PROPERTIES AND APPLICATIONS

OF THE

SHORT-SPACE FOURIER TRANSFORM

by

JEFFREY GORDON BERNSTEIN

B.S.E.E. University of Maryland
(1982)

Submitted to the Department of Electrical
Engineering and Computer Science
in Partial Fulfillment of the
Requirements of the Degree of

MASTER OF SCIENCE

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

May 1984

Signature of Author: _____
Department of Electrical Engineering and Computer Science
May 25, 1984

Certified by: _____
David H. Staelin, Thesis Supervisor

Accepted by: _____
Chairman, Departmental Committee on Graduate Students

PROPERTIES AND APPLICATIONS OF THE SHORT-SPACE FOURIER TRANSFORM

by

JEFFREY GORDON BERNSTEIN

ABSTRACT

In many applications, such as image coding and image restoration, localized spectral information is desired from multidimensional signals. One common approach is to divide the signal into rectangular blocks and perform a discrete Fourier or similar transform over each block. A problem with this approach is that discontinuities at the block boundaries are often visible after processing. One way to avoid this type of problem is to employ the short-time Fourier transform (STFT) using a smoothly decreasing window function. The STFT, however, is not well suited for signals of finite extent, such as images. In this thesis, a multi-dimensional extension of the STFT is introduced which is specifically suited for finite extent signals. This is referred to as the short-space Fourier transform (SSFT). Several important properties of this new transform are discussed, and a fast algorithm for its computation is presented. In addition, its applicability to image coding is discussed and demonstrated for both intraframe and interframe coding.

Thesis Supervisor: David H. Staelin
Professor of Electrical Engineering

ACKNOWLEDGEMENTS

I would like to express my sincere appreciation to Prof. David H. Staelin for his guidance and support during my stay at M.I.T. I also wish to thank Brian Hinman for our joint effort in much of the work presented in this thesis. In addition, I would like to thank Ali Ali for his help in producing the photographs used in the thesis. Finally, I wish to thank my family for their support during the past years.

CONTENTS

LIST OF FIGURES

1. Introduction

In image processing and many other applications, the multidimensional Fourier transform plays an important role. The Fourier transform of a signal represents its global spectral characteristics. In many cases, however, it is desirable to obtain localized spectral information. In the case of image processing, one method that is often used is to divide the image into small rectangular blocks, and perform discrete Fourier transforms independently over each block. There are several problems in taking this approach, however. Discontinuities at the block boundaries can cause extraneous spectral energy which does not truly represent the signal characteristics. This reduces its usefulness in image coding and spectral estimation applications. For image coding, this problem is somewhat avoided by using discrete cosine transforms (DCT) rather than Fourier transforms [1]. A second problem with any type of block transform in image processing applications is that, after processing, the block boundaries often appear as visible discontinuities. These are known as blocking effects.

The short-time Fourier transform (STFT) has long been used as a method of providing localized spectral information of one-dimensional, infinite extent signals [2][3][23]. With the proper choice of window function, a multidimensional extension of the STFT could be used in many of the applications discussed above, avoiding the problems of block Fourier transforms. Unfortunately, the STFT has a number of

problems when applied to signals of finite extent. In this thesis, a multidimensional extension of the STFT is introduced, referred to as the short-space Fourier transform (SSFT), which avoids these problems [13]. The specific application of image coding is used to demonstrate the effectiveness of this new transform method.

This thesis begins by reviewing the basic concepts of the short-time Fourier transform. Following this review, the short-space Fourier transform is defined and a number of its mathematical properties are presented. Next, an efficient algcrithm for computing the SSFT is presented. Finally, the application of the SSFT to both intraframe and interframe image coding is discussed and demonstrated.

2. Review of Short-Time Fourier Analysis

Because the short-space Fourier transform, as discussed in this thesis, is related to the short-time Fourier transform, it is necessary to briefly review the basic concepts of short-time Fourier analysis. This will include definitions of the transform, its interpretations, and a discussion of implementation techniques.

The short-time Fourier transform (STFT) of a discretely sampled signal, x, has been defined by the equation [2][3][23][25]:

$$X(n,\omega) = \sum_{m=-\infty}^{\infty} x(m)\, w(n-m)\, \exp\left[-j\omega m\right] \qquad (2.1)$$

where w is a localizing window function. The purpose of the STFT is to provide a simultaneous time and frequency representation of a signal. As such, the resulting function $X(n,\omega)$ is a two-dimensional function whose first argument, n, is a time index, and whose second argument, ω, corresponds to angular frequency. The ordinary Fourier transform uses the entire function in order to provide a spectral representation. In order to simultaneously provide a temporal representation, it is desired to have the spectrum corresponding to a particular time, n, represent the signal only in the local area of that position. The STFT accomplishes this by the use of the localizing window function, w. A particular region of the signal, centered at a position given by n, is isolated by

multiplying the signal by w, and performing an ordinary Fourier transform on the result.

It is important to note the limitation on spectral and temporal resolution that is inherent in short-time Fourier analysis. Since the window function multiplies the signal in the time domain, it equivalently convolves the signal in the frequency domain. This smearing effect limits the resolvability of features in the frequency dimension of the STFT. By broadening the window function in the time domain, the smearing will be reduced, thereby improving resolution. However, a broad window function will reduce the resolvability of rapid variations of the signal characteristics in the time domain. Thus, there is an inherent trade-off of spectral versus temporal resolution that is controlled by the choice of window function.

The STFT can be interpreted as a series of Fourier transforms of a signal multiplied by a localizing window. It has proven useful in the past to introduce a second interpretation of the STFT. It has been noticed that (2.1) can be viewed as a convolution between the window function and the signal, x, modulated by an exponential of frequency ω. As such, the STFT can be interpreted as a bank of filters. For each frequency, ω, the window is convolved with the signal centered over that frequency. If w is lowpass in nature, this corresponds to a bank of bandpass filters. In the frequency domain, this can be seen as the dual of the first interpretation. That is, the window spectrum isolates a

region of the signal spectrum centered at the frequency ω, before an inverse Fourier transform is taken on the result.

The use of the STFT is generally referred to as short-time Fourier analysis. From $X(n,\omega)$, it is possible to exactly reconstruct the original signal, $x(n)$. That is, the STFT is an invertible transformation. This inversion is generally referred to as short-time Fourier synthesis. The inverse STFT, in its most general form, has been found to be [23]:

$$x(n) = \frac{1}{2\pi} \int_{-\pi}^{\pi} \sum_{m=-\infty}^{\infty} X(m,\omega)\, f(n-m)\, \exp[j\omega n]\, d\omega \qquad (2.2)$$

The function, f, which appears in this expression is referred to as the the synthesis window. In order for the inverse to exactly reproduce the original function, the synthesis window must be related to w, the analysis window, by the following constraint:

$$\sum_{m=-\infty}^{\infty} f(-m)\, w(m) = 1 \qquad (2.3)$$

In order to implement the STFT, the frequency variable, ω, must be discretized. The STFT and its inverse can be rewritten to account for this discretization. Due to the lack of resolvability of the STFT between each temporal sample, it is common to decimate the STFT in the time

dimension as well. If we take M samples of the STFT in frequency between $-\pi$ and π, then we can substitute $\omega = \Omega k$ where $\Omega = 2\pi/M$. Further, if the STFT is decimated by a factor R in the temporal dimension, then the STFT can be rewritten as:

$$X(Rn,\Omega k) = \sum_{m=-\infty}^{\infty} x(m)\, w(Rn-m)\, \exp[-jk\Omega m] \qquad (2.4)$$

It has been shown that it is still possible to uniquely invert the STFT as long as $R \leq M$. That is, the number of samples skipped in the time dimension must be less than the total number of samples in the frequency dimension. If this condition is met, the synthesis equation can be expressed as:

$$x(n) = \frac{1}{M} \sum_{k=0}^{M-1} \sum_{m=-\infty}^{\infty} X(Rm,\Omega k)\, f(n-Rm)\, \exp[\, jk\Omega n\,] \qquad (2.5)$$

In this case the synthesis window must be related to the analysis window by the constraint:

$$\sum_{m=-\infty}^{\infty} f(n-Rm)\, w(Rm-n+Mp) = \delta(p) \quad \forall n \qquad (2.6)$$

When the temporal decimation ratio, R, is exactly equal to the number of frequency samples, M, the STFT is referred to as being critically sampled. That is, the STFT is

represented with the minimum possible number of samples. Unlike the general case where many synthesis windows may be used to reconstruct the signal, in the critically sampled case, only one synthesis window can be used for a given analysis window.

Analogous to the resolution trade-off discussed earlier, there is also a trade-off between the maximum allowable temporal and spectral decimation. For the case where R = M, the temporal decimation ratio, R, and the spectral decimation ratio, Ω, are related by $R\Omega = 2\pi$. Thus, in general, $R\Omega \leq 2\pi$.

For the discretely sampled STFT, it has been shown that fast Fourier transform algorithms can be used to provide an efficient implementation technique [23]. This can be demonstrated by the following analysis. By decomposing the summation in equation (2.4) such that m = Ma + b, it can be rewritten as:

$$X(Rn,\Omega k) = \sum_{a=-\infty}^{\infty} \sum_{b=0}^{M-1} x(Ma+b)\, w(Rn-Ma-b) \cdot \exp[-jk\Omega(Ma+b)] \tag{2.7}$$

Since $\Omega M = 2\pi$, we can remove the dependence of the exponential on a. By defining a new function, z, we can express the STFT as:

$$X(Rn,\Omega k) = \sum_{b=0}^{M-1} z(n,b)\ \exp[-jk\Omega b] \qquad (2.8)$$

where

$$z(n,b) = \sum_{a=-\infty}^{\infty} x(Ma+b)\ w(Rn-Ma-b) \quad 0\le b\le M-1 \qquad (2.9)$$

From this it can be seen that for each n, X can be expressed as a discrete Fourier transform, of length M, of the function z(n,b) in the variable b. The function z corresponds to x multiplied by the window function, and then time aliased in sections of length M. If w is a finite length window then this can be computed directly since the summation in (2.9) need not extend past the non-zero portion of w.

For the special case of the critically sampled STFT, it has been shown that z can be computed more efficiently than by direct implementation of equation (2.9) [5][7][22]. If we let M = R, then (2.9) can be rewritten as:

$$z(n,b) = \sum_{a=-\infty}^{\infty} x(Ma+b)\ w(M(n-a)-b) \quad 0\le b\le M-1 \qquad (2.10)$$

If we define $x_b(a) = x(Ma+b)$ and $w_b(a) = w(Ma+b)$, then equation (2.10) can be again rewritten in the following simple form:

$$z(n,b) = \sum_{a=-\infty}^{\infty} x_b(a)\, w_b(n-a) \qquad (2.11)$$

$$z(n,b) = x_b(n) * w_b(n) \qquad (2.12)$$

If we assume that w is of finite extent, then z(n,b) can be computed using the FFT to implement these convolutions by the overlap-add or overlap-save techniques. This implementation of the critically sampled STFT is known as the polyphase filtering implementation. This is because each decimated phase of the signal, $x_b(n)$, for successive values of b, is convolved with a different filter function $w_b(n)$, forming a polyphase filter network.

3. The Short-Space Fourier Transform

The short-time Fourier transform has been defined for one-dimensional, infinite extent signals. It is very straightforward to extend the STFT to account for multidimensional signals. Let $x(\underline{n})$ be a rectangularly sampled function of the D-dimensional vector, $\underline{n}$, over the region $D_\infty = (\underline{n}: \ -\infty \le n_i \le \infty \ \text{ for } i=1,2,\ldots,D)$. Let R and Ω be sampling matrices defined as:

$$R = \begin{bmatrix} R_1 & & 0 \\ & R_2 & \\ & & \ddots \\ 0 & & R_D \end{bmatrix} \qquad \Omega = \begin{bmatrix} \Omega_1 & & 0 \\ & \Omega_2 & \\ & & \ddots \\ 0 & & \Omega_D \end{bmatrix} \tag{3.1}$$

The multidimensional extension of the STFT can then be written as:

$$X(R\underline{n}, \Omega\underline{k}) = \sum_{\underline{m} \in D_\infty} x(\underline{m})\, w(R\underline{n} - \underline{m}) \exp\left[-j\underline{k}^T \Omega \underline{m}\right] \tag{3.2}$$

In general, practical multidimensional signals, and many one dimensional signals, cannot be considered to be of infinite extent. If we attempt to use (3.2) for finite extent signals, for example, by zero padding outside the region of support, two problems immediately arise. First, depending upon the extent of the window function, X may be non-zero well beyond the original region of support of x in the spatial

dimension. Second, the discontinuities that may appear at the boundaries of the signal due to zero padding would result in spectra that do not necessarily reflect the true characteristics of the signal within these boundaries.

The simplest approach to avoid the first problem is to periodically extend x, rather than zero padding. This is analogous to the use of the Fourier series for finite extent signals. We can define $\hat{x}(\underline{n})$ such that $\hat{x}(\underline{n}) = x(\underline{n}+N\underline{i})$ for all integer vectors, $\underline{i}$, where N is a periodicity matrix, defined by the region of support of x. The matrix N takes the form:

$$N = \begin{bmatrix} N_1 & & 0 \\ & N_2 & \\ & \ddots & \\ 0 & & N_D \end{bmatrix} \tag{3.3}$$

Replacing x with $\hat{x}$ in equation (3.2), we obtain:

$$X(R\underline{n}, \Omega\underline{k}) = \sum_{\underline{m} \in D_\infty} \hat{x}(\underline{m})\ w(R\underline{n} - \underline{m})\ \exp\left[-j\underline{k}^T \Omega \underline{m}\right] \tag{3.4}$$

Using this definition, X is periodic in $\underline{n}$, and thus can be uniquely determined from knowledge of only one period. Thus, even if w is of infinite extent, X need only be known over the region $R\underline{n} \in D_N$.

Periodic extension does indeed solve the problem of spatial extent, but it does not necessarily eliminate boundary

discontinuities. To avoid these discontinuities, instead of periodically extending the signal, we can reflectively extend it. The signal boundaries are then continuous and the resulting spectra will more closely represent the characteristics of the signal. Such a technique is analogous to the use of a discrete cosine transform which can be defined in terms of the Fourier transform of a reflectively extended signal block. From this point on, this definition of X will be referred to as the short-space Fourier transform (SSFT) [13]. Let us define the function $\tilde{x}$ which is x reflected in all D dimensions:

$$\tilde{x}(\underline{n}) = x(\underline{J}(\underline{n}))$$

$$J_i(\underline{n}) = J_i(n_i) = \begin{cases} n_i & 0 \le n_i \le N_i - 1 \\ 2N_i - 1 - n_i & N_i \le n_i \le 2N_i - 1 \end{cases} \quad (3.5)$$

This defines the signal $\tilde{x}$ over the region given by $D_{2N} = (n: 0 \le n_i \le 2N \text{ for } i=1,2,\ldots,D)$. Outside of this region, $\tilde{x}$ can be defined as being periodically extended with periodicity matrix 2N. Using $\tilde{x}$ rather than $\hat{x}$, equation (3.4) can be rewritten to obtain:

$$X(R\underline{n}, \Omega\underline{k}) = \sum_{\underline{m} \in D_\infty} \tilde{x}(\underline{m})\, w(R\underline{n} - \underline{m}) \exp\left[-j\underline{k}^T \Omega \underline{m}\right] \quad (3.6)$$

Since $\tilde{x}$ is periodic, it is possible to rewrite equation (3.6) in such a way that infinite summations are not needed to compute the SSFT. The summation in $\underline{m}$ can be decomposed by defining $\underline{m} = \underline{a} + 2N\underline{b}$.

$$X(R\underline{n},\Omega\underline{k}) = \sum_{\underline{a}\in D_{2N}} \sum_{\underline{b}\in D_{\infty}} \tilde{x}(\underline{a}+2N\underline{b})\, w(R\underline{n}-\underline{a}-2N\underline{b}) \cdot \exp\left[-j\underline{k}^T\Omega(\underline{a}+2N\underline{b})\right] \tag{3.7}$$

Since $\tilde{x}$ is periodic with periodicity matrix 2N, then $x(\underline{a}+2N\underline{b}) = x(\underline{a})$. We can define the matrix M, corresponding to the number of samples in each frequency dimension of X, such that $\Omega = 2\pi M^{-1}$. If M and N are related by the equation ML = N, and 2L forms an integer matrix, then the term in the exponent, $2N\Omega$, can be eliminated. Thus, we can further simplify this expression to be of the form:

$$X(R\underline{n},\Omega\underline{k}) = \sum_{\underline{a}\in D_{2N}} \tilde{x}(\underline{a}) \exp\left[-j\underline{k}^T\Omega\underline{a}\right] \cdot \sum_{\underline{b}\in D_{\infty}} w(R\underline{n}-\underline{a}-2N\underline{b}) \tag{3.8}$$

We can now define a new function, $\tilde{w}$, which is the window function spatially aliased over blocks of size 2N:

$$\tilde{w}(\underline{n}) = \sum_{\underline{b}\in D_\infty} w(\underline{n} - 2N\underline{b}) \tag{3.9}$$

Using the aliased window function rather than w itself, equation (3.8) can be rewritten in the final form:

$$X(R\underline{n}, \Omega\underline{k}) = \sum_{\underline{a}\in D_{2N}} \tilde{x}(\underline{a})\ \tilde{w}(R\underline{n} - \underline{a})\ \exp\left[-j\underline{k}^T\Omega\underline{a}\right] \tag{3.10}$$

Just as for the discretely sampled STFT, the SSFT has a simple formulation for resynthesis of the signal x from the SSFT coefficients. Since equation (3.6) represents the multidimensional STFT operating on the function $\tilde{x}$, we can use the direct multidimensional extension of the STFT synthesis equations, (2.5) and (2.6), to write the SSFT synthesis in the form:

$$\tilde{x}(\underline{n}) = \frac{1}{|M|} \sum_{\underline{k}\in D_M} \sum_{\underline{m}\in D_\infty} f(\underline{n} - R\underline{m})\, X(R\underline{m}, \Omega\underline{k})\, \exp\left[j\underline{k}^T\Omega\underline{n}\right] \tag{3.11}$$

$$\sum_{\underline{m}\in D_\infty} f(\underline{n} - R\underline{m})\ w(R\underline{m} - \underline{n} + M\underline{p}) = \delta(\underline{p}) \qquad \forall\underline{n} \tag{3.12}$$

Using the same approach as before, we can simplify these expressions and avoid the infinite summations by letting

$\underline{m} = \underline{a} + Q\underline{b}$, where Q is defined by the relation $QR = 2N$.

$$\tilde{x}(\underline{n}) = \frac{1}{|M|} \sum_{\underline{k}\in D_M} \sum_{\underline{a}\in D_Q} \sum_{\underline{b}\in D_\infty} f(\underline{n} - R(\underline{a} + Q\underline{b})) \cdot X(R(\underline{a} + Q\underline{b}), \Omega\underline{k}) \exp[\, j\underline{k}^T \Omega \underline{n}\,] \tag{3.13}$$

Noting that X is periodic in the spatial dimensions with period 2N, we can simplify this to:

$$\tilde{x}(\underline{n}) = \frac{1}{|M|} \sum_{\underline{k}\in D_M} \sum_{\underline{a}\in D_Q} X(R\underline{a}, \Omega\underline{k}) \exp[\, j\underline{k}^T \Omega \underline{n}\,] \cdot \sum_{\underline{b}\in D_\infty} f(\underline{n} - R\underline{a} - 2N\underline{b}) \tag{3.14}$$

Finally, we can define $\tilde{f}$ such that the inverse can be written:

$$\tilde{x}(\underline{n}) = \frac{1}{|M|} \sum_{\underline{k}\in D_M} \sum_{\underline{a}\in D_Q} X(R\underline{a}, \Omega\underline{k})\, \tilde{f}(\underline{n} - R\underline{a}) \exp[\, j\underline{k}^T \Omega \underline{n}\,] \tag{3.15}$$

where,

$$\tilde{f}(\underline{n}) = \sum_{\underline{b}\in D_\infty} f(\underline{n} - 2N\underline{b}) \tag{3.16}$$

The constraint on $\tilde{f}$ can be found by substituting the SSFT definition, equation (3.10), into the synthesis equation:

$$\tilde{x}(\underline{n}) = \frac{1}{|M|} \sum_{\underline{k}\in D_M} \sum_{\underline{a}\in D_Q} \sum_{\underline{m}\in D_{2N}} \tilde{x}(\underline{m})\, \tilde{w}(R\underline{a}-\underline{m}) \cdot \exp\left[-j\underline{k}^T\Omega\underline{m}\right] \tilde{f}(\underline{n}-R\underline{a})\, \exp\left[j\underline{k}^T\Omega\,\underline{n}\right] \tag{3.17}$$

Making the substitution $\underline{m} = M\underline{r} + \underline{s}$, and noting that $\Omega_M = 2\pi I$, yields:

$$\tilde{x}(\underline{n}) = \frac{1}{|M|} \sum_{\underline{a}\in D_Q} \sum_{\underline{r}\in D_{2L}} \sum_{\underline{s}\in D_M} \tilde{x}(M\underline{r}+\underline{s})\, \tilde{w}(R\underline{a}-M\underline{r}-\underline{s}) \cdot \tilde{f}(\underline{n}-R\underline{a}) \sum_{\underline{k}\in D_M} \exp\left[-j\underline{k}^T\Omega\underline{s}\right]\, \exp\left[j\underline{k}^T\Omega\,\underline{n}\right] \tag{3.18}$$

$$\tilde{x}(\underline{n}) = \sum_{\underline{r}\in D_{2L}} \sum_{\underline{s}\in D_M} \tilde{x}(M\underline{r}+\underline{s}) \sum_{\underline{a}\in D_Q} \tilde{w}(R\underline{a}-M\underline{r}-\underline{s}) \cdot \tilde{f}(\underline{n}-R\underline{a}) \sum_{\underline{i}\in D_\infty} \delta(\underline{n}-\underline{s}-M\underline{i}) \tag{3.19}$$

$$\tilde{x}(\underline{n}) = \sum_{\underline{r}\in D_{2L}} \tilde{x}(M\underline{r}+\underline{n}) \sum_{\underline{a}\in D_Q} \tilde{w}(R\underline{a}-M\underline{r}-n)\; \tilde{f}(\underline{n}-R\underline{a}) \tag{3.20}$$

Thus, for exact reconstruction of the original signal, $\tilde{f}$ must be related to $\tilde{w}$ by the constraint:

$$\sum_{\underline{q}\in D_Q} \tilde{w}(R\underline{q} - M\underline{r} - \underline{n})\, \tilde{f}(\underline{n} - R\underline{q}) = \delta(\underline{r}) \quad \forall \underline{n} \quad \underline{r}\in D_{2L} \tag{3.21}$$

4. Properites of the Short-Space Fourier Transform

4.1 Space-Frequency Duality

As discussed for the case of the short-time Fourier transform, there are dual interpretations: that of a windowed set of Fourier transforms and that of a bank of filters. Similarly, the short-space Fourier transform also has dual interpretations. The first is clear from the definition: windowed Fourier transforms of the reflectively extended signal. The exact form of the dual interpretation will be the topic of this section.

For the case of the STFT, the filter-bank interpretation could be viewed as taking a series of inverse Fourier transforms of windowed, or filtered, versions of the signal spectrum. To determine if a similar relationship exists for the SSFT, we can rewrite (3.10) as an inverse Fourier transform:

$$X(R\underline{n},\Omega\underline{k}) = \frac{1}{|Q|}\sum_{\underline{h}\in D_Q}\exp\left[j\underline{n}^T\Gamma\underline{h}\right]\sum_{\underline{m}\in D_Q}\exp\left[-j\underline{h}^T\Gamma\underline{m}\right] \cdot\sum_{\underline{a}\in D_{2N}}\tilde{x}(\underline{a})\,\tilde{w}(R\underline{m}-\underline{a})\,\exp\left[-j\underline{k}^T\Omega\underline{a}\right] \tag{4.1}$$

where $QR = 2N$, $\Gamma = 2\pi Q^{-1}$ and $|Q|$ is the determinant of Q. This can be rearranged into the following form:

$$X(R\underline{n},\Omega\underline{k}) = \frac{1}{|Q|}\sum_{\underline{h}\in D_Q}\exp\left[j\underline{n}^T\Gamma\underline{h}\right]\sum_{\underline{a}\in D_{2N}}\tilde{x}(\underline{a})\exp\left[-j\underline{k}^T\Omega\underline{a}\right]$$
$$\cdot\sum_{\underline{m}\in D_Q}\exp\left[-j\underline{h}^T\Gamma\underline{m}\right]\tilde{w}(R\underline{m}-\underline{a}) \qquad (4.2)$$

Expressing $\tilde{w}$ in terms of its discrete Fourier transform over D_{2N}, written as $\tilde{W}_{2N}$:

$$X(R\underline{n},\Omega\underline{k}) = \frac{1}{|Q|}\sum_{\underline{h}\in D_Q}\exp\left[j\underline{n}^T\Gamma\underline{h}\right]\sum_{\underline{a}\in D_{2N}}\tilde{x}(\underline{a})\exp\left[-j\underline{k}^T\Omega\underline{a}\right]$$
$$\cdot\sum_{\underline{m}\in D_Q}\exp\left[-j\underline{h}^T\Gamma\underline{m}\right]\frac{1}{2|N|}\sum_{\underline{s}\in D_{2N}}\tilde{W}_{2N}(\underline{s}) \qquad (4.3)$$
$$\cdot\exp\left[j\underline{s}^T\Psi(R\underline{m}-\underline{a})\right]$$

where $\Psi = 2\pi(2N)^{-1}$. Noting that $\Psi R = \Gamma$, and rearranging the summations yields:

$$X(R\underline{n},\Omega\underline{k}) = \frac{1}{|Q|}\frac{1}{2|N|}\sum_{\underline{s}\in D_{2N}}\sum_{\underline{h}\in D_Q}\tilde{W}_{2N}(\underline{s})\exp\left[j\underline{n}^T\Gamma\underline{h}\right]$$
$$\cdot\sum_{\underline{a}\in D_{2N}}\tilde{x}(\underline{a})\exp\left[-j\underline{k}^T\Omega\underline{a}\right]\exp\left[-j\underline{s}^T\Psi\underline{a}\right] \qquad (4.4)$$
$$\cdot\sum_{\underline{m}\in D_Q}\exp\left[j(\underline{s}-\underline{h})^T\Gamma\underline{m}\right]$$

Noting that

$$\sum_{\underline{m}\in D_Q} \exp\left[j(\underline{s}-\underline{h})^T \Gamma \underline{m}\right] = |Q| \sum_{\underline{i}\in D_\infty} \delta(\underline{s}-\underline{h}-Q\underline{i}) \qquad (4.5)$$

the summation in $\underline{h}$ can be eliminated. Thus, equation (4.4) simplifies to:

$$\begin{aligned} X(R\underline{n},\Omega\underline{k}) = \frac{1}{2|N|} \sum_{\underline{s}\in D_{2N}} \tilde{W}_{2N}(\underline{s}) \exp\left[j\underline{n}^T \Gamma \underline{s}\right] \\ \cdot \sum_{\underline{a}\in D_{2N}} \tilde{x}(\underline{a}) \exp\left[-j(\underline{k}^T\Omega + \underline{s}^T\Psi)\underline{a}\right] \end{aligned} \qquad (4.6)$$

Since $LM = N$, Ω can be expressed as $2L\Psi$. Thus the summation in $\underline{a}$ can be seen as a discrete Fourier transform of $\tilde{x}$ over D_{2N}, and can be written $\tilde{X}_{2N}$.

$$X(R\underline{n},\Omega\underline{k}) = \frac{1}{2|N|} \sum_{\underline{s}\in D_{2N}} \tilde{W}_{2N}(\underline{s})\, \tilde{X}_{2N}(2L\underline{k} + \underline{s}) \exp\left[j\underline{n}^T \Gamma \underline{s}\right] \qquad (4.7)$$

From this expression for X, it can be seen that the SSFT can be interpreted as a series of inverse Fourier transforms of $\tilde{X}_{2N}$ windowed by $\tilde{W}_{2N}$ centered over a frequency $\Omega\underline{k}$. The function $\tilde{X}_{2N}$ is not the DFT of the signal x but of $\tilde{x}$, the reflectively extended version of x. With the exception of a phase wrap and a scaling factor, this is equivalent to taking the even discrete cosine transform (DCT) of the original

function x over the region D_N, followed by reflectively extending the DCT about each zero-frequency axis. Thus, the SSFT can be interpreted as a bank of filters operating in the cosine transform domain rather than the Fourier transform domain.

4.2 Conditions for Reconstruction

One of the desired properties for a useful extension of the STFT for finite extent signals was that it should not extend beyond the spatial support of the original signal. Periodic extension of the signal resulted in a short-space spectrum which was also periodic, with a period equal to the size of the region of support of the signal. In an attempt to reduce the effect of signal boundary discontinuites, the signal was reflectively extended, resulting in a periodicity double that of the previous case. For the critically sampled SSFT, this would imply that in order to uniquely reconstruct the original signal, the total number of sample points needed would be 2^D times the number of samples in the original D-dimensional signal. It will be shown in this section that for a wide class of analysis windows, the SSFT can, in fact, uniquely represent a signal with no more sample values than are in the original signal.

In the spatial dimension, one period of $X(Rn,\Omega k)$ is defined over the region $Rn \in D_{2N}$. The most straightforward way to constrain the region in which the SSFT uniquely represents x without redundancy is to use the values of X only in the

region $R\underline{n} \in D_N$. To find a constraint on the analysis window such that this will hold, we can begin by rewritting the definition of the SSFT using the substitution $\underline{a} = M\underline{r} + \underline{s}$.

$$X(R\underline{n},\Omega\underline{k}) = \sum_{\underline{s}\in D_M} \sum_{\underline{r}\in D_{2L}} \tilde{x}(M\underline{r}+\underline{s})\ \tilde{w}(R\underline{n}-M\underline{r}-\underline{s}) \cdot \exp\left[-j\underline{k}^T\Omega(M\underline{r}+\underline{s})\right] \tag{4.8}$$

Noting that $\Omega M = 2\pi I$, and reordering the summations, this can be rewritten as:

$$X(R\underline{n},\Omega\underline{k}) = \sum_{\underline{s}\in D_M} \exp\left[-j\underline{k}^T\Omega\underline{s}\right] \sum_{\underline{r}\in D_{2L}} \tilde{x}(M\underline{r}+\underline{s})\ \tilde{w}(R\underline{n}-M\underline{r}-\underline{s}) \tag{4.9}$$

If we define a new function, z, such that:

$$z(\underline{n},\underline{s}) = \sum_{\underline{r}\in D_{2L}} \tilde{x}(M\underline{r}+\underline{s})\ \tilde{w}(R\underline{n}-M\underline{r}-\underline{s}) \qquad \underline{s}\in D_M \tag{4.10}$$

then the SSFT can be expressed as a multidimensional DFT of z:

$$X(R\underline{n},\Omega\underline{k}) = \sum_{\underline{s}\in D_M} z(\underline{n},\underline{s})\ \exp\left[-j\underline{k}^T\Omega\underline{s}\right] \tag{4.11}$$

Since there is a unique, one-to-one relationship between $X(R\underline{n},\Omega\underline{k})$ and $z(\underline{n},\underline{s})$, to show that X uniquely represents x in

the region $R\underline{n} \in D_N$, it is enough to show that x can be uniquely reconstructed using $z(\underline{n},\underline{s})$ in the same region, $R\underline{n} \in D_N$, or equivalently, $\underline{n} \in D_{Q/2}$.

In order to reduce notational complexity, the window constraints will be derived only for the case of one-dimensional signals. The extension to more than one dimension is straightforward and will be discussed briefly after the derivation.

In one dimension, z can be expressed as:

$$z(n,s) = \sum_{r=0}^{2L-1} \tilde{x}(Mr+s)\ \tilde{w}(Rn-Mr-s) \qquad 0 \leq s \leq M-1 \quad (4.12)$$

For each fixed value of s in the above equation, z can be viewed as a linear combination of the samples of one particular phase of $\tilde{x}$ decimated by a factor M. If z is known over the complete interval $0 \leq n \leq Q-1$, then x can clearly be reconstructed using the inversion equation (3.15) with one of the numerous possible choices for the synthesis window. If, however, z(n,s) is only known for $0 \leq n \leq \frac{1}{2}Q-1$, it is not clear that the decimated samples of $\tilde{x}$ can be reconstructed by an inverse transformation of z. By noting the symmetry in $\tilde{x}$, it can be shown that for two values of s, z(n,s) depends on the same decimated samples of $\tilde{x}$. If the values of z(n,s) are used simultaneously for both of these values of s, there may be enough information to reconstruct x.

To show that for two values of s, $z(n,s)$ depends on the same decimated samples of $\tilde{x}$, we can make the substitutions $a = M-1-s$ for $0 \leq s \leq M-1$, and $b = Q-1-n$ for $\frac{1}{2}Q \leq n \leq Q-1$, into the definition of z.

$$z(Q-1-b,M-1-a) = \sum_{r=0}^{2L-1} \tilde{x}(Mr+M-1-a) \cdot \tilde{w}(R(Q-1-b)-Mr-M+1+a) \tag{4.13}$$

Further, substituting $m = 2L-1-r$ for $0 \leq r \leq 2L-1$, this can be rewritten as:

$$z(Q-1-b,M-1-a) = \sum_{m=0}^{2L-1} \tilde{x}(M(2L-1-m)+M-1-a) \cdot \tilde{w}(R(Q-1-b)-M(2L-1-m)-M+1+a) \tag{4.14}$$

$$z(Q-1-b,M-1-a) = \sum_{m=0}^{2L-1} \tilde{x}(2N-Mm-1-a)\tilde{w}(Mm+1+a-R-Rb) \tag{4.15}$$

Using the definition of $\tilde{x}$ by equation (3.5), we can see that:

$$\tilde{x}(2N-1-(Mm+a)) = \tilde{x}(Mm+a) \tag{4.16}$$

Using the variables in the original definition of z, equation (4.15) can now be rewritten as:

$$z(Q-1-n,M-1-s) = \sum_{r=0}^{2L-1} \tilde{x}(Mr+s)\, \tilde{w}(Mr+1+s-R-Rn) \quad (4.17)$$

Since the argument of $\tilde{x}$ in (4.17) and in (4.12) are identical, for a given value of s, the same decimated samples of $\tilde{x}$ are used to generate both z(n,s) and z(Q-1-n,M-1-s). If we define $\hat{z}$ and $\hat{w}$ such that:

$$\hat{z}(n,s) = \begin{cases} z(n,s) & 0 \le n \le \frac{Q}{2}-1 \\ z(Q-1-n,M-1-s) & \frac{Q}{2} \le n \le Q-1 \end{cases}$$

(4.18)

$$\hat{w}(Rn,q) = \begin{cases} \tilde{w}(Rn+q) & 0 \le n \le \frac{Q}{2}-1 \\ \tilde{w}(1+q-R-Rn) & \frac{Q}{2} \le n \le Q-1 \end{cases}$$

then (4.12) and (4.17) can be combined into one equation representing a single linear combination of the decimated samples of $\tilde{x}$.

$$\hat{z}(n,s) = \sum_{r=0}^{2L-1} \tilde{x}(Mr+s)\, \hat{w}(Rn,Mr+s) \qquad \begin{matrix} 0 \le n \le Q-1 \\ 0 \le s \le M-1 \end{matrix} \quad (4.19)$$

For the SSFT to be uniquely invertable using only the region of $X(Rn,\Omega k)$ for $0 \leq n \leq \frac{1}{2}Q-1$, all that is required is that the linear transformation defined by (4.19) is non-singular. A very wide class of window functions will satisfy this constraint.

Since $\hat{w}$ is an arbitrary linear transformation, finding and implementing the inverse may be very difficult. It would be desirable to have an inverse that can be implemented in the same manner as the ordinary inverse given by equation (3.15). It can be shown that the window function can be further constrained such that given $\hat{z}$, z can be found completely by a simple relationship. Once z is known, (3.15) can be used directly to reconstruct x. As will be demonstrated in section 5, this can also help to reduce the computational complexity in implementing the SSFT.

One case where such a simple relationship exists can be found by constraining $\tilde{w}$ such that:

$$\tilde{w}(Rn-q) = \tilde{w}(1+q-R-Rn) \tag{4.20}$$

By the definition of $\hat{w}$, this implies that $\hat{w}$ can be written as:

$$\hat{w}(Rn,q) = \tilde{w}(Rn-q) \qquad 0 \leq n \leq Q-1 \tag{4.21}$$

Now, equation (4.19) can be rewritten in the simpler form:

$$\hat{z}(n,s) = \sum_{r=0}^{2L-1} \tilde{x}(Mr+s)\, \tilde{w}(Rn-Mr-s) \qquad \begin{matrix} 0 \le n \le Q-1 \\ 0 \le s \le M-1 \end{matrix} \tag{4.22}$$

Comparing this to equation (4.12) we can see that they are identical. Thus:

$$z(n,s) = \hat{z}(n,s) \qquad \begin{matrix} 0 \le n \le Q-1 \\ 0 \le s \le M-1 \end{matrix} \tag{4.23a}$$

$$z(n,s) = \begin{cases} z(n,s) & 0 \le n \le \frac{Q}{2}-1 \\ z(Q-1-n, M-1-s) & \frac{Q}{2} \le n \le Q-1 \end{cases} \tag{4.23b}$$

From the lower portion of (4.23b), it is shown that the values of $z(n,s)$ for $\frac{1}{2}Q \le n \le Q-1$ can be determined from $z(Q-1-n,M-1-s)$. Since the argument $Q-1-n$ is restricted to be between zero and $\frac{1}{2}Q-1$ for $\frac{1}{2}Q \le n \le Q-1$, the upper portion of z can be obtained directly from corresponding values in the lower portion. Since all of z can be determined from the known half of z, equation (3.15) can then be used for reconstruction.

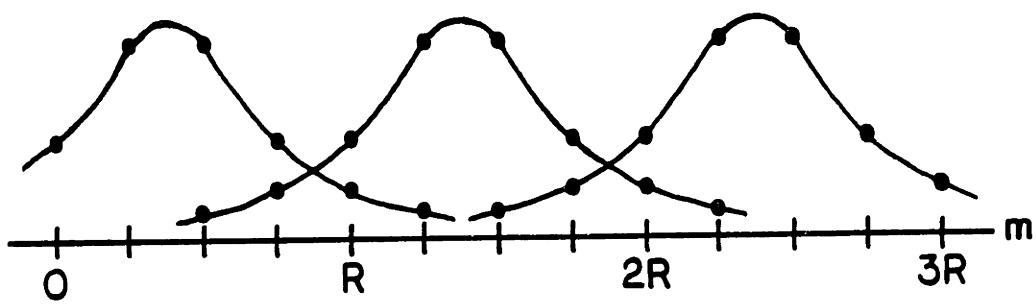
The physical meaning of the constraint on the window function can be seen by using equation (4.20) and substituting $p = Rn-q$.

$$\tilde{w}(p) = \tilde{w}(1-R-p) \tag{4.24}$$

This implies that $\tilde{w}$ must be symmetric about the point p such that p = 1-R-p. Solving for p, we obtain:

$$p = \frac{1-R}{2} \tag{4.25}$$

This constraint is very reasonable for practical situations. It simply means that $\tilde{x}$ is multiplied by the time reversed window function centered within blocks defined by the samples of Rn. An example of a window function which satisfies this constraint is shown below. Pictured is the time reversed window w(Rn-m) for several values of n with R = 4.



It is possible to also define other constraints on the window function such that z can be found from $\hat{z}$ by a simple relationship. For example, if the window function is complex valued, it may be desirable to have conjugate symmetry rather than complete symmetry. Clearly, this simply corresponds to:

$$z(n,s) = z^*(Q-1-n, M-1-s) \tag{4.26}$$

In a similar manner, many other constraints on $\tilde{w}$ may be found which result in simple relationships such as this.

For the case of multidimensional signals, these results can be extended in a straightforward manner. From the sample values for a single phase of the decimated signal $\tilde{x}(M\underline{r}+\underline{s})$, we can generate $\hat{z}(\underline{n},\underline{s})$ defined in a similar manner to equation (4.18). The difference is that instead of $\hat{z}$ being defined in terms of z on each half of the one-dimensional axis, $\hat{z}$ is defined in terms of z for each "quadrant" of the multidimensional space. Similarly, the "quadrants" of $\hat{w}$ can also be defined in terms of w.

The constraint for symmetry in w such that z can be directly obtained from $\hat{z}$ also extends easily into multiple dimensions. For more than one dimensional signals, the symmetry constraint on the window function, given by equations (4.24) and (4.25), must hold along each dimension for all values of the remaining dimensions. For example, in two dimensions, $\tilde{w}(x,y)$ must be symmetric around $y = \frac{1}{2}(1-R_y)$ for all x, and around $x = \frac{1}{2}(1-R_x)$ for all y. Note that a separably symmetric window function will satisfy this constraint. For the example of the conjugate symmetry constraint, a separably conjugate symmetric window will not satisfy this condition.

4.3 Linear Filtering Using the Short-Space Fourier Transform

There are many applications in which it is desirable to implement time-varying or spatially varying linear filters. For the case of multidimensional finite extent signals, such as images, examples of such applications include image restoration, image enhancement and non-uniform motion compensation. In general, spatially varying filters are computationally difficult to implement. Unlike spatially invariant filters which can be implemented very efficiently by direct multiplication in the frequency domain, there is not necessarily an efficient way to implement a spatially varying filter.

For one-dimensional, infinite extent signals, Portnoff [23] has shown that by direct multiplication of the short-time Fourier transform coefficients of a signal, a limited class of time-varying filters can be implemented. Because of the existence of fast algorithms for computing the STFT and its inverse, many filters can be implemented much more efficiently than they can in the time domain. Clearly, for multidimensional, finite extent signals, direct multiplication of the SSFT coefficients can be used to implement a similar class of spatially varying filters. Since the SSFT is simply the multidimensional extension of the STFT on the reflectively extended signal, $\tilde{\tilde{x}}$, the class of spatially varying filters implementable in this manner is the same class of filters discussed by Portnoff operating on $\tilde{\tilde{x}}$. Since $\tilde{\tilde{x}}$ is continuous

at the boundaries, serious problems due to edge effects may be avoided.

In his discussions, Portnoff addressed the issue of what class of filters can be implemented by direct multiplication of the STFT. A more general question to ask, however, is how the STFT coefficients, or the SSFT coefficients can be modified to implement any particular filter. From this general formulation, it may then be possible to find other classes of filters which can be implemented efficiently using these transforms. The implementation may be more complex for many of these classes than direct multiplication, but still may be more efficient than direct implementation in the spatial or temporal domains. Although equally applicable to the STFT this issue will be addressed only for the case of the SSFT.

The most general form of a spatially varying filter operating on the signal, x, over a region D_N ,can be written as:

$$y(\underline{m}) = \sum_{\underline{b} \in D_N} g(\underline{m},\underline{b})\ x(\underline{b}) \qquad \underline{m} \in D_K \tag{4.27}$$

Equivalently we can define $\tilde{y}$ as the reflectively extended version of y as in equation (3.5). By introducing a modified transformation, $\tilde{g}$, it is clearly possible to write:

$$\tilde{y}(\underline{m}) = \sum_{\underline{b}\in D_{2N}} \tilde{g}(\underline{m},\underline{b})\ \tilde{x}(\underline{b}) \qquad \underline{m}\in D_{2K} \tag{4.28}$$

This expression may be thought of as a general filter applied to the signal $\tilde{x}$. Writing the transformation in this special form is only useful because it allows us to directly use the SSFT as previously defined.

The most straightforward way to determine how the SSFT might be used to implement this transformation is to simply find the relationship between the SSFT of x and the SSFT of y. The SSFT of y is defined by:

$$Y(A\underline{n},\Phi\underline{k}) = \sum_{\underline{m}\in D_{2K}} \tilde{y}(\underline{m})\ \tilde{w}(A\underline{n}-\underline{m})\ \exp\left[-j\underline{k}^T\Phi\underline{m}\right] \tag{4.29}$$

Using the expression for $\tilde{y}$ in terms of $\tilde{x}$, we can write this as:

$$Y(A\underline{n},\Phi\underline{k}) = \sum_{\underline{m}\in D_{2K}} \sum_{\underline{b}\in D_{2N}} \tilde{g}(\underline{m},\underline{b})\ \tilde{x}(\underline{b})\ \tilde{w}(A\underline{n}-\underline{m}) \cdot \exp\left[-j\underline{k}^T\Phi\underline{m}\right] \tag{4.30}$$

Finally, we can express Y directly in terms of X by using the SSFT synthesis equation given by (3.15).

$$Y(A\underline{n},\Phi\underline{k}) = \sum_{\underline{m}\in D_{2K}} \sum_{\underline{b}\in D_{2N}} \tilde{g}(\underline{m},\underline{b}) \sum_{\underline{h}\in D_M} \sum_{\underline{a}\in D_Q} \tilde{f}(\underline{b}-R\underline{a})$$

$$\cdot X(R\underline{a},\Omega\underline{h}) \exp\left[j\underline{h}^T\Omega\underline{b}\right] \tilde{w}(A\underline{n}-\underline{m}) \qquad (4.31)$$

$$\cdot \exp\left[-j\underline{k}^T\Phi\underline{m}\right]$$

By rearranging the summations and defining a new function G, which does not depend on x, we can write (4.31) in the form:

$$Y(A\underline{n},\Phi\underline{k}) = \sum_{\underline{h}\in D_M} \sum_{\underline{a}\in D_Q} X(R\underline{a},\Omega\underline{h})\ G(A\underline{n},R\underline{a};\Phi\underline{k},\Omega\underline{h}) \qquad (4.32)$$

where

$$G(A\underline{n},R\underline{a};\Phi\underline{k},\Omega\underline{h}) = \sum_{\underline{m}\in D_{2K}} \sum_{\underline{b}\in D_{2N}} \tilde{f}(\underline{b}-R\underline{a})\ \tilde{w}(A\underline{n}-\underline{m}) \qquad (4.33)$$

$$\cdot \tilde{g}(\underline{m},\underline{b}) \exp\left[j\underline{h}^T\Omega\underline{b}\right] \exp\left[-j\underline{k}^T\Phi\underline{m}\right]$$

The SSFT of the function y, has been written in the form of a linear filtering operation on the SSFT coefficients of x. The filter, G, can be interpreted as the (2D)-dimensional SSFT of the spatial domain filter function $g(\underline{m},\underline{b})$. For the variable $\underline{b}$, the analysis window, $\tilde{w}$, is used, and for the variable $\underline{m}$, the analysis window is $\tilde{f}$, where $\tilde{f}$ is the synthesis window

ordinarily used in the definition of the D-dimensional SSFT inverse.

The conversion of the filter function, g, into the transformed filter function, G, by equation (4.33) can be seen to be similar to the bi-frequency interpretation of time-varying filters in the Fourier transform domain [7][9]. It has been shown that the effect of a time-varying filter on Fourier coefficients can be regarded as a filter operating on the frequency coefficients. This linear filter function can be found by taking the Fourier transform along one dimension of the original filter function, and an inverse Fourier transform along the other. This process can be easily understood by considering the matrix formulation of a linear transformation. Let $\underline{y}$ be a vector produced by multiplying the vector $\underline{x}$ by the matrix G. This is equivalent to the general linear filter given by equation (4.27). Let S be another matrix, coresponding, for example, to the SSFT. We can find a matrix H which can be used to express the transformed version of $\underline{y}$, $S\underline{y}$, in terms of the transformed version of $\underline{x}$, $S\underline{x}$.

$$\begin{aligned} \underline{y} &= G\underline{x} \\ S\underline{y} &= SG\underline{x} \\ S\underline{y} &= SGS^{-1}S\underline{x} \\ S\underline{y} &= H(S\underline{x}) \end{aligned} \tag{4.34}$$

where

$$H = SGS^{-1} \tag{4.35}$$

This corresponds to forward transforming the column vectors of G, and inverse transforming the row vectors. This is exactly what is being performed by equation (4.33).

By obtaining the SSFT of a signal, the new transfomation that can be applied to obtain the SSFT of the result is expressed in exactly the same form as the original transformation. To generate the SSFT of y, all of the values of X must be linearly combined, just as in the case of the spatial domain filter. In fact, if the SSFT is not critically sampled, more samples must be included. The computational savings, of course, comes when the class of filters is restricted to those which are computationally efficient in the SSFT domain. The case which Portnoff discussed for the STFT was the computationally simplest case: direct multiplication of the STFT coefficients. He showed that, by varying the choice of window function used, a fairly wide variety of useful transformations could be closely approximated. In terms of the matrix formulation, direct multiplication corresponds to the H matrix being restricted to be diagonal. This means that the spatial domain response matrix is restricted to be of the form:

$$G = S^{-1}DS \tag{4.36}$$

where D is an arbitrary diagonal matrix.

For many applications, the purely diagonal constraint may be too restrictive. For example, it has been found that for image motion compensation, implementation with a strictly diagonal operator is not sufficient to obtain a response close to the desired response. By sacrificing some of the computational savings gained by using a purely diagonal transformation, it is possible to widely expand the class of filters which can be implemented. For example, for the case of motion compensation, using the frequency coefficients from adjacent blocks could greatly improve the approximation to the desired response. In the matrix formulation, this corresponds to having non-zero terms not only along the main diagonal, but also along the two adjacent diagonals.

In this sense, the general results obtained here can be used to describe any class of filters. By not restricting the filters to be diagonal from the beginning, we can determine with more generality what types of filters can be efficiently implemented in the SSFT domain. Specific cases, and experimental tests have not yet been performed in this area. A great deal of work still remains to determine if the properties of the SSFT allow it to be of use in practical applications.

5. Implementation of The Short-Space Fourier Transform

5.1 Spatial Domain Implementation

The short-space Fourier transform would be of little value if there were not an efficient algorithm for its computation. As discussed in chapter 2, for the short-time Fourier transform, a fast computational algorithm does exist, which makes use of the fast Fourier transform [5][22][23]. In this section, a similar algorithm is developed for efficient computation of the short-space Fourier transform.

The spatial domain algorithm for computing the SSFT can be derived by beginning with the expression for the SSFT given by equation (4.11).

$$X(R\underline{n},\Omega\underline{k}) = \sum_{\underline{s}\in D_M} z(\underline{n},\underline{s}) \exp\left[-j\underline{k}^T\Omega\underline{s}\right] \qquad (5.1)$$

where

$$z(\underline{n},\underline{s}) = \sum_{\underline{r}\in D_{2L}} \tilde{x}(M\underline{r}+\underline{s})\ \tilde{w}(R\underline{n}-M\underline{r}-\underline{s}) \qquad \underline{s}\in D_M \qquad (5.2)$$

For each spatial location, $\underline{n}$, X can be computed by taking the size M discrete Fourier transform of the function $z(\underline{n},\underline{s})$ as indicated by equation (5.1). This can be efficiently implemented using any of the fast Fourier transform algorithms [8][22]. The more difficult portion of the algorithm involves the computation of $z(\underline{n},\underline{s})$. If the transform is not critically

sampled, z can only be found by direct implementation of (5.2). Physically, equation (5.2) represents multiplying $\tilde{x}$ by the spatially reversed window function centered at position $R\underline{n}$, and spatially aliasing this signal into size M blocks.

If the SSFT is critically sampled, the computation of z can be done much more efficiently than by direct implementation. If R = M then equation (5.2) can be written as:

$$z(\underline{n},\underline{s}) = \sum_{\underline{r}\in D_{2L}} \tilde{x}(M\underline{r}+\underline{s})\ \tilde{w}(M(\underline{n}-\underline{r})-\underline{s}) \tag{5.3}$$

By defining $\tilde{x}_{\underline{s}}$ and $\tilde{w}_{\underline{s}}$ as decimated versions of $\tilde{x}$ and $\tilde{w}$, where

$$\tilde{x}_{\underline{s}}(\underline{r}) = \tilde{x}(M\underline{r}+\underline{s}) \tag{5.4}$$

$$\tilde{w}_{\underline{s}}(\underline{r}) = \tilde{w}(M\underline{r}-\underline{s}) \tag{5.5}$$

equation (5.3) can be further rewritten in the simplified form:

$$z(\underline{n},\underline{s}) = \sum_{\underline{r}\in D_{2L}} \tilde{x}_{\underline{s}}(\underline{r})\ \tilde{w}_{\underline{s}}(\underline{n}-\underline{r}) \tag{5.6}$$

Since $\tilde{w}$ is periodic with periodicity matrix 2N, $\tilde{w}_{\underline{s}}$, as

defined, is periodic with periodicity matrix 2L. Thus equation (5.6) simply corresponds to a circular convolution of the decimated samples of $\tilde{x}$ and $\tilde{w}$. To implement this series of circular convolutions, the discrete Fourier transform can be used. Again, by using FFT algorithms, the convolutions can be done much more efficiently than they can be done in the spatial domain.

By using the special case symmetric window, as discussed in section 4.2, the computational complexity of the SSFT can be further reduced. This is accomplished by accounting for the redundancy in z that results. For the one-dimensional case, it was shown that

$$z(n,s) = z(2L-1-n, M-1-s) \qquad \begin{matrix} 0 \le n \le 2L-1 \\ 0 \le s \le M-1 \end{matrix} \tag{5.7}$$

under the constraint that the window function is symmetric about a point p given by equation (4.25). This was taken for the critically sampled case where Q = 2L. To compute the non-redundant portion of the SSFT it is required to compute z(n,s) for $0 \le s \le M-1$ and $0 \le n \le L-1$. This defines z only in the region of support of the original signal. We know that the other half of z is redundant. However, if we use the discrete Fourier transform to implement the circular convolutions in equation (5.6) for each value of s, the result includes the values of z(n,s) for all n over $0 \le n \le 2L-1$. Since redundant information is computed, some of the

computation is wasteful. However, if z is computed using the DFT implementation for only the values of s over $0 \leq s \leq \frac{1}{2}M-1$, equation (5.7) can be used to find all of the remaining values. Specifically, the values of z in the region $0 \leq n \leq L-1$ and $\frac{1}{2}M \leq s \leq M-1$ which are needed, correspond exactly to values in the region $L \leq n \leq 2L-1$ and $0 \leq s \leq \frac{1}{2}M-1$ which are known.

For multiple dimensions, this corresponds to computing z for only one "quadrant" in $\underline{s}$. From this, all of the remaining quadrants can be found directly.

From this description of the SSFT analysis implementation, a simple approach to SSFT synthesis is readily apparent. For the critically sampled case, the synthesis procedure simply involves reversing each step of the analysis procedure. First, inverse discrete Fourier transforms are performed, followed by the circular deconvolutions using the inverse of the analysis window function in the frequency domain.

The basic block diagram for computing the one-dimensional critically sampled SSFT for a real valued, symmetric window function is shown in figure 5.1. In figures 5.2 and 5.3, a detailed flow diagram of the algorithm is shown. Figure 5.2 shows the computation of the window coefficient matrix. Since this depends only on the window function, it can be precomputed and stored so that this portion of the algorithm does not need to be performed each time the SSFT is computed. Figure 5.3 shows the computation of the SSFT coefficients from the input signal.

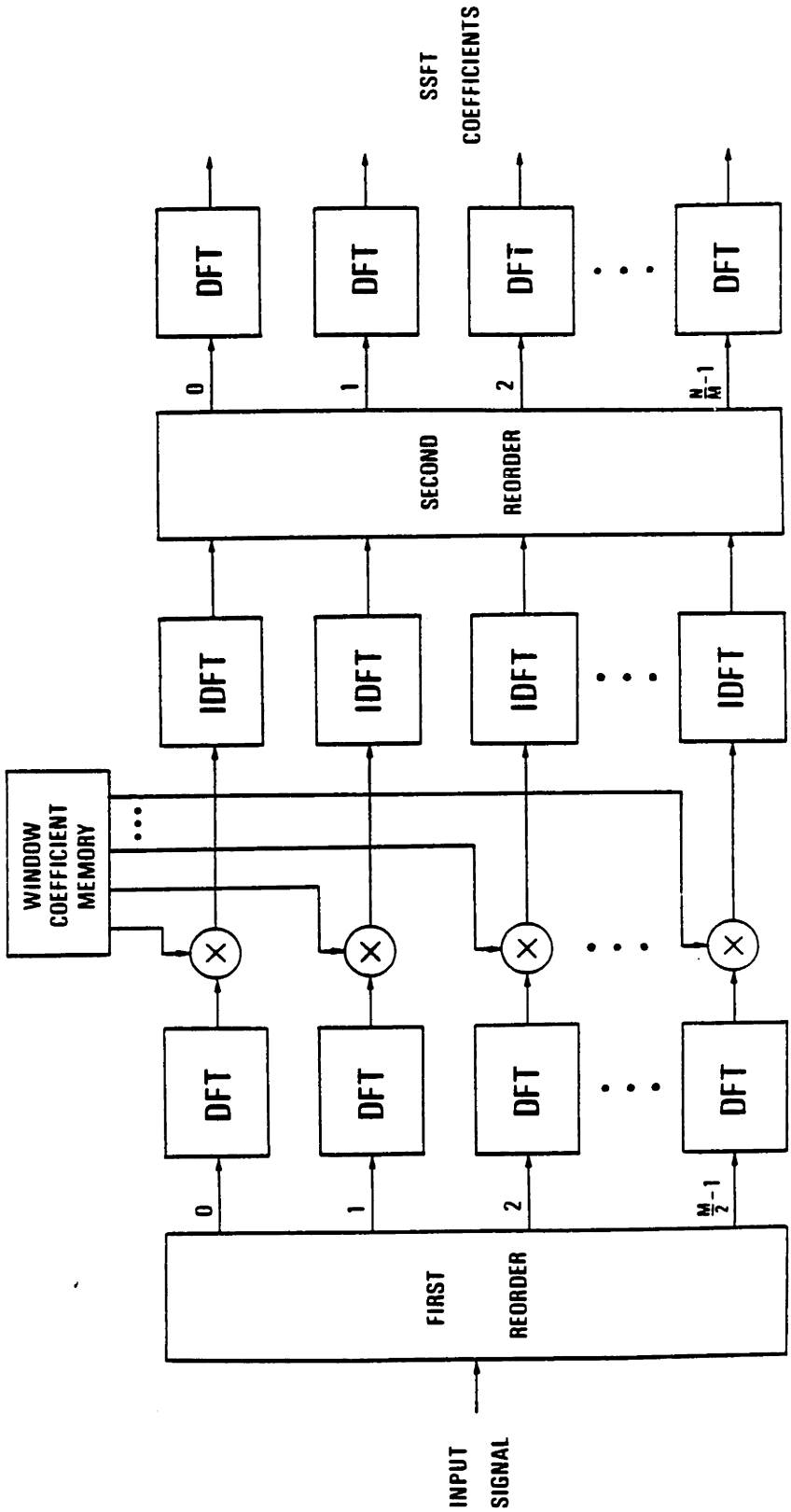


Figure 5.1 - Block diagram for the one-dimensional critically sampled SSFT using a symmetric window function as described in the text.

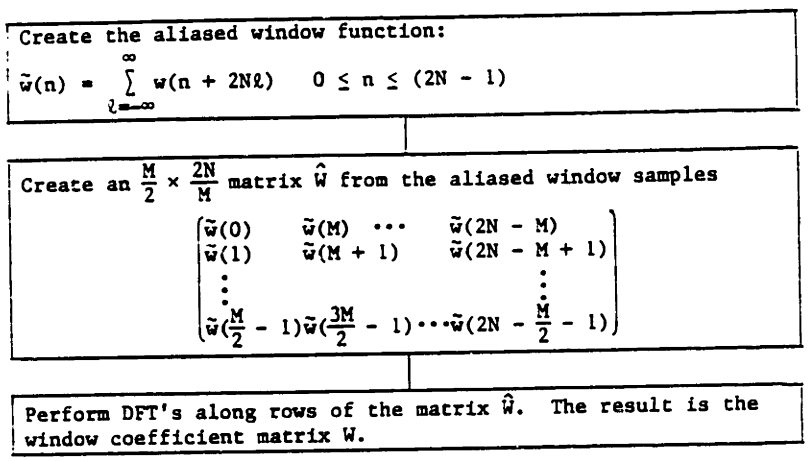


Figure 5.2 - Computation of the window coefficient matrix used in the implementation of the SSFT shown in figure 5.3.

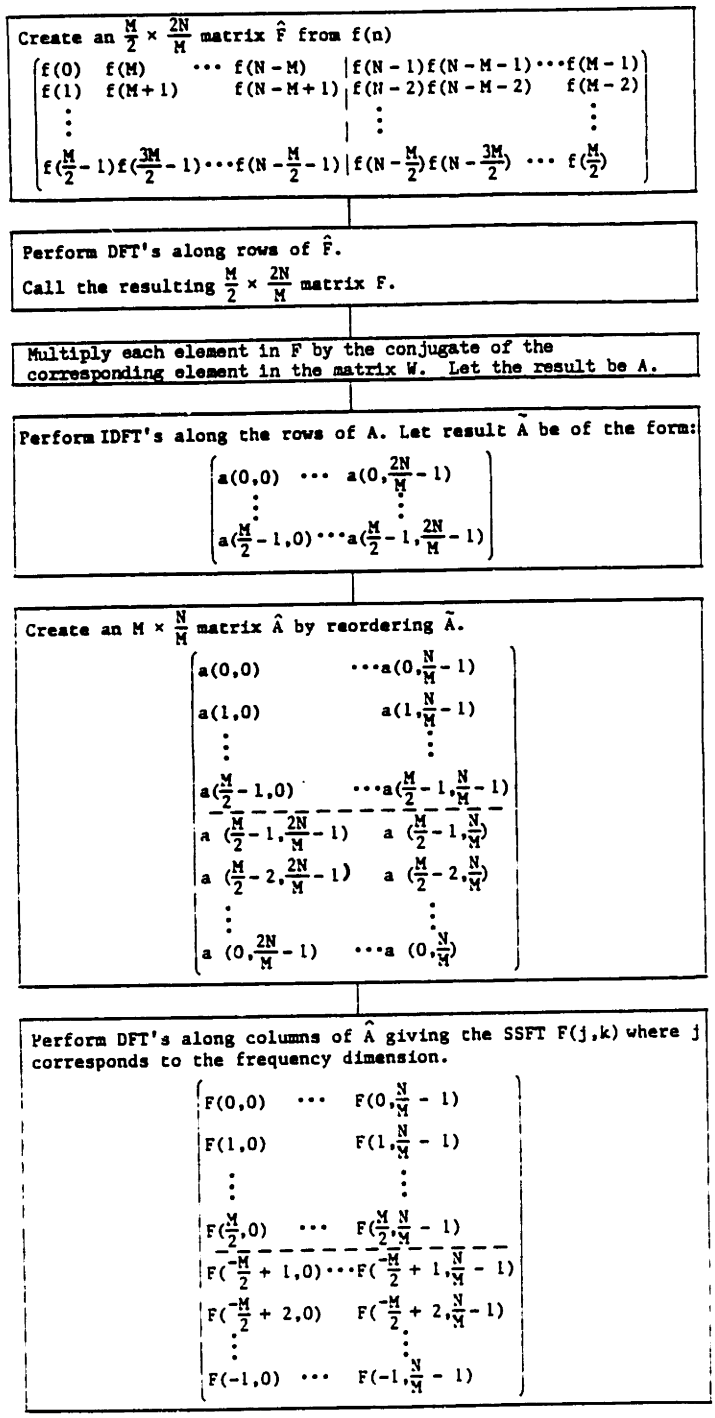


Figure 5.3 - Detailed flow diagram for computation of the one-dimensional critically sampled SSFT using a real, symmetric window function as described in the text.

5.2 Frequency Domain Implementation

An alternative to implementing the SSFT by the use of equations (5.1) and (5.2) is to use the dual formulation given by equation (4.7). With the exception of the scale factor and the fact that the shift is on the signal term rather than the window term, this equation is in the same form as the SSFT expressed in the spatial domain. Thus, given $\tilde{X}_{2N}$ and $\tilde{W}_{2N}$, the amount of computation required to compute the SSFT in the frequency domain would be approximately the same as it would be to compute it in the spatial domain in the manner described in the last section. However, if the functions $\tilde{x}$ and $\tilde{w}$ are only known in the spatial domain, the Fourier transforms of these must be computed first. Clearly, for an arbitrary window function, this approach requires much more computation than the spatial domain approach. For a particular class of window functions, however, this is not true. Specifically, if the window function is an ideal lowpass window, then the frequency domain implementation becomes efficient. Such a window will be shown in chapter 6 to be very useful for image coding applications.

To show why implementation using an ideal lowpass window is efficient to compute, we can decompose the summation in equation (4.7) in a similar manner to what was done for the time domain implementation. If we define, $\underline{s} = Q\underline{a} + \underline{b}$, then equation (4.7) can be expressed in the form:

$$X(R\underline{n},\Omega\underline{k}) = \frac{1}{2|N|}\sum_{\underline{a}\in D_R}\sum_{\underline{b}\in D_Q}\tilde{W}_{2N}(Q\underline{a}+\underline{b}) \cdot \tilde{X}_{2N}(2L\underline{k}+Q\underline{a}+\underline{b})\exp\left[j\underline{n}^T\Gamma(Q\underline{a}+\underline{b})\right] \tag{5.8}$$

Rearranging the summations and noting that $\Gamma_Q = 2\pi I$, we get:

$$X(R\underline{n},\Omega\underline{k}) = \frac{1}{2|N|}\sum_{\underline{b}\in D_Q}\exp\left[j\underline{n}^T\Gamma\underline{b}\right]\sum_{\underline{a}\in D_R}\tilde{W}_{2N}(Q\underline{a}+\underline{b}) \cdot \tilde{X}_{2N}(2L\underline{k}+Q\underline{a}+\underline{b}) \tag{5.9}$$

Analogous to the spatial-domain approach, we can define a new function, $y(\underline{b},\underline{k})$, such that

$$X(R\underline{n},\Omega\underline{k}) = \frac{1}{|Q|}\sum_{\underline{b}\in D_Q} y(\underline{b},\underline{k})\exp\left[j\underline{n}^T\Gamma\underline{b}\right] \tag{5.10}$$

where

$$y(\underline{b},\underline{k}) = \frac{1}{|R|}\sum_{\underline{a}\in D_R}\tilde{W}_{2N}(Q\underline{a}+\underline{b})\ \tilde{X}_{2N}(2L\underline{k}+Q\underline{a}+\underline{b}) \tag{5.11}$$

If $\tilde{W}_{2N}$ is non-zero only in a region of size Q, then for each value of $\underline{b}$, there will only be one non-zero term in the

summation to obtain y. Thus the computation of the SSFT will simply consist of computing $\tilde{X}_{2N}$, then, for each value of $\underline{k}$, windowing a block of $\tilde{X}_{2N}$ of size Q, and taking a size Q inverse Fourier transform. This can be done so that the computational complexity is similar to that the spatial-domain algorithm. As was previously stated, $\tilde{X}_{2N}$ is closely related to the discrete cosine transform. Thus, it can be computed using one of the numerous fast cosine transform algorithms in existence [16]. Computing it in this manner is considerbly more efficient than directly computing the size 2N discrete Fourier transform.

One useful choice for $\tilde{W}_{2N}$ is a rectangular window with constant amplitude over the band. In continuous frequency, a rectangular window with a constant value of one over the band would correspond to a separable sinc function (i.e. sin(x)/x) in the spatial domain. If this were used as the window function, $w(\underline{n})$, then we can define $w(\underline{n})$ as the original window spatially aliased over size 2N blocks as given by equation (3.9). In the frequency domain, the aliasing operation corresponds to sampling. For a sinc function corresponding exactly to a bandwidth of size Q, some of the sample points will lie exactly at the discontinuities at the edges of the rectangular window. The result is a discrete window function of size larger than Q, and whose sample points are not of constant amplitude. In order to obtain a discrete window function that does have these properties, the continuous

frequency window must be first shifted by some odd multiple of half a sample point. This corresponds to multiplying the sinc window by a separable phase wrap in the spatial domain. That is, the spatial domain function corresponding to a constant valued rectangular window is:

$$w(\underline{n}) = \prod_{i=1}^{D} \exp\left[j\pi \frac{n_i(2c_i+1)}{2N_i}\right] \frac{\sin(\pi n_i/R_i)}{(\pi n_i/R_i)} \qquad (5.12)$$

where the c_i's are arbitrary integer constants which determine the center frequency of the bands.

One important factor to notice about equation (5.13) is that the window is separably conjugate symmetric. It was noted in section 4.2 that for more than one dimensional signals, a separably conjugate symmetric window does not satisfy the condition found for simple reconstruction of the SSFT from its values in the region $R\underline{n} \in D_N$. For this particular case, however, it is possible to find a different region from which the signal can easily be reconstructed. Because $\tilde{X}_{2N}$ is the Fourier transform of a reflectively extended signal, it has a symmetry of the form:

$$\tilde{X}_{2N}(\underline{k}) = \exp\left[j\underline{f}^T \Psi \underline{k}\right] \tilde{X}_{2N}(\underline{h}) \qquad (5.13)$$

where $\Psi = 2\pi(2N)^{-1}$, for all integer valued vectors, $\underline{f}$, where $\underline{f} \in D_I$, and $\underline{h}$ is found by:

$$h_i = \begin{cases} k_i & f_i = 0 \\ 2N_i - k_i & f_i = 1 \end{cases} \qquad (5.14)$$

Thus, $\tilde{X}_{2N}$ is completely defined from only one "quadrant." By choosing the window function to be positioned such that $c_i = \frac{1}{2}Q_i - 1$, then only the frequency coefficients of $X(R\underline{n}, \Omega_{\underline{k}})$ in the region $\underline{k} \in D_{M/2}$ are required. These coefficients will define $\tilde{X}_{2N}(\underline{k})$ over the region $\underline{k} \in D_N$. The remainder of $\tilde{X}_{2N}$ can be reconstructed by employing equation (5.13).

If the original signal, x, is real valued, then the total number of sample points needed for simple reconstruction can be further reduced. If x is real, then $\tilde{X}_{2N}$ can be expressed as a real valued function multiplied by a constant phase wrap:

$$\tilde{X}_{2N}(\underline{k}) = C(\underline{k}) \exp\left[\frac{1}{2} j \underline{i}^T \Psi \underline{k}\right] \qquad (5.15)$$

where all of the elements of the vector $\underline{i}$ are equal to one and $C(\underline{k})$ is a real valued function. If the rectangular window to be used, instead of being constant and real valued, also has a specific phase wrap corresponding to:

$$W_{2N}(\underline{k}) = \begin{cases} \exp\left[\frac{1}{2} j \underline{i}^T(\Gamma - \Psi)\underline{k}\right] & \underline{k} \in D_Q \\ 0 & \text{other} \end{cases} \quad (5.16)$$

then (5.11) can be expressed in the form:

$$y(\underline{b},\underline{k}) = \frac{1}{|R|} \tilde{W}_{2N}(\underline{b})\ \tilde{X}_{2N}(2L\underline{k} + \underline{b}) \qquad \underline{b} \in D_Q \quad (5.17)$$

$$y(\underline{b},\underline{k}) = \frac{1}{|R|} C(2L\underline{k} + \underline{b}) \cdot \exp\left[\frac{1}{2} j \left(\underline{i}^T(\Gamma - \Psi)\underline{b} + \underline{i}^T\Psi(2L\underline{k} + \underline{b})\right)\right] \quad (5.18)$$

$$y(\underline{b},\underline{k}) = \frac{1}{|R|} C(2L\underline{k} + \underline{b}) \exp\left[\frac{1}{2} j \underline{i}^T(\Gamma\underline{b} + \Omega\underline{k})\right] \quad (5.19)$$

It can be shown that after performing the size Q inverse DFT over the variable $\underline{b}$, the result, $X(R\underline{n},\Omega\underline{k})$ is symmetric in $\underline{n}$ such that only half of the region $R\underline{n} \in D_{2N}$ is needed for reconstruction. That is, for one of the dimensions, i, X_i is only needed for $0 \leq R_i n_i \leq N_i$. The remaining portion can be reconstructed by the equation:

$$X(R\underline{m},\Omega\underline{k}) = \exp\left[j \underline{i}^T \Omega \underline{k}\right]\ X^*(R\underline{n},\Omega\underline{k}) \quad (5.20)$$

where $m_i = Q_i - 1 - n_i$.

An interesting fact to note is that the phase wrap on the window function corresponds to a shift of the point of symmetry of the window in the spatial domain to exactly the same position that was required in the spatial-domain technique for simple reconstruction.

To summarize, the frequency-domain computation of the SSFT, using the sinc window function described above, involves first obtaining $\tilde{X}_{2N}$ by either Fourier or cosine transform algorithms, followed by performing inverse DFT's of size Q of $\tilde{X}_{2N}$ multiplied by the the phase wrap given in equation (5.16) at the position 2Lk.

One possible advantage to using the frequency-domain approach to implement the SSFT is that it is very simple to use a frequency dependent window function. That is, for each frequency sample, which may not even be uniformly spaced, a different window function may be used. This can be done very easily since the transform is computed independently for each frequency sample. One simple example of this, which has proven to be useful in image coding applications, is to use non-overlapping rectangular bands which are of different sizes in different portions of the frequency spectrum. This will be discussed further is chapter 6.

6. Application of the Short-Space Fourier Transform to Intraframe Image Coding

6.1 Review of Transform Image Coding Techniques

The short-space Fourier transform, as described in previous chapters, has proven to be very useful as a method of transform image coding. With the correct choice of analysis window, the SSFT has several properties which enable it to perform better than traditionally used transform coding techniques. Before describing the process of SSFT image coding, the basic ideas of traditional transform coding will be briefly reviewed.

The goal of any image coding system is to encode an image, using as few bits of information as possible, with the smallest possible distortion between the original image and the reconstructed image. In general, the measure of distortion that is used to determine the performance of an image coding system is the statistical mean squared error. This error measure is a relatively good approximation of visual error sensitivity and a mathematically simple one to analyze.

There are a wide variety of techniques used in the field of image coding. One of the most successful of these is transform coding. The usefulness of linear transformations in image coding arises by modeling an image as a sample of a two-dimensional random process. Specifically, it has been experimentally determined that natural images are approximately two-dimensional first-order Gauss-Markov

processes. For any Gaussian process, it has been shown that, given the correlation matrix describing the process, the optimal method of scalar quantization involves first performing a Karhunen-Loeve (K-L) transformation on the image data, then quantizing using indepenent Max quantizers [17] on each of the K-L coefficients. The number of bits used to quantize each coefficient must be, within an additive constant, the base-two log of the standard deviation of that coefficient [24]. The K-L transform serves to diagonalize the correlation matrix of the signal. For a Gaussian random process, this results in K-L coefficients which are independent Gaussian random variables. If the process is not jointly Gaussian, the K-L transformation is still the optimal linear processing that can be applied prior to independent scalar quantization.

For a two-dimensional first order Markov process, the K-L transform can be closely approximated by a two-dimensional discrete cosine transform (DCT), provided that the correlation coefficient of the signal is close to unity [1]. In fact, as the correlation coefficient approaches unity, the K-L transform becomes exactly the DCT. The DCT has an important advantage over the K-L transform, in that a fast algorithm exist for its computation [16]. For the general K-L transform, no fast algorithm exists. For this reason, the DCT is widely used in image coding systems.

To this point, it has not been explicitly stated over what regions these transforms should be taken. There are

several reasons to use many DCT's over small blocks of the image rather than one large DCT over the entire image. The first reason involves computational considerations. It is more computationally efficient to perform many small DCT's than it is to perform one large DCT. The loss in coding performance resulting from the use of small transforms is not extremely significant. This is due to the fact that widely spaced picture elements (pels) have little correlation. Thus, there is little to be lost by not accounting for this correlation. The second reason for using block transforms is a more important one. The K-L diagonalization only takes into account second-order statistics of the image. If the image is not jointly Gaussian, it is possible to improve coding performance by taking advantage of its true statistics. One characteristic of natural images that would not be expected in a jointly Gaussian process is the high probability that concentrated regions of high detail and large smooth regions with little detail will appear. This characteristic is not accounted for by the K-L transformation. By using block transforms, however, it is possible to quantize each block differently by noting the characteristics of the region that the block covers [30]. This technique is known as adaptive transform coding. A simple example is the categorization of blocks into two types: smooth or detailed. This can be done by a measure of variance, for example. The smooth blocks may be assigned very few bits for quantization leaving more bits free for use in the detailed regions. Adaptive techniques

such as this generally perform well. A reduction of up to two to one in data rate may be expected over non-adaptive techniques for comparable distortion.

The most apparent problem with block transform coding is that the boundaries of the blocks are often visible in the reconstructed image. At very low data rates, the block boundaries clearly stand out as the most obvious defect, giving a tiled appearance to the image. Such artifacts are generally referred to as blocking effects. The cause of blocking effects can be explained in simple terms. When a block of the image is reconstructed there is a resulting error field which represents the difference between the reconstructed block values and those of the original. The values in this error field may be correlated with each other. For example, a large error in a low-frequency term of the DCT may result in a large tilt in the error function. However, since each block is quantized independently, the resulting error in adjacent blocks is independent. Since the error at a pel at the edge of one block may be correlated with with the error at an adjacent pel within its own block, but completely independent of the error at an adjacent pel in another block, it is relatively likely that there will be discontinuities in the error field from block to block.

There are several approaches that have been taken in the past to reduce the visibility of the block boundaries. One approach, discussed by Reeve and Lim [26][27], is based on post-processing of the reconsructed image. This technique

involves the use of a smoothing filter which is used only on pels adjacent to a block boundary. In this way, the boundary discontinuities are smoothed. The problem with this method is that any details which happen to fall at a block boundary will also be smoothed, resulting in a possible loss of resolution.

Other techniques that have been proposed involve slight modifications on the block transform method itself. One technique of this type, also discussed by Reeve and Lim [26][27], involves overlapping the transform blocks by one row of pels along each border. In the reconstruction process, the pels which overlap are averaged to arrive at their final reconstructed value. This technique smooths any discontinuities at the block boundaries while preserving the detail in the original image along these boundaries. The drawback to this method is that more data is required to encode the image since each block contains redundant information. Another technique involves a "two-component source model" of images [18][19][29][31]. A low-frequency component of the image is obtained first. In general, this is accomplished by averaging over blocks and bilinearly interpolating. This may be transmitted directly in sampled form by PCM or coded by a transform technique. The remaining signal is then coded using ordinary block transform coding. By removing the low-frequency component, the correlation within the blocks will be reduced. After coding, the resulting error field will also be less correlated within each block. Thus the independence of the noise between blocks will

not cause the block boundaries to be as visible as they would otherwise be.

6.2 Image Coding Using the Short-Space Fourier Transform

None of the techniques mentioned in the previous section completely eliminate blocking effects. They merely reduce their visibility somewhat. With a specific choice of analysis window, the critically sampled SSFT can be used for both adaptive and non-adaptive transform image coding and completely avoid the problem of blocking effects. Before discussing the relative advantages and disadvantages of the proposed method in comparison to the traditional techniques, the SSFT coding technique will be presented in detail.

The characteristics of the block DCT that make it useful for adaptive image coding are that it approximately decorrelates the image data, making it suitable for independent quantization, and that it provides localized information which can be exploited by adaption techniques. By its very nature, the SSFT can also be used to provide localized information over an image. Using any analysis window in which most of the energy is concentrated over a central position, this property will be satisfied. The choice of analysis window which provides a set of uncorrelated frequency coefficients can be determined by making the following observation. In section 4.1 it was noted that the SSFT can be interpreted as taking the slightly modified DCT over the entire image, followed by windowing at the selected

frequencies and performing inverse Fourier transforms at each frequency. Since the entire image can be modeled as a first order autoregressive process, as with the block DCT, the full image DCT results in approximately decorrelated coefficients. Noting this fact, it is clear that one choice of windows that can be multiplied by the transform are non-overlapping rectangular windows. From equation (4.7), taken for the critically sampled case where R = M, it can be seen that the frequency-domain window function $\tilde{W}_{2N}$ should be of width 2L. This ensures that the windowed bands of $\tilde{X}_{2N}$ do not overlap. Since all of the frequency components are uncorrelated, and the bands do not overlap, it is clear that the coefficients in the resulting inverse Fourier transform, taken for each windowed segment, will not be correlated with each other. Since these bands correspond to the frequency components of the SSFT, then, using this window function, it will provide a set of uncorrelated coefficients at each spatial position.

Due to the need to use the minimum amount of data to represent the signal, the exact choice of window function should be that given by equation (5.16) for the critically sampled case where Q = 2L.

As stated in section 5.2, the frequency-domain implementation of the SSFT involves taking inverse Fourier transforms over these bands of $\tilde{X}_{2N}$ with the proper phase wrap. After the transform, assuming the image is real valued, it was stated that the values for one of the spatial variables is needed for only half of the ordinary range. That is, the

signal can be simply reconstructed from $X(M\underline{n}, \Omega\,\underline{k})$ over $0 \leq M_1 n_1 \leq N_1$, $0 \leq M_2 n_2 \leq 2N_2$ for $0 \leq k_i \leq \frac{1}{2}M_i$. Effectively, this corresponds to using one quarter of the Fourier spectrum for each block, but using blocks taken over a double-sized image. This condition can be related to the case of using a real-valued window function of the form described in section 4.2. In this case, the Fourier spectrum would be symmetric in such a way that only half of the sample points, or two quadrants, are needed in each block. However, these would only be needed for blocks covering the original, unreflected image. That is, $0 \leq M_i n_i \leq N_i$. If one considers the meaning of the blocks in one of the other spatial quadrants, they simply correspond to windows over the reflected image. Thus, for each block in one spatial quadrant, there is a corresponding block in the others which windows the same portion of the image. Using one quarter of the frequency domain in two of these corresponding blocks is equivalent to the other case where two quarters of the each block are needed, but only from blocks in one spatial quadrant.

To this point, all that has been shown about the SSFT and its use in image coding is that it provides the same qualities as the block DCT. That is, it provides localization and decorrelation. There are, however, several important advantages to using the SSFT. The most important of these is the complete elimination of blocking effects. The most apparent reason for the lack of blocking effects is that the window functions are smooth and overlapping. Thus the

transition from one "block" to the next is gradual. It was stated previously that the cause of blocking effects was the non-stationary characteristic of the quantization noise. That is, the noise may be correlated within a block, but between blocks it will be independent. Quantization of the SSFT coefficients can be viewed as adding noise in independent bands of a bank of ideal bandpass filters. Since the bands are non-overlapping, the resulting noise field is obtained by simply summing the noise components from each band. Viewing the coding process in this way, it is clear that the noise will be stationary over all positions, within or between "blocks," and no blocking effects can occur.

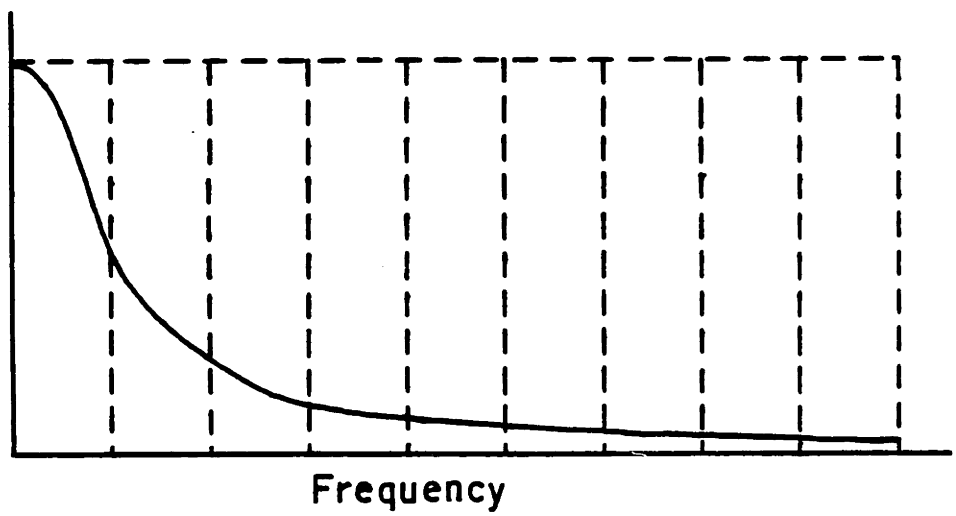
A second advantage of the SSFT is that it provides better energy compaction. This can be seen by considering how the SSFT is formed. The first step is to perform a DCT of the entire image. This closely approximates the K-L transform which provides the best decorrelation and compaction of any unitary transformation. Since the frequency bands of the SSFT are simply the successive, non-overlapping bands of the DCT, the energy within each band must be the maximum possible for any localized transform. All other localized transforms can also be expressed in terms of the DCT of the entire image. But these must all have cross terms which serve to reduce the energy in the low-frequency components and increase the energy in the high-frequency components, thus poorer compaction will result.

Another advantage of the SSFT is its insensitivity to the correlation coefficient of the image. It has been shown that for a relatively large image, the DCT closely approximates the K-L transform for nearly any value of the correlation coefficient. Small DCT's however, are poor approximations for values of the correlation coefficient not very close to unity [6]. The decorrelation of the SSFT depends only on the decorrelation of the DCT of the entire image since the bands are non-overlapping in this domain. As a result, the SSFT coefficients will maintain good decorrelation for any value of the correlation coefficient. In chapter 7, this property will be shown to be of value for interframe coding.

The final advantage of the SSFT is that, like a Fourier transform, it can be defined in terms of magnitude and phase components. Natural images are often composed of objects with sharp edge features. It has been noted that Fourier transform phase is more important in accurately representing these edge features than is magnitude. Since the SSFT is simply a windowed Fourier transform, the same property applies. For Fourier transform coding of natural images, it has been determined that by coding the phase component with one bit more than the magnitude results in improved performance over independent quantization of the real and imaginary parts [4]. Since the SSFT coefficients can also be separated into phase and magnitude, this method can also be used to improve its coding performance. The DCT does not have

the property of separability into phase and magnitude and thus the coding performance cannot be improved by using this technique.

As with the block cosine transform, there is an inherent trade-off in determining the proper choice of "block" size. For the SSFT, block size refers to the spacing between samples of the SSFT in the spatial domain. The basis for this trade-off can most easily be seen in the frequency domain. The variance of each coefficient in $\tilde{X}_{2N}$ for a typical set of images is shown below.



The blocks indicate one possible choice of window function width. By taking the inverse Fourier transforms over each windowed segment, the result is a spatial domain function with samples representing each spatial block of the SSFT. It is in the spatial domain that one can take advantage of the adaptability for coding. That is, it is possible to send few bits for those spatial regions that are smooth and many bits

in those that are detailed. If the frequency bands are widened, this corresponds to making the spacing of the blocks tighter. Since it is then possible to adapt to more rapid changes in the signal characteristics, the adaptability can be improved. One loss in doing this, however, is that, since the total number of blocks increases, more overhead information is needed to transmit the adaption information. Disregarding the effect of overhead, there is still an important factor which limits the minimum spacing of the blocks. Taking the inverse Fourier transform over each band "recorrelates" the coefficients within the band. The decorrelation and compaction occurs strictly between the bands. Thus, the wider the bands become, the less overall decorrelation and compaction that can be exploited. For the advantages of adaption, and decorrelation and compaction, a compromise window size must be used.

One important fact regarding the trade-off is that this "recorrelation" occurs differently at different frequencies. In the high-frequency range, the transform coefficients have relatively constant variance. This can be seen by the flatness of the high-frequency portion of the variance pattern shown above. As a result of this flatness, the inverse Fourier transforms produce coefficients which are still relatively uncorrelated. That is, the high-frequency components have very little correlation from one block to the next. In the very low frequencies, however, the slope of the spectrum is very steep. The inverse Fourier transforms in

this region of the spectrum will therefore produce coefficients which are highly correlated. That is, the low-frequency components of the SSFT may be highly correlated from block to block.

One simple way to account for this effect is to divide the frequency scale into two regions, where the bandwidth used in each region is different. If narrow bands are used for the low frequencies and wide bands are used for the high frequencies, the overall performance can be improved. By using narrow bands for the low-frequency coefficients, the high degree of compaction in this region can be exploited. Since the low-frequency components of most images are more uniformly distributed than the high-frequency components, there is very little that could be gained by using a high degree of adaption for these frequencies. But there is much to be gained by using the additional compaction. In the high frequencies, wide bands can be used in order to provide adaptability to rapid variations in the amount of image detail. Since the high-frequency portion of the spectrum is relatively flat, the low correlation of the high-frequency SSFT components between blocks may be maintained. It can be noted that this approach is similar to the two-component coding procedure discussed earlier. One choice for the two windows, which was used in the example images presented in the next section, is to use one by one bands in the lowest frequency block. Thus each point in what would otherwise be the lowest frequency band is coded independently.

It is not practical to vary the size of the bands over the entire range of frequencies. This is true since, for each group of frequencies with a specific block size, separate adaption information must be sent. If few frequency bands have a common width, or if all are of different widths, the amount of overhead information required would be unreasonably high.

6.3 Experimental Results

In order to demonstrate the effectiveness of the SSFT for low data rate image coding a comparison was made between block DCT coding and SSFT coding. To demonstrate the elimination of blocking effects, the same 256x256 pel image was first coded non-adaptively at a data rate of 0.35 bit/pel using each of the two methods described above. The DCT coder involved using 16x16 blocks of the image. The log-variance procedure was used to determine the proper bit assignment, and the coefficients were quantized with the optimal Max quantizer for Gaussian random variables [17]. The DC coefficient for each block was quantized with a uniform quantizer.

The SSFT coding was performed using a sample spacing of 16x16 pels. The SSFT coefficients were coded as phase and magnitude, using one extra bit for phase on each coefficient. The phase was quantized using a uniform quantizer and the magnitude was quantized using a Max quantizer for a Rayleigh random variable. The lowest band of frequencies was coded directly in the frequency domain of the entire image as

discussed in section 6.2. These coefficients were quantized using a Gaussian Max quantizer.

Figure 6.1 shows the original 256x256 test image. Figure 6.2 shows the DCT coded image and figure 6.3 shows the SSFT coded image. The signal-to-noise ratio of the DCT coded image was found to be 20.0 dB while that of the SSFT coded image was 21.3 dB. The most noticeable difference between the two images is the complete lack of blocking effects in the SSFT coded image. The increase in SNR of 1.3 dB can be attributed to several factors. One such factor is the coding of phase with one extra bit. Since the test image has a number of sharp edge features, the extra accuracy in the phase provides better alignment of the edges, thus improving the SNR.

To show that the SSFT lends itself well to adaptive coding techniques, a second SSFT coded image was produced. The method of adaption implemented is similar to that used by Wintz for block transform coding. Each SSFT block is coded with one of four possible bit and quantization assignments. The category that each block is put into is determined by a measure of the total energy within each of the two frequency quadrants corresponding to a particular spatial position. The energy in each quadrant is compared to a threshold and the result determines into which of the four categories the block should be placed. The bit and quantization assignment for each category is determined by using the log-variance procedure on the blocks in a test image which fall in that

category. By adapting, not only for the overall energy of a block, but also for the relative energy in the two quadrants of the frequency domain, variations in directionality as well as in detail are accounted for. Figure 6.4 shows the test image coded at a data rate of 0.29 bit/pel, a lower data rate than was used for the previous examples. The image was found to have a signal to noise ratio of 22.0 dB. This clearly demonstrates that the SSFT can be successfully used for adaptive coding.

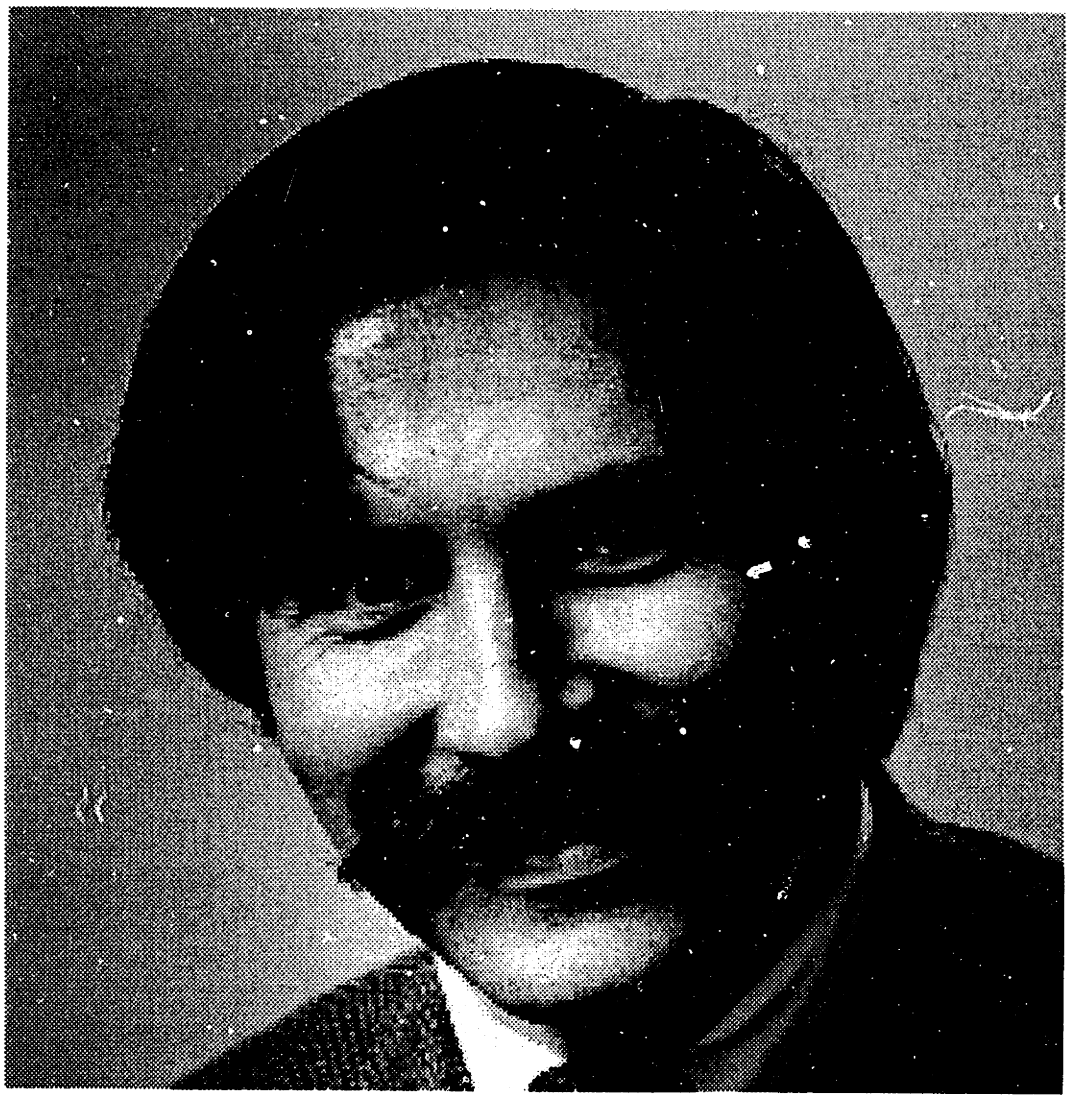


Figure 6.1 - The original 256x256 image.

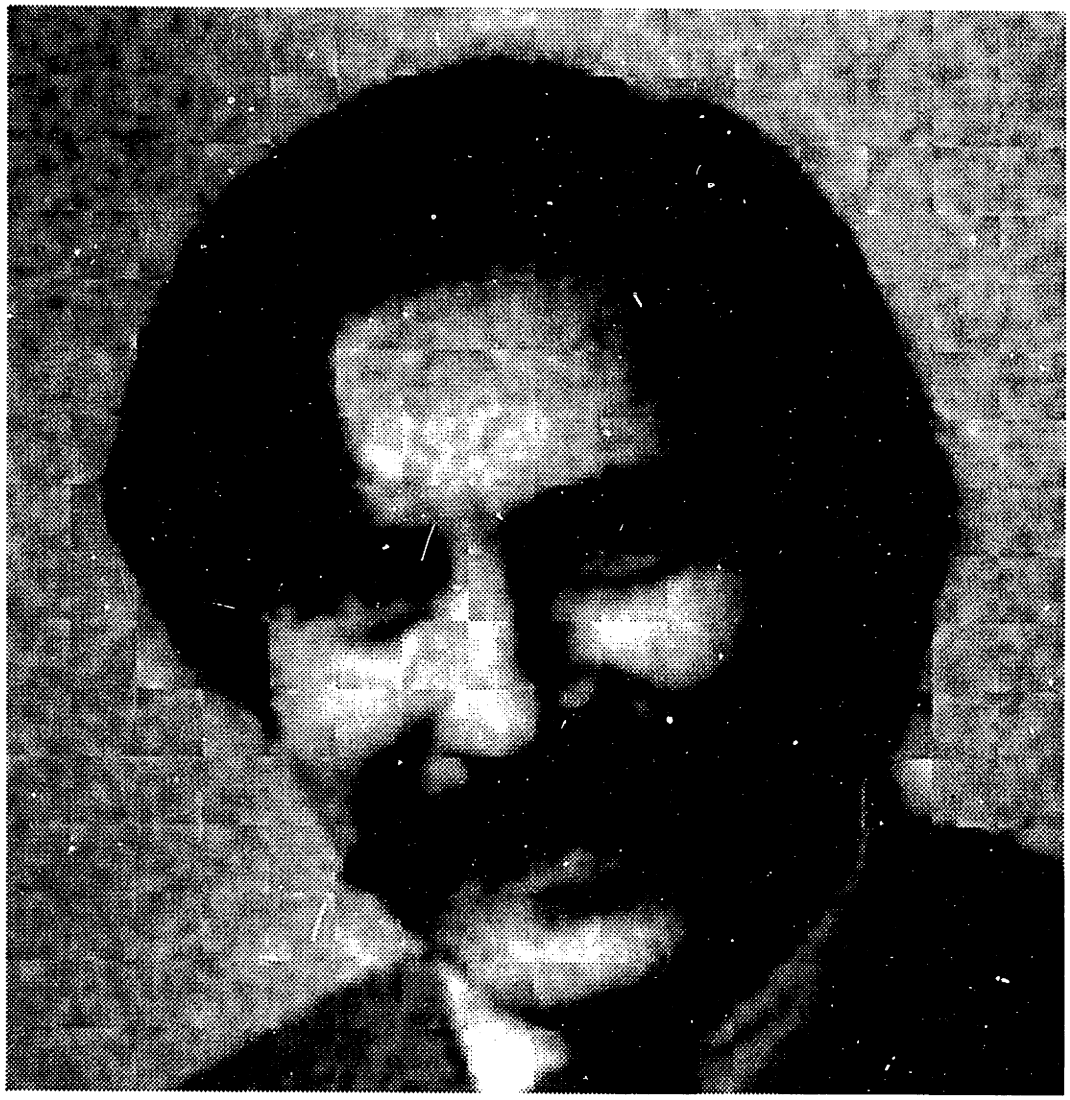


Figure 6.2 - Non-adaptive DCT coded at 0.35 bit/pel.
SNR = 20.0 dB.

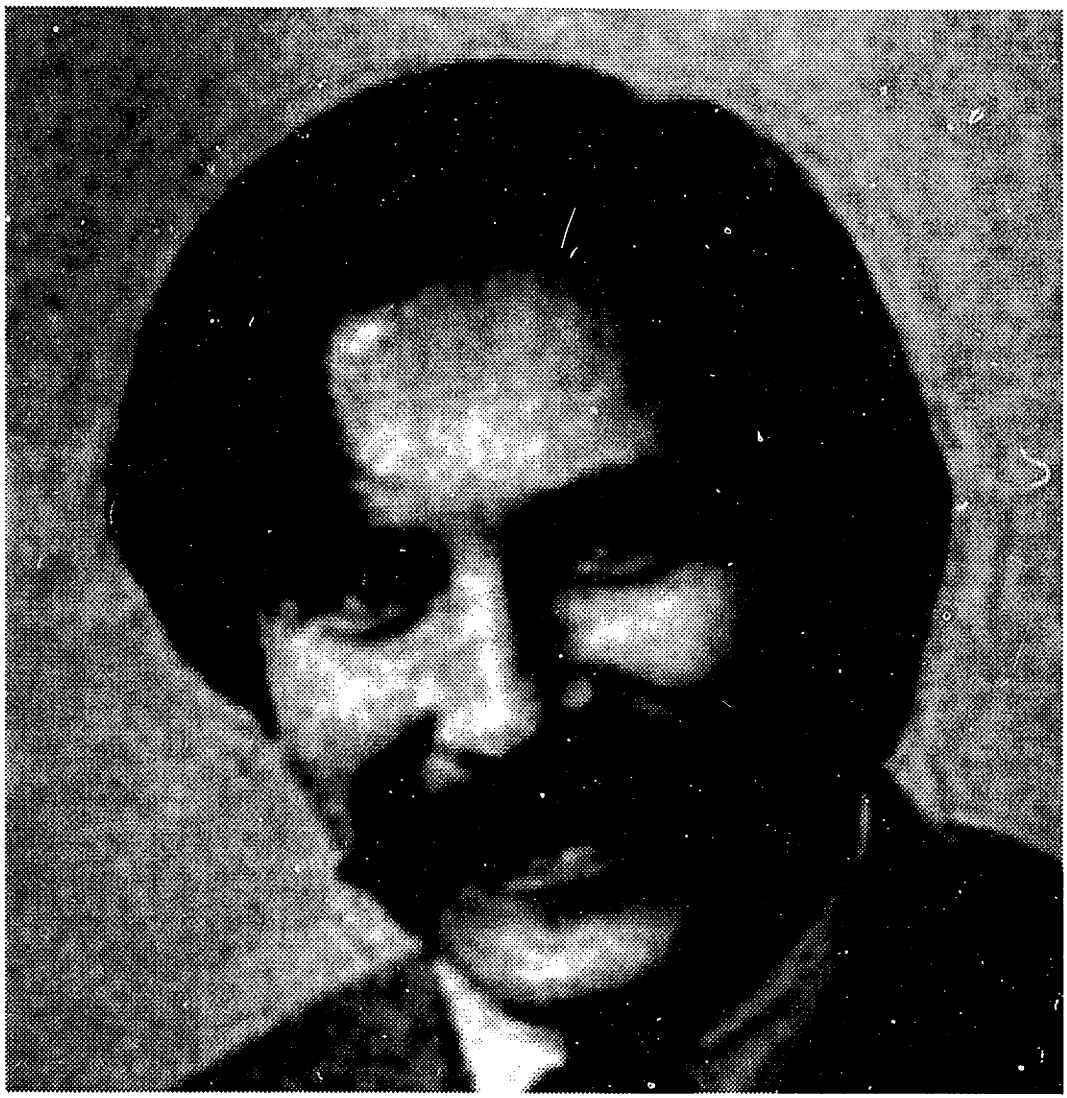


Figure 6.3 - Non-adaptive SSFT coded at 0.35 bit/pel. SNR = 21.3 dB.

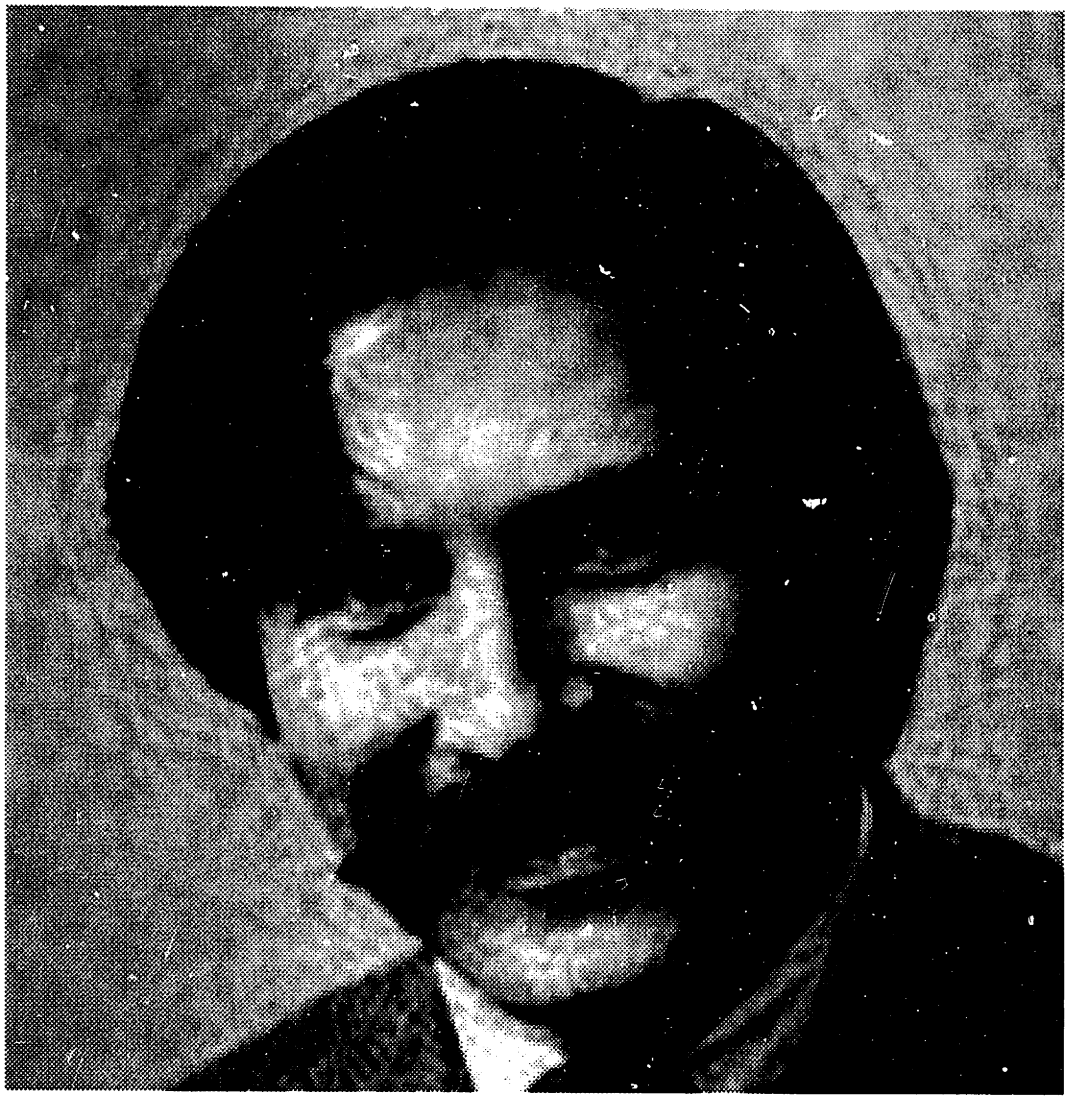


Figure 6.4 - Adaptive SSFT coded at 0.29 bit/pel.
SNR = 22.0 dB.

7. Application of the Short-Space Fourier Transform to Interframe Image Coding

7.1 Review of Interframe Image Coding Techniques

It is often the case that images to be encoded do not result from single, still images, but instead are part of a sequence of related images. For example, in broadcast television or video conferencing applications, each frame in a sequence is, in general, only slightly different from the previous frame. By accounting for the similarities between frames, it is possible to greatly reduce the amount of data required for transmission from what would otherwise be required. There are a wide variety of techniques that have been used in the past to perform efficient interframe coding. In this section, only those methods which involve transform coding will be reviewed. In the following section, it will be shown that the SSFT can be used with all of these techniques in order to prevent blocking effects.

In interframe coding, high efficiency is achieved by accounting for the correlation in both the spatial and temporal dimensions. One approach that has been successfully demonstrated is that of directly extending the method of block transform coding to three dimensions [24][28]. That is, by thinking of an image sequence as simply a three-dimensional random process, the correlation in all three dimensions can be exploited by using a three-dimensional block transform such as a DCT. This technique can be made to provide excellent coding efficiency. However, there are two problems which make this

method much less practical than others. First, all of the frames corresponding to the width of one block in the time dimension must be simultaneously stored before the encoding process can begin. For a fairly high resolution image sequence, this may require an extremely large amount of storage, making the cost of such a system very high. Second, because all of these frames must be stored before the blocks are coded and transmitted, there may be a sizable delay from the source to the destination. For one-way transmission such delays are acceptable. However, for two-way transmission, such as video conferencing, long delays are not tolerable.

The remaining techniques avoid these two problems by coding each frame using only knowledge of the single previous frame. In this way, only one frame need be stored at a time, and the delay can be made arbitrarily small, depending only on the computation time. The first of these methods is referred to as selective replenishment. In such a system, the image is first divided into blocks. The total error between each block and the corresponding block from the previosly transmitted frame is measured and compared to a threshold. If the error is less than the threshold, this means that there was very little change in the contents of the block since it was last sent, and thus does not need to be updated. All that is sent in this case would be the proper code informing the receiver that the block will not be updated. If the error is above the threshold, implying that a large change between frames, then the new block must be transmitted. Generally this is done by

block transform coding the new block. Since successive frames of a sequence often change only slightly, and often only in isolated areas, only a small percentage of the total number of blocks will need to be updated at each frame, thus the total amount of data that is sent may be very small.

Another technique for interframe coding is referred to as hybrid transform/DPCM coding [10][11]. As implied by its name, this technique employs differential coding in the time dimension combined with transform coding in the spatial dimension. Hybrid coding is very similar to selective replenishment coding. The difference between the two methods involves the way the blocks are updated. In selective replenishment, the actual updated blocks are transform coded and transmitted, while in hybrid coding, what is transmitted is the transform coded difference signal between a block to be updated and the previously transmitted block.

The operation of a hybrid coder can be viewed as involving a simple, fixed, linear predictor to predict the value of each frame from its predecessor. The predicted value at each pel is simply the value of the corresponding pel in the previous frame. It has been noted that in natural images, successive frames often consist of the same objects, moved non-uniformly in position. Instead of using a fixed predictor directly along the time axis, it is possible to use a variable predictor along the local direction of motion. This technique is known as motion compensation [12][15][20]. One implementation of this method involves first estimating a

single motion vector for each block of the image. The next frame is predicted using the pel values from the previous frame projected in the direction of motion. Since the prediction will in general be much better that for the DPCM system, the energy in the error signal will be greatly reduced, thereby greatly increasing coding efficiency. For each block, the value of the motion vector must be transmitted as overhead information. However, since natural images closely fit the motion-compensation model, the savings in coding the error signal greatly outweighs the expense of sending the motion vectors. A further advantage of motion-compensation coding is that it is possible to use lower frame rates than would be practical in the other coding techniques. For the other methods, if the frame rate is very low, the image will appear "jerky." It is possible to linearly interpolate between frames to remove this jerkiness. However, this results in blurring of the image. If motion compensation is used, interpolation can easily be done in the direction of the motion with little additional computation since the motion vectors are already known to the receiver. This results in a smooth transition between frames, even at very low frame rates [12].

7.2 The Use of the Short-Space Fourier Transform in Interframe Coding

By using block transform coding with all of these methods, blocking effects cannot be avoided. In moving sequences, blocking effects are especially annoying since, except for the motion-compensation technique, the blocks remain stationary while the image appears to move behind them, creating a "dirty window effect." In the same way that the SSFT was used to eliminate blocking effects in intraframe coding, it can also be used in interframe coding. In each of the four interframe transform coding techniques described in section 7.1, the SSFT can be substituted for the block transform.

For the case of three-dimensional transform coding, the SSFT cannot be used directly as defined in chapter 3. The signal to be coded is of finite extent in the two spatial dimensions and of infinite extent in the time dimension. Since the multidimensional transform can be defined in a separable manner, it is possible to use the SSFT in the two spatial dimensions and the STFT in the time dimension. This implies that a finite extent window function must be used in the time dimension. In order to compute this portion efficiently, the overlap-add or overlap-save techniques for polyphase filtering must be used, as discussed in chapter 2. While the performance of such a system would be very good, and blocking effects would be absent, the problems of the three-dimensional transform method discussed in section 7.1

are significantly worse with the SSFT. That is, if the analysis window is long in the time dimension, more frames of the image need to be stored simultaneously than for the block transform case. This further increases the memory requirements and the delay. One possible alternative to this approach is to implement a three-dimensional transform consisting of the SSFT in the spatial dimensions and the DCT in the time dimension. In this way, blocking effects will be avoided while the total amount of storage need not be any larger than for the three-dimensional DCT approach.

Application the SSFT to a selective replenishment system is straightforward. The frequency spectra at the sample positions of the SSFT are updated rather than the actual spatial blocks. Because the analysis windows localize the signal, only those samples close to areas of large change in the signal will generally need to be updated, thus selective replenishment can still be used. Similarly, in a DPCM system, the difference between the SSFT coefficients at a particular location and the coefficients for that location that were previously transmitted is sent to update the block.

Finally, the SSFT can also be used in similar fashion in a motion-compensated coding system. Such a system would operate by first making the motion-compensated prediction, generating the error image, and coding the resulting image using the SSFT, selectively transmitting the SSFT coefficients only at location of high energy in the error signal.

7.3 Experimental Results

In order to determine the performance of interframe coding systems using the SSFT, several experimental simulations were performed. The DPCM and motion-compensation techniques, as described above, were implemented using both the block DCT and the SSFT. The coding was done using a 128x120, 15 frame per second original image sequence. All of the simulated coding systems, were operating at a constant data rate of 56 Kbits/second. In the coding process, only alternate frames were transmitted, resulting in an actual frame rate of 7.5 frames per second. The intermediate frames were reconstructed at the receiver by interpolation. For the DPCM case, the interpolation simply involved averaging of the two surrounding frames. For the motion-compensation case, however, the transmitted motion vectors were used to perform motion-compensated interpolation.

For both the DCT and SSFT coders using DPCM, a constant data rate was achieved by updating a fixed number of blocks in each frame. For both cases, this was approximately twenty percent of the total number of blocks. The block size used was 8x8. Figure 7.1 shows the basic structure of the DPCM transmitter. The corresponding receiver is shown in figure 7.2. For each incoming frame, the transform is performed over the image and the difference is taken between the resulting transform coefficients and those of the reconstructed previous frame. The total energy in each block is then computed. The blocks are sorted by energy and only

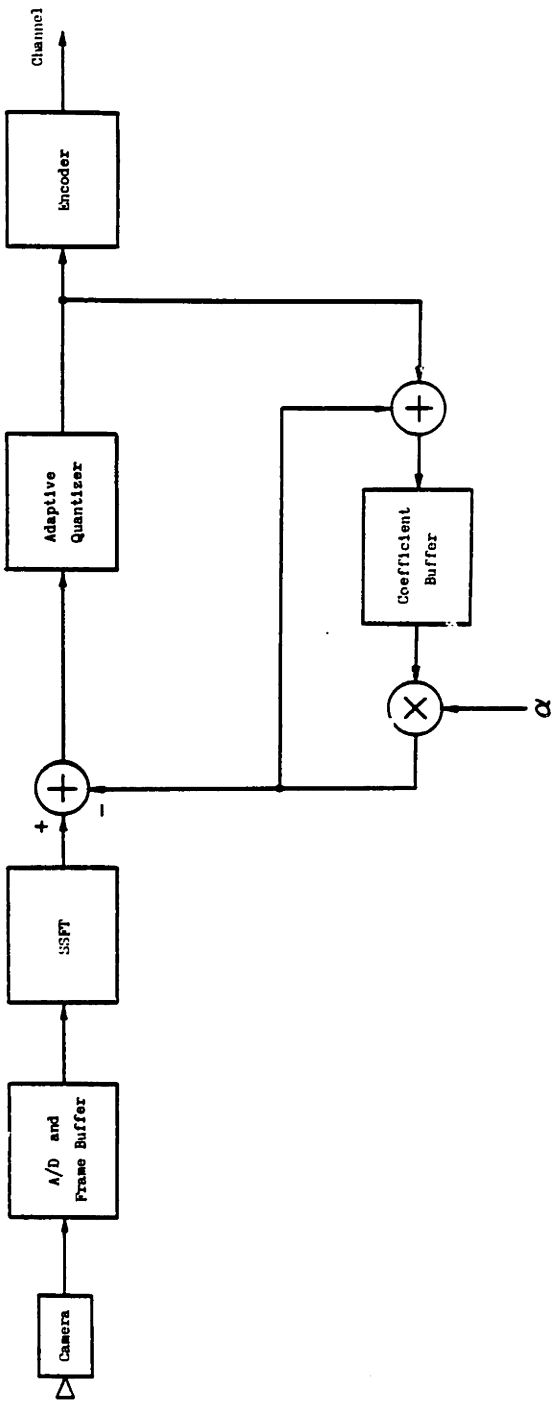


Figure 7.1 - Block diagram for a hybrid transform/DPCM transmitter.

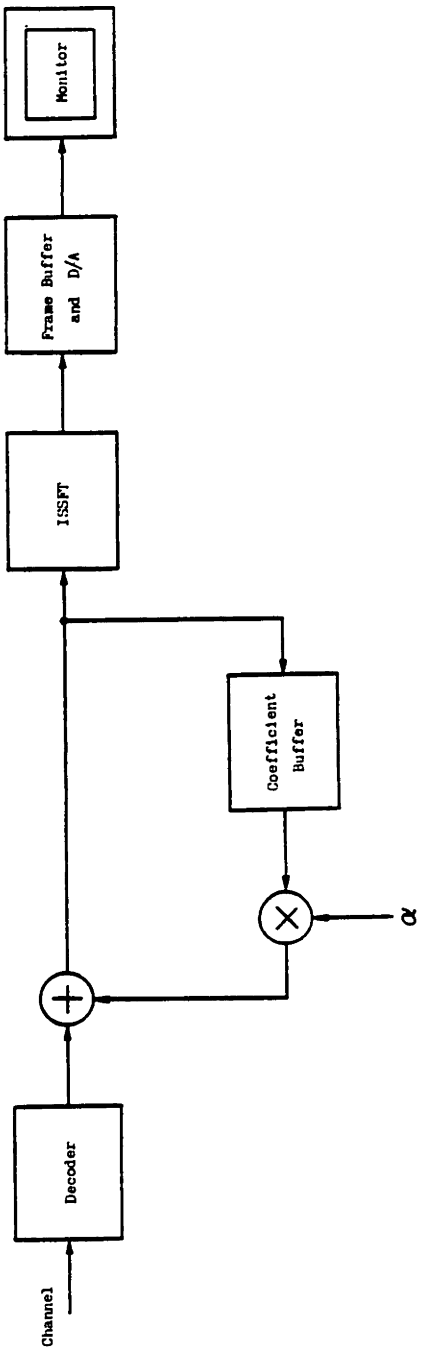


Figure 7.2 - Block diagram for a hybrid transform/DPCM receiver.

the highest twenty percent of the blocks are updated. Of the blocks transmitted, adaption is used to provide a further improvement in performance. The blocks are divided into four categories depending on their energies. In each of the four categories, a different bit assignment is used so that the most bits are assigned to the blocks with the largest error. To ensure a constant data rate, a fixed number of blocks are placed in each category. In order to account for both periods of inactivity and periods of rapid motion, the scaling of the quantizers is also adapted on the basis of the average energy of the blocks in each category.

For both transforms the low-frequency information was coded separately. For the DCT coder, the two-component source coding method [18][19][29][31], as described is section 6.1, was used. The block averages were coded using PCM, updating twenty five percent of these coefficients per frame. In the coding process, the bilinearly interpolated low-frequency components are subtracted from the image before the DCT coding begins. For the SSFT coder, frequency-dependent window functions were used, as discussed in section 6.2 and 6.3. For the lowest frequency band, rather than using one-by-one blocks as was used in the intraframe coding experiments, narrow bands were used which correspond to spatial block sizes of 32x32. This was done in order to allow for selective updating of coefficients. Again, only twenty five percent of these low-frequency blocks were updated in each frame.

In the motion-compensated coders, the method used to code the error signal is identical to that described above. The difference between the two coders is the addition of a motion-compesated predictor. The basic form of the motion-compensated transmitter is shown in figure 7.3. Figure 7.4 shows the corresponding reciever. Because of the fact that the motion compensation must be performed in the spatial domain, the additional computation of an inverse transform in the predictor loop is required. The motion estimation is done using an iterative technique based on the steepest-descent minimization algorithm [12][21]. Motion vectors are estimated using 4x4 blocks of the image. For each block, only a single vector is computed. In order to prevent discontinuities, the vector field is interpolated using a raised-cosine interpolation filter. The interpolated vector field is then used to generate the motion-compensated prediction which is subtracted from the incoming frame to produce the error signal. In order for the reciever to perform the motion compensation on the transmitted frames, the motion vectors must be coded and transmitted as overhead information. Because there is often a high degree of correlation in the motion vector field, it is possible to reduce the amount of data needed to transmit these vectors from what would be required if PCM were used to independently code each vector [12]. In the experimental system demonstrated here, adaptive transform coding was employed for this purpose. Specifically, the horizontal and vertical components of the initial 32x32

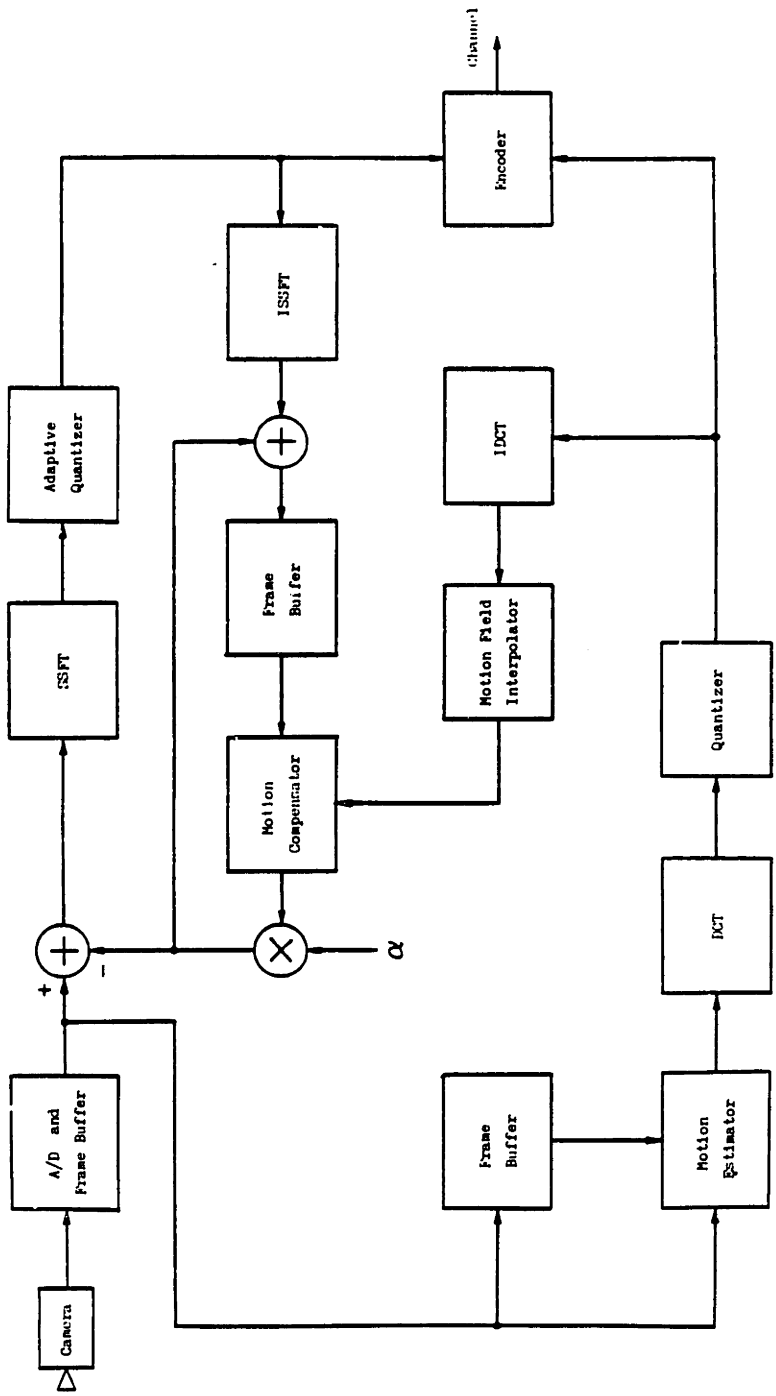


Figure 7.3 - Block diagram for a transform/motion compensation transmitter.

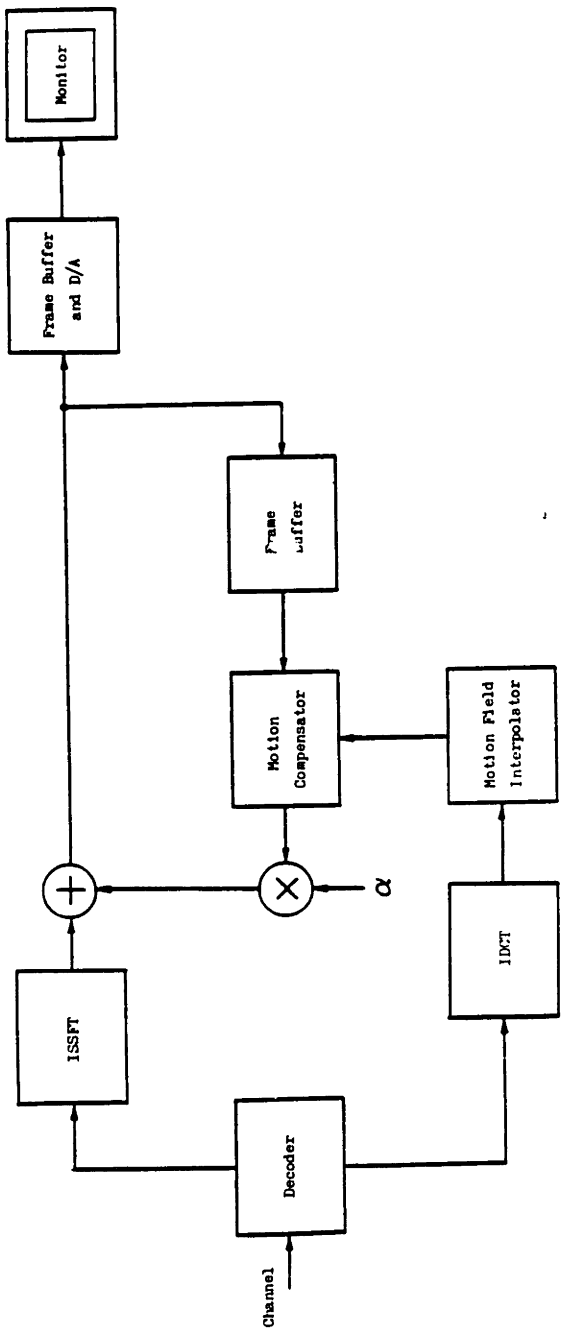


Figure 7.4 - Block diagram for a transform/motion compensation receiver.

vector field are coded using 8x8 block DCT's. The adaption is done by dividing these blocks into two groups, based on their total energy, and quantizing the blocks in each group using a different bit assignment. As stated above, intermediate frames were generated by interpolation in order to increase the frame rate to the original 15 frames per second. For the motion-compensated coders, this was done by using the motion vectors to generate an image moved half the distance used to generate the prediction image.

The four coding systems described above were simulated using 150 frames, or ten seconds, of an image sequence consisting of a head-and-shoulder subject. In order to account for the lack of an initial coded frame at the beginning of the sequence, a blank frame was used for this purpose and the coder was allowed to settle by repeating the first frame in the sequence twenty times before the remainder of the sequence was coded. For the DPCM coders, the resulting signal-to-noise ratios are plotted in figure 7.5 as a function of frame number. For clarity, the SNR's of the interpolated frames are not shown in the plot. Figure 7.6 shows the same comparison for the motion-compensated coders.

The sequence that was coded consisted of a period of inactivity (until approximately frame 10) followed by a period of moderate motion (from frame 10 to frame 70) and a period of extreme motion (after frame 70). From these plots it can be seen that for both the DPCM and the motion-compensated coders, the SSFT performed better than the DCT in the period

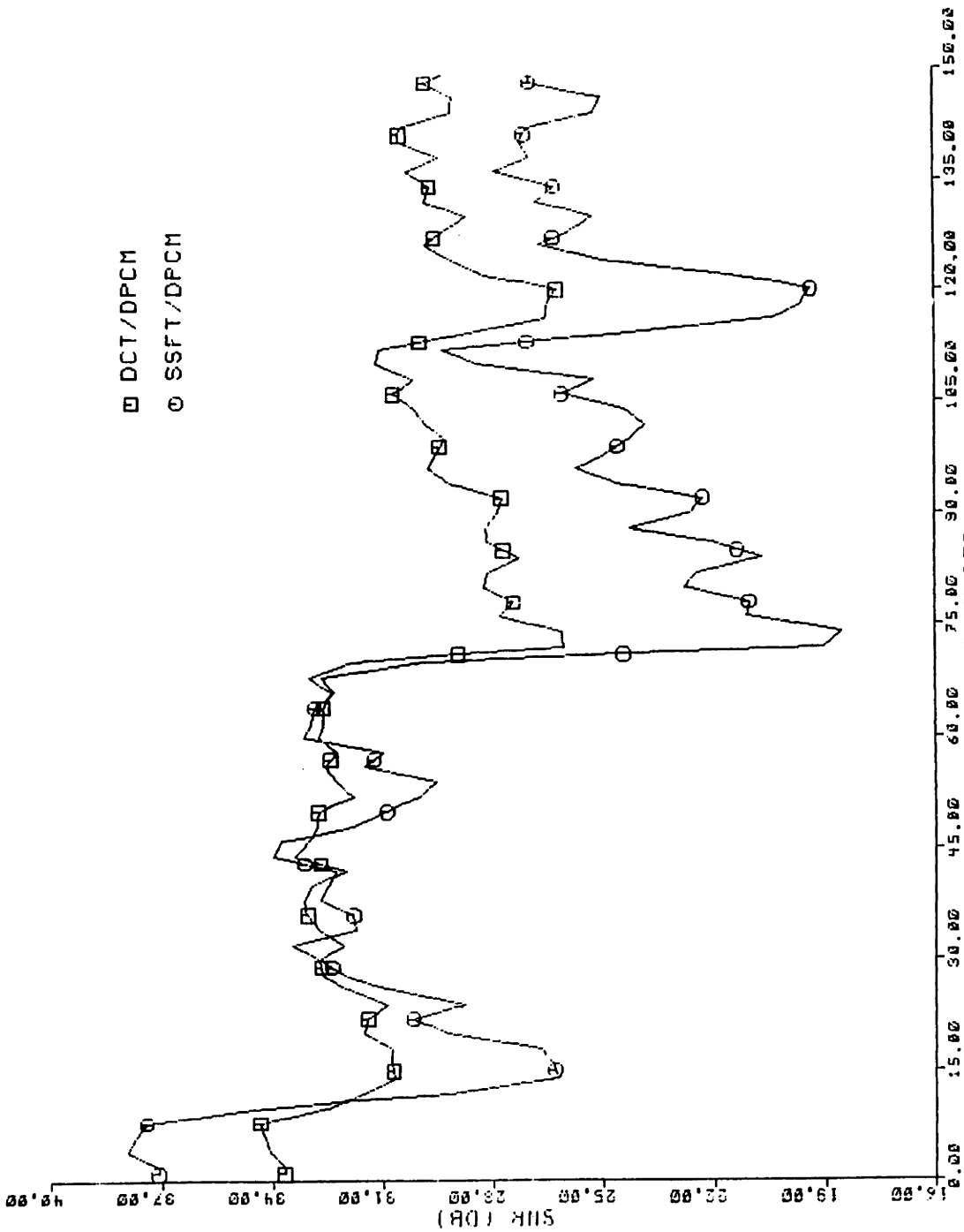


Figure 7.5 - A comparison of SNR vs. frame number for the DPCM coder using both the DCT and the SSFT.

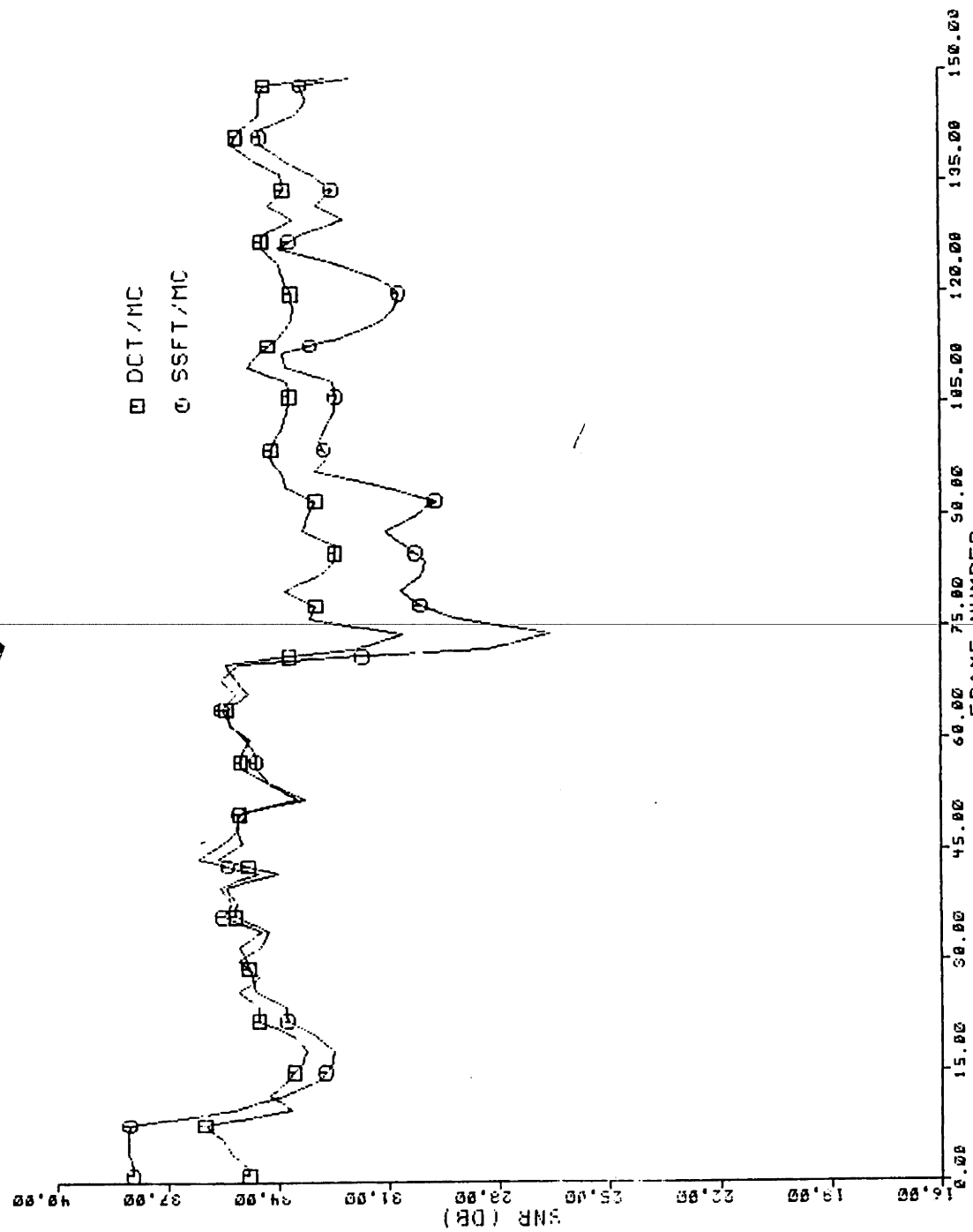


Figure 7.6 - A comparison of SNR vs. frame number for the motion compensated coder using both the DCT and the SSFT.

of inactivity, approximately the same as the DCT in the moderate motion period and worse than the DCT in the periods of high motion. From the characteristics of the DCT and the SSFT it is easy to explain this behavior. In the region of little motion, the error signal to be coded has very little energy. In addition, it has been found that as the error signals become smaller, the amount of spatial correlation also decreases. In fact, for the motion-compensated coder, the correlation coefficient may drop to nearly 0.55. It was noted earlier that the DCT does not closely approximate the optimal K-L transform for low-correlation signals [6]. However, the SSFT coefficients remain nearly uncorrelated for any value of correlation coefficient. This effect seems to explain the excellent performance of the SSFT in periods of inactivity. The poor performance of the SSFT during periods of extreme motion can also easily be explained. This is due to the spatial extent of the SSFT window functions. In background regions of the image, where there is very little change from frame to frame, one would expect that the blocks would rarely need updating. However, due to the fact that the window functions for these blocks extend into the regions of motion, there will be a small, but finite error introduced into these blocks by changes elsewhere in the image. Because of this, these blocks need to be updated more frequently than would be expected, using some of the data that could be used elsewhere. Since the basis functions of the DCT do not extend past the block boundaries, this problem is avoided.

In order to more clearly demonstrate the relative performance of the simulated coding systems, several frames from each sequence are shown. In Figure 7.7 the fifth frame from each of the four coding techniques is shown. This frame demonstrates the superior performance of the SSFT in regions of relative inactivity. The two frames from the DCT coded sequences are clearly noisier than those from the SSFT sequences. The next set of images, taken from frame 43 of each of the sequences, demonstrates the relative performance in a region of moderate motion. This is shown ir figure 7.8. There is very little visible difference between the images using the DCT and those using the SSFT. Figure 7.9, consisting of images taken from frame 91 of the four sequences clearly shows the poorer performance of the SSFT for periods of extreme motion. For both the DPCM and motion-compenstated coders, the noise within the area of motion is approximately the same for both the DCT and the SSFT. The main difference being the noticeable lack of blocking effects in the SSFT coded images. However, in the background regions there is clearly much more noise in the SSFT coded images. In addition to the effect indicated above, the fact that the noise is distributed into the backgrounds is also due to another factor. Another result of the large spatial extent of the window functions is that the background regions contain a small amount of information from the blocks in the regions of motion. In these blocks, there may be large quantization errors introduced in the coding process. After the inverse

SSFT is taken, a small amount of the error in these blocks is redistributed into the background areas.

To further demonstrate the coding performace, frame 5 and frame 73, from the motion-compensated coding simulation only, are compared. These frames represent the two extremes of motion: frame 5 occurs in a period of nearly complete inactivity while frame 73 occurs in the most active portion of the sequence. Figure 7.10 shows all four of these frames. These images further emphasize the relative advantages and disadvantages of the two coding techniques. In order to allow for more detailed inspection of these images, they are individually repeated in a much larger format in figures 7.11-7.14.

It seems from these results that, for interframe coding, the SSFT does not provide the overall increase in performance over the DCT that was hoped. The SSFT does have the advantages of improved performance in periods of inactivity and the complete elimination of blocking effects. However, the poor image quality in periods of rapid motion make its overall performance lower than the that of the DCT.

Figure 7.7 - Frame 5 from each of the four coded sequences.
Top left: SSFT/motion compensation.
Top right: DCT/motion compensation.
Bottom left: SSFT/DPCM.
Bottom right: DCT/DPCM.

Figure 7.8 - Frame 43 from each of the four coded sequences.
Top left: SSFT/motion compensation.
Top right: DCT/motion compensation.
Bottom left: SSFT/DPCM.
Bottom right: DCT/DPCM.

Figure 7.9 - Frame 91 from each of the four coded sequences.
Top left: SSFT/motion compensation
Top right: DCT/motion compensation.
Bottom left: SSFT/DPCM.
Bottom right: DCT/DPCM.

Figure 7.10 - Frames 5 and 73 from each of the motion compensation coded sequences.
Top left: SSFT/motion compensation, frame 5.
Top right: DCT/motion compensation, frame 5.
Bottom left: SSFT/motion compensation, frame 73.
Bottom right: DCT/motion compensation, frame 73.

Figure 7.11 - Frame 5 from the SSFT/motion compensation coded sequence.

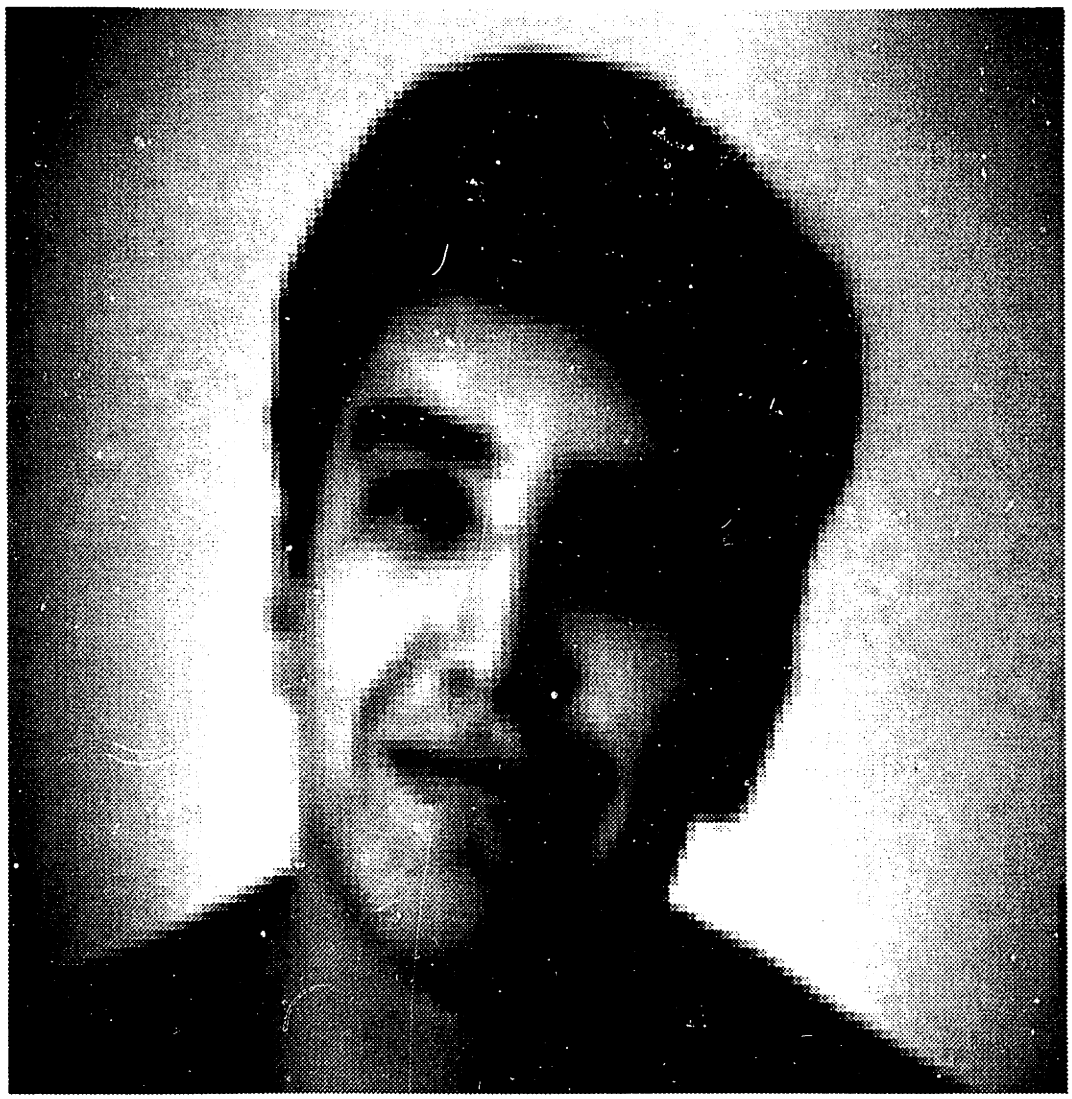


Figure 7.12 - Frame 5 from the DCT/motion compensation coded sequence.

Figure 7.13 - Frame 73 from the SSFT/motion compensation coded sequence.

Figure 7.14 - Frame 73 from the DCT/motion compensation coded sequence.

8. Conclusions

This thesis has demonstrated the value of the short-space Fourier transform for several applications. The properties of the SSFT were presented, along with a fast algorithm for its computation. It was shown that the SSFT could be used for intraframe image coding with better performance than traditional transform methods. Specifically, it can improve coding accuracy and completely eliminate blocking effects. Moreover, it has been demonstrated that the SSFT can also be used for interframe coding in order to avoid blocking effects. However, due to the spread of the SSFT basis functions, the overall performance of the SSFT interframe coding systems implemented was slighly poorer than that of traditional coding techniques.

As a suggestion for further research, it would be desirable to correct the one major flaw with the SSFT for purposes of interframe image coding. That is, the coding efficiency could be much improved by using a transform with basis functions that do not extend throughout the image. If the basis functions only extend into the surrounding blocks, for example, this problem would be avoided. If these basis functions smoothly overlapped between adjacent blocks, then such a transform would still avoid the problem of blocking effects.

Another possible area of future research includes further investigation into the uses of the SSFT for linear filtering. It has not yet been demonstrated, for example, if

the SSFT could successfully be used for image restoration or motion compensation applications.

REFERENCES

[1] N. Ahmed, T. Natarajan and K.R. Rao, "Discrete Cosine Transform," IEEE Transactions on Computers, Vol. C-23 No. 1, pp. 90-93, Jan. 1974.

[2] J.B. Allen, "Short-Term Spectral Analysis, Synthesis and Modification by Discrete Fourier Transform," IEEE Transactions on Acoustics, Speech and Signal Processing, Vol. ASSP-25, No. 3, pp. 235-238, June 1977.

[3] J.B. Allen and L.R. Rabiner, "A Unified Approach to Short-Time Fourier Analysis and Synthesis," Proceedings IEEE, Vol. 65, No. 11, pp. 1558-1564, Nov. 1977.

[4] H.C. Andrews and A.G. Tescher, "The Role of Adaptive Phase Coding in Two and Three Dimensional Fourier and Walsh Image Compression," Proceedings of the Walsh Function Symposium, Washington DC, March 1974.

[5] M. Bellanger and J.L. Daguet, "TDM-FDM Transmultiplexer: Digital Polyphase and FFT," IEEE Transactions on Communications, Vol. COM-22, No. 9, pp.1199-1205, Sept. 1974.

[6] R.J. Clark, "Performance of Karhunen-Loeve and Discrete Cosine Transforms for Data Having Widely Varying Values of Intersample Correlation Coefficient," Electronic Letters, Vol. 19, No. 7, pp. 251-253, March 31, 1983.

[7] R.E. Crochiere and L.R. Rabiner, Multirate Digital Signal Processing, Englewood Cliffs, NJ: Prentice Hall, 1983.

[8] D.E. Dudgeon and R.M. Messereau, Multidimensional Digital Signal Processing, Englewood Cliffs, NJ: Prentice Hall, 1984.

[9] A. Gersho, "Characterization of Time-Varying Linear Systems," Proceedings IEEE, p. 238, Jan. 1963.

[10] A. Habibi, "Hybrid Coding of Pictorial Data," IEEE Transactions on Communications, Vol. COM-22, No. 5, pp. 614-624, May 1974.

[11] A. Habibi, "An Adaptive Strategy for Hybrid Image Coding," IEEE Transactions on Communications, Vol. COM-29, No. 12, pp. 1736-1740, Dec. 1981.

[12] B.L. Hinman, "Theory and Applications of Image Motion Estimation," MSEE Thesis, MIT, May 1984.

[13] B.L. Hinman, J.G. Bernstein and D.H. Staelin, "Short-Space Fourier Transform Image Processing," Proceedings International Conference on Acoustics, Speech and Signal Processing, San Diego, CA, pp. 4.8.1-4.8.4, March 1984.

[14] A.K. Jain, "Image Data Compression: A Review," Proceedings IEEE, Vol. 69, No. 3, pp. 349-389, March 1981.

[15] J.R. Jain and A.K. Jain, "Displacement Measurement and its Application to Interframe Image Coding," IEEE Transactions on Communications, Vol. COM-29, No. 12, pp. 1799-1808, Dec 1981.

[16] B.G. Lee, "FTC-A Fast Cosine Transform," Proceedings International Conference on Acoustics, Speech and Signal Processing, San Diego, CA, pp. 28A.3.1-28A.3.4, March 1984.

[17] J. Max, "Quantizing for Minimum Distortion," IRE Transactions on Information Theory, Vol. IT-6, No. 1, pp. 7-12, March 1960.

[18] A.Z. Meiri, "The Pinned Karhunen-Loeve Transform of a Two-Dimensional Gauss-Markov Field," Proceedings SPIE Conference on Image Processing, San Diego, CA, 1976.

[19] A.Z. Meiri and E. Yudilevich, "A Pinned Sine Transform Image Coder," IEEE Transactions on Communications, Vol. COM-29, No. 12, pp. 1728-1735, Dec. 1981.

[20] A.N. Netravali and J.O. Limb, "Picture Coding: A Review," Proceedings IEEE, Vol. 68, No. 3, pp. 336-406, March 1980.

[21] A.N. Netravali and J.D. Robbins, "Motion-Compensated Television Coding: Part 1," Bell Systems Technical Journal, Vol. 58, No. 3, pp. 631-669, March 1979.

[22] H.J. Nussbaumer, "Polynomial Transform Implementation of Digital Filter Banks," IEEE Transactions on Acoustics, Speech and Signal Processing, Vol. ASSP-31, No. 3, pp. 616-622, June 1983.

[23] M.R. Portnoff, "Time-Frequency Representation of Digital Signals and Systems Based on Short-Time Fourier Analysis," IEEE Transactions on Acoustics, Speech and Signal Processing, Vol. ASSP-28, No. 1, pp. 55-69, Feb 1980.

[24] W.K. Pratt, Digital Image Processing, New York: Wiley, 1978.

[25] L.R. Rabiner and R.W. Schafer, Digital Processing of Speech Signals, Englewood Cliffs, NJ: Prentice Hall, 1978.

[26] H.C. Reeve, "Reduction of Blocking Effect in Image Coding," MSEE Thesis, MIT, Jan. 1983.

[27] H.C. Reeve and J.S. Lim, "Reduction of Blocking Effects in Image Coding," Proceedings International Conference on Acoustics, Speech and Signal Processing, Boston, MA, pp. 1212-1215, April 1983.

[28] J.A. Roese, W.K. Pratt and G.S. Robinson, "Interframe Cosine Transform Image Coding," IEEE Transactions on Communications, pp. 1329-1339, Nov. 1977.

[29] J.M. Schumpert and R.J. Jenkins, "A Two-Component Image Coding Scheme Based on Two-Dimensional Interpolation and the Discrete Cosine Transform," Proceedings International Conference on Acoustics, Speech and Signal Processing, Boston, MA, pp. 1232-1235, April 1983.

[30] P.A. Wintz, "Transform Picture Coding," Proceedings IEEE, Vol. 60, pp. 809-820, July 1972.

[31] J.K. Yan and D.J. Sakrison, "Encoding of Images Based on a Two-Component Source Model," IEEE Transactions on Communications, Vol. COM-25, pp. 1315-1322, Nov. 1977.