969	3.3699	4.1298	6.3392	5.0302
LDT-HVDD (l_2)	5.4316	5.1911	6.0757	4.6658	5.3083	5.2080	4.7786	5.1505	4.8918	3.4513	4.1295	6.6609	5.0786

A spatial self-similarity is observed, however, to exist between multiresolution subbands. This self-similarity has been exploited and taken advantage of in the set partitioning in hierarchical trees (SPIHT) coding [54]. This coding reduces some correlation in the interpolation error and leads to better comparison results [7]. Based on this coding technique, the comparative SPIHT coding of 3-level and 5-level lossless image compression are therefore given in Table 5.5 and 5.6 for 3-level and 5-level decomposition, respectively.

Table 5.5 Comparative SPIHT coding (bits/pixel) of 3-level lossless image decomposition.

Comparative SPIHT coding (bits/pixel) of 3-level image decomposition obtained with different image transformations													
Methods/Images	Airplane	Camera	Barbara	Face	Lena	Peppers	Milkdrop	Tiffany	Zelda	CT	X-rays	Fingerprint	Average (bpp)
S+P	4.9506	4.9772	5.1926	4.4993	4.7862	4.8226	4.4913	4.8249	4.6050	2.8226	3.3891	5.3673	4.5607
Modified S+P (l_1)	5.1369	5.0890	5.2540	4.5941	4.8501	4.9617	4.6405	4.9247	4.8229	2.8755	3.6585	5.6545	4.7052
Modified S+P (l_2)	5.1709	5.0982	5.4047	4.6116	4.9082	5.0049	4.6079	4.9756	4.8737	2.8854	3.7088	5.6424	4.7410
2D-HVDD (l_1)	5.0170	4.9455	5.2038	4.4798	4.8164	4.9989	4.7079	4.8323	4.6617	2.8122	3.5187	5.4176	4.6177
2D-HVDD (l_2)	5.0576	4.9581	5.2138	4.4974	4.8330	5.0541	4.7347	4.8351	4.6668	2.9094	3.5342	5.7208	4.6679
OS-HVD (l_1)	5.3552	5.2121	6.0433	4.9349	5.4939	5.4984	5.1090	5.3098	5.3614	2.9248	4.0699	5.5508	5.0720
OS-HVD (l_2)	5.4788	5.3483	6.1299	5.0103	5.5809	5.5613	5.0852	5.3098	5.3632	3.0910	4.1483	7.2813	5.2824
OS-HVVD (l_1)	5.3260	5.2490	5.9453	4.9601	5.3775	5.3882	4.9179	5.2667	5.2675	2.9272	4.0139	5.6620	5.0251
OS-HVVD (l_2)	5.4187	5.3392	5.9950	5.0054	5.4698	5.4354	4.9455	5.2687	5.2705	3.0891	4.0630	7.1910	5.2076
OS-HVDD (l_1)	5.5803	5.3492	6.1683	5.095	5.6334	5.7073	5.2328	5.4310	5.5121	3.0445	4.2258	5.7090	5.2241
OS-HVDD (l_2)	5.6998	5.5077	6.2507	5.1815	5.7209	5.7537	5.2733	5.4315	5.5148	3.2126	4.3137	7.3063	5.4305
LDT-HVDD (l_1)	5.2640	5.1165	5.7155	4.681	5.1513	5.1760	4.8405	5.0837	4.9286	2.8978	3.7500	6.0005	4.8838
LDT-HVDD (l_2)	5.2757	5.1237	5.8025	4.6878	5.1630	5.1841	4.8643	5.1105	4.9265	2.9456	3.7524	6.8235	4.9716

Table 5.6 Comparative SPIHT coding (bits/pixel) of 5-level lossless image decomposition.

Comparative SPIHT coding (bits/pixel) of 5-level image decomposition obtained with different image transformations													
Methods/Images	Airplane	Camera	Barbara	Face	Lena	Peppers	Milkdrop	Tiffany	Zelda	CT	X-rays	Fingerprint	Average (bpp)
S+P	4.9060	4.9304	5.1487	4.4471	4.7419	4.7877	4.4334	4.7820	4.5450	2.7409	3.3092	5.3655	4.5115
Modified S+P (l_1)	5.1021	5.0504	5.2167	4.5510	4.8118	4.9355	4.5913	4.8916	4.7738	2.7993	3.5835	5.6564	4.6636
Modified S+P (l_2)	5.1371	5.0602	5.3663	4.5735	4.8742	4.9775	4.5582	4.9392	4.8248	2.8099	3.6135	5.6401	4.6979
2D-HVDD (l_1)	4.9784	4.9042	5.1728	4.4371	4.7783	4.9684	4.6588	4.7936	4.6102	2.7581	3.4516	5.4141	4.5771
2D-HVDD (l_2)	5.0204	4.9128	5.1822	4.4541	4.7939	5.0284	4.6858	4.7960	4.6152	2.8549	3.4651	5.7187	4.6273
OS-HVD (l_1)	5.3245	5.1783	6.0206	4.9013	5.4721	5.4841	4.9808	5.2789	5.3233	2.8619	4.0093	5.5460	5.0318
OS-HVD (l_2)	5.4496	5.3174	6.1070	4.9781	5.5599	5.5467	5.0483	5.2793	5.3255	3.0282	4.0960	7.2814	5.2515
OS-HVVD (l_1)	5.2976	5.2197	5.9230	4.9298	5.3575	5.3751	4.8551	5.2385	5.2294	2.8504	3.9570	5.6572	4.9909
OS-HVVD (l_2)	5.3917	5.3116	5.9727	4.9752	5.4501	5.4221	4.9098	5.2411	5.2321	3.0123	4.0119	7.1911	5.1768
OS-HVDD (l_1)	5.5556	5.3205	6.1486	5.0682	5.6170	5.6980	5.2008	5.4037	5.4780	2.9926	4.1725	5.7042	5.1966
OS-HVDD (l_2)	5.6762	5.4829	6.2310	5.1552	5.7047	5.7443	5.2420	5.4043	5.4806	3.1607	4.2680	7.3063	5.4047
LDT-HVDD (l_1)	5.2357	5.0835	5.6880	4.6422	5.1230	5.1531	4.8006	5.0520	4.8876	2.8389	3.6847	5.9995	4.8491
LDT-HVDD (l_2)	5.2477	5.0991	5.7781	4.6462	5.1418	5.1599	4.8205	5.0852	4.8842	2.8894	3.6848	6.8351	4.9393

The lossless compression results shown in Table 5.3 – Table 5.6 indicates that the S+P-transform is superior to the other techniques. Nevertheless, the 2D-HVDD method gives improved results in comparison to the modified S+P-transform, the optimum scalar transform and the LDT method. Furthermore, it is seen that lossless compression of using SPIHT coding improves the average bits/pixel over that of using Huffman coding. The SPIHT coding not only reduces the statistical dependency between detail subbands but also provides embedded codes supporting progressive transmission. In progressive transmission, the bits receiving process can be halted at any time to reconstruct the image. Nowadays, this feature is important, especially for viewing images on the Internet.

Lossy Image Compression Results

For the experiment of lossy image compression, the transformed image coefficients based on the proposed techniques are encoded by the SPIHT coding. The reconstructed images at specified bits per pixel compression resulting from the proposed methods are compared with those of the wavelet transform and the LDT-HVDD method in which the following criterion is employed. The power of signal to noise ratio (PSNR (dB)) criterion is defined by

$$PSNR\ (dB)\ = 10\log_{10}\left(\frac{255^2}{MSE}\right) \tag{6.1}$$

where $MSE = \left(\frac{\sum_{m=1}^{M}\sum_{n=1}^{N}\left(p(m,n)-\hat{p}(m,n)\right)^2}{MN}\right)$ for which $p(m,n)$ and $\hat{p}(m,n)$ are original and reconstructed image pixel of size M x N image, respectively. This expression is employed to measure the fidelity of the reconstructed images compared with the original version. The measurement provided by this criterion, however, may not correspond to the human perception. The percentage of normalized absolute error (PNE_1(%)) and mean squared error (PNE_2(%)) are additional criteria to measure the goodness of compressed image reconstruction as designated by

$$PNE_1\ (\%) = 100\times\left(\frac{\sum_{m=1}^{M}\sum_{n=1}^{N}\left|p(m,n)-\hat{p}(m,n)\right|}{\sum_{m=1}^{M}\sum_{n=1}^{N}\left|p(m,n)\right|}\right) \tag{6.2}$$

and

$$PNE_2\ (\%) = 100\times\left(\sqrt{\frac{\sum_{m=1}^{M}\sum_{n=1}^{N}\left|p(m,n)-\hat{p}(m,n)\right|^2}{\sum_{m=1}^{M}\sum_{n=1}^{N}\left|p(m,n)\right|^2}}\right). \tag{6.3}$$

Furthermore, it is interesting to investigate in the Fourier characteristic of the reconstructed images, especially the Fourier phase. The importance of image phase information has been studied in [79][80][81] and it was found that much of the important information is contained in the Fourier transform's phase component [80]. Additionally, a similar conclusion

was found in [82]. Namely, for an equivalent distortion, phase encoding of the discrete Fourier transform of random sequences used 1.37 more bits than magnitude encoding. These studies imply that phase information plays an important role in frequency synthesis. The magnitude and phase characteristics of the reconstructed images therefore has been examined and compared with those of the original version in term of the signal-to-noise ratio (dB) evaluation as defined by

$$SNR(dB) = 10\log_{10}\left(\frac{\sigma_s^2}{\sigma_d^2}\right) \tag{6.4}$$

where σ_s^2 is the average squared value of signal and σ_d^2 is the mean squared error.

Additionally, due to the fact that human perception tends to be more sensitive to image's edges distortion than smooth area distortion, the edge preservation of the reconstructed images resulted by different tested schemes was investigated. This process is accomplished by applying the edge detection operator to the original and the reconstructed images. The result of the reconstructed image is then compared with that of the original version using the SNR measurement. The well-known Sobel operator is employed [83] for image edge detector in this research. In the conduct of lossy image compression, the four widely used images, *Airplane, Lena, Barbara, and Peppers* are employed in three-level image decomposition. Table 5.7 illustrates the numerical results of the reconstructed Airplane image obtained from tested techniques at 0.5, 1.0, and 1.5 bits/pixel of compression while Figure 5.1 and 5.2 represent the reconstructed Lena images and their Sobel gradients at 1.0 bits/pixel, respectively. Similarly, the lossy experiment is also performed on Lena, Barbara, and Peppers images. The comparative numerical results of the reconstructed Barbara, Lena, and Peppers images obtained from different tested techniques at 0.5, 1.0, and 1.5 bits/pixel of compression are shown in Table 5.8, 5.9, and 5.10, respectively. The reconstructed Barbara image and the Sobel gradients of the reconstructed Barbara images at 1.0 bit/pixel are respectively shown in Figure 5.3 and 5.4. For Lena image, the reconstructed images at 1.5 bits/pixel are depicted in Figure 5.5 whereas their associated Sobel gradients are shown in Figure 5.6. Additionally, Figure 5.7 and 5.8 successively demonstrate the reconstructed Peppers images at 1.5 bits/pixel and the related Sobel gradients.

Table 5.7 Comparative numerical results of reconstructed Airplane images for 3-level image decomposition obtained with different tested methods at 0.5, 1.0, and 1.5 bits/pixel of lossy compression.

Airplane	Comparative reconstructed images			Comparative the Fourier characteristic of reconstructed images		Comparative edge preservation of reconstructed images
				Magnitude component	Phase component	
0.5 bpp	**PSNR(dB)**	**PNE_1(%)**	**PNE_2(%)**	**SNR(dB)**	**SNR(dB)**	**SNR(dB)**
DWT	26.5305	4.7958	6.5151	25.7259	-1.0335	10.4416
2D-HVDD(l_1)	19.4221	12.6983	14.7688	17.4365	-0.3738	8.4537
2D-HVDD(l_2)	19.4433	12.6743	14.7328	17.4653	-0.3171	8.4606
OS-HVD(l_1)	16.6505	16.4005	20.3200	15.5619	-1.3810	5.1844
OS-HVD(l_2)	16.0862	18.3536	21.6838	14.6938	-1.3564	5.3458
OS-HVVD(l_1)	16.6664	16.3655	20.2828	15.5947	-1.4441	5.1357
OS-HVVD(l_2)	16.0787	18.3489	21.7026	14.7016	-1.4016	5.2330
OS-HVDD(l_1)	16.5970	16.4825	20.4455	15.5797	-1.4824	4.9830
OS-HVDD(l_2)	16.1129	18.1608	21.6172	14.8123	-1.4572	5.1508
LDT-HVDD(l_1)	17.6547	15.3576	18.1015	15.8679	-0.9933	7.3303
LDT-HVDD(l_2)	17.6114	15.5705	18.1919	15.7913	-0.9572	7.3783
1.0 bpp	**PSNR(dB)**	**PNE_1(%)**	**PNE_2(%)**	**SNR(dB)**	**SNR(dB)**	**SNR(dB)**
DWT	30.8513	2.9647	3.9617	29.8659	1.0080	14.5545
2D-HVDD(l_1)	26.0252	5.7619	6.9054	24.3138	1.4878	12.5039
2D-HVDD(l_2)	25.9782	5.7930	6.9428	24.2723	1.6440	12.4866
OS-HVD(l_1)	20.8791	10.8390	12.4880	19.2304	0.2327	10.0951
OS-HVD(l_2)	19.8082	12.6669	14.1267	17.8639	0.2005	10.2550
OS-HVVD(l_1)	20.8179	10.8884	12.5763	19.2008	0.1974	9.9541
OS-HVVD(l_2)	19.7463	12.7471	14.2276	17.8238	0.1493	10.0974
OS-HVDD(l_1)	20.9010	10.8017	12.4565	19.2501	0.0442	10.1302
OS-HVDD(l_2)	19.9460	12.4588	13.9043	18.0128	0.0480	10.2943
LDT-HVDD(l_1)	21.6546	10.1348	11.4214	19.5714	0.8124	11.2908
LDT-HVDD(l_2)	21.5047	10.3845	11.6202	19.3754	0.8481	11.2590
1.5 bpp	**PSNR(dB)**	**PNE_1(%)**	**PNE_2(%)**	**SNR(dB)**	**SNR(dB)**	**SNR(dB)**
DWT	34.6109	1.9579	2.5698	33.6908	2.7256	18.3026
2D-HVDD(l_1)	27.4799	5.0078	5.8405	25.5049	3.1791	15.1225
2D-HVDD(l_2)	27.2765	5.1204	5.9789	25.3099	3.1762	15.0293
OS-HVD(l_1)	26.3966	5.5626	6.6164	25.1549	1.9382	14.4041
OS-HVD(l_2)	24.6226	7.1225	8.1155	22.8502	1.8689	14.1748
OS-HVVD(l_1)	26.2947	5.6231	6.6944	25.0871	1.7971	14.1903
OS-HVVD(l_2)	24.5462	7.1800	8.1872	22.7877	1.7986	13.9986
OS-HVDD(l_1)	26.4111	5.5563	6.6053	25.1833	1.5901	14.3652
OS-HVDD(l_2)	24.8898	6.9061	7.8697	23.1616	1.4982	14.3474
LDT-HVDD(l_1)	27.3626	5.0825	5.9199	25.5656	2.5464	15.5249
LDT-HVDD(l_2)	27.1566	5.2705	6.0620	25.2813	2.5177	15.5382

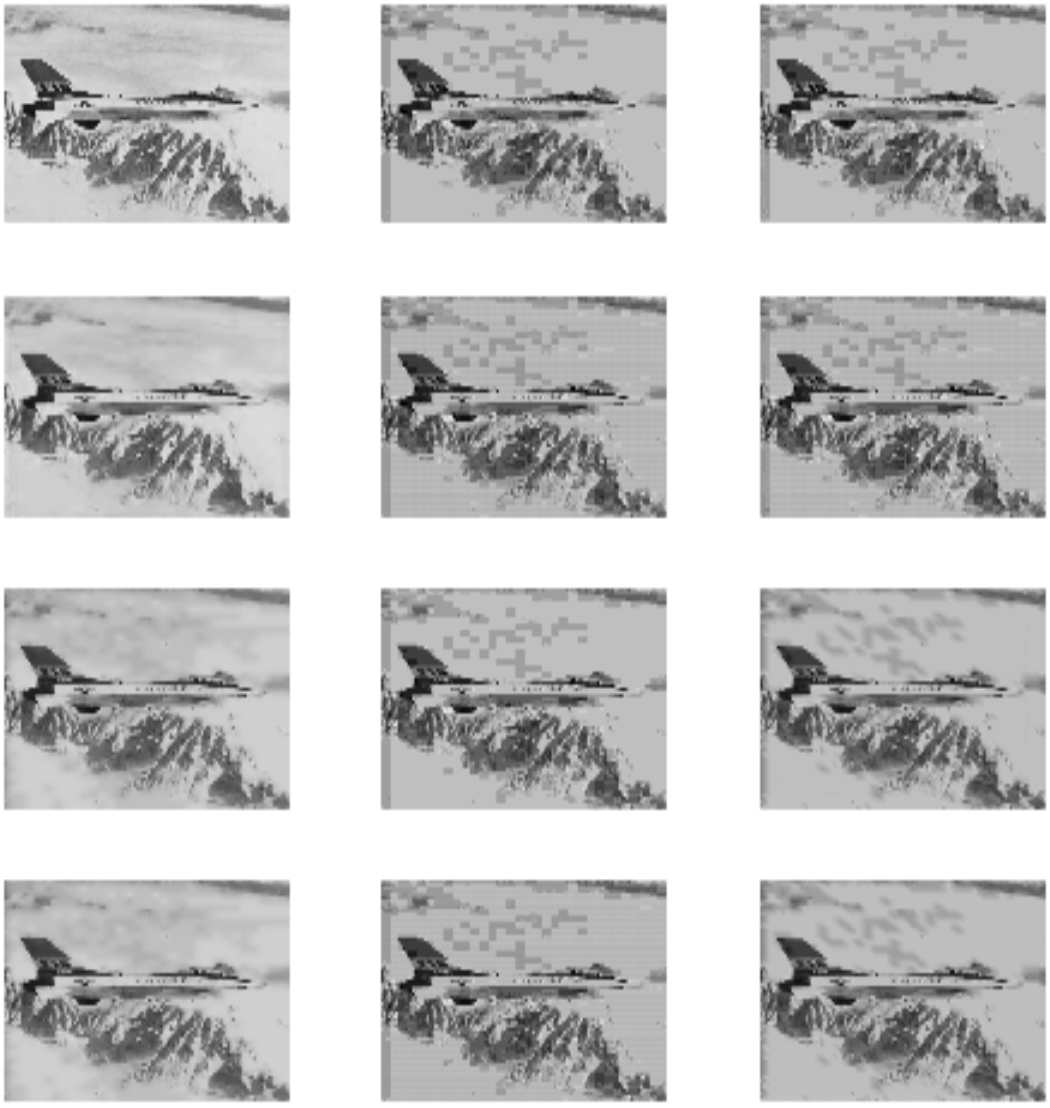

Figure 5.1 The reconstructed Airplane images at 1.0 bit/pixel of lossy compression: original image (*top row-left*), DWT(2^{nd} *row-left*), l_1 norm based 2D-HVDD method (3^{rd} *row-left*), l_2 norm based 2D-HVDD method (*bottom row-left*), l_1 norm based OS-HVD method (*top row-middle*), l_2 norm based OS-HVD method (2^{nd} *row-middle*), l_1 norm based OS-HVDD method (3^{rd} *row-middle*), l_2 norm based OS-HVDD method (*bottom row-middle*), l_1 norm based OS-HVVD method (*top row-right*), l_2 norm based OS-HVVD method (2^{nd} *row-right*), l_1 norm based LDT-HVDD method (3^{rd} *row-right*), and l_2 norm based LDT-HVDD method (*bottom row-right*).

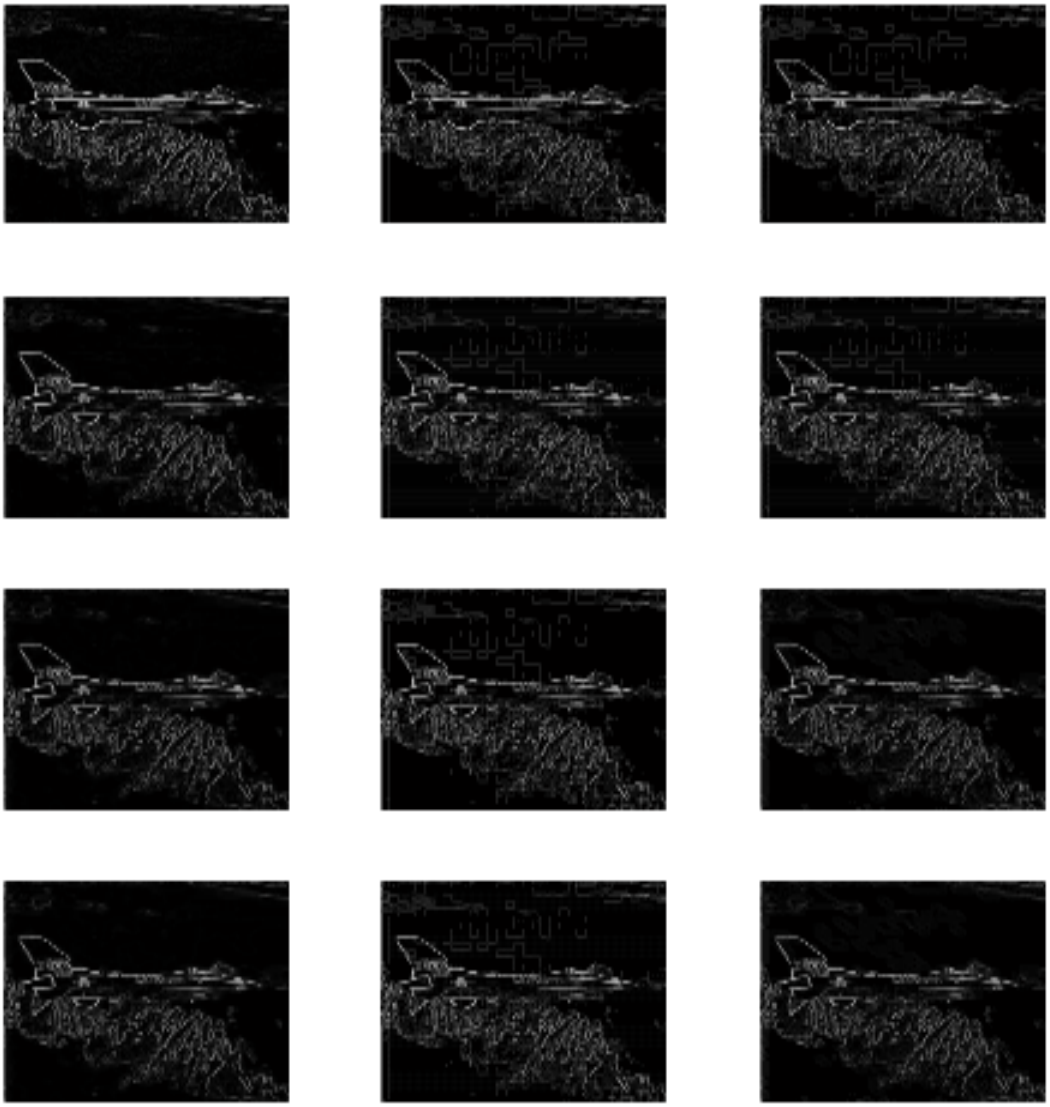

Figure 5.2 Sobel gradients of the reconstructed Airplane images at 1.0 bit/pixel of lossy compression: original image (*top row-left*), DWT(2nd *row-left*), l_1 norm based 2D-HVDD method (3rd *row-left*), l_2 norm based 2D-HVDD method (*bottom row-left*), l_1 norm based OS-HVD method (*top row-middle*), l_2 norm based OS-HVD method (2nd *row-middle*), l_1 norm based OS-HVDD method (3rd *row-middle*), l_2 norm based OS-HVDD method (*bottom row-middle*), l_1 norm based OS-HVVD method (*top row-right*), l_2 norm based OS-HVVD method (2nd *row-right*), l_1 norm based LDT-HVDD method (3rd *row-right*), and l_2 norm based LDT-HVDD method (*bottom row-right*).

Table 5.8 Comparative numerical results of reconstructed Barbara images for 3-level image decomposition obtained with different tested methods at 0.5, 1.0, and 1.5 bits/pixel of lossy compression.

Barbara	Comparative reconstructed images			Comparative the Fourier characteristic of reconstructed images		Comparative edge preservation of reconstructed images
				Magnitude component	Phase component	
0.5 bpp	**PSNR(dB)**	**PNE_1(%)**	**PNE_2(%)**	**SNR(dB)**	**SNR(dB)**	**SNR(dB)**
DWT	25.9097	8.1188	10.0703	21.6646	-1.7094	8.1718
2D-HVDD(l_1)	21.6377	14.5856	16.4680	16.7818	-1.2838	8.0822
2D-HVDD(l_2)	21.7049	14.4429	16.3411	16.9410	-1.2255	8.2687
OS-HVD(l_1)	16.3317	26.9926	30.3348	11.8575	-2.2187	4.7533
OS-HVD(l_2)	15.7554	29.2789	32.4158	10.9820	-2.2009	5.0700
OS-HVVD(l_1)	16.4046	26.7778	30.0813	11.9091	-2.1947	4.8906
OS-HVVD(l_2)	15.8405	28.9958	32.0998	11.0509	-2.1665	5.1762
OS-HVDD(l_1)	16.3184	26.9962	30.3811	11.8550	-2.2065	4.5273
OS-HVDD(l_2)	15.8418	28.9083	32.0947	11.1273	-2.2177	4.8062
LDT-HVDD(l_1)	17.3598	24.6164	26.9485	12.1909	-1.9775	6.2552
LDT-HVDD(l_2)	17.2420	24.9707	27.3163	11.9976	-1.9738	5.7581
1.0 bpp	**PSNR(dB)**	**PNE_1(%)**	**PNE_2(%)**	**SNR(dB)**	**SNR(dB)**	**SNR(dB)**
DWT	29.9999	5.0677	6.2882	25.7097	-0.7936	12.1183
2D-HVDD(l_1)	26.7218	8.0436	9.1714	22.0033	-0.3097	12.2402
2D-HVDD(l_2)	26.8699	7.8891	9.0164	22.3410	-0.3503	12.5841
OS-HVD(l_1)	20.3225	16.5971	19.1602	16.3118	-1.5759	7.6990
OS-HVD(l_2)	19.5381	18.6378	20.9709	15.0236	-1.5574	7.9248
OS-HVVD(l_1)	20.5267	16.2417	18.7150	16.3796	-1.4891	7.9882
OS-HVVD(l_2)	19.7663	18.2040	20.4272	15.1883	-1.4961	8.1474
OS-HVDD(l_1)	20.3999	16.5150	18.9903	16.3568	-1.5928	7.7043
OS-HVDD(l_2)	19.7483	18.2304	20.4696	15.2941	-1.6048	7.9035
LDT-HVDD(l_1)	22.0166	14.2560	15.7650	16.9163	-1.2180	9.6751
LDT-HVDD(l_2)	21.7871	14.6833	16.1870	16.6041	-1.2447	9.1285
1.5 bpp	**PSNR(dB)**	**PNE_1(%)**	**PNE_2(%)**	**SNR(dB)**	**SNR(dB)**	**SNR(dB)**
DWT	33.7499	3.3448	4.0835	29.6822	-0.0643	16.1927
2D-HVDD(l_1)	30.2275	5.2285	6.1256	25.9353	0.6968	14.3231
2D-HVDD(l_2)	30.1819	5.2384	6.1579	26.1883	0.6889	14.7177
OS-HVD(l_1)	24.0671	10.6450	12.4501	20.4624	-0.9044	10.3959
OS-HVD(l_2)	23.0058	12.3452	14.0681	18.7319	-0.9792	10.3818
OS-HVVD(l_1)	24.6450	10.0297	11.6486	20.9300	-0.7149	11.2387
OS-HVVD(l_2)	23.5731	11.6156	13.1786	19.2031	-0.7471	11.3071
OS-HVDD(l_1)	21.2895	15.0882	17.1416	16.7917	-1.0830	9.4035
OS-HVDD(l_2)	20.6663	16.5987	18.4166	15.8748	-1.0882	9.5573
LDT-HVDD(l_1)	26.8532	8.0201	9.0337	22.0012	-0.4352	13.3394
LDT-HVDD(l_2)	26.5070	8.3974	9.4010	21.5880	-0.5423	12.7334

Table 5.9 Comparative numerical results of reconstructed Lena images for 3-level image decomposition obtained with different tested methods at 0.5, 1.0, and 1.5 bits/pixel of lossy compression.

Lena	Comparative reconstructed images			Comparative the Fourier characteristic of reconstructed images		Comparative edge preservation of reconstructed images
				Magnitude component	Phase component	
0.5 bpp	**PSNR(dB)**	**PNE_1(%)**	**PNE_2(%)**	**SNR(dB)**	**SNR(dB)**	**SNR(dB)**
DWT	27.7782	6.3004	7.8616	24.0263	-0.8746	9.9721
2D-HVDD(l_1)	22.2448	13.4871	14.8656	17.2964	0.0492	9.0359
2D-HVDD(l_2)	22.2377	13.5064	14.8777	17.3169	0.0295	9.2004
OS-HVD(l_1)	16.7823	24.8849	27.8809	12.3760	-1.4537	4.1614
OS-HVD(l_2)	16.2866	26.9612	29.5182	11.6615	-1.4301	4.4156
OS-HVVD(l_1)	16.7265	25.0336	28.0607	12.3453	-1.3394	4.1454
OS-HVVD(l_2)	16.2466	27.0730	29.6548	11.6369	-1.3516	4.3752
OS-HVDD(l_1)	16.8295	24.7235	27.7300	12.4478	-1.4548	4.1103
OS-HVDD(l_2)	16.3705	26.6210	29.2347	11.7707	-1.4414	4.3250
LDT-HVDD(l_1)	21.4446	14.5107	16.3001	16.8431	-0.5861	8.3434
LDT-HVDD(l_2)	21.3729	14.7174	16.4352	16.7187	-0.5948	8.2521
1.0 bpp	**PSNR(dB)**	**PNE_1(%)**	**PNE_2(%)**	**SNR(dB)**	**SNR(dB)**	**SNR(dB)**
DWT	33.2254	3.3356	4.1991	29.8728	1.0318	15.0341
2D-HVDD(l_1)	27.4894	7.1927	8.1274	22.7149	2.0121	12.8523
2D-HVDD(l_2)	27.4305	7.2512	8.1827	22.7331	1.8882	13.1440
OS-HVD(l_1)	21.1127	15.0176	16.9351	16.9172	-0.0810	7.6687
OS-HVD(l_2)	20.3492	16.8738	18.4911	15.8548	-0.1535	7.8845
OS-HVVD(l_1)	21.0800	15.0689	16.9990	16.8736	-0.0706	7.7821
OS-HVVD(l_2)	20.3598	16.8253	18.4686	15.8465	-0.0798	7.9265
OS-HVDD(l_1)	21.2357	14.8150	16.6969	17.0536	-0.1823	7.7530
OS-HVDD(l_2)	20.5473	16.4770	18.0741	16.0654	-0.3021	7.9405
LDT-HVDD(l_1)	26.7523	7.7587	8.8472	22.2925	1.1170	12.5214
LDT-HVDD(l_2)	26.5702	8.0171	9.0347	22.0105	1.0998	12.4142
1.5 bpp	**PSNR(dB)**	**PNE_1(%)**	**PNE_2(%)**	**SNR(dB)**	**SNR(dB)**	**SNR(dB)**
DWT	35.4812	2.6038	3.2386	31.6391	2.5909	16.9363
2D-HVDD(l_1)	32.4090	3.9967	4.6129	27.9657	3.6406	16.4918
2D-HVDD(l_2)	32.1847	4.0818	4.7335	27.8777	3.4581	16.7365
OS-HVD(l_1)	25.8842	8.4551	9.7771	21.8562	1.2215	11.8924
OS-HVD(l_2)	24.8071	9.9019	11.0679	20.4204	1.0709	11.8549
OS-HVVD(l_1)	25.8394	8.4822	9.8278	21.8350	1.2255	11.7759
OS-HVVD(l_2)	24.7603	9.9287	11.1277	20.4013	1.2216	11.6632
OS-HVDD(l_1)	26.1105	8.2781	9.5257	22.0512	1.0572	12.0335
OS-HVDD(l_2)	24.9870	9.7209	10.8411	20.5894	0.9425	11.9771
LDT-HVDD(l_1)	27.7548	7.0035	7.8829	23.0083	2.4313	14.3375
LDT-HVDD(l_2)	27.5683	7.2297	8.0540	22.7423	2.3332	14.2782

Table 5.10 Comparative numerical results of reconstructed Peppers images for 3-level image decomposition obtained with different tested methods at 0.5, 1.0, and 1.5 bits/pixel of lossy compression.

Peppers	Comparative reconstructed images			Comparative the Fourier characteristic of reconstructed images		Comparative edge preservation of reconstructed images
				Magnitude component	Phase component	
0.5 bpp	**PSNR(dB)**	**PNE_1(%)**	**PNE_2(%)**	**SNR(dB)**	**SNR(dB)**	**SNR(dB)**
DWT	28.3801	6.0507	7.5230	24.4547	-0.6872	10.7124
2D-HVDD(l_1)	22.1805	13.8898	15.3591	17.0811	-0.4089	8.7580
2D-HVDD(l_2)	22.0045	14.1200	15.6736	16.9184	-0.3899	8.7318
OS-HVD(l_1)	16.3969	27.8864	29.8917	11.6922	-1.5106	4.7308
OS-HVD(l_2)	15.8787	29.9658	31.7292	10.9798	-1.5462	5.0516
OS-HVVD(l_1)	16.3605	28.0085	30.0170	11.6755	-1.4916	4.7061
OS-HVVD(l_2)	15.8371	30.0964	31.8816	10.9368	-1.4744	5.0275
OS-HVDD(l_1)	16.4069	27.8851	29.8574	11.7266	-1.7419	4.5961
OS-HVDD(l_2)	15.9479	29.7453	31.4775	11.0803	-1.7615	4.9019
LDT-HVDD(l_1)	21.8058	14.4296	16.0363	16.9609	-0.5623	8.8985
LDT-HVDD(l_2)	21.7475	14.5739	16.1443	16.8901	-0.6242	8.9049
1.0 bpp	**PSNR(dB)**	**PNE_1(%)**	**PNE_2(%)**	**SNR(dB)**	**SNR(dB)**	**SNR(dB)**
DWT	33.8335	3.2703	4.0153	30.2520	0.9377	16.0987
2D-HVDD(l_1)	27.2815	7.5502	8.5373	22.3806	1.4420	12.8762
2D-HVDD(l_2)	26.7769	7.9331	9.0480	21.8957	1.3700	12.6972
OS-HVD(l_1)	21.3136	15.1441	16.9712	16.8808	-0.3762	8.5698
OS-HVD(l_2)	20.5830	16.8449	18.4605	15.8432	-0.3028	8.7635
OS-HVVD(l_1)	22.5534	12.6632	14.7138	18.8836	-0.2969	8.6212
OS-HVVD(l_2)	22.7746	12.7879	14.3438	18.7300	-0.2428	9.3142
OS-HVDD(l_1)	21.4056	15.0188	16.7925	16.9985	-0.5395	8.5589
OS-HVDD(l_2)	20.8002	16.4558	18.0045	16.1052	-0.5138	8.8543
LDT-HVDD(l_1)	27.2053	7.7007	8.6125	22.4579	1.0125	13.4527
LDT-HVDD(l_2)	27.0436	7.8686	8.7743	22.2918	0.9554	13.4370
1.5 bpp	**PSNR(dB)**	**PNE_1(%)**	**PNE_2(%)**	**SNR(dB)**	**SNR(dB)**	**SNR(dB)**
DWT	36.4498	2.4518	2.9710	32.8688	2.1965	18.5082
2D-HVDD(l_1)	31.9674	4.3281	4.9776	27.4109	2.6995	16.5586
2D-HVDD(l_2)	30.8400	4.7985	5.6675	26.4257	2.4487	16.1189
OS-HVD(l_1)	26.2201	8.5540	9.6469	22.1130	0.9231	12.7566
OS-HVD(l_2)	25.2500	9.7915	10.7868	20.7132	0.9307	12.7167
OS-HVVD(l_1)	26.0922	8.6637	9.7901	22.0094	1.0291	12.5687
OS-HVVD(l_2)	25.1487	9.9038	10.9133	20.6181	1.0608	12.5672
OS-HVDD(l_1)	26.3091	8.4940	9.5486	22.1771	0.6825	12.8014
OS-HVDD(l_2)	25.3515	9.7025	10.6616	20.8258	0.6447	12.8130
LDT-HVDD(l_1)	30.0469	5.2910	6.2094	25.6241	2.2343	15.0238
LDT-HVDD(l_2)	29.4132	5.7310	6.6794	24.8840	2.1522	14.9202

Figure 5.3 The reconstructed Barbara image at 1.0 bit/pixel of lossy compression: original image (*top row-left*), DWT(2nd *row-left*), l_1 norm based 2D-HVDD method (3rd *row-left*), l_2 norm based 2D-HVDD method (*bottom row-left*), l_1 norm based OS-HVD method (*top row-middle*), l_2 norm based OS-HVD method (2nd *row-middle*), l_1 norm based OS-HVDD method (3rd *row-middle*), l_2 norm based OS-HVDD method (*bottom row-middle*), l_1 norm based OS-HVVD method (*top row-right*), l_2 norm based OS-HVVD method (2nd *row-right*), l_1 norm based LDT-HVDD method (3rd *row-right*), and l_2 norm based LDT-HVDD method (*bottom row-right*).

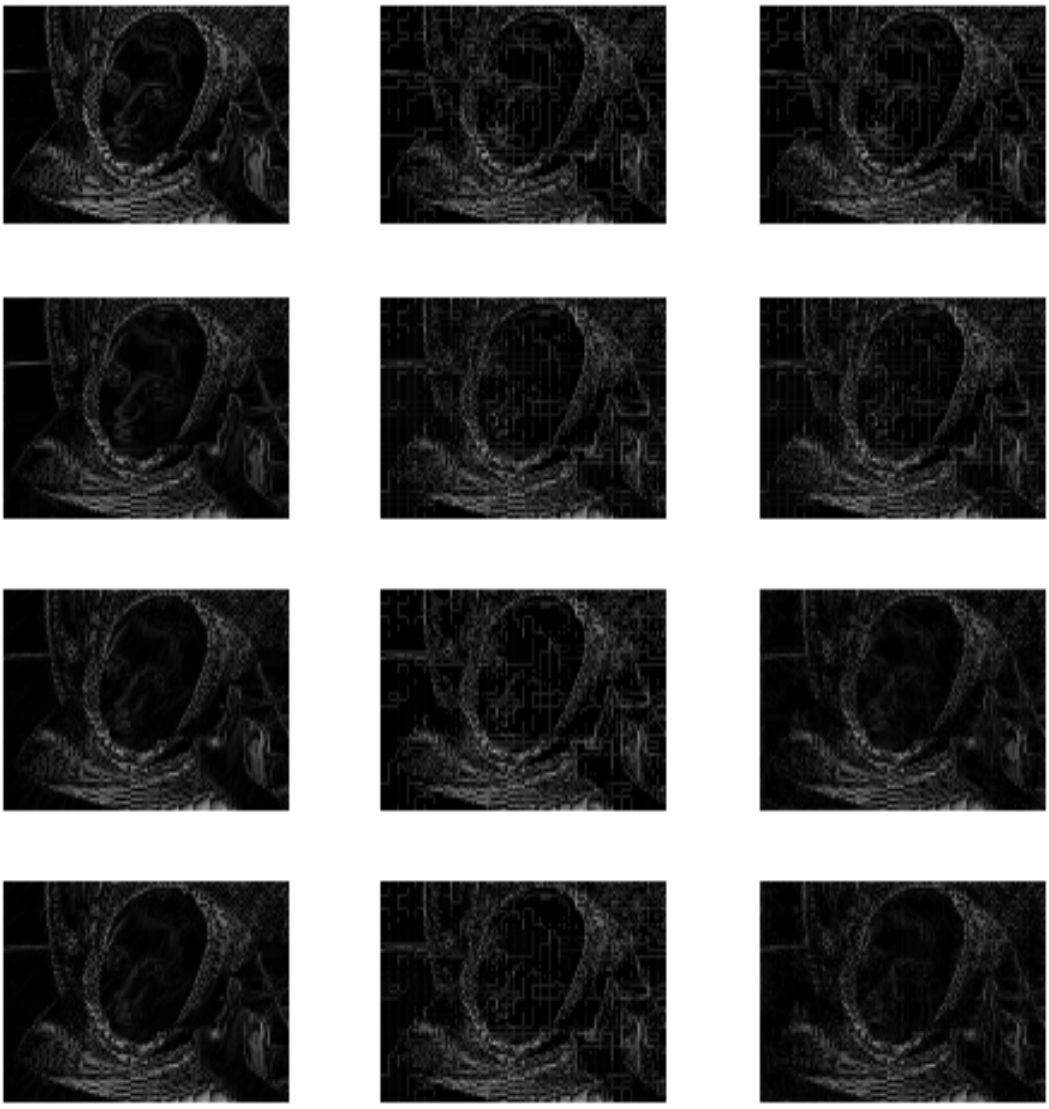

Figure 5.4 Sobel gradients of the reconstructed Barbara image at 1.0 bit/pixel of lossy compression: original image (*top row-left*), DWT(2^{nd} *row-left*), l_1 norm based 2D-HVDD method (3^{rd} *row-left*), l_2 norm based 2D-HVDD method (*bottom row-left*), l_1 norm based OS-HVD method (*top row-middle*), l_2 norm based OS-HVD method (2^{nd} *row-middle*), l_1 norm based OS-HVDD method (3^{rd} *row-middle*), l_2 norm based OS-HVDD method (*bottom row-middle*), l_1 norm based OS-HVVD method (*top row-right*), l_2 norm based OS-HVVD method (2^{nd} *row-right*), l_1 norm based LDT-HVDD method (3^{rd} *row-right*), and l_2 norm based LDT-HVDD method (*bottom row-right*).

Figure 5.5 The reconstructed Lena images at 1.5 bits/pixel of lossy compression: original image (*top row-left*), DWT(2nd *row-left*), l_1 norm based 2D-HVDD method (3rd *row-left*), l_2 norm based 2D-HVDD method (*bottom row-left*), l_1 norm based OS-HVD method (*top row-middle*), l_2 norm based OS-HVD method (2nd *row-middle*), l_1 norm based OS-HVDD method (3rd *row-middle*), l_2 norm based OS-HVDD method (*bottom row-middle*), l_1 norm based OS-HVVD method (*top row-right*), l_2 norm based OS-HVVD method (2nd *row-right*), l_1 norm based LDT-HVDD method (3rd *row-right*), and l_2 norm based LDT-HVDD method (*bottom row-right*).

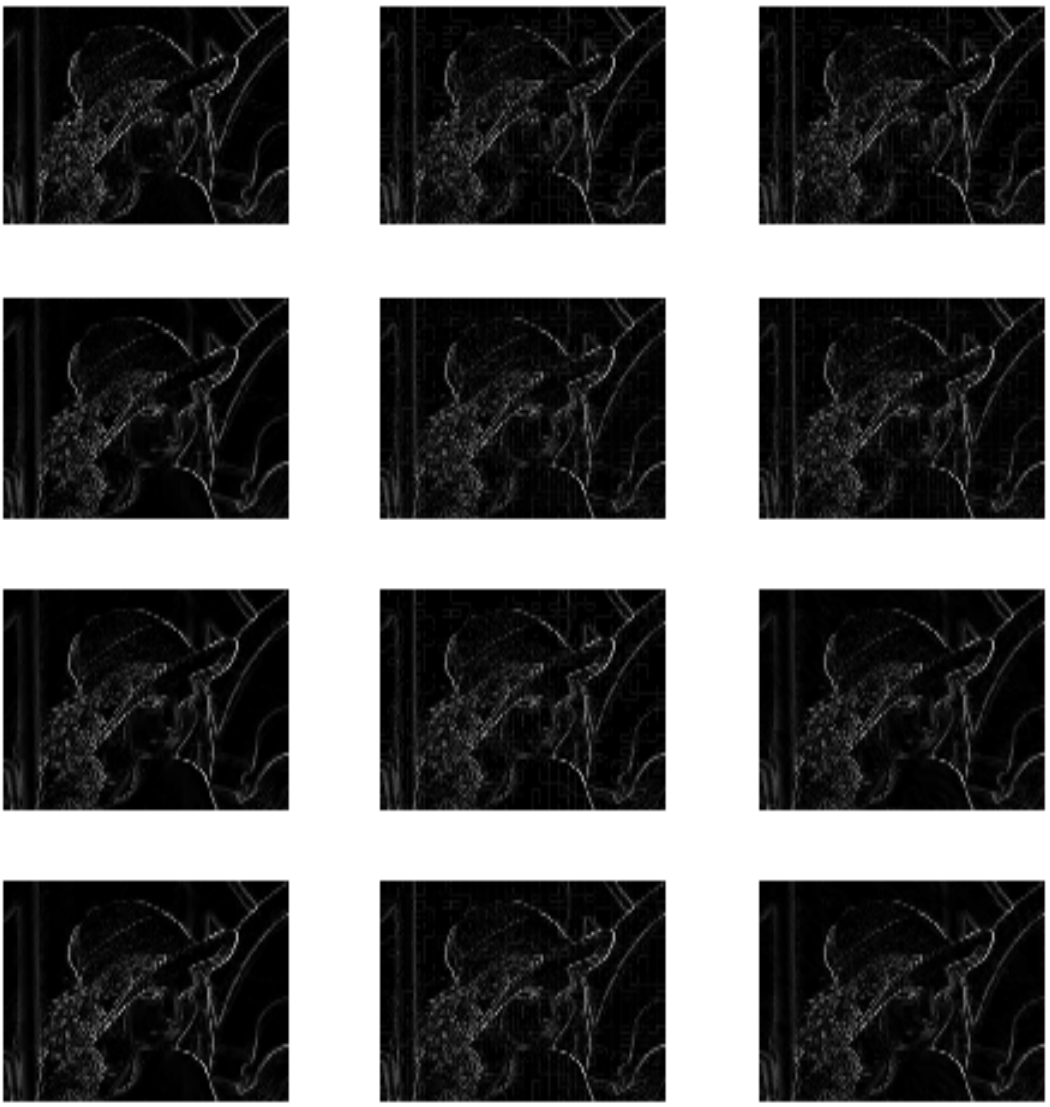

Figure 5.6 Sobel gradients of the reconstructed Lena image at 1.5 bits/pixel of lossy compression: original image (*top row-left*), DWT(2^{nd} *row-left*), l_1 norm based 2D-HVDD method (3^{rd} *row-left*), l_2 norm based 2D-HVDD method (*bottom row-left*), l_1 norm based OS-HVD method (*top row-middle*), l_2 norm based OS-HVD method (2^{nd} *row-middle*), l_1 norm based OS-HVDD method (3^{rd} *row-middle*), l_2 norm based OS-HVDD method (*bottom row-middle*), l_1 norm based OS-HVVD method (*top row-right*), l_2 norm based OS-HVVD method (2^{nd} *row-right*), l_1 norm based LDT-HVDD method (3^{rd} *row-right*), and l_2 norm based LDT-HVDD method (*bottom row-right*).

Figure 5.7 The reconstructed Peppers images at 1.5 bits/pixel of lossy compression: original image (*top row-left*), DWT(2nd *row-left*), l_1 norm based 2D-HVDD method (3rd *row-left*), l_2 norm based 2D-HVDD method (*bottom row-left*), l_1 norm based OS-HVD method (*top row-middle*), l_2 norm based OS-HVD method (2nd *row-middle*), l_1 norm based OS-HVDD method (3rd *row-middle*), l_2 norm based OS-HVDD method (*bottom row-middle*), l_1 norm based OS-HVVD method (*top row-right*), l_2 norm based OS-HVVD method (2nd *row-right*), l_1 norm based LDT-HVDD method (3rd *row-right*), and l_2 norm based LDT-HVDD method (*bottom row-right*).

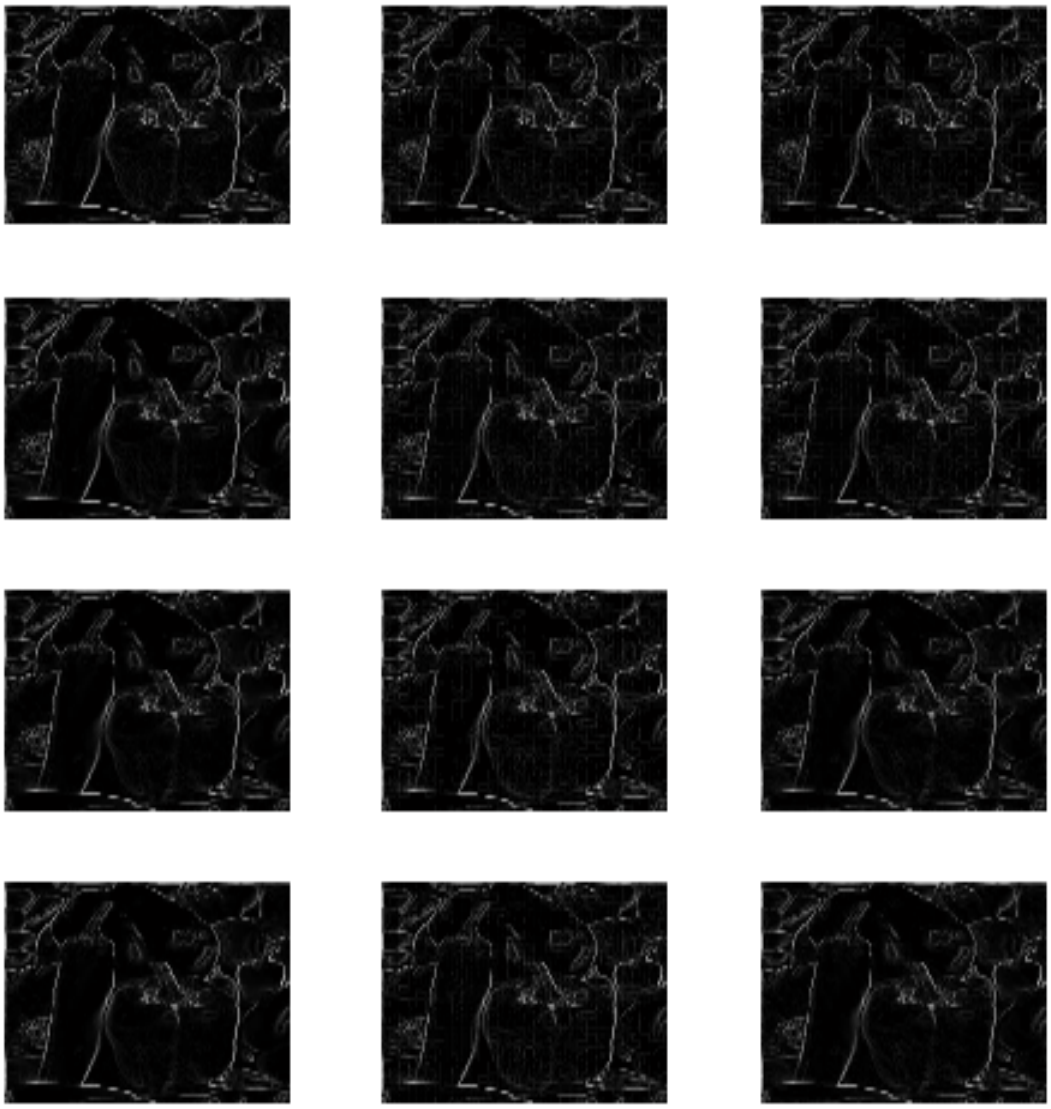

Figure 5.8 Sobel gradients of the reconstructed Peppers image at 1.5 bits/pixel of lossy compression: original image (*top row-left*), DWT(2nd *row-left*), l_1 norm based 2D-HVDD method (3rd *row-left*), l_2 norm based 2D-HVDD method (*bottom row-left*), l_1 norm based OS-HVD method (*top row-middle*), l_2 norm based OS-HVD method (2nd *row-middle*), l_1 norm based OS-HVDD method (3rd *row-middle*), l_2 norm based OS-HVDD method (*bottom row-middle*), l_1 norm based OS-HVVD method (*top row-right*), l_2 norm based OS-HVVD method (2nd *row-right*), l_1 norm based LDT-HVDD method (3rd *row-right*), and l_2 norm based LDT-HVDD method (*bottom row-right*).

The numerical results represented in Table 5.7 – 5.10 imply that the wavelet transform generally provides best results in terms of PSNR (dB), PNE_1 (%), and PNE_2 (%) measurements of the reconstructed images relative to the original version. In addition, the comparative SNR (dB) of the Fourier magnitude characteristic indicates that the wavelet transform is superior to the other techniques. On the other hand, the 2D-HVDD method always provides best SNR (dB) results when the reconstructed images are compared in term of the Fourier phase characteristic respected to the original image. As results shown for PEPPERS image at 1.5 bits/pixel in Table 5.10, the numerical PSNR(dB), PNE_1(%), and PNE_2(%) results of using between the wavelet transform and the 2D-HVDD method are relatively different; however, it is difficult to distinguish the difference of visual image quality in Figure 5.7. Contrarily, even though the numerical results of using the 2D-HVDD and the LDT-HVDD techniques are quite close, the difference between the images resulted by these methods is perceptually noticeable. This observation suggests that the comparative numerical criteria are not directly related to visual image quality. The Sobel edge operator was then employed to examine Sobel gradients of the reconstructed images. It was found that the wavelet transform preserves image edge relative to the original image better than the other methods. In this study, the results indicate that the 2D-HVDD and the LDT-HVDD methods are not significantly different while the optimum scalar transform does not perform well in general.

CHAPTER VI

CONCLUSIONS

In this research, two decorrelation techniques are proposed for the application of lossless and lossy image compression. The first method, called the optimum scalar decomposition, simply decomposes image into subbands and then approximates the decomposed subbands as a scalar multiple of the retained subband. Another decorrelation technique motivated by the linear decomposition transform (LDT) employs a two-dimensional decorrelation structure to improve the decorrelation performance over one-dimensional row-wise and column-wise processing techniques as used in the LDT. The basic concept of these methods is based on the interpolative subband decomposition. Due to the fact that every image has its own distintive characteristic, the advisability of using a fixed universal interpolation filter is questionable. The proposed techniques adapt such filters to the image being compressed. The adapted interpolation filters are optimally designed to reduce statistical dependence between the prediction errors as much as possible based on l_1 and l_2 norm criteria.

In the decorrelation performance examination, it was shown that, without the S+P-transform, the 2D-HVDD method provides better decorrelation performance than the modified S+P-transform, the optimum scalar transform and the LDT method in one-level image decomposition. However, this performance tends to be decreased (which is always be the case for other tested methods) in two-level of decomposition due to the fact that neighboring image pixels tend to be less correlated at higher levels of decomposition. As expected, the two-dimensional interpolation filter used in the 2D-HVDD method has shown the better decorrelation performance compared to one-dimensional interpolation filter employed in the LDT-HVDD technique. It also suggests that the 2D-HVDD method will provide the promising results for lossless and lossy image compression.

As shown in lossless image compression results using Huffman or SPIHT coding, even though the 2D-HVDD method inferior produced results relative to those obtained by the S+P-transform, it resulted in an improvement over other methods, especially the LDT. When the SPIHT algorithm is employed as a coding tool, the average bits/pixel improves compared to the results employing Huffman coding. This improvement indicates that the SPIHT coding has

taken advantage of the self-similarity between subbands and reduced the correlation in those subbands.

With the progressive transmission property of the SPIHT coding, this coding is then employed to encode the transformed image coefficients obtained with the wavelet transform, the 2D-HVDD and the LDT-HVDD methods for lossy image compression. It is found that the wavelet transform leads the other techniques based on the numerical PSNR(dB), PNE_1(%), and PNE_2(%) results. These measurements are mathematically tractable; unfortunately, they do not provide a very accurate indication of the perceptible fidelity of image reconstruction. Therefore, better numerical results do not always imply better visual image quality and vice versa. To measure the perceptible fidelity of the reconstructed image, an accurate model for human perception must be employed in order that the transform of the reconstructed images in the perceptual space be compared. However, the human perception is very complex and an accurate model has not yet established. This topic is one of the active area research in the present [78]. In this dissertation, the Sobel edge detector was used to investigate the edge preservation in the reconstructed images compared to the original version. The SNR (dB) result generally shows that the wavelet transform achieves the smallest error between the Sobel gradients of the reconstructed image and that of the original image.

It is interesting to note that, however, the 2D-HVDD method is superior to the other schemes for lossy image compression in term of the SNR (dB) of the Fourier phase characteristic of the reconstructed image relative to the original version. On the other hand, the SNR (dB) of the Fourier magnitude characteristics of the 2D-HVDD based reconstructed image is very inferior to the wavelet. Even though the image phase component of the Fourier transform is found to be more important than the magnitude component [79][80][81], much distortion in the magnitude component arose using the 2D-HVDD method. This resulted in inferior numerical result and perceptual image quality in comparison to that of the wavelet transform. Furthermore, the property of the SPIHT coding is to first transmit the most significant bits of the transformed coefficients. When the bit receiving process is halted, the bits not yet been received are assumed to be zero. This is akin to truncating the smallest detail coefficients which does not greatly impact the reconstructed image for an orthogonal transformation (e.g., wavelet). Contrarily, the 2D-HVDD method, the optimum scalar transform, and the LDT-HVDD technique are not orthogonal transforms. Truncating a number of the smallest non-orthogonal transform based

detail coefficients does not always result in an optimum reconstruction. Recently, the selection of a set of truncated detail coefficients for the LDT method has been extensively investigated in [84]. The results of using this developed selection technique have shown an improvement over the conventional smallest coefficients truncation. This study may be applied for the 2D-HVDD method and the optimum scalar transform for improving both numerical results and visual image quality. Based on this described reason and poor decorrelation structure, the optimum scalar transform generally does not provide the potential results in lossless and lossy image compression. However, the 2D-HVDD technique has shown an improved capability and encourages further development.

APPENDIX A

TESTED IMAGES

Figure A.1 Tested images: Airplane (*top left-row*), Barbara (*top right-row*), Camera (*middle left row*), Face (*middle right-row*), Lena (*bottom left-row*), and Milkdrop (*bottom right-row*).

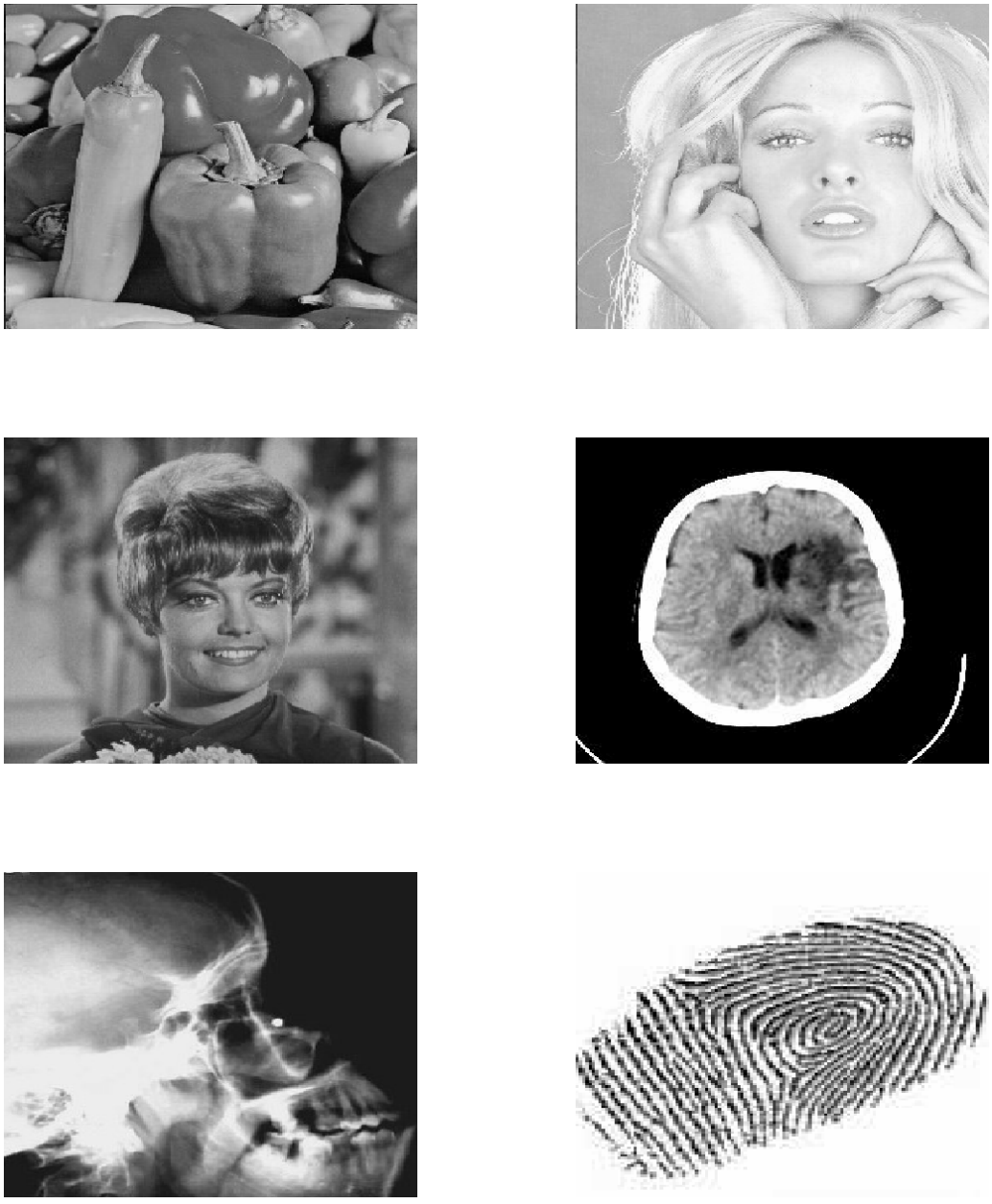

Figure A.2 Tested images: Peppers (*top left-row*), Tiffany (*top right-row*), Zelda (*middle left row*), CT (*middle right-row*), X-rays (*bottom left-row*), and Fingerprint (*bottom right-row*).

APPENDIX B

LISTS OF THE OPTIMUM PARAMETERS

Table B.1 The optimum scalars of Airplane image for 3-level and 5-level image decomposition using the l_1 and l_2 norm based modified S+P transform, the l_1 and l_2 norm based 2D-HVDD, the l_1 and l_2 norm based LDT-HVDD, the l_1 and l_2 norm based OS-HVD, the l_1 and l_2 norm based OS-HVVD, and the l_1 and l_2 norm based OS-HVDD methods.

Airplane	3-level image decomposition				5-level image decomposition			
Method								
Modified S+P(l_1)	-0.0528	0.1682	0.0530	-0.1679	-0.0527	0.1670	0.0500	-0.1724
Modified S+P(l_2)	-0.0813	0.2155	0.0663	-0.2383	-0.0817	0.2140	0.0638	-0.2427
2D-HVDD(l_1) : hv filter	0.2427	0.2996	0.2231	0.2353	0.2398	0.3062	0.2220	0.2327
: d filter	0.1947	0.2104	0.2931	0.3028	0.1921	0.2095	0.2941	0.3053
2D-HVDD(l_2) : hv filter	0.2348	0.2744	0.2487	0.2443	0.2344	0.2801	0.2478	0.2399
: d filter	0.2134	0.1931	0.2965	0.2992	0.2126	0.1934	0.2966	0.2995
LDT-HVDD(l_1) : h filter	-0.0170	0.5197	0.5058	-0.0086	-0.0157	0.5182	0.5061	-0.0086
: v filter	-0.0158	0.5075	0.5282	-0.0198	-0.0150	0.5059	0.5284	-0.0192
: d filter	-0.0109	0.5079	0.5089	-0.0058	-0.0099	0.5051	0.5100	-0.0052
LDT-HVDD(l_2) : h filter	-0.0026	0.5102	0.4956	-0.0039	-0.0002	0.5055	0.4967	-0.0029
: v filter	-0.0109	0.4935	0.5287	-0.0124	-0.0067	0.4868	0.5276	-0.0088
: d filter	0.0003	0.4974	0.4884	0.0124	0.0056	0.4890	0.4875	0.0164
	Level-1	Level-2	Level-3	Level-1	Level-2	Level-3	Level-4	Level-5
OS-HVD(l_1) : alpha	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	0.9955	1.0000
: beta	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
OS-HVD(l_2) : alpha	0.9970	0.9941	0.9901	0.9970	0.9941	0.9901	0.9871	1.0302
: beta	0.9954	0.9892	0.9651	0.9954	0.9892	0.9651	0.9805	1.0064
OS-HVVD(l_1) : alpha	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	0.9955	1.0000
: beta	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
: gamma	0.5857	0.4182	0.2632	0.5857	0.4182	0.2632	0.3333	0.7786
OS-HVVD(l_2) : alpha	0.9970	0.9941	0.9901	0.9970	0.9941	0.9901	0.9871	1.0302
: beta	0.9954	0.9892	0.9651	0.9954	0.9892	0.9651	0.9805	1.0064
: gamma	0.6699	0.5330	0.4044	0.6699	0.5330	0.4044	0.5337	0.7871
OS-HVDD(l_1) : alpha	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0046
: beta	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
: gamma	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0152
OS-HVDD(l_2) : alpha	0.9969	0.9948	0.9870	0.9969	0.9948	0.9870	0.9849	1.0496
: beta	0.9954	0.9892	0.9651	0.9954	0.9892	0.9651	0.9805	1.0064
: gamma	0.9948	0.9883	0.9717	0.9948	0.9883	0.9717	0.9822	1.0377

Table B.2 The optimum scalars of Barbara image for 3-level and 5-level image decomposition using the l_1 and l_2 norm based modified S+P transform, the l_1 and l_2 norm based 2D-HVDD, the l_1 and l_2 norm based LDT-HVDD, the l_1 and l_2 norm based OS-HVD, the l_1 and l_2 norm based OS-HVVD, and the l_1 and l_2 norm based OS-HVDD methods.

Barbara	3-level image decomposition				5-level image decomposition			
Method								
Modified S+P(l_1)	-0.0612	0.2658	0.2660	0.2368	-0.0611	0.2639	0.2599	0.2296
Modified S+P(l_2)	-0.0392	0.1877	0.1885	0.2919	-0.0410	0.1852	0.1843	0.2855
2D-HVDD(l_1) : hv filter	-0.0341	-0.0325	0.5311	0.5367	-0.0328	-0.0310	0.5308	0.5342
: d filter	-0.0052	0.5342	0.5042	-0.0312	-0.0009	0.5300	0.5014	-0.0285
2D-HVDD(l_2) : hv filter	-0.0216	-0.0238	0.5185	0.5274	-0.0157	-0.0177	0.5151	0.5187
: d filter	0.0512	0.4718	0.4450	0.0279	0.0616	0.4611	0.4361	0.0374
LDT-HVDD(l_1) : h filter	0.0899	0.3986	0.4073	0.1056	0.0856	0.4004	0.4105	0.1049
: v filter	-0.0993	0.5861	0.6084	-0.0952	-0.0983	0.5849	0.6066	-0.0932
: d filter	0.1834	0.3004	0.3307	0.1872	0.1773	0.3057	0.3344	0.1842
LDT-HVDD(l_2) : h filter	0.2342	0.2495	0.2637	0.2513	0.2271	0.2559	0.2703	0.2457
: v filter	-0.0696	0.5645	0.5661	-0.0621	-0.0657	0.5598	0.5619	-0.0571
: d filter	0.2889	0.1914	0.2234	0.2946	0.2837	0.1953	0.2265	0.2929
	Level-1	Level-2	Level-3	Level-1	Level-2	Level-3	Level-4	Level-5
OS-HVD(l_1) : alpha	0.9931	0.9868	0.9774	0.9931	0.9868	0.9774	0.9778	0.9657
: beta	1.0000	0.9939	0.9895	1.0000	0.9939	0.9895	0.9886	0.9691
OS-HVD(l_2) : alpha	0.9725	0.9579	0.9543	0.9725	0.9579	0.9543	0.9601	0.9651
: beta	0.9940	0.9794	0.9597	0.9940	0.9794	0.9597	0.9430	0.9146
OS-HVVD(l_1) : alpha	0.9931	0.9868	0.9774	0.9931	0.9868	0.9774	0.9778	0.9657
: beta	1.0000	0.9939	0.9895	1.0000	0.9939	0.9895	0.9886	0.9691
: gamma	0.8571	0.5088	0.3214	0.8571	0.5088	0.3214	0.3077	0.2222
OS-HVVD(l_2) : alpha	0.9725	0.9579	0.9543	0.9725	0.9579	0.9543	0.9601	0.9651
: beta	0.9940	0.9794	0.9597	0.9940	0.9794	0.9597	0.9430	0.9146
: gamma	0.8262	0.4229	0.1457	0.8262	0.4229	0.1457	0.2659	0.2487
OS-HVDD(l_1) : alpha	0.9932	0.9867	0.9777	0.9932	0.9867	0.9777	0.9833	0.9558
: beta	1.0000	0.9939	0.9895	1.0000	0.9939	0.9895	0.9886	0.9691
: gamma	0.9896	0.9821	0.9703	0.9896	0.9821	0.9703	0.9684	0.9897
OS-HVDD(l_2) : alpha	0.9726	0.9563	0.9489	0.9726	0.9563	0.9489	0.9601	0.9514
: beta	0.9940	0.9794	0.9597	0.9940	0.9794	0.9597	0.9430	0.9146
: gamma	0.9650	0.9644	0.9473	0.9650	0.9644	0.9473	0.9471	0.9773

Table B.3 The optimum scalars of Camera image for 3-level and 5-level image decomposition using the l_1 and l_2 norm based modified S+P transform, the l_1 and l_2 norm based 2D-HVDD, the l_1 and l_2 norm based LDT-HVDD, the l_1 and l_2 norm based OS-HVD, the l_1 and l_2 norm based OS-HVVD, and the l_1 and l_2 norm based OS-HVDD methods.

Camera	3-level image decomposition				5-level image decomposition			
Method								
Modified S+P(l_1)	-0.0224	0.1336	0.0206	-0.0694	-0.0236	0.1320	0.0178	-0.0696
Modified S+P(l_2)	-0.0733	0.2237	0.0922	-0.0943	-0.0745	0.2223	0.0880	-0.0958
2D-HVDD(l_1) : hv filter	0.1316	0.1334	0.3820	0.3535	0.1276	0.1295	0.3860	0.3573
: d filter	0.2264	0.2855	0.2832	0.2052	0.2184	0.2898	0.2905	0.2015
2D-HVDD(l_2) : hv filter	0.1420	0.1356	0.3691	0.3565	0.1445	0.1370	0.3677	0.3538
: d filter	0.2162	0.2988	0.2847	0.2026	0.2120	0.3028	0.2842	0.2027
LDT-HVDD(l_1) : h filter	-0.0010	0.4988	0.5066	-0.0044	-0.0001	0.4996	0.5044	-0.0039
: v filter	-0.0105	0.4981	0.5297	-0.0173	-0.0093	0.5010	0.5242	-0.0160
: d filter	0.0000	0.4884	0.5165	-0.0050	0.0001	0.4896	0.5153	-0.0051
LDT-HVDD(l_2) : h filter	0.0048	0.4814	0.5069	0.0056	0.0088	0.4770	0.5024	0.0101
: v filter	-0.0112	0.5047	0.5210	-0.0139	-0.0074	0.5052	0.5154	-0.0130
: d filter	0.0307	0.4603	0.4896	0.0188	0.0297	0.4616	0.4854	0.0220
	Level-1	Level-2	Level-3	Level-1	Level-2	Level-3	Level-4	Level-5
OS-HVD(l_1) : alpha	1.0000	1.0000	0.9946	1.0000	1.0000	0.9946	0.9938	0.9834
: beta	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	0.9940	0.9934
OS-HVD(l_2) : alpha	0.9869	0.9703	0.9465	0.9869	0.9703	0.9465	0.9349	0.8966
: beta	0.9922	0.9743	0.9652	0.9922	0.9743	0.9652	0.9130	0.9161
OS-HVVD(l_1) : alpha	1.0000	1.0000	0.9946	1.0000	1.0000	0.9946	0.9938	0.9834
: beta	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	0.9940	0.9934
: gamma	0.6116	0.5410	0.2588	0.6116	0.5410	0.2588	0.6702	0.0667
OS-HVVD(l_2) : alpha	0.9869	0.9703	0.9465	0.9869	0.9703	0.9465	0.9349	0.8966
: beta	0.9922	0.9743	0.9652	0.9922	0.9743	0.9652	0.9130	0.9161
: gamma	0.6864	0.5983	0.3891	0.6864	0.5983	0.3891	0.4933	0.3303
OS-HVDD(l_1) : alpha	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	0.9887	0.9721
: beta	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	0.9940	0.9934
: gamma	1.0000	0.9946	1.0000	1.0000	0.9946	1.0000	0.9882	0.9822
OS-HVDD(l_2) : alpha	0.9871	0.9687	0.9571	0.9871	0.9687	0.9571	0.9320	0.8936
: beta	0.9922	0.9743	0.9652	0.9922	0.9743	0.9652	0.9130	0.9161
: gamma	0.9825	0.9578	0.9275	0.9825	0.9578	0.9275	0.8903	0.8585

Table B.4 The optimum scalars of Face image for 3-level and 5-level image decomposition using the l_1 and l_2 norm based modified S+P transform, the l_1 and l_2 norm based 2D-HVDD, the l_1 and l_2 norm based LDT-HVDD, the l_1 and l_2 norm based OS-HVD, the l_1 and l_2 norm based OS-HVVD, and the l_1 and l_2 norm based OS-HVDD methods.

Face	3-level image decomposition				5-level image decomposition			
Method								
Modified S+P(l_1)	-0.0305	0.2551	0.1387	-0.0876	-0.0326	0.2564	0.1353	-0.0917
Modified S+P(l_2)	-0.0727	0.2818	0.1350	-0.1171	-0.0782	0.2861	0.1274	-0.1282
2D-HVDD(l_1) : hv filter	0.1580	0.1635	0.3397	0.3402	0.1618	0.1646	0.3375	0.3373
: d filter	0.1623	0.3362	0.3328	0.1718	0.1624	0.3348	0.3310	0.1747
2D-HVDD(l_2) : hv filter	0.1364	0.1321	0.3669	0.3691	0.1522	0.1427	0.3550	0.3548
: d filter	0.1734	0.3256	0.3237	0.1852	0.1820	0.3155	0.3190	0.1911
LDT-HVDD(l_1) : h filter	-0.0344	0.5361	0.5342	-0.0355	-0.0322	0.5327	0.5354	-0.0356
: v filter	-0.0502	0.5663	0.5210	-0.0373	-0.0470	0.5609	0.5202	-0.0343
: d filter	-0.0335	0.5384	0.5241	-0.0280	-0.0314	0.5358	0.5236	-0.0270
LDT-HVDD(l_2) : h filter	-0.0413	0.5428	0.5414	-0.0439	-0.0385	0.5410	0.5359	-0.0393
: v filter	-0.0547	0.5658	0.5373	-0.0495	-0.0481	0.5522	0.5392	-0.0447
: d filter	-0.0379	0.5431	0.5330	-0.0384	-0.0327	0.5372	0.5281	-0.0329
	Level-1	Level-2	Level-3	Level-1	Level-2	Level-3	Level-4	Level-5
OS-HVD(l_1) : alpha	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	0.9888
: beta	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	0.9921	0.8000
OS-HVD(l_2) : alpha	0.9899	0.9702	0.9451	0.9899	0.9702	0.9451	0.8492	0.9450
: beta	0.9912	0.9873	0.9490	0.9912	0.9873	0.9490	0.8975	0.6815
OS-HVVD(l_1) : alpha	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	0.9888
: beta	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	0.9921	0.8000
: gamma	0.6563	0.5946	0.4615	0.6563	0.5946	0.4615	0.4945	0.4783
OS-HVVD(l_2) : alpha	0.9899	0.9702	0.9451	0.9899	0.9702	0.9451	0.8492	0.9450
: beta	0.9912	0.9873	0.9490	0.9912	0.9873	0.9490	0.8975	0.6815
: gamma	0.7089	0.6279	0.5242	0.7089	0.6279	0.5242	0.5728	0.5237
OS-HVDD(l_1) : alpha	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0114
: beta	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	0.9921	0.8000
: gamma	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	0.9646	0.7303
OS-HVDD(l_2) : alpha	0.9888	0.9712	0.9482	0.9888	0.9712	0.9482	0.8533	1.0025
: beta	0.9912	0.9873	0.9490	0.9912	0.9873	0.9490	0.8975	0.6815
: gamma	0.9839	0.9631	0.9168	0.9839	0.9631	0.9168	0.8010	0.7164

Table B.5 The optimum scalars of Lena image for 3-level and 5-level image decomposition using the l_1 and l_2 norm based modified S+P transform, the l_1 and l_2 norm based 2D-HVDD, the l_1 and l_2 norm based LDT-HVDD, the l_1 and l_2 norm based OS-HVD, the l_1 and l_2 norm based OS-HVVD, and the l_1 and l_2 norm based OS-HVDD methods.

Lena	3-level image decomposition				5-level image decomposition			
Method								
Modified S+P(l_1)	-0.0420	0.2142	0.2205	0.0515	-0.0438	0.2110	0.2173	0.0526
Modified S+P(l_2)	-0.0958	0.1923	0.2060	0.0559	-0.0973	0.1905	0.2021	0.0560
2D-HVDD(l_1) : hv filter	0.0352	0.0381	0.4610	0.4661	0.0340	0.0376	0.4612	0.4676
: d filter	0.1185	0.3969	0.3641	0.1227	0.1198	0.3945	0.3654	0.1225
2D-HVDD(l_2) : hv filter	0.0726	0.0853	0.4157	0.4278	0.0718	0.0865	0.4131	0.4297
: d filter	0.1717	0.3376	0.3244	0.1684	0.1738	0.3332	0.3247	0.1705
LDT-HVDD(l_1) : h filter	-0.0218	0.5123	0.5414	-0.0315	-0.0205	0.5112	0.5391	-0.0293
: v filter	-0.0271	0.5232	0.5212	-0.0173	-0.0248	0.5221	0.5173	-0.0145
: d filter	-0.0151	0.4979	0.5365	-0.0186	-0.0141	0.4954	0.5363	-0.0170
LDT-HVDD(l_2) : h filter	-0.0063	0.5063	0.5073	-0.0075	-0.0029	0.5035	0.4997	-0.0007
: v filter	-0.0196	0.5281	0.4967	-0.0057	-0.0152	0.5275	0.4891	-0.0022
: d filter	0.0043	0.4901	0.4990	0.0052	0.0075	0.4856	0.4964	0.0092
	Level-1	Level-2	Level-3	Level-1	Level-2	Level-3	Level-4	Level-5
OS-HVD(l_1) : alpha	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0134	1.0000
: beta	1.0000	0.9952	0.9924	1.0000	0.9952	0.9924	0.9868	0.9530
OS-HVD(l_2) : alpha	0.9914	0.9798	0.9698	0.9914	0.9798	0.9698	0.9608	0.8956
: beta	0.9937	0.9884	0.9733	0.9937	0.9884	0.9733	0.9380	0.8734
OS-HVVD(l_1) : alpha	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0134	1.0000
: beta	1.0000	0.9952	0.9924	1.0000	0.9952	0.9924	0.9868	0.9530
: gamma	0.7000	0.6023	0.5563	0.7000	0.6023	0.5563	0.4451	0.1159
OS-HVVD(l_2) : alpha	0.9914	0.9798	0.9698	0.9914	0.9798	0.9698	0.9608	0.8956
: beta	0.9937	0.9884	0.9733	0.9937	0.9884	0.9733	0.9380	0.8734
: gamma	0.5774	0.5239	0.4770	0.5774	0.5239	0.4770	0.4328	0.2357
OS-HVDD(l_1) : alpha	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0127	0.9800
: beta	1.0000	0.9952	0.9924	1.0000	0.9952	0.9924	0.9868	0.9530
: gamma	1.0000	1.0000	0.9937	1.0000	1.0000	0.9937	1.0151	0.9574
OS-HVDD(l_2) : alpha	0.9912	0.9811	0.9670	0.9912	0.9811	0.9670	0.9422	0.8739
: beta	0.9937	0.9884	0.9733	0.9937	0.9884	0.9733	0.9380	0.8734
: gamma	0.9878	0.9747	0.9605	0.9878	0.9747	0.9605	0.9503	0.8458

Table B.6 The optimum scalars of Milkdrop image for 3-level and 5-level image decomposition using the l_1 and l_2 norm based modified S+P transform, the l_1 and l_2 norm based 2D-HVDD, the l_1 and l_2 norm based LDT-HVDD, the l_1 and l_2 norm based OS-HVD, the l_1 and l_2 norm based OS-HVVD, and the l_1 and l_2 norm based OS-HVDD methods.

Milkdrop	3-level image decomposition				5-level image decomposition			
Method								
Modified S+P(l_1)	-0.0417	0.2199	0.0718	-0.1491	-0.0404	0.2173	0.0684	-0.1533
Modified S+P(l_2)	-0.0686	0.3153	0.1219	-0.1303	-0.0661	0.3122	0.1195	-0.1390
2D-HVDD(l_1) : hv filter	0.1010	0.1316	0.3797	0.3886	0.1038	0.1321	0.3790	0.3860
: d filter	0.1374	0.1386	0.3491	0.3753	0.1367	0.1409	0.3497	0.3731
2D-HVDD(l_2) : hv filter	0.1274	0.1633	0.3577	0.3549	0.1326	0.1692	0.3564	0.3452
: d filter	0.1472	0.1312	0.3465	0.3787	0.1471	0.1367	0.3451	0.3741
LDT-HVDD(l_1) : h filter	-0.0209	0.6044	0.4363	-0.0203	-0.0195	0.6035	0.4351	-0.0195
: v filter	-0.0171	0.5275	0.5128	-0.0235	-0.0159	0.5282	0.5084	-0.0210
: d filter	-0.0157	0.5941	0.4377	-0.0168	-0.0148	0.5914	0.4392	-0.0166
LDT-HVDD(l_2) : h filter	-0.0558	0.6761	0.4108	-0.0331	-0.0453	0.6610	0.4136	-0.0313
: v filter	-0.0423	0.5468	0.5405	-0.0460	-0.0305	0.5355	0.5301	-0.0365
: d filter	-0.0343	0.6247	0.4394	-0.0328	-0.0290	0.6134	0.4408	-0.0288
	Level-1	Level-2	Level-3	Level-1	Level-2	Level-3	Level-4	Level-5
OS-HVD(l_1) : alpha	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	0.9935	1.0058
: beta	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	0.9835
OS-HVD(l_2) : alpha	0.9910	0.9791	0.9548	0.9910	0.9791	0.9548	0.9472	0.9400
: beta	0.9966	0.9909	0.9886	0.9966	0.9909	0.9886	0.9446	0.9118
OS-HVVD(l_1) : alpha	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	0.9935	1.0058
: beta	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	0.9835
: gamma	0.9836	0.9753	0.9504	0.9836	0.9753	0.9504	0.9211	0.0593
OS-HVVD(l_2) : alpha	0.9910	0.9791	0.9548	0.9910	0.9791	0.9548	0.9472	0.9400
: beta	0.9966	0.9909	0.9886	0.9966	0.9909	0.9886	0.9446	0.9118
: gamma	0.8801	0.8496	0.7159	0.8801	0.8496	0.7159	0.5666	-0.0473
OS-HVDD(l_1) : alpha	1.0000	1.0000	0.9957	1.0000	1.0000	0.9957	0.9920	1.0000
: beta	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	0.9835
: gamma	1.0000	1.0000	0.9912	1.0000	1.0000	0.9912	0.9825	0.9906
OS-HVDD(l_2) : alpha	0.9909	0.9786	0.9545	0.9909	0.9786	0.9545	0.9460	0.9368
: beta	0.9966	0.9909	0.9886	0.9966	0.9909	0.9886	0.9446	0.9118
: gamma	0.9890	0.9757	0.9556	0.9890	0.9757	0.9556	0.9146	0.9167

Table B.7 The optimum scalars of Peppers image for 3-level and 5-level image decomposition using the l_1 and l_2 norm based modified S+P transform, the l_1 and l_2 norm based 2D-HVDD, the l_1 and l_2 norm based LDT-HVDD, the l_1 and l_2 norm based OS-HVD, the l_1 and l_2 norm based OS-HVVD, and the l_1 and l_2 norm based OS-HVDD methods.

Peppers	3-level image decomposition				5-level image decomposition			
Method								
Modified S+P(l_1)	-0.0434	0.2498	0.1390	-0.1094	-0.0435	0.2477	0.1342	-0.1124
Modified S+P(l_2)	-0.0681	0.2829	0.1433	-0.1844	-0.0708	0.2783	0.1379	-0.1858
2D-HVDD(l_1) : hv filter	0.1711	0.1871	0.3135	0.3294	0.1676	0.1813	0.3178	0.3346
: d filter	0.1613	0.2989	0.3132	0.2288	0.1612	0.2942	0.3159	0.2309
2D-HVDD(l_2) : hv filter	0.2116	0.2196	0.2705	0.3027	0.2121	0.2154	0.2746	0.3024
: d filter	0.1336	0.2883	0.3030	0.2816	0.1347	0.2820	0.3049	0.2856
LDT-HVDD(l_1) : h filter	-0.0237	0.5179	0.5310	-0.0251	-0.0229	0.5176	0.5292	-0.0238
: v filter	-0.0199	0.5152	0.5149	-0.0101	-0.0178	0.5113	0.5160	-0.0095
: d filter	-0.0199	0.5147	0.5202	-0.0144	-0.0196	0.5139	0.5206	-0.0143
LDT-HVDD(l_2) : h filter	-0.0265	0.5233	0.5414	-0.0400	-0.0193	0.5138	0.5396	-0.0356
: v filter	-0.0289	0.5291	0.5114	-0.0122	-0.0217	0.5164	0.5137	-0.0087
: d filter	-0.0246	0.5282	0.5160	-0.0217	-0.0187	0.5212	0.5145	-0.0189
	Level-1	Level-2	Level-3	Level-1	Level-2	Level-3	Level-4	Level-5
OS-HVD(l_1) : alpha	1.0000	1.0048	1.0000	1.0000	1.0048	1.0000	1.0110	1.0396
: beta	1.0000	0.9947	0.9896	1.0000	0.9947	0.9896	0.9890	0.9550
OS-HVD(l_2) : alpha	0.9940	0.9823	0.9588	0.9940	0.9823	0.9588	0.9509	0.9759
: beta	0.9964	0.9896	0.9724	0.9964	0.9896	0.9724	0.9851	0.9492
OS-HVVD(l_1) : alpha	1.0000	1.0048	1.0000	1.0000	1.0048	1.0000	1.0110	1.0396
: beta	1.0000	0.9947	0.9896	1.0000	0.9947	0.9896	0.9890	0.9550
: gamma	0.7910	0.6944	0.6727	0.7910	0.6944	0.6727	0.6831	0.4545
OS-HVVD(l_2) : alpha	0.9940	0.9823	0.9588	0.9940	0.9823	0.9588	0.9509	0.9759
: beta	0.9964	0.9896	0.9724	0.9964	0.9896	0.9724	0.9851	0.9492
: gamma	0.7447	0.6753	0.6211	0.7447	0.6753	0.6211	0.6429	0.4630
OS-HVDD(l_1) : alpha	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0097	1.0326
: beta	1.0000	0.9947	0.9896	1.0000	0.9947	0.9896	0.9890	0.9550
: gamma	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0088	1.0761
OS-HVDD(l_2) : alpha	0.9939	0.9820	0.9566	0.9939	0.9820	0.9566	0.9517	0.9755
: beta	0.9964	0.9896	0.9724	0.9964	0.9896	0.9724	0.9851	0.9492
: gamma	0.9923	0.9791	0.9470	0.9923	0.9791	0.9470	0.9676	1.0126

Table B.8 The optimum scalars of Tiffany image for 3-level and 5-level image decomposition using the l_1 and l_2 norm based modified S+P transform, the l_1 and l_2 norm based 2D-HVDD, the l_1 and l_2 norm based LDT-HVDD, the l_1 and l_2 norm based OS-HVD, the l_1 and l_2 norm based OS-HVVD, and the l_1 and l_2 norm based OS-HVDD methods.

Tiffany	3-level image decomposition				5-level image decomposition			
Method								
Modified S+P(l_1)	-0.0424	0.1660	0.0894	-0.0480	-0.0430	0.1598	0.0832	-0.0496
Modified S+P(l_2)	-0.0903	0.1403	0.0676	-0.0506	-0.0907	0.1378	0.0652	-0.0505
2D-HVDD(l_1) : hv filter	0.0855	0.1186	0.3889	0.4077	0.0847	0.1180	0.3896	0.4085
: d filter	0.1836	0.2498	0.3253	0.2431	0.1818	0.2492	0.3276	0.2432
2D-HVDD(l_2) : hv filter	0.1236	0.1552	0.3509	0.3704	0.1241	0.1570	0.3503	0.3687
: d filter	0.1622	0.2164	0.3439	0.2786	0.1600	0.2153	0.3459	0.2800
LDT-HVDD(l_1) : h filter	0.0028	0.5081	0.4779	0.0116	0.0033	0.5072	0.4783	0.0116
: v filter	0.0000	0.5000	0.5000	0.0000	0.0000	0.5000	0.5000	0.0000
: d filter	0.0040	0.5101	0.4761	0.0116	0.0054	0.5078	0.4760	0.0127
LDT-HVDD(l_2) : h filter	0.0375	0.4786	0.4392	0.0447	0.0407	0.4762	0.4391	0.0439
: v filter	0.0248	0.4864	0.4738	0.0151	0.0295	0.4826	0.4692	0.0188
: d filter	0.0368	0.4719	0.4473	0.0443	0.0415	0.4670	0.4436	0.0482
	Level-1	Level-2	Level-3	Level-1	Level-2	Level-3	Level-4	Level-5
OS-HVD(l_1) : alpha	1.0000	1.0000	1.0041	1.0000	1.0000	1.0041	1.0047	1.0086
: beta	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
OS-HVD(l_2) : alpha	1.0003	1.0012	1.0054	1.0003	1.0012	1.0054	1.0086	1.0191
: beta	0.9989	0.9989	1.0003	0.9989	0.9989	1.0003	0.9948	0.9723
OS-HVVD(l_1) : alpha	1.0000	1.0000	1.0041	1.0000	1.0000	1.0041	1.0047	1.0086
: beta	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
: gamma	0.5641	0.5238	0.5000	0.5641	0.5238	0.5000	0.3750	0.6667
OS-HVVD(l_2) : alpha	1.0003	1.0012	1.0054	1.0003	1.0012	1.0054	1.0086	1.0191
: beta	0.9989	0.9989	1.0003	0.9989	0.9989	1.0003	0.9948	0.9723
: gamma	0.5462	0.5161	0.5014	0.5462	0.5161	0.5014	0.4732	0.4408
OS-HVDD(l_1) : alpha	1.0000	1.0000	1.0040	1.0000	1.0000	1.0040	1.0045	1.0086
: beta	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
: gamma	1.0000	1.0000	1.0043	1.0000	1.0000	1.0043	1.0043	1.0000
OS-HVDD(l_2) : alpha	1.0001	1.0013	1.0058	1.0001	1.0013	1.0058	1.0052	1.0034
: beta	0.9989	0.9989	1.0003	0.9989	0.9989	1.0003	0.9948	0.9723
: gamma	1.0004	1.0020	1.0088	1.0004	1.0020	1.0088	1.0119	1.0233

Table B.9 The optimum scalars of Zelda image for 3-level and 5-level image decomposition using the l_1 and l_2 norm based modified S+P transform, the l_1 and l_2 norm based 2D-HVDD, the l_1 and l_2 norm based LDT-HVDD, the l_1 and l_2 norm based OS-HVD, the l_1 and l_2 norm based OS-HVVD, and the l_1 and l_2 norm based OS-HVDD methods.

Zelda	3-level image decomposition				5-level image decomposition			
Method								
Modified S+P(l_1)	-0.0233	0.2699	0.1240	-0.1269	-0.0249	0.2695	0.1197	-0.1310
Modified S+P(l_2)	-0.0575	0.2620	0.0970	-0.1577	-0.0606	0.2633	0.0948	-0.1626
2D-HVDD(l_1) : hv filter	0.1206	0.1218	0.3817	0.3766	0.1225	0.1231	0.3800	0.3752
: d filter	0.2471	0.2513	0.2659	0.2374	0.2471	0.2525	0.2664	0.2356
2D-HVDD(l_2) : hv filter	0.1563	0.1633	0.3403	0.3424	0.1620	0.1670	0.3377	0.3357
: d filter	0.2357	0.2535	0.2647	0.2491	0.2364	0.2560	0.2658	0.2449
LDT-HVDD(l_1) : h filter	-0.0422	0.5429	0.5389	-0.0394	-0.0398	0.5396	0.5377	-0.0373
: v filter	-0.0370	0.5337	0.5507	-0.0473	-0.0359	0.5322	0.5522	-0.0484
: d filter	-0.0317	0.5334	0.5309	-0.0322	-0.0294	0.5309	0.5295	-0.0306
LDT-HVDD(l_2) : h filter	-0.0263	0.5407	0.5005	-0.0143	-0.0194	0.5310	0.5001	-0.0113
: v filter	-0.0364	0.5334	0.5528	-0.0492	-0.0314	0.5276	0.5526	-0.0480
: d filter	-0.0251	0.5312	0.5049	-0.0101	-0.0183	0.5229	0.5034	-0.0070
	Level-1	Level-2	Level-3	Level-1	Level-2	Level-3	Level-4	Level-5
OS-HVD(l_1) : alpha	1.0000	1.0000	0.9937	1.0000	1.0000	0.9937	0.9677	0.9675
: beta	1.0000	1.0000	0.9926	1.0000	1.0000	0.9926	0.9944	0.9172
OS-HVD(l_2) : alpha	0.9959	0.9919	0.9804	0.9959	0.9919	0.9804	0.9432	0.9841
: beta	0.9978	0.9940	0.9857	0.9978	0.9940	0.9857	0.9883	0.9309
OS-HVVD(l_1) : alpha	1.0000	1.0000	0.9937	1.0000	1.0000	0.9937	0.9677	0.9675
: beta	1.0000	1.0000	0.9926	1.0000	1.0000	0.9926	0.9944	0.9172
: gamma	0.7500	0.6667	0.6296	0.7500	0.6667	0.6296	0.6176	0.2174
OS-HVVD(l_2) : alpha	0.9959	0.9919	0.9804	0.9959	0.9919	0.9804	0.9432	0.9841
: beta	0.9978	0.9940	0.9857	0.9978	0.9940	0.9857	0.9883	0.9309
: gamma	0.7256	0.5544	0.5065	0.7256	0.5544	0.5065	0.5282	0.1380
OS-HVDD(l_1) : alpha	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	0.9762	1.0000
: beta	1.0000	1.0000	0.9926	1.0000	1.0000	0.9926	0.9944	0.9172
: gamma	1.0000	0.9932	0.9811	1.0000	0.9932	0.9811	0.9512	0.9854
OS-HVDD(l_2) : alpha	0.9959	0.9928	0.9811	0.9959	0.9928	0.9811	0.9494	0.9596
: beta	0.9978	0.9940	0.9857	0.9978	0.9940	0.9857	0.9883	0.9309
: gamma	0.9950	0.9882	0.9743	0.9950	0.9882	0.9743	0.9402	0.9610

Table B.10 The optimum scalars of CT image for 3-level and 5-level image decomposition using the l_1 and l_2 norm based modified S+P transform, the l_1 and l_2 norm based 2D-HVDD, the l_1 and l_2 norm based LDT-HVDD, the l_1 and l_2 norm based OS-HVD, the l_1 and l_2 norm based OS-HVVD, and the l_1 and l_2 norm based OS-HVDD methods.

CT	3-level image decomposition				5-level image decomposition			
Method								
Modified S+P(l_1)	-0.0138	0.2034	0.1389	0.0215	-0.0142	0.2037	0.1379	0.0246
Modified S+P(l_2)	-0.0391	0.2932	0.2035	-0.0520	-0.0429	0.2924	0.2004	-0.0524
2D-HVDD(l_1) : hv filter	0.0143	0.0264	0.4906	0.4690	0.0153	0.0278	0.4936	0.4635
: d filter	0.1394	0.3677	0.3547	0.1396	0.1349	0.3700	0.3641	0.1324
2D-HVDD(l_2) : hv filter	0.1161	0.1607	0.3889	0.3403	0.1219	0.1629	0.3883	0.3334
: d filter	0.1832	0.3059	0.2912	0.2263	0.1854	0.3005	0.2942	0.2260
LDT-HVDD(l_1) : h filter	-0.0144	0.5216	0.4960	-0.0032	-0.0131	0.5194	0.4956	-0.0019
: v filter	-0.0156	0.4978	0.5273	-0.0095	-0.0141	0.4906	0.5332	-0.0097
: d filter	-0.0061	0.5191	0.4870	0.0000	-0.0057	0.5131	0.4926	0.0000
LDT-HVDD(l_2) : h filter	-0.0317	0.5536	0.5054	-0.0301	-0.0282	0.5473	0.5020	-0.0237
: v filter	-0.0261	0.5260	0.5214	-0.0228	-0.0221	0.5117	0.5331	-0.0239
: d filter	-0.0110	0.5269	0.4893	-0.0074	-0.0084	0.5219	0.4896	-0.0056
	Level-1	Level-2	Level-3	Level-1	Level-2	Level-3	Level-4	Level-5
OS-HVD(l_1) : alpha	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	0.9692	0.8844
: beta	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	0.8185
OS-HVD(l_2) : alpha	0.9783	0.9523	0.9026	0.9783	0.9523	0.9026	0.8349	0.6721
: beta	0.9840	0.9724	0.9280	0.9840	0.9724	0.9280	0.9134	0.7626
OS-HVVD(l_1) : alpha	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	0.9692	0.8844
: beta	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	0.8185
: gamma	0.7992	0.5882	0.3858	0.7992	0.5882	0.3858	0.3000	0.9804
OS-HVVD(l_2) : alpha	0.9783	0.9523	0.9026	0.9783	0.9523	0.9026	0.8349	0.6721
: beta	0.9840	0.9724	0.9280	0.9840	0.9724	0.9280	0.9134	0.7626
: gamma	0.5797	0.5595	0.4722	0.5797	0.5595	0.4722	0.4760	0.8177
OS-HVDD(l_1) : alpha	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	0.9647	0.9098
: beta	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	0.8185
: gamma	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	0.9725	0.6196
OS-HVDD(l_2) : alpha	0.9765	0.9529	0.9083	0.9765	0.9529	0.9083	0.8488	0.6989
: beta	0.9840	0.9724	0.9280	0.9840	0.9724	0.9280	0.9134	0.7626
: gamma	0.9689	0.9394	0.8680	0.9689	0.9394	0.8680	0.7784	0.5348

Table B.11 The optimum scalars of X-rays image for 3-level and 5-level image decomposition using the l_1 and l_2 modified S+P transform, the l_1 and l_2 2D-HVDD, the l_1 decomposition using the l_1 and l_2 norm based modified S+P transform, the l_1 and l_2 norm based 2D-HVDD, the l_1 and l_2 norm based LDT-HVDD, the l_1 and l_2 norm based OS-HVD, the l_1 and l_2 norm based OS-HVVD, and the l_1 and l_2 norm based OS-HVDD methods.

X-rays	3-level image decomposition				5-level image decomposition			
Method								
Modified S+P(l_1)	-0.0408	0.2991	0.1173	-0.1902	-0.0376	0.2936	0.1230	-0.1812
Modified S+P(l_2)	-0.0667	0.3219	0.1009	-0.2548	-0.0646	0.3122	0.1203	-0.2203
2D-HVDD(l_1) : hv filter	0.2500	0.2500	0.2500	0.2500	0.2500	0.2500	0.2500	0.2500
: d filter	0.2242	0.2669	0.2829	0.2260	0.2215	0.2694	0.2859	0.2232
2D-HVDD(l_2) : hv filter	0.2549	0.2548	0.2436	0.2484	0.2486	0.2495	0.2461	0.2574
: d filter	0.2146	0.2870	0.2931	0.2073	0.2097	0.2940	0.2983	0.2000
LDT-HVDD(l_1) : h filter	-0.0564	0.5551	0.5652	-0.0639	-0.0565	0.5578	0.5586	-0.0598
: v filter	-0.0518	0.5454	0.5601	-0.0536	-0.0505	0.5448	0.5576	-0.0519
: d filter	-0.0415	0.5365	0.5555	-0.0505	-0.0392	0.5341	0.5535	-0.0484
LDT-HVDD(l_2) : h filter	-0.0654	0.5691	0.5594	-0.0634	-0.0608	0.5682	0.5391	-0.0469
: v filter	-0.0597	0.5534	0.5658	-0.0598	-0.0567	0.5507	0.5581	-0.0523
: d filter	-0.0458	0.5420	0.5592	-0.0558	-0.0360	0.5294	0.5531	-0.0470
	Level-1	Level-2	Level-3	Level-1	Level-2	Level-3	Level-4	Level-5
OS-HVD(l_1) : alpha	1.0000	1.0000	0.9958	1.0000	1.0000	0.9958	0.9882	0.9609
: beta	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
OS-HVD(l_2) : alpha	0.9943	0.9878	0.9719	0.9943	0.9878	0.9719	0.9390	0.8673
: beta	0.9982	0.9928	0.9878	0.9982	0.9928	0.9878	0.9615	0.9350
OS-HVVD(l_1) : alpha	1.0000	1.0000	0.9958	1.0000	1.0000	0.9958	0.9882	0.9609
: beta	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
: gamma	0.8333	0.6818	0.5000	0.8333	0.6818	0.5000	0.3733	0.0183
OS-HVVD(l_2) : alpha	0.9943	0.9878	0.9719	0.9943	0.9878	0.9719	0.9390	0.8673
: beta	0.9982	0.9928	0.9878	0.9982	0.9928	0.9878	0.9615	0.9350
: gamma	0.8274	0.6770	0.4842	0.8274	0.6770	0.4842	0.4088	0.2310
OS-HVDD(l_1) : alpha	1.0000	1.0000	0.9958	1.0000	1.0000	0.9958	0.9874	0.9609
: beta	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
: gamma	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	0.9948	0.9874
OS-HVDD(l_2) : alpha	0.9942	0.9873	0.9734	0.9942	0.9873	0.9734	0.9361	0.8355
: beta	0.9982	0.9928	0.9878	0.9982	0.9928	0.9878	0.9615	0.9350
: gamma	0.9926	0.9825	0.9644	0.9926	0.9825	0.9644	0.9215	0.8953

Table B.12 The optimum scalars of Fingerprint image for 3-level and 5-level image decomposition using the l_1 and l_2 norm based modified S+P transform, the l_1 and l_2 norm based 2D-HVDD, the l_1 and l_2 norm based LDT-HVDD, the l_1 and l_2 norm based OS-HVD, the l_1 and l_2 norm based OS-HVVD, and the l_1 and l_2 norm based OS-HVDD methods.

Fingerprint	3-level image decomposition				5-level image decomposition			
Method								
Modified S+P(l_1)	-0.0459	0.1142	0.0132	0.0727	-0.0510	0.1107	0.0110	0.0693
Modified S+P(l_2)	-0.0348	0.1431	0.0279	0.0396	-0.0367	0.1414	0.0265	0.0376
2D-HVDD(l_1) : hv filter	0.0928	0.0884	0.4155	0.4033	0.0956	0.0900	0.4146	0.3997
: d filter	0.4203	0.1011	0.0859	0.3927	0.4165	0.1044	0.0859	0.3931
2D-HVDD(l_2) : hv filter	0.1548	0.1589	0.3466	0.3407	0.1565	0.1587	0.3467	0.3392
: d filter	0.3532	0.1562	0.1643	0.3186	0.3503	0.1575	0.1665	0.3176
LDT-HVDD(l_1) : h filter	0.0089	0.4929	0.5088	-0.0106	0.0108	0.4924	0.5079	-0.0111
: v filter	0.0000	0.5143	0.4969	-0.0112	0.0000	0.5124	0.4976	-0.0100
: d filter	0.0705	0.4090	0.4300	0.0905	0.0679	0.4092	0.4292	0.0938
LDT-HVDD(l_2) : h filter	0.0798	0.4176	0.4251	0.0687	0.0810	0.4160	0.4235	0.0710
: v filter	0.0473	0.4497	0.4534	0.0445	0.0500	0.4484	0.4507	0.0454
: d filter	0.1559	0.3224	0.3529	0.1604	0.1561	0.3224	0.3520	0.1604
	Level-1	Level-2	Level-3	Level-1	Level-2	Level-3	Level-4	Level-5
OS-HVD(l_1) : alpha	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
: beta	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
OS-HVD(l_2) : alpha	0.9644	0.9194	0.9626	0.9644	0.9194	0.9626	0.9433	0.8777
: beta	0.9721	0.9460	0.9378	0.9721	0.9460	0.9378	0.9218	0.9806
OS-HVVD(l_1) : alpha	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
: beta	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
: gamma	0.5682	0.0678	0.0390	0.5682	0.0678	0.0390	0.1296	0.3873
OS-HVVD(l_2) : alpha	0.9644	0.9194	0.9626	0.9644	0.9194	0.9626	0.9433	0.8777
: beta	0.9721	0.9460	0.9378	0.9721	0.9460	0.9378	0.9218	0.9806
: gamma	0.5643	0.1090	0.0433	0.5643	0.1090	0.0433	0.0416	0.2403
OS-HVDD(l_1) : alpha	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
: beta	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
: gamma	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
OS-HVDD(l_2) : alpha	0.9644	0.9174	0.9752	0.9644	0.9174	0.9752	0.9588	0.8883
: beta	0.9721	0.9460	0.9378	0.9721	0.9460	0.9378	0.9218	0.9806
: gamma	0.9592	0.9434	0.9411	0.9592	0.9434	0.9411	0.9150	0.8702

APPENDIX C

WAVELET AND S+P-TRANSFORM FILTER COEFFICIENTS

Table C.1 Wavelet Daubechies length-4 filter coefficients.

Wavelet Daubechies (N = 4) scaling function and wavelet coefficients		
n	$h(n)$	$g(n)$
0	0.48296291314453	0.12940952255126
1	0.83651630373781	0.22414386804201
2	0.22414386804201	-0.83651630373781
3	-0.12940952255126	0.48296291314453

Table C.2 S+P-transform universal filter coefficients.

S+P-transform universal filter coefficients			
	Predictor A	Predictor B	Predictor C
α_{-1}	0	0	- 1/16
α_0	1/4	1/4	1/4
α_1	1/4	3/8	1/2
β_1	0	1/4	3/8

APPENDIX D

SPIHT CODING ALGORITHM

Progressive Image Transmission

Let the original image be defined by a set of pixel values $p(i, j)$, where (i,j) is the pixel coordinate and let T represent a subband transformation. The coding is normally performed on the two-dimensional transform coefficients as given by

$$c(i, j) = T\big(p(i, j)\big). \tag{D.1}^{*}$$

Let $\hat{c}(i, j)$ and $\hat{p}(i, j)$ be the received coefficients and reconstructed image coefficients, respectively. In a progressive transmission scheme, $\hat{p}(i, j)$ are initially set to be zero. Each component is updated according to the received coded message when after receiving the value of some coefficients, a reconstructed image is obtained by

$$\hat{p}(i, j) = T\big(\hat{c}(i, j)\big). \tag{D.2}$$

The main concept of the progressive transmission scheme is to select and primarily transmit the most important information, which yields the smallest distortion. Hence, the transform coefficients $c(i, j)$ are first ranked according to their binary representation then the most significant bits are first transmitted. This idea is also known as a bit-plane method for progressive transmission.

Set Partitioning Sorting Algorithm

In the progressive transmission, the ordering data is not explicitly transmitted to the decoder. However, the execution path of any algorithm is defined on the results of the

---

[*] For coding, it is assumed that each transform coefficient can be treated as an integer.

comparisons on its branching points. It is then possible to have the same execution path for both encoder and decoder. By the knowledge of the execution path and the received information, the recovered process can be completed by the decoder. However, not all coefficients need to be ranked. In fact, it is necessary to design a sorting algorithm such that it divides the set of pixels into portioning subsets *P* and then performs the magnitude comparison as given by

$$S_n(P) = \begin{cases} 1, & \text{for } \max\limits_{(i,j)\in P}\{|c(i,j)|\} \geq 2^n \\ 0, & \text{otherwise} \end{cases} \tag{D.3}$$

where $n = \left\lfloor \log_2\left(\max\limits_{(i,j)}\{|c(i,j)|\} \right) \right\rfloor$. The objective of these partition subsets is to have insignificant subsets ($S_n(P) = 0$) contain a relatively large number of elements while significant subsets ($S_n(P) = 1$) contain a relatively small number of elements. For the significant case, a certain rule, shared by the encoder and the decoder, is used to partition *P* into new subsets and the magnitude comparison is the repeated to the new subsets. This division process continues until all single coefficients are identified to be either significant or insignificant.

Spatial Orientation Trees

In the subband pyramid, the image energy is normally concentrated in the low frequency components. Accordingly, the variance decreases from the highest to the lowest levels of the subband pyramid [54]. Additionally, it has a spatial self-similarity between subbands. Hence, the coefficients are expected to be better magnitude-ordered if the direction moves downward following the same spatial orientation in the subband pyramid. The spatial orientation tree is spatial relationship defined on the hierarchical pyramid. It is constructed with recursive four-subband splitting. Each node of the tree is defined by the pixel coordinate and the pixels of the same spatial orientation in the next finer level of the pyramid are defined to be its descendants (offspring). However, each node of the tree can have either no offspring or four offspring, which always produces a group of 2 x 2 arrays. Figure D.1 illustrates the described spatial orientation tree defined in a pyramid [78]. The arrows indicate the oriented direction from lowest resolution (Band I) to its four offspring. It is interesting to note that the coefficient in the top-left corner of the lowest resolution has no descendants.

The trees are further partitioned into the following sets of coordinates used to present the new coding method:

- $O(i,j)$: the set of coordinates of all offspring of node (i,j) where each node can either have four offspring or none.
- $D(i,j)$: the set of coordinates of all descendants of the node (i,j) which include the offspring, the offspring of the offspring and so on.
- H: the set of coordinates of all spatial orientation tree roots in the lowest resolution band.
- $L(i,j)$: the set of coordinates of all the descendants of node (i,j) except for the immediate offspring of node (i,j), in other words

$$L(i,j) = D(i,j) - O(i,j). \tag{D.4}$$

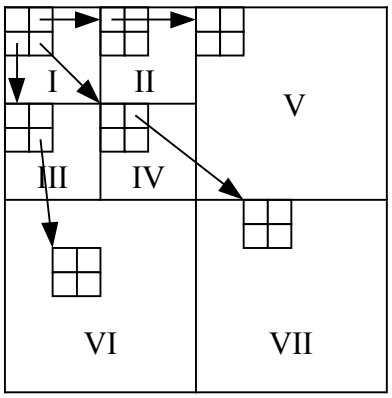

Figure D.1 The data structure used in the SPIHT algorithm.

The parts of the spatial orientation trees are employed as the partitioning subsets in the sorting algorithm in which the set partitioning rules are

1) The initial partition is formed with the sets $\{(i,j)\}$ and $D(i,j)$ for all $(i,j) \in H$.
2) If $D(i,j)$ is significant, then it is partitioned into $L(i,j)$ and the four single-element sets with $(k,l) \in O(i,j)$.

3) If $L(i,j)$ is significant, then it is partitioned into the four sets $D(k,l)$, with $(k,l) \in O(i,j)$.

Coding Algorithm

Due to the fact that the order in which the subsets are tested for significance is important, practically, the significance information is stored in three ordered lists called list of insignificant sets (LIS), list of insignificant pixels (LIP), and list of significant pixels (LSP). Each element in all lists is identified by a coordinate (i,j), which in the LIP and LSP represents individual pixels whereas in the LIS represents either the set $D(i,j)$ (type A) or $L(i,j)$ (type B). The sorting pass algorithm using the set partitioning approach is described as follows.

1) Initialization:

1.1) Output $n = \left\lfloor \log_2 \left(\max_{(i,j)} \{|c(i,j)|\} \right) \right\rfloor$

1.2) Set the LSP as an empty list, add all coordinates $(i,j) \in H$ to the LIP and only those with descendants to the LIS as type A entries.

2) Sorting Pass:

2.1) For each entry (i,j) in the LIP do:

2.1.1) Output $S_n(i,j)$;

2.1.2) If $S_n(i,j) = 1$ then move (i,j) to the LSP and output the sign of $c(i,j)$;

2.2) For each entry (i,j) in the LIS do:

2.2.1) If the entry is type A then

- Output $S_n(D(i,j))$;
- If $S_n(D(i,j)) = 1$ then
  - For each $(k,l) \in O(i,j)$ do:
    - Output $S_n(k,l)$;
    - If $S_n(k,l) = 1$ then add (k,l) to the LSP and output the sign of $c(k,l)$;
    - If $S_n(k,l) = 0$ then add (k,l) to the end of the LIP;

- ❖ If $L(i,j) \neq 0$ then move (i,j) to the end of the LIS as an entry of type B and go to step 2.2.2); otherwise, remove entry (i,j) from the LIS;

2.2.2) If the entry is of type B then

- ➢ Output $S_n\big(L(i,j)\big)$;
- ➢ If $S_n\big(L(i,j)\big)=1$ then add each $(k,l) \in O(i,j)$ to the end of the LIS as an entry of type A and remove (i,j) from the LIS.

3) **Refinement Pass:** For each entry (i,j) in the LSP, except those included in the last sorting pass, output the n^{th} most significant bit of $|c(i,j)|$.

4) **Quantization-Step Update:** Decrement n by 1 and go to step 2.

APPENDIX E

OPTIMUM SCALAR TRANSFORM ON THE CHECKERBOARD IMAGES

A set of 32 x 32 pixels checkerboard images representing different frequencies binary images shown in Figure E.1. They are employed as a test for 6-level image decomposition using the l_1 norm based OS-HVDD technique. The resulted optimum scalars are represented in Table E.1. Furthermore, the results of using checkerboard images#1, #3, and #5 are shown in Figure E.2, E.3, and E.4, respectively.

As the optimum scalars shown in Table E.1, they imply that more levels of optimum scalar decomposition are required (until the subsampled binary image cannot be further decomposed) to decompose low frequency binary image than of high frequency binary images.

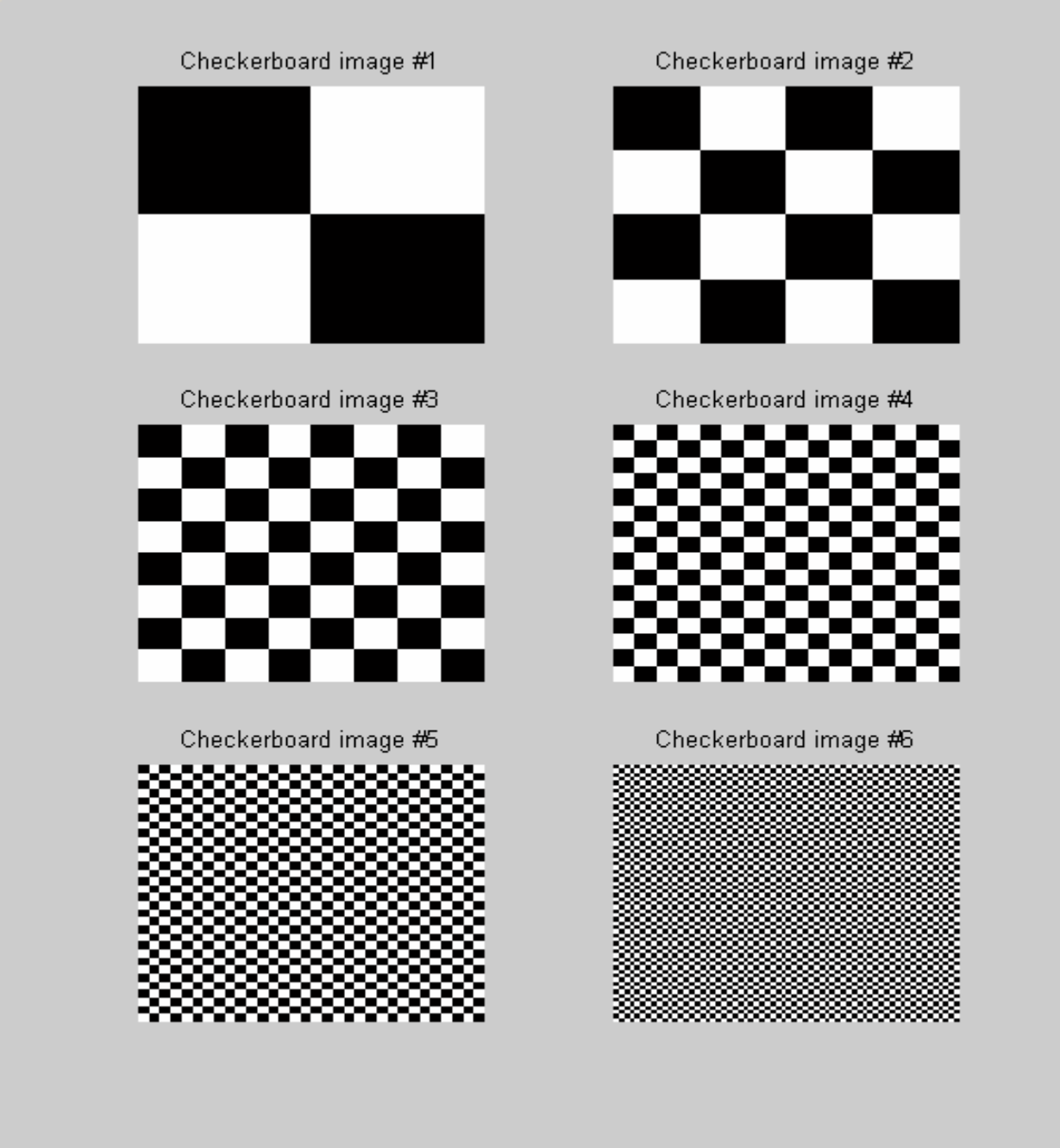

Figure E.1 A set of tested checkerboard images.

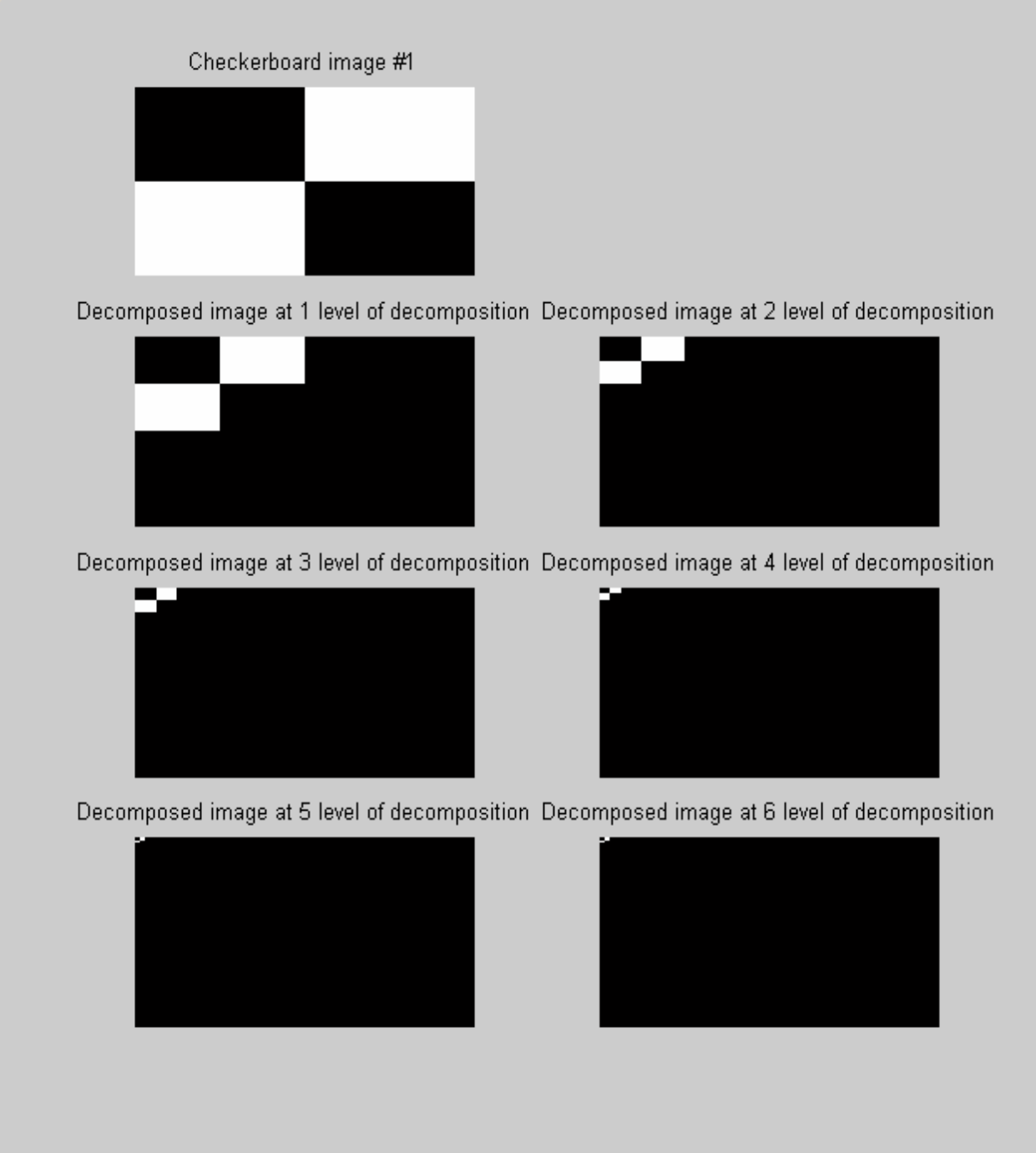

Figure E.2 The 6-level OS-HVDD(l_1) method for checkerboard image #1.

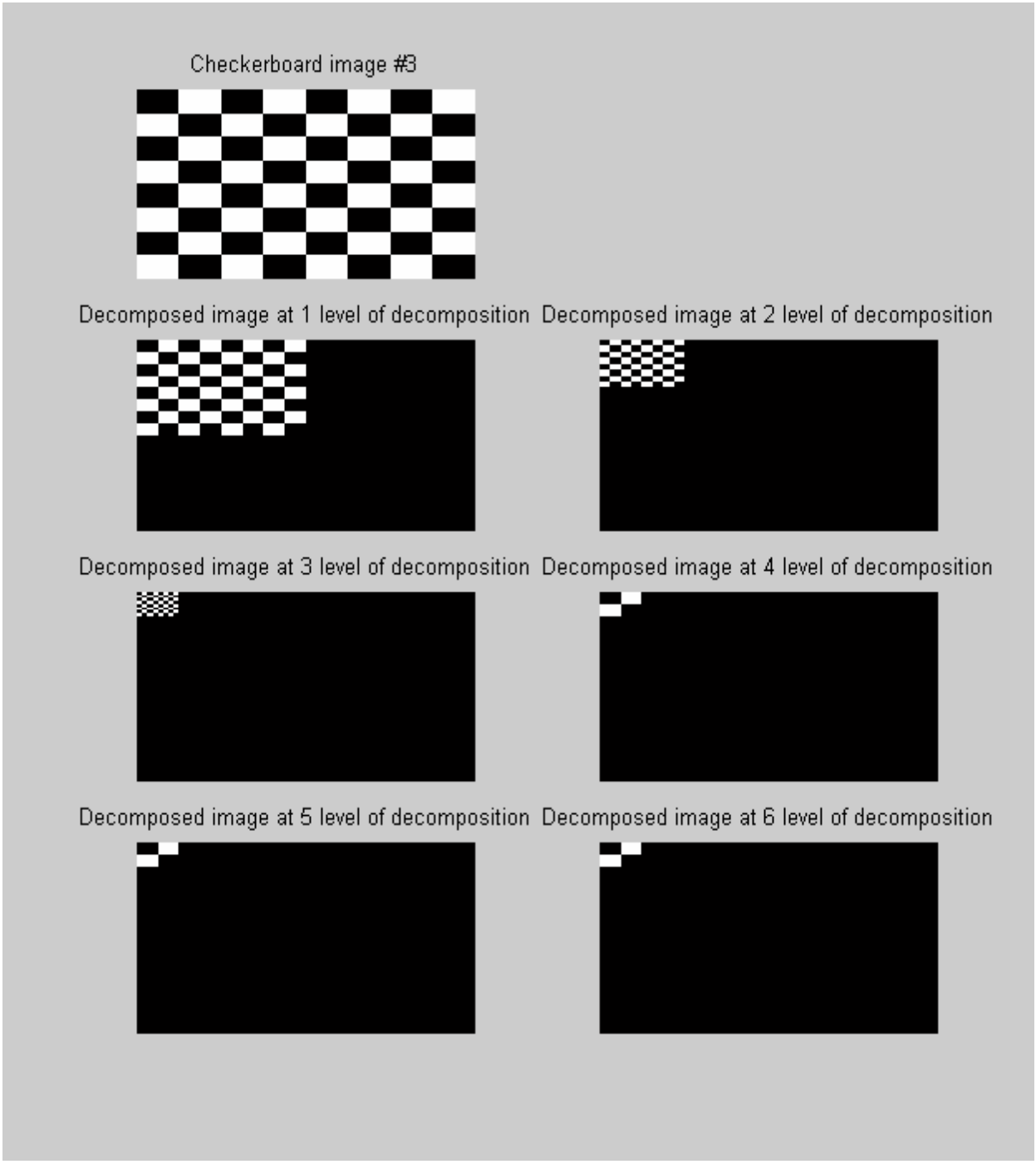

Figure E.3 The 6-level OS-HVDD(l_1) method for checkerboard image #3.

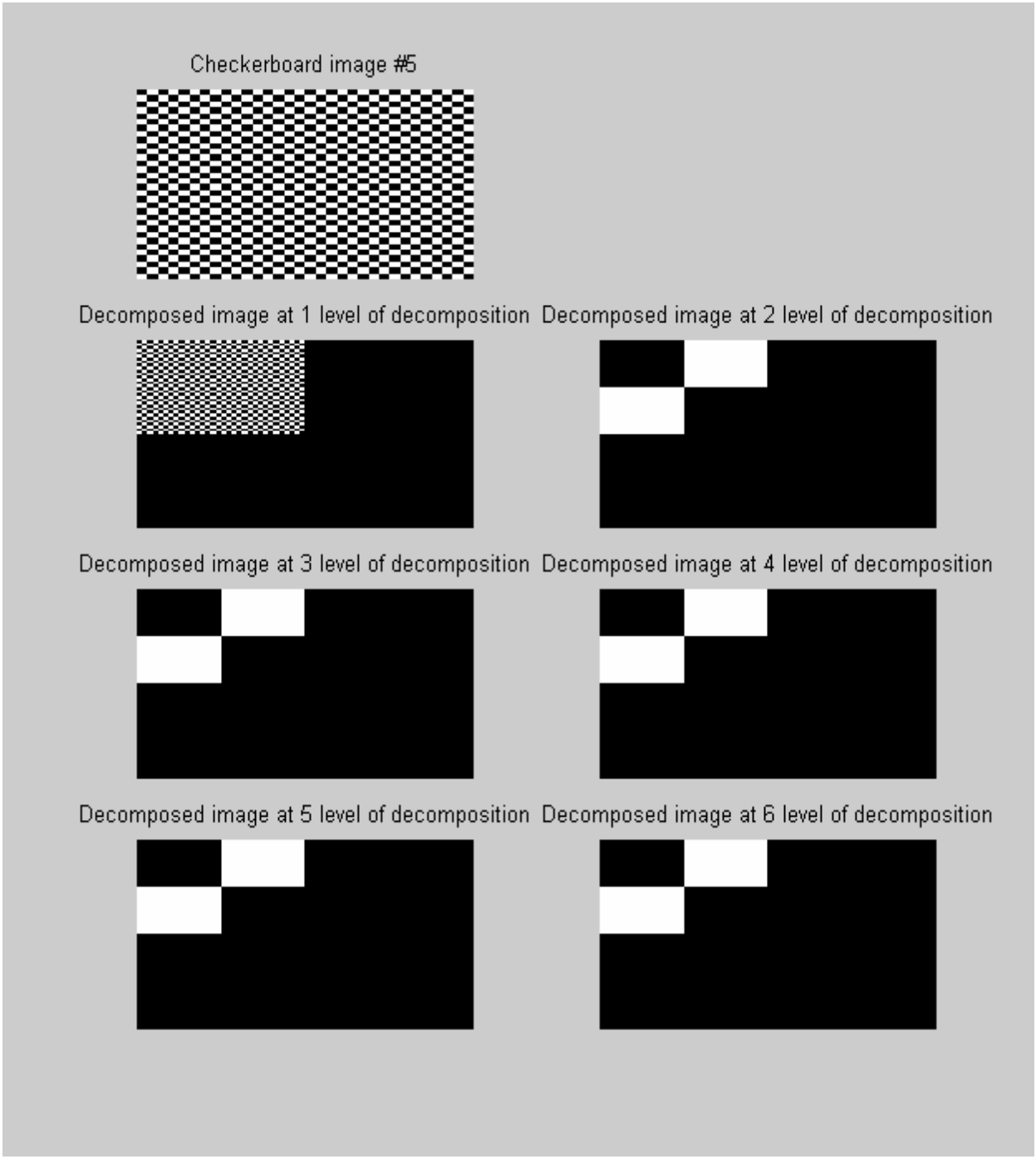

Figure E.4 The 6-level OS-HVDD(l_1) method for checkerboard image #5.

Table E.1 The optimum scalars of 6-level image decomposition using l_1 norm based OS-HVDD method to a set of 32 x 32 checkerboard images.

Checkerboard image	Level of decomposition	Optimum scalar		
		Horizontal orientation	Vertical orientation	Diagonal orientation
#1	1	1	1	1
	2	1	1	1
	3	1	1	1
	4	1	1	1
	5	1	1	1
	6	0	0	0
#2	1	1	1	1
	2	1	1	1
	3	1	1	1
	4	1	1	1
	5	0	0	0
	6	0	0	0
#3	1	1	1	1
	2	1	1	1
	3	1	1	1
	4	0	0	0
	5	0	0	0
	6	0	0	0
#4	1	1	1	1
	2	1	1	1
	3	0	0	0
	4	0	0	0
	5	0	0	0
	6	0	0	0
#5	1	1	1	1
	2	0	0	0
	3	0	0	0
	4	0	0	0
	5	0	0	0
	6	0	0	0
#6	1	0	0	0
	2	0	0	0
	3	0	0	0
	4	0	0	0
	5	0	0	0
	6	0	0	0

REFERENCES

[1] C.M. Brislawn and J.N. Bradley, "The FBI compression standard for digitized fingerprint images", *Proc. SPIE Signal and Image Processing*, vol. 2847, pp. 344-355, 1996.

[2] M.G. Albanesi, "Thresholding wavelets for image compression", *Proc. Compression and Complexity of Sequence*, pp. 374-389, June 1997.

[3] X. Wu, N. Memon, "CALIC – A Context based adaptive lossless image codec", *Proc. IEEE Int. Conf. Acoustics, Speech and Signal Processing*, vol. 4, pp. 1890-1893, 1996.

[4] V.K. Heer and H.-E. Reinfelder, "A comparison of reversible methods for data compression", *Proc. SPIE-Medical Imaging IV*, vol. 1233, pp. 354-365, 1990.

[5] S. Ranganath and H. Blume, "Hierarchical image decomposition and filtering using the S-transform", *Proc. SPIE Medical Imaging II*, vol. 914, pp. 799-814, 1988.

[6] A. Said and W.A. Pearlman, "An image multiresolution representation for lossless and lossy compression", *IEEE Trans. Image Processing*, vol. 5, pp. 1303-1310, 1996.

[7] G. Deng, "An interpolative subband coding algorithm for lossless compression", *Signal Processing: Image Communication*, vol. 14, pp. 721-736, 1999.

[8] M.J. Weinberger, G. Seroussi, and G. Shapiro, "LOCO-I : A low complexity, context-based, lossless image compression algorithm", *Proc. Data Compression Conference*, pp. 141-150, 1996.

[9] X. Wu, "An algorithmic study on lossless image compression", *Proc. Data Compression Conference*, pp. 150-159, 1996.

[10] G. Deng, H. Ye, and L.W. Cahill, "Adaptive combination of linear predictors for lossless image compression", *IEE Proc. Sciences Measurements Technologies*, vol. 147, pp. 414-419, Nov. 2000.

[11] T. Seemann and P.E. Tischer, "Generalized locally adaptive DPCM", *Technical Report No. 97/301*, Monash University, Australia, 1997.

[12] W.S. Lee, "Edge adaptive prediction for lossless image coding", *Proc Data Compression Conference*, pp. 483-490, 1999.

[13] W. Woods and S.D. O'Neil, "Subband coding of images", *IEEE Trans. Acoustics, Speech, Signal Processing*, vol. 43, pp. 1278-1288, Oct. 1986.

[14] S. Marusic and G. Deng, "New prediction schemes for lossless coding of fullband and subband images", *Signal Processing : Image Communication*, vol. 14, pp. 869-878, 1999.

[15] S. Weiss, M. Harteneck, and R.W. Stewart, "On implementation and design of filter banks for subband adaptive systems", *Proc. IEEE Workshop Signal Processing System*, Cambridge, MA, pp. 172-181, 1998.

[16] M. Harteneck, S. Weiss, and R.W. Stewart, "Design of near perfect reconstruction oversampled filter banks for subband adaptive filters", *IEEE Trans. Circuits System II*, vol. 46, pp. 1081-1085, Aug. 1999.

[17] S. Hosur and A.H. Tewfik, "Wavelet transform domain adaptive FIR filtering", *IEEE Trans. Signal Processing*, vol. 45, pp. 617-630, Mar. 1997.

[18] N. Erdol and F. Basbug, "Performance of wavelet transform based adaptive filters", *Proc. IEEE Int. Conf. Acoustics, Speech, and Signal Processing*, vol. 3, pp. 500-503, Apr. 1993.

[19] G. Strang, *Wavelets and Filter Banks*, Wellesley, MA : Wellesley-Cambridge Press, 1996.

[20] P.P. Vaidyanathan, *Multirate Systems and Filter Banks*, Englewood Cliffs, NJ : Prentice-Hall, 1983.

[21] O.N. Gerek and A.E. Cetin, "Adaptive polyphase subband decomposition structures for image compression", *IEEE Trans. Image Processing*, vol. 9, no. 10, pp. 1649-1660, 2000.

[22] R. Oktem, O.N. Gerek, and E. Cetin, L. Oktem, and K. Egiazarian, "Adaptive filter bank for lossless image compression", *Proc. IEEE Int. Conf. Acoustics, Speech, Signal Processing*, vol. 3, pp. 1809-1812, 2001.

[23] D. Taubman, "Adaptive non-separable lifting transform for image compression", *IEEE Int. Conf. Image Processing*, vol. 3, pp.772-776, 1999.

[24] F.J. Hampson and J.-C. Pesquet, "M-band nonlinear subband decompositions with perfect reconstruction", *IEEE Trans. Image Processing*, vol. 7, pp. 1547-1560, Nov. 1998.

[25] S. Agaian, D.S. Choi, and J. Noonan, "Image compression using fuzzy subband decomposition", *Proc. IEEE Int. Conf. Fuzzy Systems*, vol. 2, pp. 894-899, 2000.

[26] A.Benazza-Benyahia, J.-C. Pesquet, and M. Hamdi, "Lossless coding for progressive archival of multispectral images", *Proc. IEEE Int. Conf. Acoustics, Speech, and Signal Processing*, vol. 3, pp. 1817-1820, 2001.

[27] E.S. Cardoso Jr. and E.A.B. da Silva, “Non-separable non-linear decompositions with applications to image compression”, *Proc. IEEE Int. Conf. Image Processing*, vol. 2, pp. 387-390, 2000.

[28] M. Effros and P.A. Chou, “Weighted universal transform coding: universal image compression with the Karhunen-Loeve transform”, *Proc. IEEE Int. Conf. Image Processing*, vol. 2, pp. 61-64, 1995.

[29] Y. Yamashita, “Image compression by weighted Karhunen-Loeve transform”, *Proc. IEEE Int. Conf. Pattern Recognition*, vol. 2, pp. 636-640, 1996.

[30] J. Lee, “Optimized quadtree for Karhunen-Loeve transform in multispectral image coding”, *IEEE Trans. Image Processing*, vol. 8, no. 4, pp. 453-461, Apr. 1999.

[31] H. Feng and M. Effros, “On the rate-distortion performance and computational efficiency of the Karhunen-Loeve transform for lossy data compression”, *IEEE Trans. Image Processing*, vol. 11, no. 2, pp. 113-122, Feb. 2002.

[32] A. Baskurt and R. Goutte, “Encoding the location of spectral coefficients using quadtrees in transform image compression”, *Proc. IEEE Int. Conf. Acoustics, Speech, and Signal Processing*, vol. 3, pp. 1842-1845, 1989.

[33] D.M. Monro and G.J. Dickson, “Zerotree coding of DCT coefficients”, *Proc. IEEE Int. Conf. Image Processing*, vol. 2, pp. 625-628, 1997.

[34] W. Philips, “The lossless DCT for combined lossy/lossless image coding”, *Proc. IEEE Int. Conf. Image Processing*, vol. 3, pp. 871-875, 1998.

[35] N. I. Cho and S.K. Mitra, “An image compression algorithm using warped discrete cosine transform”, *Proc. IEEE Int. Conf. Image Processing*, vol. 2, pp. 834-837, 1999.

[36] M Antonini, M. Barlaud, P. Mathieu, and I. Daubechies, “Image coding using wavelet transform”, *IEEE Trans. Image Processing*, vol. 1, no. 2, pp. 205-220, Apr. 1992.

[37] A.S. Lewis and G. Knowles, “Image compression using the 2-D wavelet transform”, *IEEE Trans. Image Processing*, vol. 1, no. 2, pp. 244-250, Apr. 1992.

[38] J.M. Shapiro, “An embedded hierarchical image coder using zerotrees of wavelet coefficients”, *Data Compression Conference*, pp. 214-223, 1993.

[39] O. Rioul, “On the choice of 'wavelet' filter for still image compression”, *Proc. IEEE Int. Conf. Acoustics, Speech, and Signal Processing*, vol. 5, pp. 550-553, 1993.

[40] J.N. Bradley and C.M. Brislawn, "The wavelet/scalar quantization compression standard for digital fingerprint images", *Proc. IEEE Int. Symp. Circuits and Systems*, vol. 3, pp. 205-208, June 1994.

[41] A. Manduca, "Compressing images with wavelet/subband coding", *IEEE Engineering in Medicine and Biology Magazine*, vol. 14, no. 5, pp. 639-646, 1995.

[42] A.R. Calderbank, I. Daubechies, W. Sweldens, and Boon-Lock Yeo, "Lossless image compression using integer to integer wavelet transforms", *Proc. IEEE Int. Conf. Image Processing*, vol. 1, pp. 596-599, 1997.

[43] R.H.G. Tan, T.J.F. Zhang, R. Morgan, and A. Greenwood, "Still image compression based on 2D discrete wavelet transform", *Electronics Letters*, vol. 35, no. 22, pp. 1934-1935, Oct. 1999.

[44] L. Oktem, R. Oktem, K. Eguiazarian, and J. Astola, "Efficient encoding of the significance maps in wavelet based image compression", *Proc. IEEE Int. Symp. Circuits and Systems*, vol. 3, pp. 25-28, 2000.

[45] D. Marpe, G. Blattermann, J. Ricke, and P. Maab, "A two-layered wavelet-based algorithm for efficient lossless and lossy image compression", *IEEE Trans. Circuits and Systems for Video Technology*, vol. 10, no. 7, pp. 1094-1102, Oct. 2000.

[46] S. Saha and R. Vemuri, "Analysis-based adaptive wavelet filter selection in lossy image coding schemes", *Proc. IEEE Int. Symp. Circuits and Systems*, vol. 3, pp. 29-32, 2000.

[47] S. Yammen, *A data dependent linear decomposition transform*, Ph.D. Dissertation, Vanderbilt University, Nashville, 2001.

[48] A. Grossmann and J. Morlet, "Decomposition of hardy functions into square integrable wavelets of constant shape", *SIAM J. Math.*, vol. 15, pp. 723-736, 1984.

[49] C.S. Burrus, R.A. Gopinath, and H. Guo, *Introduction to Wavelets and Wavelet Transform A Primer*, Upper Saddle River, NJ : Prentice-Hall, 1998.

[50] Y. Meyer, *Wavelets, Algorithms and Applications*, SIAM, Philadelphia, 1993. Translated by R.D. Ryan based on lectures given for the Spanish Institute in Madrid, Feb. 1991.

[51] S.G. Mallat, "Multiresolution approximation and wavelet orthogonal bases of $L^{2"}$, *Trans. American Mathematical Society*, vol. 315, pp. 69-87, 1989.

[52] S.G. Mallat, "A theory for multiresolution signal decomposition : the wavelet representation", *IEEE Trans. Pattern Recognition and Machine Intelligence*, vol. 17, no. 7, pp. 674-693, July 1989.

[53] I. Daubechies, *Ten lectures on wavelets*, SIAM, Philadelphia, 1993. Notes from the 1990 CBMS-NSF Conference on Wavelets and Applications at Lowell, MA.

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

[55] The MathWorks, Inc., *MATLAB: The Language of Technical Computing Version 6.1.0.450 Release 12.1*, Instructional Computing Lab of Vanderbilt University School of Engineering, May 18, 2001.

[56] P. Roos, M.A. Viergever, M.C.A. Van Dijke, and J.H. Peters, "Reversible interframe compression of medical images", *IEEE TransMedical Imaging*, vol. 7, pp 328-336, 1989.

[57] D. Houlding and J. Vaisey, "Low entropy image pyramids for efficient lossless coding", *IEEE Trans. Image Processing*, vol. 4, pp 1150-1153, 1995.

[58] N. Memon, V. Sippy, and Xiaolin Wu, "A comparison of prediction schemes proposed for a new lossless image compression standard", *Proc. IEEE Int. Symp. Circuits and Systems*, vol. 2, pp 309-312, 1996.

[59] N.V. Boulgouris, D. Tzovaras, and M.G. Strintzis, "Lossless image compression based on optimal prediction, adaptive lifting, and conditional arithmetic coding", *IEEE Trans. Image Processing*, vol. 10, pp 1-14, Jan. 2001.

[60] X.M. Lu, J.M. Reid, and S.H. Wang, "A practical application of L1 deconvolution for medical ultrasound", *Proc. IEEE Ultrasonics Symposium*, vol. 2, pp. 1177-1180, 1991.

[61] J. Xin and N.M. Bilgutay, "Ultrasonic range resolution enhancement using L1 norm deconvolution", *Proc. IEEE Ultrasonics Symposium*, vol. 2, pp. 711-714, 1993.

[62] S.R. Zhao and H. Halling, "Minimum L1 norm MEG reconstruction minimizing signal deviation reduced lead field", *Proc. IEEE Int. Conf. Engineering in Medicine and Biology Society*, vol. 2, pp. 814-816, 1996.

[63] A. Megretski, , "Combining L1 and L2 methods in the robust stability and performance analysis of nonlinear systems", *Proc. IEEE Conf. Decision and Control*, vol. 3, pp. 3176-3181, 1995.

[64] S. Umeyama, "Least-squares estimation of transformation parameters between two point patterns", *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 13, pp. 376-380, Apr.1991.

[65] H. Ochi, Y. Higa, and S. Kinjo, "A subband adaptive filter with the optimum analysis filter bank", *Int. Conf. Acoustics, Speech, and Signal Processing*, vol. 2, pp. 993-996, 1995.

[66] A. Reko, V.K. Jones, and B. Parkinson, "LMS identification of systems with dynamics and an output deadzone", *Proc. American Control Conference*, vol. 5, pp. 2770-2774, 1998.

[67] S. Marusic and G. Deng, "New Prediction Schemes for Lossless Coding of Fullband and Subband Images", *Signal Processing: Image Communication*, vol. 14, pp. 869-878, 1999.

[68] P. Bloomfield and W.L. Steiger, "Least Absolute Deviations Curve-fitting", *SIAM Journal of Scientific and Statistical Computing*, 1, pp. 290-301, 1980.

[69] P. Bloomfiled and W.L. Steiger, *Least Absolute Deviations, Theory, Applications and Algorithms*, Boston, MA: Birkhauser, 1983.

[70] R.J. Schilling and H. Lee, *Engineering Analysis a Vector Space Approach*, New York: John Wiley & Sons, Inc., 1988.

[71] D.G. Luenberger, *Optimization by Vector Space Methods*, New York, NY: John Wiley & Sons, Inc., 1969.

[72] J.A. Cadzow, *Application of The l_1 Norm in Signal Processing*, Department of Electrical Engineering, Vanderbilt University, Nashville, 1998.

[73] J.A. Cadzow, *Minimum Norm Approximate Solution : System of Linear Equations*, Department of Electrical Engineering, Vanderbilt University, Nashville, 2000.

[74] I. Barrodale and F.D.K. Roberts, "An Improved Algorithm for Discrete l_1 Linear Approximation", *SIAM Journal of Numerical Analysis*, vol. 10, pp. 839-848, 1973.

[75] R.H. Bartels, A.R. Conn, and J.W. Sinclair, "Minimization Techniques for Piecewise Differential Functions : the l_1 Solution to an Overdetermined System", *SIAM Journal of Numerical Analysis*, vol. 15, pp. 224-241, 1978.

[76] J.M. Shapiro, "Embedded image coding using zerotrees of wavelets coefficients", *IEEE Trans. Signal Processing*, vol. 41, pp. 3445-3462, Dec. 1993.

[77] A. Said and W.A. Pearlman, "Image compression using the spatial orientation tree", *IEEE Int. Symp. Circuits and Systems*, Chicago, IL, pp. 279-282, May 1993.

[78] K. Sayood, *Introduction to Data Compression*, San Francisco, CA: Morgan Kaufmann Publishers, 2000.

[79] A.V. Oppenheim and J.S. Lim, "The importance of phase in signals", *IEEE Proceedings*, vol. 69, pp. 529-541, May. 1981.

[80] F.M. Wahl, *Digital image signal processing*, Norwood, MA: Artech House, Inc., 1987.

[81] T.S. Huang, J.W. Burnett and A.G. Deczky, "The importance of phase in image processing filters", *IEEE Trans. Acoustics, Speech, and Signal Processing*, vol. 23, pp. 529-542, Dec. 1975.

[82] W.A. Pearlman and R.M. Gray, "Source coding of the discrete Fourier transform", *IEEE Trans. Information Theory*, vol. 24, pp. 683-692, Nov. 1978.

[83] W.K. Pratt, *Digital image processing*, Los Altos, CA: John Wiley & sons, Inc., 2001.

[84] L. Wang, *A study of the one-dimensional linear decomposition transform*, Master thesis, Vanderbilt university, Nashville, 2002.