Video Teleconferencing Algorithms

by

Sarah A. Rajala

Center for Communications and Signal Processing

Department of Electrical and Computer Engineering
North Carolina State University

November 1985

CCSP-TR-85/16

ACKNOWLEDGMENTS

Dr. Sarah A. Rajala would like to acknowledge the efforts of the following graduate students who made this research possible.

Robert J. Moorhead
Alireza F. Faryar
Peter Santago
Larry W. Cook
Raman A. Nobahht

Table of Contents

I. INTRODUCTION

Much research has been done in recent years in an attempt to find better ways to transmit video information over low bandwidth channels. This research has resulted in the development of full motion video systems for the T1 carrier bandwidths. However, with these high compression factors the quality of the reconstructed imagery at the receiver is significantly lower than "broadcast quality". In addition, there is a continuing demand for further reductions in bandwidths without further loss in quality.

Video compression can be viewed as a problem in determining the minimum information necessary to reconstruct a signal at a receiver which represents, within some specified accuracy, the signal which entered the transmitter. The calculation of the minimum information needed to exactly reconstruct any signal has been defined by Shannon. We cannot sample at a rate less than twice the highest frequency component (spatial or temporal) in the signal without a loss in information. Abiding by Shannon's sampling theorem guarantees zero error reconstruction. The problem at hand, however, forces us below the rate that Shannon proves is necessary for exact reconstruction. As a result, we are faced with the problem of finding the subset of information to transmit which can subsequently be used to reconstruct a signal at the receiver with some desired accuracy.

This report describes the research performed in the CCSP Enhancement Project entitled, "Video Teleconferencing" from July 1, 1984—June 30, 1985. The research concentrated on finding new methods for compressing the video information to yield better picture quality at lower bandwiths. To accomplish this goal, three areas were address: the collection and description of statistical information contained in video sequences; the

development of a new pel-recursive motion-compensated video teleconferencing algorithm; and the evaluation of the use of deconvolution techniques for improved image reconstruction at the receiver. Specifically, Section II presents the results of the study of image statistics. In Section III, a description of the new pel-recursive motion-compensated algorithm is given along with the analysis of its convergence properties and simulation results. Section IV contains the analysis of the use of deconvolution techniques, Section V states the conclusions, and Section VI indicates several areas for further research.

II. THE CHARACTERIZATION OF IMAGE STATISTICS

II.1 Introduction

Part of the research in this project was dedicated to collecting and categorizing the information contained in video sequences in a statistical way. This task was intended to enhance our understanding of the kinds of information in a sequence and determine which of that information is important for transmission. The types of information characterized were percent motion; mean and standard deviation of the entire image, the noise, and the motion window; mean and standard deviation of the difference sequence for a full image sequence and for the motion regions; correlation information for the noise, the entire image, the motion window; and correlation information for the difference sequence for a full image sequence and for the motion regions in a sequence.

II.2 Percent Motion

Three image sequences were used as a data base for the collection of the image statistics. The three sequences each contained 60 frames (two seconds). A pair of pictures for each sequence (bobsjob, map, and robot) is shown in Figures II.1 - II.3

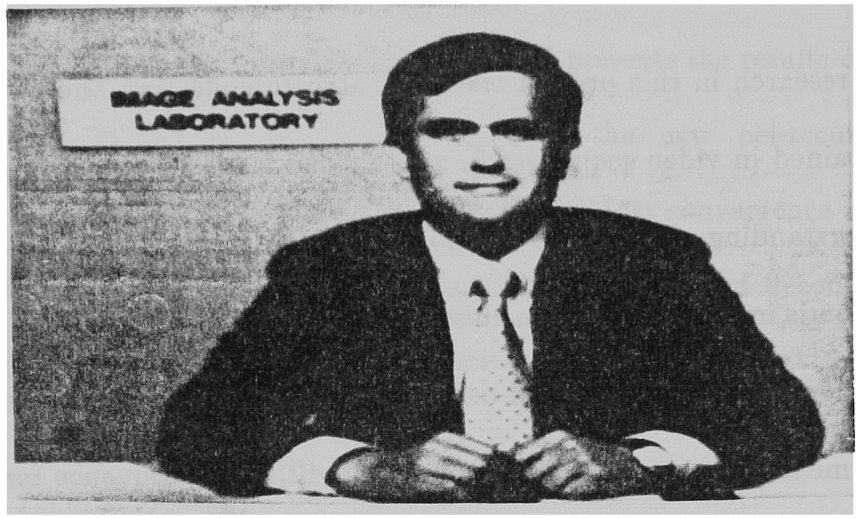

Figure II.1a: Bobsjob, frame 0

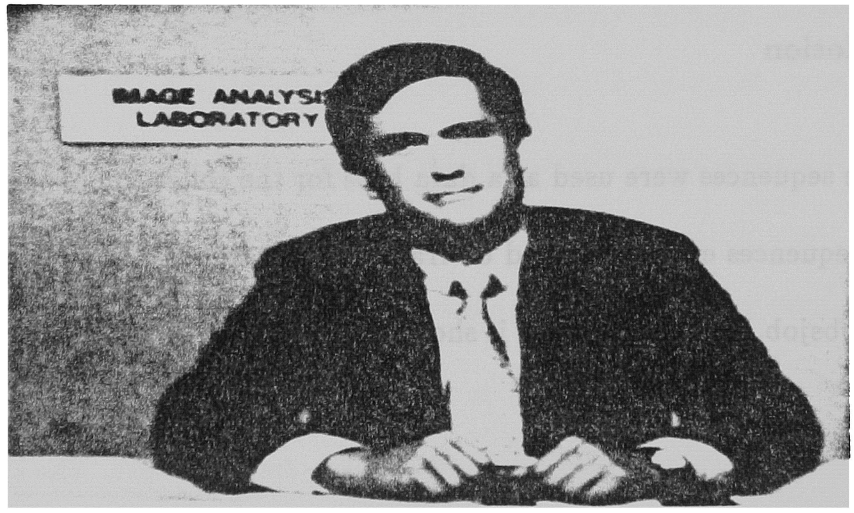

Figure II.2a: Map, frame 0

Figure II.2b: Map, frame 60

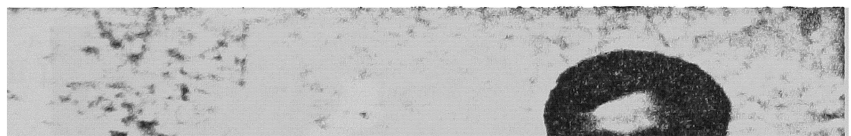

Figure II.3a: Robot, frame 0

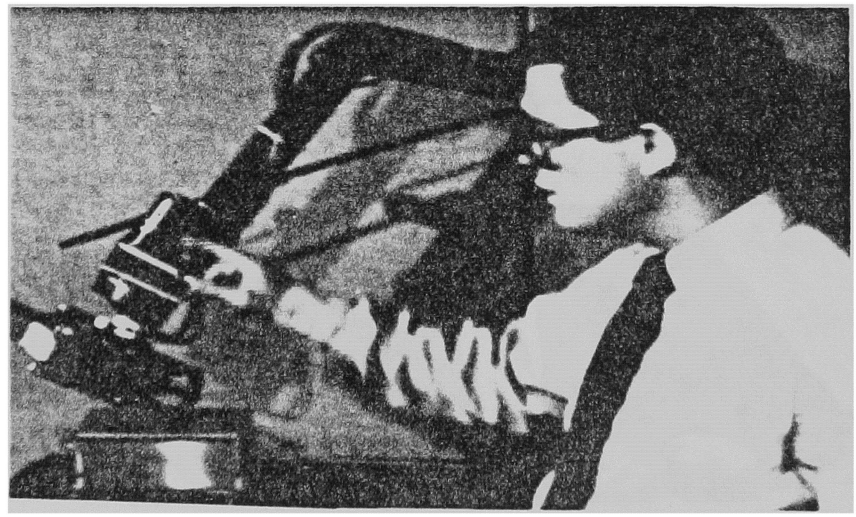

Figure II.3b: Robot, frame 60

The original sequences have the standard intensity resolution of 8 bits/pel and a spatial resolution of 282 lines/frame by 448 pels/line. This yields an uncompressed bit rate of 1.01 Mbits/frame. Each frame contains two interlaced fields.

The percentage of interframe motion for each of the three sequences is plotted in Figure II.4 -II.6. Moving pixels were defined as those that had a magnitude intensity change of more than three between consecutive frames. It is clear from these figures that percent motion can vary considerably over the length of the sequence, but in each of these cases the maximum percent motion is less than 25%.

II.3 Histograms

The histograms of the entire image, the noise, the the motion window, the difference picture, and the difference of the motion window were calculated. From these histograms estimates were made of the mean and the standard deviation. The results are shown in Figure II.7 - II.11.

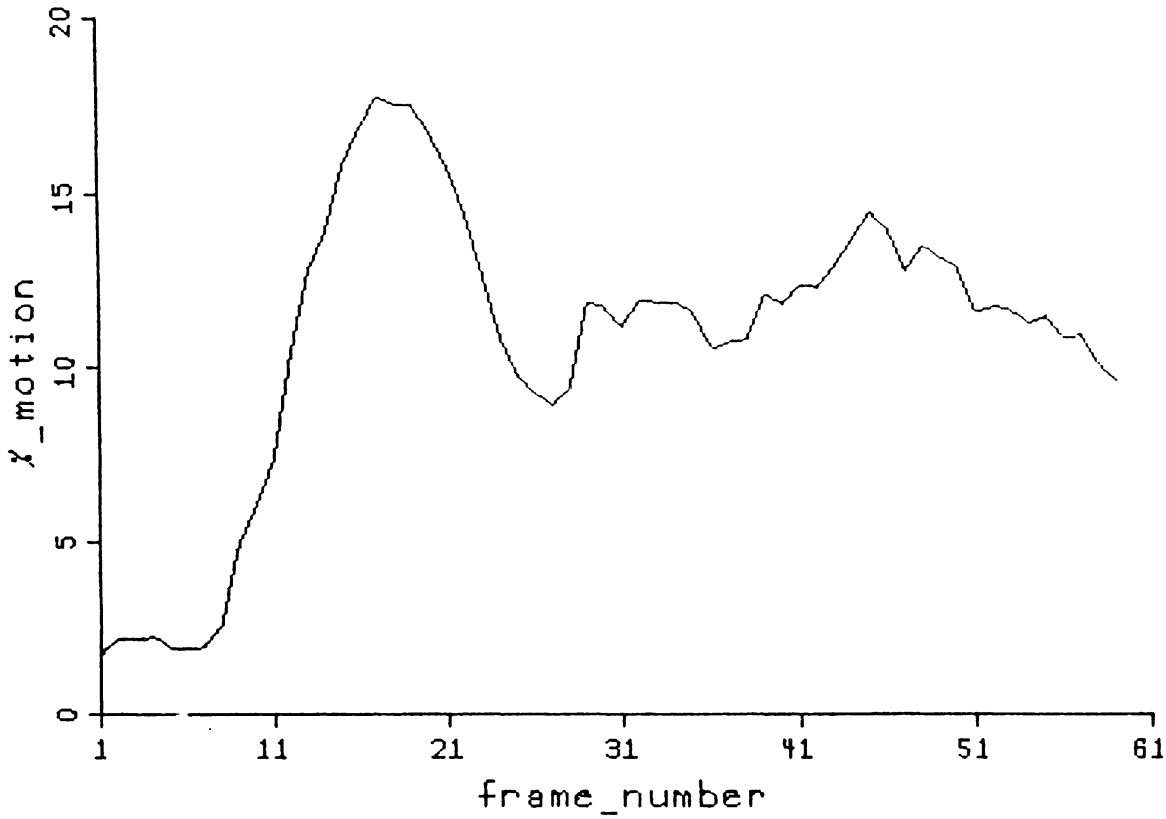

Figure II.4: Percent Interframe Motion, bobsjob sequence

Figure II.5: Percent Interframe Motion, map sequence

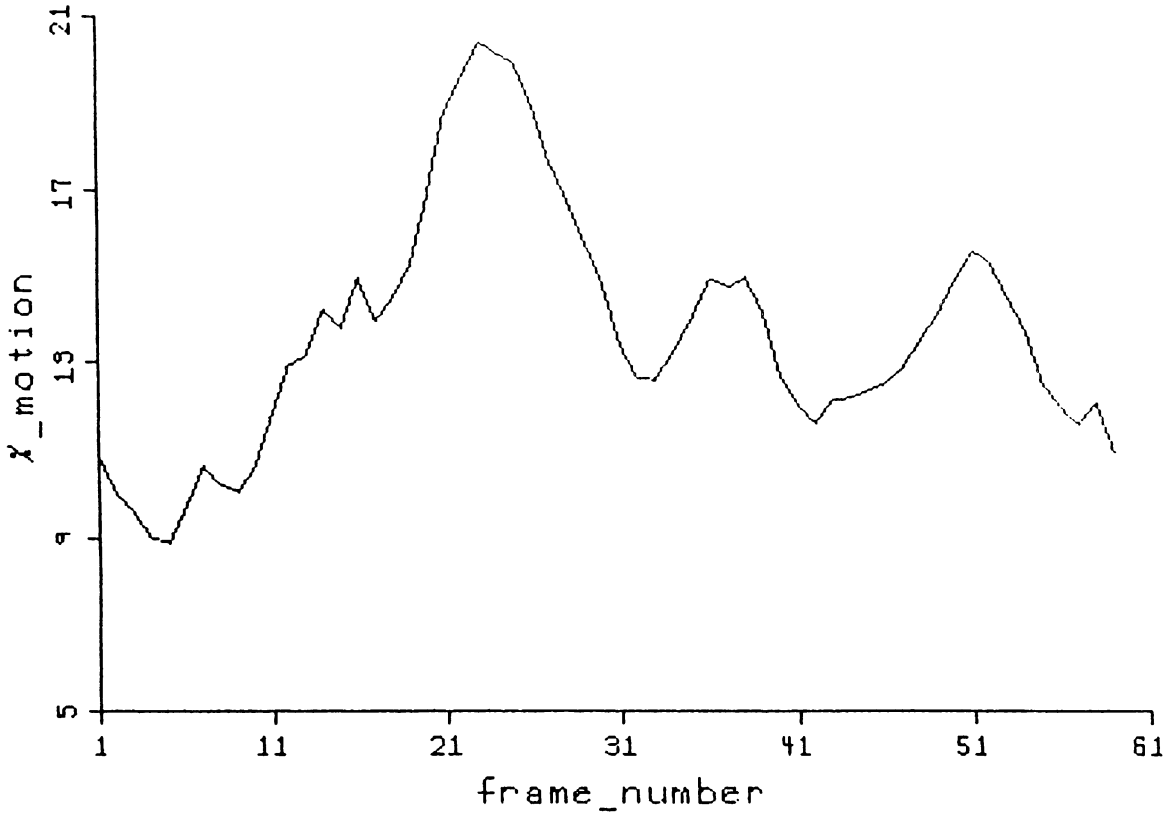

Figure II.6: Percent Interframe Motion, robot sequence

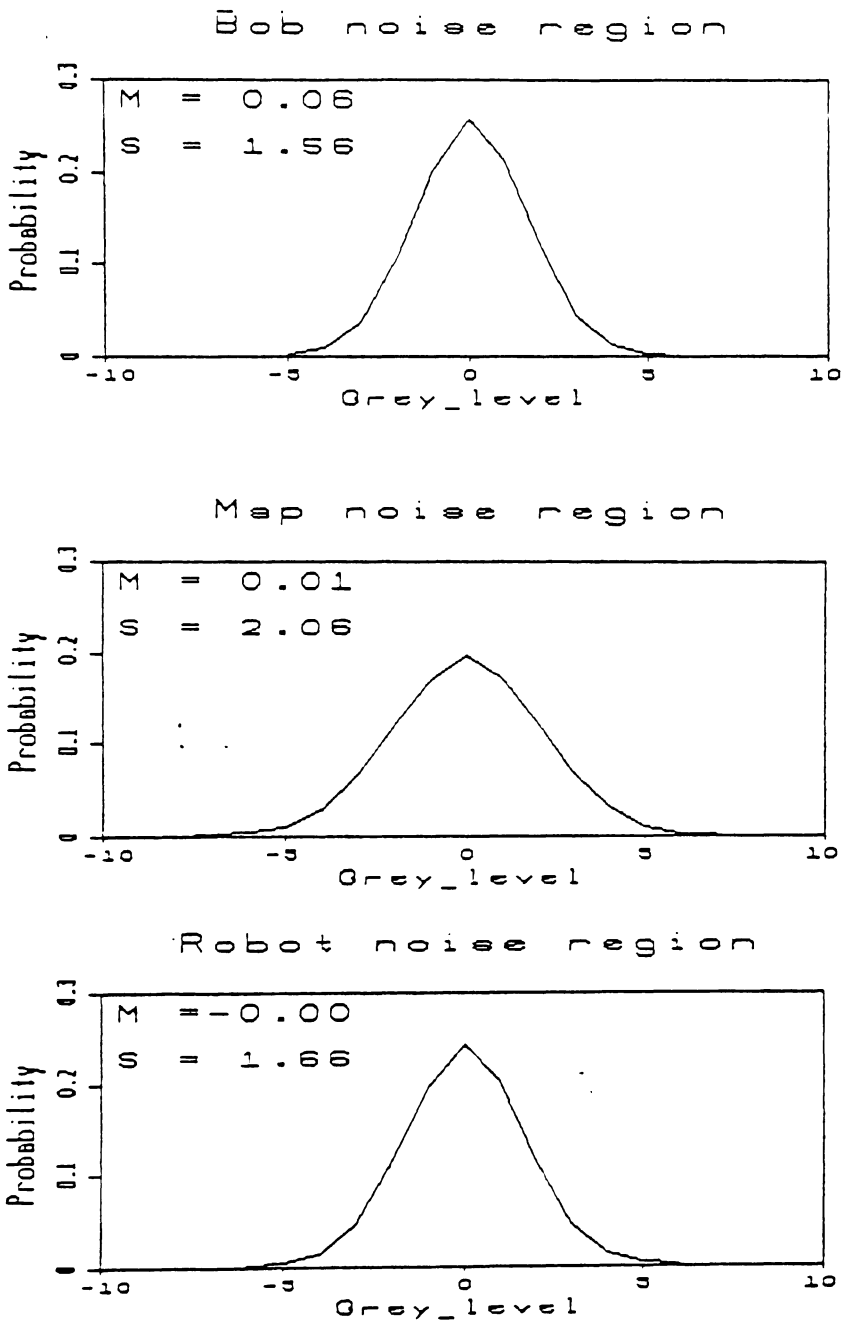

Figure II.7: Image Sequence Noise Statistics

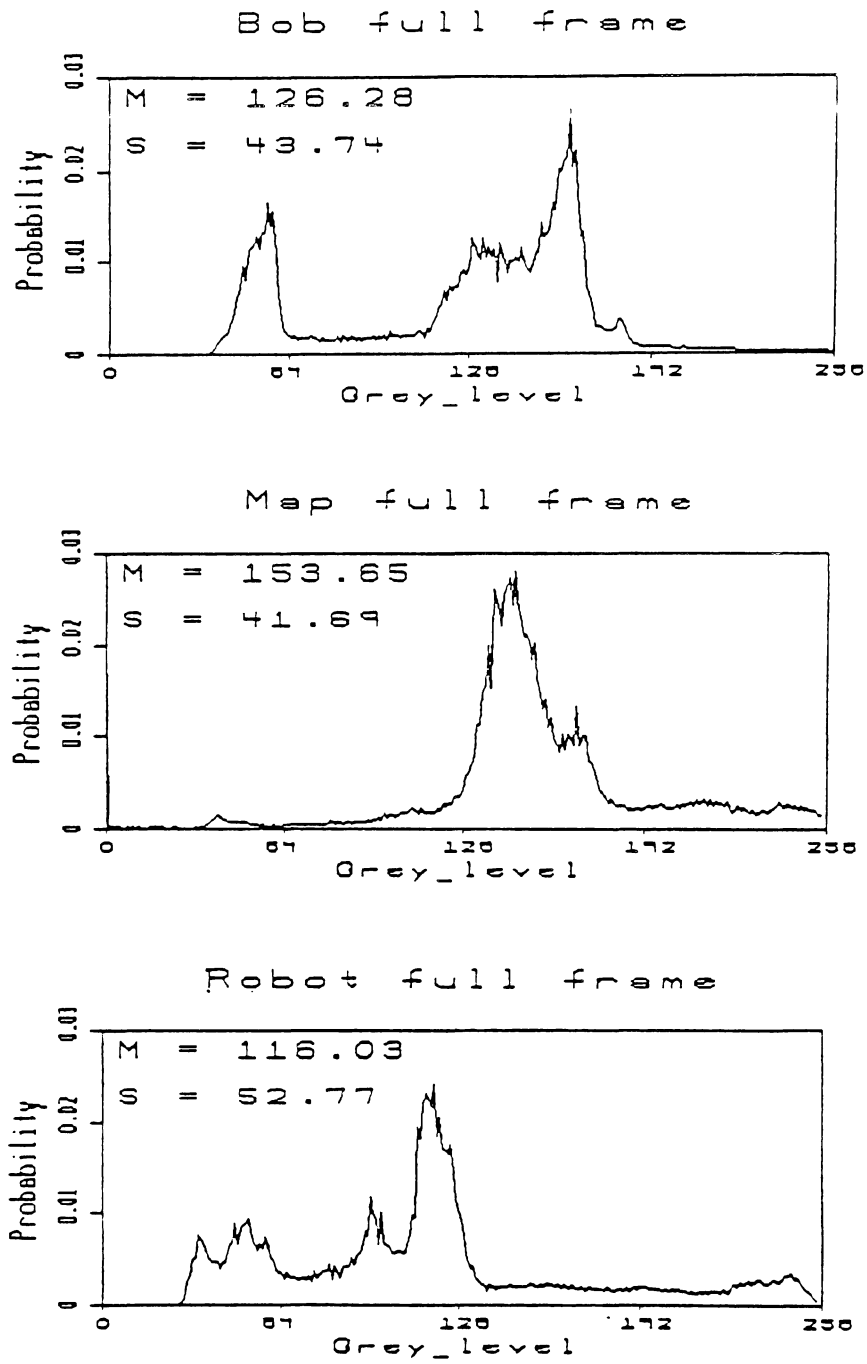

Figure II.8: Image Sequence Grey Level Histograms (Full Frame)

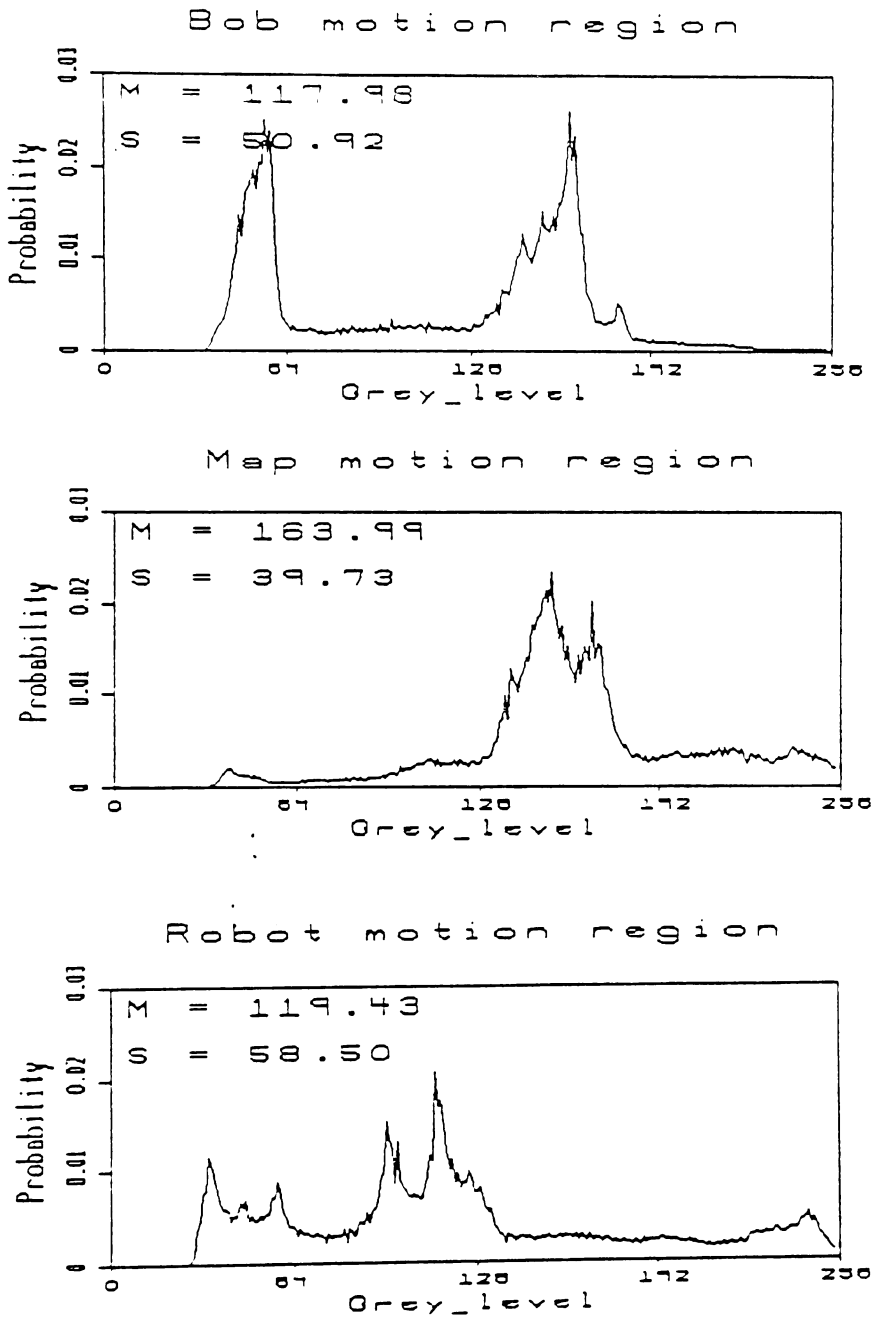

Figure II.9: Image Sequence Grey Level Histograms (Motion Window)

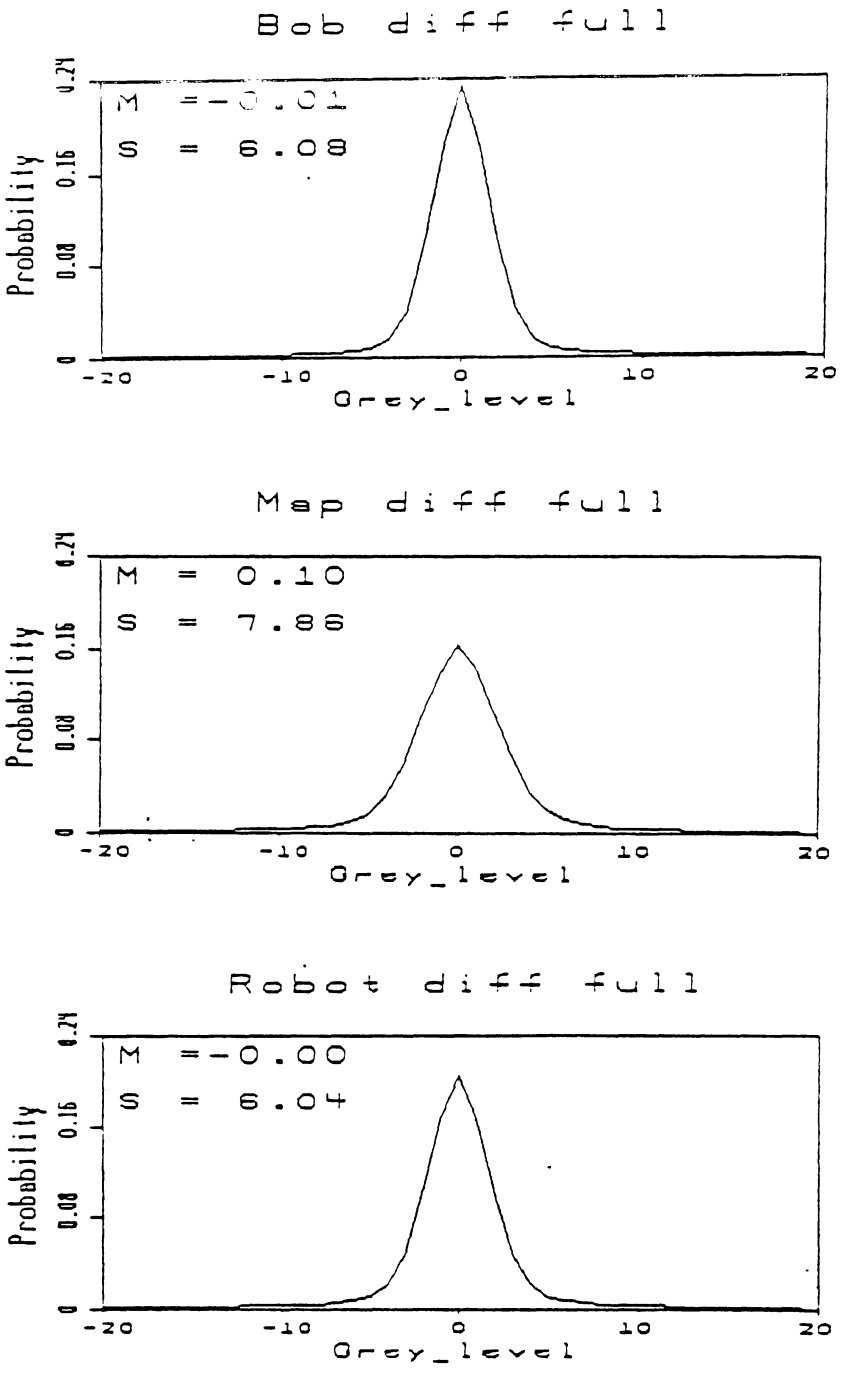

Figure II.10: Difference Sequence Grey Level Histograms (Full Frame)

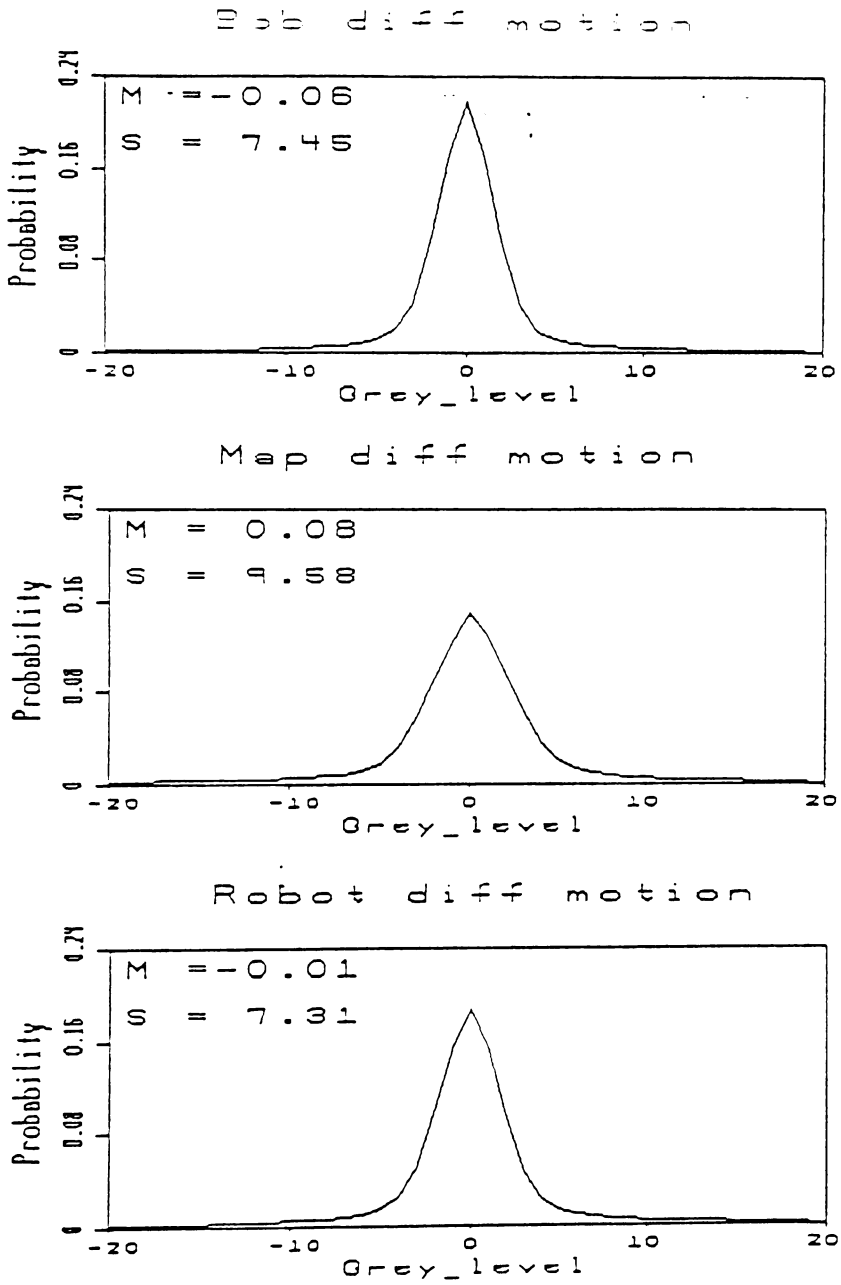

Figure II.11: Difference Sequence Grey Level Historgrams (Motion Window)

Figure II.7 contains the histograms of the noise for each of the three image sequences. In each case the histograms closely approximate a Gaussian waveform with a mean equal to zero. It should be noted that the standard deviations for each sequence are slightly different. The noise was determined by forming a temporal differencing operation in a background (non-moving) region. This should insure that any changes were due to noise only. A motion compensation scheme was applied to this region in order to compensate for any possible camera jitter. The algorithm had no effect on the statistics.

Figure II.8 contains the histograms of the entire frame for each of the three image sequences. Here, the histograms appear to be more irregular with little likelihood of characterizing this information with a standard waveform.

Figure II.9 contains the histograms of the motion windows for each of the three image sequences. Again the histograms are irregular. These motion windows were 256x320 pixels. The windows contain all of the sequence motion as well enough background to be consistent with a realistic teleconferencing situation. The predominant information that is missing from the windows is the background nonmoving parts of the picture, with variations due only to the noise.

Figure II.10 contains the histograms of the difference picture sequences generated from the full-sized images. In this case, the histograms appear to be more Laplacian in shape, with means very close to zero.

Figure II.11 contains the histograms of the difference picture sequences generated from the motion windows. The histograms, once again, appear to be Laplacian with means approximately equal to zero. No Laplacian fit was performed with this data, but the Laplacian density has been predicted for difference frames in other publications.

II.4 Correlation Information

The covariance and the normalized correlation coefficient were calculated for the noise, the full-sized image, the motion window, the difference sequence, and the difference sequence of the motion window. The results are tabulated in Figures II.12 - II.16.

Figure II.12 contains a table of the covariance and the correlation coefficient values for the noise in the three sequences. It can be seen from these results that the noise is almost perfectly uncorrelated except for the coefficients which only have a temporal time lag. In this case, the correlation of 0.5 is most likely due to 60 cycle noise from the power source.

Figure II.13 contains a table of the covariance and the correlation coefficient values for the full-sized image sequences. It is clear that all of the pels in these images are highly correlated with their adjacent neighbors. In fact, the correlation is higher than is normally used with Gauss-Markov sources employed to model image sequences.

Figure II.14 contains a table of the covariance and the correlation coefficient values for the motion window. Even in this situation, the correlation is extremely high. In fact, it is not statistically different than the previous case. This is probably due to the fact that the motion window still contains a large amount of nonmoving background.

Figure II.15 contains a table of the covariance and the correlation coefficient values for the difference sequence of the full-sized image. The correlation in these sequences is much lower.

Figure II.16 contains a table of the covariance and the correlation coefficient values for the difference sequence of the motion window.

Bobsjob difference sequence - noise region.								
(k',l',t')	(0,0,0)	(0,1,0)	(1,0,0)	(1,1,0)	(0,0,1)	(0,1,1)	(1,0,1)	(1,1,1)
covariance	2.417	-.2422	-.0251	-.0380	-1.228	.1408	.0529	.0307
normalized	1	.1002	.0104	.0157	.5078	.0582	.0219	.0127

Map difference sequence - noise region.								
(k',l',t')	(0,0,0)	(0,1,0)	(1,0,0)	(1,1,0)	(0,0,1)	(0,1,1)	(1,0,1)	(1,1,1)
covariance	4.248	-.3153	-.0100	-.0193	-2.115	.2087	-.0103	.0019
normalized	1	.0742	.0023	.0045	.4979	.0491	.0024	.0004

Robot difference sequence - noise region.								
(k',l',t')	(0,0,0)	(0,1,0)	(1,0,0)	(1,1,0)	(0,0,1)	(0,1,1)	(1,0,1)	(1,1,1)
covariance	2.761	-.2876	-.0028	-.0107	-1.381	.1655	-.0022	.0301
normalized	1	.1042	.0010	.0039	.5002	.0600	.0008	.0109

Figure II.12: Image Sequence Noise Correlation

Bobsjob actual sequence - full frame								
(k',l',t')	(0,0,0)	(0,1,0)	(1,0,0)	(1,1,0)	(0,0,1)	(0,1,1)	(1,0,1)	(1,1,1)
covariance	1998	1974	1964	1949	1977	1960	1962	1949
normalized	1	.9880	.9826	.9755	.9895	.9807	.9819	.9755

Map actual sequence - full frame								
(k',l',t')	(0,0,0)	(0,1,0)	(1,0,0)	(1,1,0)	(0,0,1)	(0,1,1)	(1,0,1)	(1,1,1)
covariance	1275	1259	1252	1242	1241	1244	1231	1232
normalized	1	.9875	.9823	.9743	.9738	.9755	.9660	.9665

Robot actual sequence - full frame								
(k',l',t')	(0,0,0)	(0,1,0)	(1,0,0)	(1,1,0)	(0,0,1)	(0,1,1)	(1,0,1)	(1,1,1)
covariance	2867	2834	2831	2801	2846	2806	2816	2777
normalized	1	.9888	.9876	.9771	.9927	.9790	.9824	.9689

Figure II.13: Image Sequence Pixel Correlation (Full Frame)

Bobsjob actual sequence - window region								
(k',l',t')	(0,0,0)	(0,1,0)	(1,0,0)	(1,1,0)	(0,0,1)	(0,1,1)	(1,0,1)	(1,1,1)
covariance	2676	2645	2632	2611	2648	2623	2630	2612
normalized	1	.9882	.9832	.9756	.9892	.9800	.9827	.9761

Map actual sequence - window region								
(k',l',t')	(0,0,0)	(0,1,0)	(1,0,0)	(1,1,0)	(0,0,1)	(0,1,1)	(1,0,1)	(1,1,1)
covariance	1697	1677	1667	1653	1648	1653	1635	1637
normalized	1	.9882	.9822	.9740	.9711	.9742	.9635	.9646

Robot actual sequence - window region								
(k',l',t')	(0,0,0)	(0,1,0)	(1,0,0)	(1,1,0)	(0,0,1)	(0,1,1)	(1,0,1)	(1,1,1)
covariance	3507	3479	3479	3446	3477	3438	3456	3412
normalized	1	.9921	.9919	.9826	.9914	.9802	.9854	.9727

Figure II.14: Image Sequence Pixel Correlation (Motion Window)

Bobsjob difference sequence - full frame								
(k',l',t')	(0,0,0)	(0,1,0)	(1,0,0)	(1,1,0)	(0,0,1)	(0,1,1)	(1,0,1)	(1,1,1)
covariance	36.80	31.85	23.68	22.65	12.34	14.44	20.20	19.96
normalized	1	.8653	.6434	.6154	.3354	.3924	.5490	.5424

Map difference sequence - full frame								
(k',l',t')	(0,0,0)	(0,1,0)	(1,0,0)	(1,1,0)	(0,0,1)	(0,1,1)	(1,0,1)	(1,1,1)
covariance	61.91	51.95	38.20	35.19	9.877	16.98	14.64	16.58
normalized	1	.8392	.6171	.5684	.1595	.2742	.2365	.2679

Robot difference sequence - full frame								
(k',l',t')	(0,0,0)	(0,1,0)	(1,0,0)	(1,1,0)	(0,0,1)	(0,1,1)	(1,0,1)	(1,1,1)
covariance	36.34	29.19	23.02	19.74	9.074	4.672	13.32	8.257
normalized	1	.8032	.6336	.5431	.2497	.1286	.3666	.2272

Figure II.15: Difference Sequence Pixel Correlation (Full Frame)

Bobsjob difference sequence - window region								
(k',l',t')	(0,0,0)	(0,1,0)	(1,0,0)	(1,1,0)	(0,0,1)	(0,1,1)	(1,0,1)	(1,1,1)
covariance	55.32	49.13	36.43	34.87	19.60	22.15	31.07	30.73
normalized	1	.8880	.6585	.6303	.3542	.4003	.5617	.5555

Map difference sequence - window region								
(k',l',t')	(0,0,0)	(0,1,0)	(1,0,0)	(1,1,0)	(0,0,1)	(0,1,1)	(1,0,1)	(1,1,1)
covariance	91.95	79.26	57.98	53.50	15.63	25.24	21.86	24.78
normalized	1	.8619	.6306	.5818	.1700	.2744	.2377	.2695

Robot difference sequence - window region								
(k',l',t')	(0,0,0)	(0,1,0)	(1,0,0)	(1,1,0)	(0,0,1)	(0,1,1)	(1,0,1)	(1,1,1)
covariance	53.30	44.24	36.70	29.87	13.87	6.575	19.68	12.26
normalized	1	.8301	.6886	.5604	.2603	.1234	.3693	.2299

Figure II.16: Difference Sequence Pixel Correlation (Motion Window)

II.5 CONCLUSION

The statistics generated are in reasonable agreement with those used for modeling and analysis, although the correlation may be a little high. Also it should that the variance for the noise is for the difference sequence and that the variance for the actual sequence would be one half of this value. Further, the motion percentage is probably over estimated. This problem is being studied.

These statistics provide valuable informaton when analyzing algorithms and generating image sequence models. They can be used to confirm any assumptions used which before were only educated guesses or based on other data.

III. CONVERGENCE CRITERIA ANALYSIS FOR A PEL-RECURSIVE MOTION-COMPENSATED CODEC

III.1 Introduction

The objective in image sequence compression is to minimize the amount of information that must be transferred. Previous work [51,55,75] has shown that predicting the motion and accessing the intensity values from the previous frame (or field) generally results in a better prediction of the intensity values than trying to predict the intensity values solely from previous intensity information.

Although motion-compensated interframe coding was introduced in the 1970's, relatively little analytical work has been done to prove convergence of the motion estimate or to develop rates of convergence. As a result, many of the proposed algorithms and their real-time implementations are ad hoc. By analyzing the convergence requirements and the convergence rates of the basic motion-compensated technique, a more rigorous algorithm can be developed.

In Section III.2 the basic technique is reviewed to establish the notation. In Section III.3 the convergence analysis is performed. In Section III.4 the new pel-recursive motion-compensated algorithm is introduced and in Section III.5 some simulation results are presented. The gradient estimation noise covariance is derived in an Appendix A.

III.2. The Basic Pel-recursive Motion-compensated Algorithm

Netravali and Robbins [51] developed a pel-recursive technique for motion-compensated coding. The intensity values within a frame are represented by $I(\mathbf{z},t)$, where $\mathbf{z}$ is a two-dimensional spatial vector and t is the frame at time t. If an object moves with purely translational motion, then for some $\mathbf{d}$, where $\mathbf{d}$ is the two-dimensional spatial translation displacement vector of the object point during the time interval $[t-1,t]$,

$$I(\mathbf{z},t) = I(\mathbf{z}-\mathbf{d},t-1). \tag{III-1}$$

Define a function called the displaced frame difference:

$$DFD(\mathbf{z},\hat{\mathbf{d}}^i) = I(\mathbf{z},t) - I(\mathbf{z}-\hat{\mathbf{d}}^i,t-1), \tag{III-2}$$

where $\hat{\mathbf{d}}^i$ is an estimate of the translation vector. The DFD converges to zero as $\hat{\mathbf{d}}^i$ converges to the actual displacement, $\mathbf{d}$, of the object point. An iterative equation to find $\mathbf{d}$ using the DFD function can be developed using the gradient method [73]:

$$\hat{\mathbf{d}}^{i+1} = \hat{\mathbf{d}}^i - \frac{\epsilon}{2}\nabla_{\hat{\mathbf{d}}}[DFD(\mathbf{z}_a,\hat{\mathbf{d}})]^2\Big|_{\hat{\mathbf{d}}=\hat{\mathbf{d}}^i}, \tag{III-3}$$

where $\nabla_{\hat{\mathbf{d}}}$ is the two-dimensional gradient operator with respect to displacement $\hat{\mathbf{d}}$ and ϵ is the convergence coefficient. After algebraic manipulations, the resulting iterative equation for correcting the displacement estimation is:

$$\hat{\mathbf{d}}^{i+1} = \hat{\mathbf{d}}^i - \epsilon DFD(\mathbf{z}_a,\hat{\mathbf{d}}^i)\nabla_{\mathbf{z}}I(\mathbf{z}-\hat{\mathbf{d}}^i,t-1)\Big|_{\mathbf{z}=\mathbf{z}_a}. \tag{III-4}$$

where $\nabla_{\mathbf{z}}I$ is the two-dimensional spatial gradient.

III.3 Convergence Analysis

Two of the important issues of an iterative algorithm such as the one presented in Section III.2 are to guarantee convergence and to determine the rate of convergence. Almost all previously presented pel-recursive techniques have used only one iteration per pel and assumed convergence over time [1-3,4,5,6,7]. The compression rate would be greater if the displacement-estimate algorithm in (III-4) converged to $\mathbf{d}$ at each pel. In other words, the information rate would be less if the algorithm obtained the correct displacement at every moving pel, not just converged over time within the moving area.

A requirement for the displacement estimation algorithm to converge at every pel is that the surface between $|DFD(\mathbf{z}_a,\hat{\mathbf{d}}^0)|^2$ and $|DFD(\mathbf{z}_a,\mathbf{d})|^2$ must be a concave surface, a minimum point of which is $|DFD(\mathbf{z}_a,\mathbf{d})|^2$. $\hat{\mathbf{d}}^0$ is the initial estimate of the displacement and $\mathbf{d}$ is the actual displacement.

Figure III.1 is a one-dimensional example of how $|DFD|$ might vary with $\hat{d}^i$, where $\hat{d}^i$ is an estimate of the true one-dimensional displacement, d. x_c is the spatial position at which the $|DFD|$ is minimized, $(x-d)$. If x_b or x_d is the initial spatial location estimate, $(x-\hat{d}^0)$, then the displacement estimation algorithm should converge to d. However, if $(x-\hat{d}^0)$ is at x_a or x_e, problems occur. With an initial spatial location estimate of x_a, the displacement estimate algorithm would probably converge to the location of the local minimum, x_f. With x_e as the initial spatial location estimate, no correction would occur

since the spatial gradient at x_e ($\nabla_x I|_{x_e}$) is zero.

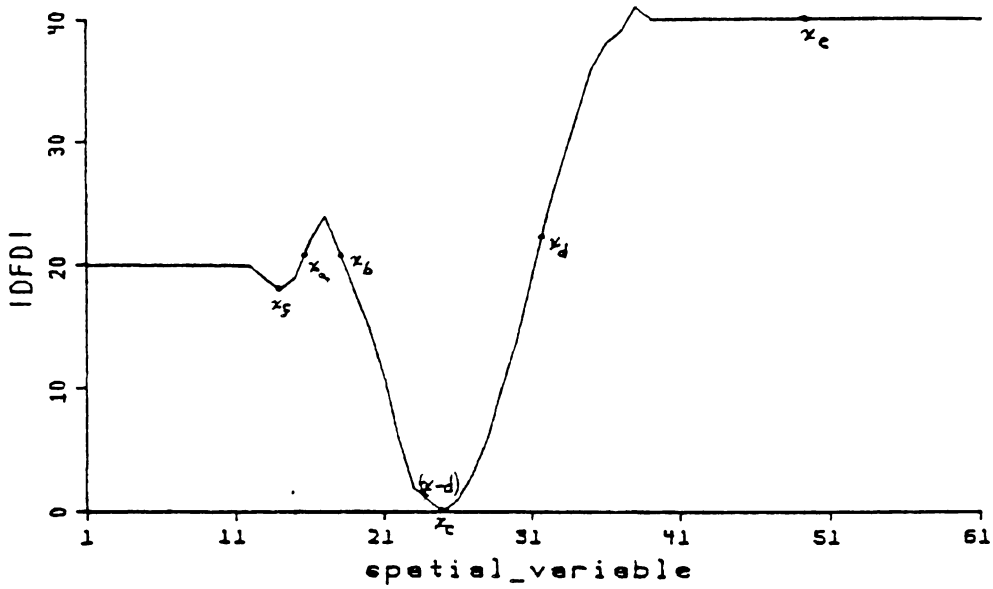

Figure III.1: A Sample Plot of $|DFD|$

One problem with an algorithm which converges to the correct displacement at every pel is a large computational expense. Horn and Schunck [8,9] and Nagel [10] have developed algorithms which do converge to the correct displacement for every pel in the image. Their algorithms require a system of coupled equations to be iterated over the image half as many times as the widest dimension of any moving object. The large number of iterations is required to insure that the displacement estimates from the perimeter of the moving object are propagated into the interior of the moving object and that the displacement estimates vary smoothly over the interior of the moving object. The coupled equations are due to the fact that $|DFD(\mathbf{z}_a, \hat{\mathbf{d}}^i)|^2$ is potentially minimized locally

for a multitude of $\hat{\mathbf{d}}^i$ values, only one of which is the "correct" one. Minimizing $|DFD(\mathbf{z}_a,\hat{\mathbf{d}}^i)|^2$ using only local information does not necessarily produce a unique (or correct) solution. In fact when using only local information, a unique displacement vector is obtained only when $|DFD(\mathbf{z}_a,\hat{\mathbf{d}}^i)|^2$ has a locally unique minimum. For $|DFD(\mathbf{z}_a,\hat{\mathbf{d}}^i)|^2$ to have a locally unique minimum, $\mathbf{z}_a$ must be the intersection of two edges of different orientation, e.g., a corner. It is only by considering a set of edges with different orientations that a unique solution can be found. However, if these edges have almost the same orientation, the problem will be ill-conditioned, i.e., noise will affect the solution greatly.

One way to get around the problem of $|DFD(\mathbf{z}_a,\hat{\mathbf{d}}^i)|^2$ being minimized with many values of $\hat{\mathbf{d}}^i$ is to use an independent second constraint on the displacement vector. For example, the second constraint used by Horn and Schunck [8,9] and Nagel [10] was that the displacement vectors had to vary smoothly.

In its initial passes over the image, Nagel's algorithm [10], which was an enhancement of Horn and Schunck's [8,9], ultimately had the same problem as an algorithm based only on minimizing $|DFD(\mathbf{z}_a,\hat{\mathbf{d}}^i)|^2$. Namely a linear set of solutions is obtained in regions having only one moving edge orientation. Nagel attempts to solve this problem by using an operator [11] which in effect finds the average gradient within an area. This is similar to the technique proposed by Huang [12] and does not necessarily lead to a well-conditioned problem. In fact, Netravali and Robbins [3] have shown that using a

least-squares approach to determine the gradient does not decrease the information rate appreciably when compared with single-point evaluation. Since image compression, not motion estimation, is the ultimate goal in this research, it should be reiterated that the techniques of Horn and Schunck [8,9] and Nagel [10,13] have only been applied to the motion-estimation/image-segmentation problem and have not been extended to the information compression problem.

There is a way to get around the problem of needing a second constraint: let the displacement vector converge in the mean over time within the moving area. This is the approach that will be used in this research. Although this is not without its drawbacks, iterating over the whole image 10-32 times appears to be too computationally expensive and using a least squares approach to gradient estimation has not been shown to be worth the additional computation involved.

Two proofs of convergence will be presented. The first is solely to determine the convergence requirements. The second is required to develop a framework from which the displacement estimate variance can be determined.

III.3.1 First Proof of Convergence of the Displacement Vector Estimate

In this section, it will be proved that the displacement estimation algorithm defined by equation (III-4) converges under certain conditions to the true displacement as a moving object is scanned. The proof procedure is similar to one used in [1]. The assumptions are that the motion is purely translational and that the uncovered background is neglected. Start by substituting equation (III-2) into (III-4) to obtain

$$\hat{\mathbf{d}}^{i+1} = \hat{\mathbf{d}}^{i} - \epsilon\{I(\mathbf{z}_a,t) - I(\mathbf{z}_a - \hat{\mathbf{d}}^{i},t-1)\}\nabla_{\mathbf{z}} I(\mathbf{z} - \hat{\mathbf{d}}^{i},t-1)\Big|_{\mathbf{z}=\mathbf{z}_a}. \qquad \text{(III-5)}$$

Substituting from (1) for $I(\mathbf{z}_a,t)$,

$$\hat{\mathbf{d}}^{i+1} = \hat{\mathbf{d}}^{i} - \epsilon\{I(\mathbf{z}_a - \mathbf{d},t-1) - I(\mathbf{z}_a - \hat{\mathbf{d}}^{i},t-1)\}\nabla_{\mathbf{z}} I(\mathbf{z} - \hat{\mathbf{d}}^{i},t-1)\Big|_{\mathbf{z}=\mathbf{z}_a}. \qquad \text{(III-6)}$$

The term in braces can be expanded using a Taylor series expansion, i.e.,

$$\begin{aligned} I(\mathbf{z}_a - \mathbf{d},t-1) - I(\mathbf{z}_a - \hat{\mathbf{d}}^{i},t-1) = {} & (\hat{\mathbf{d}}^{i} - \mathbf{d})^T \nabla_{\mathbf{z}} I(\mathbf{z} - \hat{\mathbf{d}}^{i},t-1)\Big|_{\mathbf{z}=\mathbf{z}_a} \\ & + \tfrac{1}{2}(\hat{\mathbf{d}}^{i} - \mathbf{d})^T \nabla_{\mathbf{z}}^2 I(\mathbf{z} - \hat{\mathbf{d}}^{i},t-1)(\hat{\mathbf{d}}^{i} - \mathbf{d})\Big|_{\mathbf{z}=\mathbf{z}_a} \qquad \text{(III-7)} \\ & + O(\hat{\mathbf{d}}^{i} - \mathbf{d})^3, \end{aligned}$$

where $\nabla_{\mathbf{z}}^2 I()$ is the 2x2 matrix of second partial derivatives of I() and $O(\hat{\mathbf{d}}^{i} - \mathbf{d})^3$ represents the higher order terms in $(\hat{\mathbf{d}}^{i} - \mathbf{d})$.

The $O(\hat{\mathbf{d}}^{i} - \mathbf{d})^3$ terms cannot be expressed in matrix notation; an open form must be used. Let

$$(\hat{\mathbf{d}}^{i} - \mathbf{d}) = \begin{bmatrix} \Delta d_x^i \\ \Delta d_y^i \end{bmatrix} \qquad \text{and} \qquad \nabla_{\mathbf{z}} I(i)\Big|_{\mathbf{z}=\mathbf{z}_a} = \begin{bmatrix} \dfrac{\partial I(i)}{\partial x} \\ \dfrac{\partial I(i)}{\partial y} \end{bmatrix}, \qquad \text{(III-8)}$$

where $(i) = (\mathbf{z}_a - \hat{\mathbf{d}}^{i},t-1)$. The Taylor series expansion of the difference in the two intensity values can now be rewritten in open form as

$$I(\mathbf{z}_a - \mathbf{d},t-1) - I(\mathbf{z}_a - \hat{\mathbf{d}}^{i},t-1) = \sum_{j=1}^{\infty} \frac{1}{j!} \left\{ \Delta d_x^i \left[\frac{\partial I(i)}{\partial x} \right] + \Delta d_y^i \left[\frac{\partial I(i)}{\partial y} \right] \right\}^j, \qquad \text{(III-9)}$$

where Δd_x^i is the error in the x displacement estimate at the ith iteration and Δd_y^i is the error in the y displacement estimate at the ith iteration.

The number of terms in the Taylor series that are required to obtain a good estimate of the difference in the two intensity values is dependent on two factors: 1) the error in the displacement estimation, and 2) the magnitude of the higher order derivatives of the intensity function. There is no reason to assume that a sufficient estimate is always obtained by retaining only the first term of the Taylor series expansion [14].

Substituting (III-9) into (III-6),

$$\hat{\mathbf{d}}^{i+1} = \hat{\mathbf{d}}^{i} - \epsilon \sum_{j=1}^{\infty} \frac{1}{j!} \left\{ \Delta d_x^i \left[\frac{\partial I(i)}{\partial x} \right] + \Delta d_y^i \left[\frac{\partial I(i)}{\partial y} \right] \right\}^j \nabla_{\mathbf{z}} I(\mathbf{z} - \hat{\mathbf{d}}^{i}, t-1) \Big|_{\mathbf{z}=\mathbf{z}_a}, \quad \text{(III-10)}$$

where mixed notation is used for compactness. After some intermediate algebra and regrouping of factors,

$$\hat{\mathbf{d}}^{i+1} = \hat{\mathbf{d}}^{i} - \epsilon \{ \nabla_{\mathbf{z}} I(i) \mathbf{f}(i) \} (\hat{\mathbf{d}}^{i} - \mathbf{d}), \Big|_{\mathbf{z}=\mathbf{z}_a} \quad \text{(III-11)}$$

where $\mathbf{f}(i)$ is a 1x2 matrix defined as

$$\mathbf{f}(i) = \sum_{j=1}^{\infty} \frac{1}{j!} \left\{ \Delta d_x^i \left[\frac{\partial I(i)}{\partial x} \right] + \Delta d_y^i \left[\frac{\partial I(i)}{\partial y} \right] \right\}^{j-1} \left\{ \frac{\partial I(i)}{\partial x} \quad \frac{\partial I(i)}{\partial y} \right\} \quad \text{(III-12)}$$

The first two terms of $\mathbf{f}(i)$ are indicated in Figure III.2.

Subtracting $\mathbf{d}$ from both sides of (III-11),

$$(\hat{\mathbf{d}}^{i+1} - \mathbf{d}) = [\mathbf{J} - \epsilon \{ \nabla_{\mathbf{z}} I(i) \mathbf{f}(i) \}] (\hat{\mathbf{d}}^{i} - \mathbf{d}), \Big|_{\mathbf{z}=\mathbf{z}_a} \quad \text{(III-13)}$$

where $\mathbf{J}$ is the appropriate size identity matrix.

Equation (III-13) is of the form $\mathbf{e}_{i+1} = \mathbf{A}_i\mathbf{e}_i$ where $\mathbf{A}$ is a 2x2 matrix and $\mathbf{e}$ is the 2x1 error vector which is to be reduced as $i \to \infty$. This can be rewritten as $\mathbf{e}_k = (\prod_{i=0}^{k} \mathbf{A}_i)\mathbf{e}_0$. If it can be shown that the $||\mathbf{e}_k||$ tends to zero as $k \to \infty$, convergence will be proved. Assume $\mathbf{B} = \prod_{i=0}^{k} \mathbf{A}_i$ as $k \to \infty$. $\mathbf{Be}_0 = 0$ if $\mathbf{e}_0 \in N(\mathbf{B})$, the null space of $\mathbf{B}$. Alternatively it can be shown that $\mathbf{Be}_0 = 0$ if the spectral radius of $\mathbf{B}$ is less than one. Unfortunately, neither of these is necessarily the case.

Closer inspection of the matrix $\mathbf{A}$ reveals that it can be rewritten as $[\mathbf{J} - \epsilon\mathbf{C}]$, where $\mathbf{C}$ is rank deficient, independent of the number of terms retained in the Taylor series expansion. Thus the spectral radius of $\mathbf{A}$ is greater than or equal to one depending on ϵ. If the spectral radius of $\mathbf{A}$ is one, further analysis must be performed to determine if the algorithm converges.

Investigation of stochastic gradient algorithms reveals that convergence properties are frequently determined by analyzing the behavior of the ensemble average [15-18]. Using this approach and assuming the two factors on the right-hand side are uncorrelated, take the expected value of both sides and apply Schwartz inequality,

$$||E\{\hat{\mathbf{d}}^{i+1}\} - \mathbf{d}|| \leq ||\mathbf{J} - \epsilon E\{\nabla_{\mathbf{x}} I(i)\mathbf{f}(i)\}|| \cdot ||E\{\hat{\mathbf{d}}^{i}\} - \mathbf{d}||, \tag{III-14}$$

where E{ } denotes expected value. This can be rewritten as

$$||E\{\hat{\mathbf{d}}^{i+1}\} - \mathbf{d}|| \leq |1 - \epsilon\lambda_{\max}| \cdot ||E\{\hat{\mathbf{d}}^{i}\} - \mathbf{d}||, \tag{III-15}$$

where $\lambda_{\max}$ is the maximum eigenvalue of the positive semidefinite symmetric matrix

$$j=1: \quad \mathbf{f}_1(i) = \left\{ \frac{\partial I(i)}{\partial x} \quad \frac{\partial I(i)}{\partial y} \right\}$$

$$= \left\{ \frac{\partial I}{\partial x} \quad \frac{\partial I}{\partial y} \right\} \Bigg|_{(\mathbf{z}_a - \hat{\mathbf{d}}^i, t-1)}$$

$$= \nabla_{\mathbf{z}} I^T(\mathbf{z} - \hat{\mathbf{d}}^i, t-1) \Big|_{\mathbf{z}=\mathbf{z}_a}$$

$$j=2: \quad \mathbf{f}_2(i) = \tfrac{1}{2} \left\{ \Delta d_x^i \left[\frac{\partial I(i)}{\partial x} \right] + \Delta d_y^i \left[\frac{\partial I(i)}{\partial y} \right] \right\} \left\{ \frac{\partial I(i)}{\partial x} \quad \frac{\partial I(i)}{\partial y} \right\}$$

$$= \tfrac{1}{2} \left\{ \Delta d_x^i \left[\frac{\partial^2 I}{\partial x^2} \right] + \Delta d_y^i \left[\frac{\partial^2 I}{\partial x \partial y} \right] \quad \Delta d_x^i \left[\frac{\partial^2 I}{\partial y \partial x} \right] + \Delta d_y^i \left[\frac{\partial^2 I}{\partial y^2} \right] \right\} \Bigg|_{(\mathbf{z}_a - \hat{\mathbf{d}}^i, t-1)}$$

$$= \tfrac{1}{2} (\hat{\mathbf{d}}^i - \mathbf{d})^T \nabla_{\mathbf{z}}^2 I(\mathbf{z} - \hat{\mathbf{d}}^i, t-1) \Big|_{\mathbf{z}=\mathbf{z}_a}.$$

Figure III.2: Expansion of the First Two Terms of $\mathbf{f}(i)$

$E\{\nabla_{\mathbf{z}} I(i) \mathbf{f}(i)\}$.

For convergence of the algorithm, $|1 - \epsilon \lambda_{\max}|$ must be less than unity or

$$\frac{2}{\lambda_{max}} > \epsilon > 0. \tag{III-16}$$

Since there are only two eigenvalues and the sum of the eigenvalues of any matrix equals the trace, λ_{max} is upper bounded by the trace and lower bounded by one half of the trace. Therefore the following condition is sufficient for the iterative algorithm to converge:

$$\frac{4}{tr\mathbf{e}\{\nabla_z I(i)\mathbf{f}(i)\}} > \epsilon > 0, \tag{III-17}$$

where *tr* indicates the trace of the 2x2 matrix.

If $\mathbf{f}_1(i) \approx \mathbf{f}(i)$ (see Figure III.2), then equation (III-17) can be rewritten as

$$\frac{4M}{\sum_{i=1}^{M} [\{\nabla_x I(i)\}^2 + \{\nabla_y I(i)\}^2]} > \epsilon > 0, \tag{III-18}$$

where M iterations are performed within the moving area. The number of iterations per pel is not required to be constant. $\nabla_x I(i)$ and $\nabla_y I(i)$ are the orthogonal components of $\nabla_z I(i)$, i.e., the gradient vector elements. The $\{\nabla I(i)\}^2$ notation indicates the square of the real-valued gradient component. Note the difference between $\{\nabla I(i)\}^2$ in equation (III-18) and $\nabla^2 I(i)$ in equation (III-7) and in Figure III.2. In Section III.5, it will be shown that for a set of typical sequences, ϵ must be less than 0.0040 for low interframe motion and less than 0.0200 for high interframe motion.

III.3.2 Second Proof of Convergence of the Displacement Vector Estimate

There is another way to approach the proof of convergence, which yields a procedure to determine the displacement variance at convergence. Equation (III-11) can be rewritten as

$$\hat{\mathbf{d}}^{i+1} = [\mathbf{J} - \epsilon\{\nabla_{\mathbf{z}} I(i)\mathbf{f}(i)\}]\hat{\mathbf{d}}^{i}\Big|_{\mathbf{z}=\mathbf{z}_a} + \epsilon\{\nabla_{\mathbf{z}} I(i)\mathbf{f}(i)\}\mathbf{d}\Big|_{\mathbf{z}=\mathbf{z}_a}. \tag{III-19}$$

Taking the expected values of both sides and assuming the gradient and the displacement are uncorrelated,

$$E\{\hat{\mathbf{d}}^{i+1}\} = [\mathbf{J} - \epsilon E\{\nabla_{\mathbf{z}} I(i)\mathbf{f}(i)\}]E\{\hat{\mathbf{d}}^{i}\} + \epsilon E\{\nabla_{\mathbf{z}} I(i)\mathbf{f}(i)\}\mathbf{d}. \tag{III-20}$$

Although $E\{\nabla_{\mathbf{z}} I(i)\mathbf{f}(i)\}$ may be spatially varying, it is real and symmetric. Therefore, it may be decomposed by a similarity transform into the form

$$E\{\nabla_{\mathbf{z}} I(i)\mathbf{f}(i)\} = \mathbf{U}(i)\mathbf{S}(i)\mathbf{U}^{T}(i), \tag{III-21}$$

where $\mathbf{U}(i)$ is the orthonormal matrix whose columns are the normalized eigenvectors of $E\{\nabla_{\mathbf{z}} I(i)\mathbf{f}(i)\}$, and $\mathbf{S}(i)$ is the diagonal matrix of eigenvalues of $E\{\nabla_{\mathbf{z}} I(i)\mathbf{f}(i)\}$. Further

$$\mathbf{U}^{T}(i)\mathbf{U}(i) = \mathbf{J}. \tag{III-22}$$

Note that $E\{\nabla_{\mathbf{z}} I(i)\mathbf{f}(i)\}$ is full rank only if there are multiple edge orientations within the random field over which the expected value is taken. The purpose of this transform is to "rotate" (III-19) for the expected displacement vector into a coordinate system in which the expected displacement components are uncoupled. If one were to expand (III-19), it would be seen that the behavior of $E\{d_x^{i+1}\}$ (and $E\{d_y^{i+1}\}$) is dependent upon both $E\{d_x^{i}\}$ and $E\{d_y^{i}\}$. The transformation in (III-21) provides a set of uncou-

pled displacements, $\mathbf{v}(i)$, for which the convergence properties of the gradient algorithm are more easily displayed.

Therefore, let $\mathbf{v}(i)$ be the set of uncoupled displacements and be defined by the pair of transformations

$$E\{\hat{\mathbf{d}}^i\} = \mathbf{U}(i)\mathbf{v}(i) \tag{III-23}$$

and

$$\mathbf{v}(i) = \mathbf{U}^T(i)E\{\hat{\mathbf{d}}^i\}. \tag{III-24}$$

Substituting (III-21) and (III-23) into (III-20),

$$\mathbf{v}(i+1)=\mathbf{G}(i+1)[\mathbf{J}-\epsilon\mathbf{S}(i)]\mathbf{v}(i) + \epsilon\mathbf{U}^T(i+1)\mathbf{S}(i)\mathbf{d}, \tag{III-25}$$

where

$$\mathbf{G}(i+1) = \mathbf{U}^T(i+1)\mathbf{U}(i). \tag{III-26}$$

Analytically, the product $\mathbf{U}^T(i+1)\mathbf{U}(i)$ is not easily defined. Since $\mathbf{U}^T(i)\mathbf{U}(i)$ is the identity matrix and $\hat{\mathbf{d}}^{i+1}\approx\hat{\mathbf{d}}^i$ within a moving area, as a first approximation assume $\mathbf{G}(i+1)$ is approximately equal to $\mathbf{J}$, the identity matrix.

Assume the $\mathbf{U}(i)$ and $\mathbf{S}(i)$ matrices are constant. (III-25) becomes

$$\mathbf{v}(i+1) = [\mathbf{J}-\epsilon\mathbf{S}]\mathbf{v}(i) + \epsilon\mathbf{U}^T\mathbf{S}\mathbf{d}, \tag{III-27}$$

where the i-dependence has been dropped for $\mathbf{U}$ and $\mathbf{S}$. Note that one way to make $\mathbf{S}(i)$ constant is to make ϵ equal to the product of a constant and the inverse of $\mathbf{S}(i)$. To obtain the closed-form solution for the expected displacement vector $E\{\hat{\mathbf{d}}^{i+1}\}$, it is neces-

sary to solve (III-25) for uncoupled displacement vector, $\mathbf{v}(i+1)$, and utilize (III-23) for the transformation back to the $E\{\hat{\mathbf{d}}^{i+1}\}$.

To solve (III-27), it is instructive to expand a few terms in the $\mathbf{v}(i)$ sequence:

$$\begin{aligned}
i=0:\quad \mathbf{v}(1) &= [\mathbf{J}-\epsilon\mathbf{S}]\mathbf{v}(0) + \epsilon\mathbf{U}^T\mathbf{S}\mathbf{d} \\
i=1:\quad \mathbf{v}(2) &= [\mathbf{J}-\epsilon\mathbf{S}]\mathbf{v}(1) + \epsilon\mathbf{U}^T\mathbf{S}\mathbf{d} \\
&= [\mathbf{J}-\epsilon\mathbf{S}]\{[\mathbf{J}-\epsilon\mathbf{S}]\mathbf{v}(0) + \epsilon\mathbf{U}^T\mathbf{S}\mathbf{d}\} + \epsilon\mathbf{U}^T\mathbf{S}\mathbf{d} \\
&= [\mathbf{J}-\epsilon\mathbf{S}]^2\mathbf{v}(0) + [\mathbf{J}-\epsilon\mathbf{S}]\epsilon\mathbf{U}^T\mathbf{S}\mathbf{d} + \epsilon\mathbf{U}^T\mathbf{S}\mathbf{d} \\
i=2:\quad \mathbf{v}(3) &= [\mathbf{J}-\epsilon\mathbf{S}]\mathbf{v}(2) + \epsilon\mathbf{U}^T\mathbf{S}\mathbf{d} \\
&= [\mathbf{J}-\epsilon\mathbf{S}]\{[\mathbf{J}-\epsilon\mathbf{S}]^2\mathbf{v}(0) + [\mathbf{J}-\epsilon\mathbf{S}]\epsilon\mathbf{U}^T S\mathbf{d} + \epsilon\mathbf{U}^T\mathbf{S}\mathbf{d}\} + \epsilon\mathbf{U}^T\mathbf{S}\mathbf{d}.
\end{aligned} \tag{III-28}$$

Following this procedure, the expression for $\mathbf{v}(i)$ becomes

$$\mathbf{v}(i) = [\mathbf{J}-\epsilon\mathbf{S}]^i\mathbf{v}(0) + \sum_{n=0}^{i-1}\epsilon[\mathbf{J}-\epsilon\mathbf{S}]^n\mathbf{U}^T\mathbf{S}\mathbf{d}, \qquad i>0. \tag{III-29}$$

Since $[\mathbf{J}-\epsilon\mathbf{S}]$ is a diagonal matrix, the $v_j(i)$ in (III-29) are uncoupled. Thus, (III-29) written in terms of a single $v_j(i)$ becomes

$$v_j(i) = (1-\epsilon\lambda_j)^i v_j(0) + \epsilon c_j \sum_{n=0}^{i-1}(1-\epsilon\lambda_j)^n \tag{III-30}$$

where

$$c_j = \lambda_j d_j \mathbf{u}_j \qquad j=1,2 \tag{III-31}$$

and $\mathbf{u}_j$ is the jth eigenvector of the $\mathbf{U}$ matrix, λ_j is the jth eigenvalue of $E\{\nabla_{\mathbf{z}} I(i)\mathbf{f}(i)\}$, and d_j is the jth component of the true displacement vector.

Performing the summation in (III-30) produces

$$v_j(i) = (1-\epsilon\lambda_j)^i v_j(0) + \epsilon c_j\left[\frac{1-(1-\epsilon\lambda_j)^i}{1-(1-\epsilon\lambda_j)}\right]. \tag{III-32}$$

Simplifying and using the substitution

$$\gamma_j = 1 - \epsilon\lambda_j, \tag{III-33}$$

(III-32) becomes

$$v_j(i) = \gamma_j^i v_j(0) + \frac{c_j}{\lambda_j}(1-\gamma_j^i), \qquad i \geq 0. \tag{III-34}$$

Several facts are noteworthy:

(a) Note $\frac{c_j}{\lambda_j} = d_j \mathbf{u}_j$.

(b) For convergent $v_j(i)$ solutions, each of the quantities $|\gamma_j|$ must be less than unity. This leads to the same constraints on ϵ as in (III-16) through (III-18).

(c) There are two modes of convergence.

Each uncoupled displacement, $v_j(i)$ convergences in the mean toward its final value, $\frac{c_j}{\lambda_j}$, at a rate controlled by the product $\epsilon\lambda_j$. One may define a convergence constant, τ_j, for each of the two modes as the number of iterations needed for $v_j(i)$ in (III-27) to be within e^{-1} of its final value, $v_j(\infty)$. This produces the following equation:

$$\begin{aligned} v_j(\tau_j) &= v_j(\infty) + e^{-1}[v_j(0) - v_j(\infty)] \\ &= v_j(0) + (1 - e^{-1})[v_j(\infty) - v_j(0)] \end{aligned} \tag{III-35}$$

which, with the use of (III-34), may be solved for τ_j:

$$\begin{aligned} \gamma_j^{\tau_j} v_j(0) + \frac{c_j}{\lambda_j}(1-\gamma_j^{\tau_j}) &= v_j(0)+(1-e^{-1})[\frac{c_j}{\lambda_j} - v_j(0)] \\ &= e^{-1}v_j(0)+\frac{c_j}{\lambda_j}(1-e^{-1}). \end{aligned} \tag{III-36}$$

This implies that

$$\begin{aligned} \gamma_j^{\tau_j} &= e^{-1} \\ (1-\epsilon\lambda_j)^{\tau_j} &= e^{-1} \\ \tau_j ln(1-\epsilon\lambda_j) &= -1 \\ \tau_j &= \frac{-1}{ln(1-\epsilon\lambda_j)}. \end{aligned} \tag{III-37}$$

Since each of the λ_j may be quite different, the convergence constants τ_j may vary considerably, resulting in diverse convergence rates for each mode. The largest eigenvalue corresponds to the dominant mode which produces the shortest convergence time. See Example 1.

In the next section, an analytical measure of the variance of the displacement estimate will be derived. This measure will reveal that choosing the maximum allowable value for ϵ is usually not the best choice.

Consider an image edge which has

$$\mathbf{d} = \begin{bmatrix} 2 \\ 0 \end{bmatrix}$$

and

$$E\{\nabla_z I(i)\mathbf{f}(i)\} = \begin{bmatrix} 400 & 100 \\ 100 & 200 \end{bmatrix}.$$

The eigenvalues and eigenvectors of $E\{\nabla_z I(i)\mathbf{f}(i)\}$ are:

$$\begin{bmatrix} \lambda_1 & \lambda_2 \end{bmatrix} = \begin{bmatrix} 441.4214 & 158.5786 \end{bmatrix}$$

$$\mathbf{U} = \begin{bmatrix} -0.3827 & 0.9239 \\ 0.9239 & 0.3827 \end{bmatrix}$$

Using equation (III-17), ϵ is constrained to be less than 0.0067. Arbitrarily choose $\epsilon = 0.0010$. Then

$$\tau_1 = \frac{-1}{ln(1-.4414214)} = 1.7172$$

$$\tau_2 = \frac{-1}{ln(1-.1585786)} = 5.7916$$

This analysis indicates both uncoupled displacement estimates should be within e^{-1} of their final value within six iterations.

Example 1: Convergence Rate for Displacement Estimates

III.3.3 Variance of the Displacement Vector Estimate At Convergence

While it has been shown that the expected value of the displacement vector converges to the correct value, no consideration has been given to the variance of the displacement vector estimation about the correct value. From (III-16) it can be seen that if $\epsilon < 2/\lambda_{max}$, convergence of the displacement vector in the mean sense is assured. However, it will be shown in the following that the variance of the displacement vector about the correct value is proportional to ϵ, resulting in conflicting requirements: large ϵ for rapid convergence and small ϵ for small displacement estimate variance.

Assume the estimated gradient equals the true gradient plus noise. In this case, (III-3) can be rewritten as

$$\hat{\mathbf{d}}^{i+1} = \hat{\mathbf{d}}^{i} - \frac{\epsilon}{2}\{\nabla_{\mathbf{d}}[DFD(\mathbf{z}_a, \hat{\mathbf{d}}^{i})]^2 + \mathbf{N}(i)\}. \tag{III-38}$$

After working this change through the intermediate equations, (III-13) can be rewritten as

$$(\hat{\mathbf{d}}^{i+1} - \mathbf{d}) = [\mathbf{J} - \epsilon\{\nabla_{\mathbf{z}} I(i)\mathbf{f}(i)\}](\hat{\mathbf{d}}^{i} - \mathbf{d}) + \frac{\epsilon}{2}\{\mathbf{N}(i)\}. \tag{III-39}$$

The eigenvector matrix $\mathbf{U}$ is used to provide a transformation similar to equations (III-23) and (III-24), which decouple the displacement vector equations.

Thus letting

$$\boldsymbol{\delta}(i) = \mathbf{U}^T(i)(\hat{\mathbf{d}}^{i} - \mathbf{d}) \tag{III-40}$$

and

$$(\hat{\mathbf{d}}^i - \mathbf{d}) = \mathbf{U}(i)\boldsymbol{\delta}(i) \tag{III-41}$$

and substituting (III-41) into (III-39), the following uncoupled difference vector iterative equation is obtained:

$$\boldsymbol{\delta}(i+1) = [\mathbf{J} - \epsilon\mathbf{S}(i)]\boldsymbol{\delta}(i) - \frac{\epsilon}{2}\mathbf{U}^T(i)\mathbf{N}(i). \tag{III-42}$$

Equation (III-42) can now be analyzed to determine information about the magnitude of the displacement variance about the true displacement value, $\mathbf{d}$. It has already been shown that the $\hat{\mathbf{d}}^i$ estimate converges for large values of i to the true solution, $\mathbf{d}$. This result requires that the expected value of $\boldsymbol{\delta}(i)$ in (III-42) go to zero as i increases. Taking the expected value of both sides of (III-42) and enforcing this requirement yields

$$E\{\mathbf{N}(i)\}=0. \tag{III-43}$$

In other words, the gradient estimation noise is zero mean.

To find the variance of the displacement estimates for stationary statistics, take the outer product of equation (III-42):

$$\boldsymbol{\delta}(i)\boldsymbol{\delta}^T(i) = \{[\mathbf{J} - \epsilon\mathbf{S}]\boldsymbol{\delta}(i) - \frac{\epsilon}{2}\mathbf{U}^T\mathbf{N}(i)\}\{[\mathbf{J} - \epsilon\mathbf{S}]\boldsymbol{\delta}(i) - \frac{\epsilon}{2}\mathbf{U}^T\mathbf{N}(i)\}^T. \tag{III-44}$$

Performing the matrix multiplication and then simplifying produces:

$$\begin{aligned}\boldsymbol{\delta}(i)\boldsymbol{\delta}^T(i) = {} & [\mathbf{J} - \epsilon\mathbf{S}]\boldsymbol{\delta}(i)\boldsymbol{\delta}^T(i)[\mathbf{J} - \epsilon\mathbf{S}]^T \\ & + (\frac{\epsilon}{2})^2\mathbf{U}^T\mathbf{N}(i)\mathbf{N}^T(i)U \\ & - \frac{\epsilon}{2}\{[\mathbf{J} - \epsilon\mathbf{S}]\boldsymbol{\delta}(i)\mathbf{N}^T(i)\mathbf{U} + \mathbf{U}^T\mathbf{N}(i)\boldsymbol{\delta}(i)[\mathbf{J} - \epsilon\mathbf{S}]\}.\end{aligned} \tag{III-45}$$

Next take the expected value of both sides. Since $\delta(i)$ is only affected by gradient estimation noise from the previous iterations, the vectors $\delta(i)$ and $N(i)$ are uncorrelated and the expected value of the term in brackets in (III-45) is zero. Using this result, the expectation of both sides of (III-45) becomes

$$E\{\delta(i+1)\delta^T(i+1)\} = [J-\epsilon S]E\{\delta(i)\delta^T(i)\}[J-\epsilon S] + \frac{\epsilon^2}{4}U^T E\{N(i)N^T(i)\}U. \tag{III-46}$$

To derive useful results from (III-46) some assumptions must be made. Recall that $N(i)$ is the gradient estimation "noise" and is the difference between the true value of the gradient and the computed value. Denote $E\{N(i)N^T(i)\}$ as Σ_{gn}^2, where Σ_{gn}^2 is a 2x2 matrix. (See Appendix A for derivation of Σ_{gn}^2.)

Substituting Σ_{gn}^2 into (III-46) and recognizing that at convergence the expectation matrices are no longer time dependent, that is

$$E\{\delta(i+1)\delta^T(i+1)\} = E\{\delta(i)\delta^T(i)\} = E\{\delta\delta^T\}, \tag{III-47}$$

the following result is obtained:

$$E\{\delta\delta^T\} = [J-\epsilon S]E\{\delta\delta^T\}[J-\epsilon S] + \frac{\epsilon^2}{4}U^T\Sigma_{gn}^2 U. \tag{III-48}$$

Solving for $E\{\delta\delta^T\}$ produces

$$E\{\delta\delta^T\} = \frac{\epsilon^2}{4}[2\epsilon S-\epsilon^2 S^2]^{-1}U^T\Sigma_{gn}^2 U. \tag{III-49}$$

Since the matrix to be inverted is diagonal, its inverse may be found easily, producing:

$$E\{\delta\delta^T\} = \frac{\epsilon}{4}S^{-1}\Lambda U^T\Sigma_{gn}^2 U, \tag{III-50}$$

where the nonzero elements of the diagonal matrix Λ are given by

$$\Lambda_{jj} = (2 - \epsilon\lambda_j)^{-1} = (1+\gamma_j)^{-1} \tag{III-51}$$

and the λ_j are the non-zero elements of the diagonal $\mathbf{S}$ matrix.

Using (III-40) and the substitution $\Delta = E\{(\hat{\mathbf{d}}^i - \mathbf{d})(\hat{\mathbf{d}}^i - \mathbf{d})^T\}$, it can be derived that

$$\Delta = \frac{\epsilon}{4}\Sigma_{gn}^2 \mathbf{S}^{-1}\Lambda. \tag{III-52}$$

The diagonal elements of the matrix Δ are the required displacement variances at convergence:

$$Var[\hat{\mathbf{d}}_j^i - \mathbf{d}_j] = \Delta_{jj}. \tag{III-53}$$

See Example 2.

It can now be seen that the value of the convergence coefficient, ϵ, is constrained, that the constraints can be determined given certain a priori information, and that choosing the maximum allowable ϵ is not prudent.

Consider the edge in Example 1 again. Assume $E\{\nabla_z I(i)\mathbf{f}(i)\} \approx E\{\nabla_z I(i)\nabla_z^T I(i)\}$. To determine the variance of $\hat{\mathbf{d}}$ at convergence, the noise variance of the image, σ_n^2, must be known. Assume $\sigma_n^2 = 0.0001$. Therefore from Appendix A,

$$\Sigma_{gn}^2 = 4\begin{bmatrix} 2(0.0001)(400) + (0.0001)^2 & 0 \\ 0 & 2(0.0001)(200) + (0.0001)^2 \end{bmatrix}$$

$$\Sigma_{gn}^2 \approx \begin{bmatrix} 0.32 & 0 \\ 0 & 0.16 \end{bmatrix}$$

$$\mathbf{S}^{-1} = \begin{bmatrix} \frac{1}{\lambda_1} & 0 \\ 0 & \frac{1}{\lambda_2} \end{bmatrix} \qquad \text{(III-51)}$$

$$\Lambda = \begin{bmatrix} (2-\epsilon\lambda_1)^{-1} & 0 \\ 0 & (2-\epsilon\lambda_2)^{-1} \end{bmatrix}. \qquad \text{(III-52)}$$

From (III-52)

$$\Delta = \frac{\epsilon}{4}\Sigma_{gn}^2 \mathbf{S}^{-1}\Lambda$$

$$\Delta = \epsilon\begin{bmatrix} 0.08 & 0 \\ 0 & 0.04 \end{bmatrix}\begin{bmatrix} \frac{1}{\lambda_1} & 0 \\ 0 & \frac{1}{\lambda_2} \end{bmatrix}\begin{bmatrix} (2-\epsilon\lambda_1)^{-1} & 0 \\ 0 & (2-\epsilon\lambda_2)^{-1} \end{bmatrix}.$$

Letting $\epsilon = 0.0010$ as in Example 1,

$$\Delta = 0.001\begin{bmatrix} 0.08 & 0 \\ 0 & 0.04 \end{bmatrix}\begin{bmatrix} 0.0023 & 0 \\ 0 & 0.0063 \end{bmatrix}\begin{bmatrix} 0.642 & 0 \\ 0 & 0.543 \end{bmatrix}$$

$$= \begin{bmatrix} 1.16 & 0 \\ 0 & 1.37 \end{bmatrix} \times 10^{-7}.$$

Therefore the displacement variance at convergence is

$$Var[\hat{\mathbf{d}}_x^i - \mathbf{d}_x] = 1.16 \times 10^{-7}$$

$$Var[\hat{\mathbf{d}}_y^i - \mathbf{d}_y] = 1.37 \times 10^{-7}.$$

Example 2: Displacement Estimate Variance at Convergence

III.4 An Improved Motion Prediction Technique

There have been two predominant methods of displacement estimation: spatial and temporal. Most researchers have used a spatially adjacent displacement vector as an initial estimate [1-7]. Other researchers, mostly from Bell Northern Research [4,19], proposed predicting the displacement along the temporal axis. A third approach is proposed in this dissertation: project the motion estimation forward along the motion trajectory (PAMT). This would have four advantages and require a minimal increase in computation and memory over the temporal projection procedure. The problems with the present schemes will be discussed first.

III.4.1 Existing Motion Prediction Techniques

By always using a spatially adjacent displacement vector as an initial estimate for the displacement vector under consideration, an implicit assumption is being made that the displacement vectors always have a high spatial correlation. This is not what the original image model implies. The original model assumed that an object is moving over a fixed stationary background. Although the displacement vectors are highly correlated within the moving object and in the stationary background, the displacement vectors are highly uncorrelated at the edges of that moving object. It has been shown that the edges of an object may be what contribute most to the estimation of the displacement vectors [20]. In Section III.3 it was indicated that the number of terms in the Taylor series expansion that are required

to obtain a good estimate of the DFD is dependent on the error in the displacement estimation. It is questionable whether a spatially adjacent displacement vector is a sufficiently accurate initial estimate to assure convergence of the displacement estimation equation. Consider a one-dimensional example.

In Figure III.3 an edge has moved three units to the left between frames t and $t-1$. Scanning from left to right in frame t, a nonzero DFD is first encountered at point x_a (assume $\hat{\mathbf{d}}^0=0$). This incorrect displacement estimate of 0 will not be corrected since $\nabla_x I|_{t-1}=0$. No matter how many times the algorithm iterates at x_a, the correct displacement value cannot be found for $I(x_a,t)$. It is not until point x_b is reached that the motion estimation can be corrected (i.e., when $\nabla_x I|_{t-1}$ becomes nonzero). If the correct $\mathbf{d}$ has not been determined by the time $x_{\mathbf{d}}$ is reached in the scan line, $\hat{\mathbf{d}}$ cannot be corrected further until the spatial gradient in frame $t-1$ becomes nonzero again, which may not occur until much later in the scan line.

Dubois et al [4,19] suggested using the temporally adjacent displacement vector as an initial estimate. By projecting the displacement vector estimates over time rather than space, the displacement estimates at the edges can exhibit a sharp discontinuity and this discontinuity can be sharpened over time. However, this approach does not fully solve the problem. It assumes that the location of the moving objects remain the same frame-to-frame.

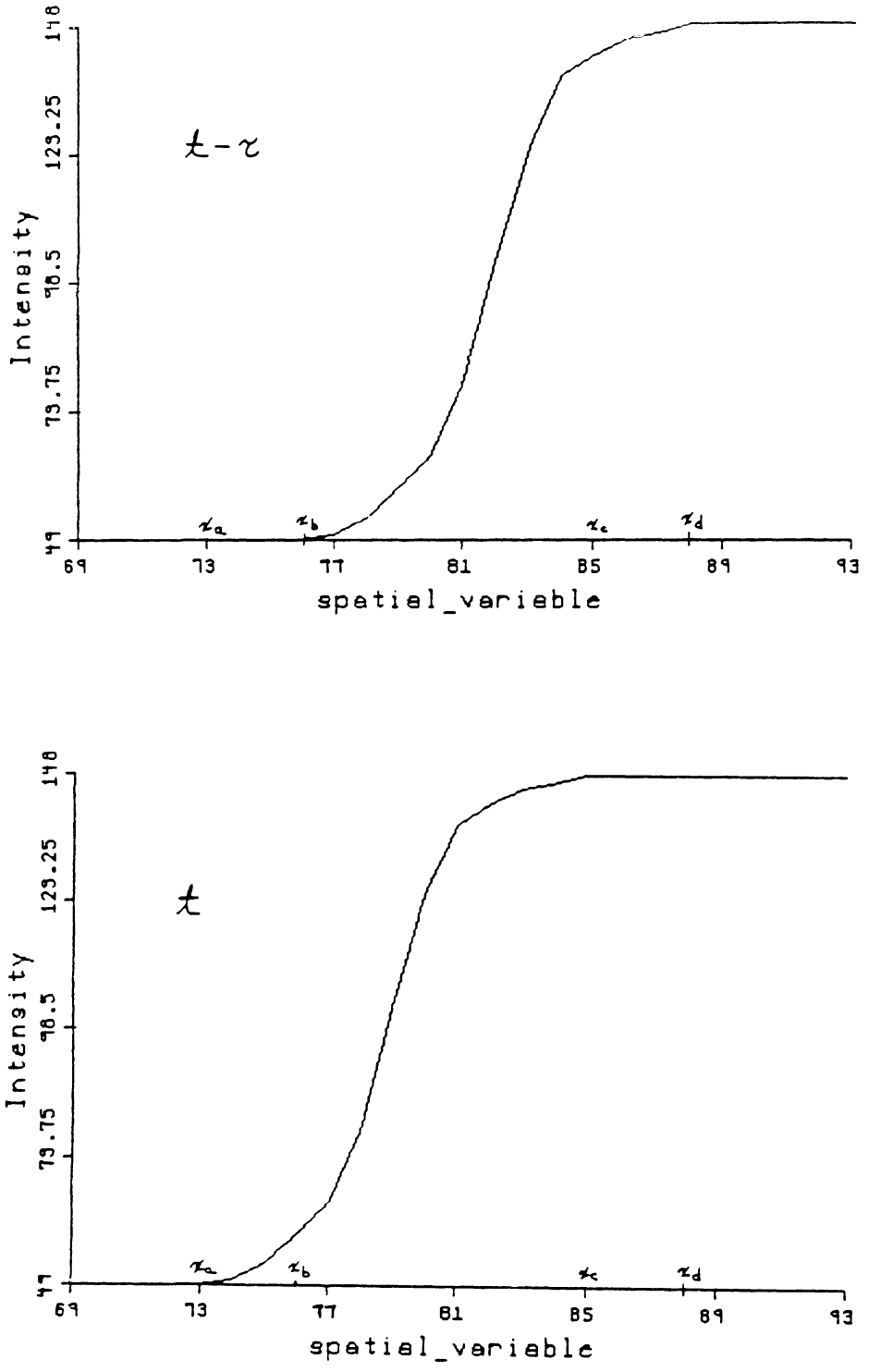

Figure III.3: A Moving Edge in One Dimension

As one example of the problem with temporal prediction, look again at Figure III.3. The same problem exists here as with spatially adjacent estimation: $\hat{\mathbf{d}}^0 = 0$ at x_a. There is no way to converge to $\mathbf{d}$ at x_a. The improvement of temporal prediction occurs at x_b where $\hat{\mathbf{d}}^0$ is not necessarily zero. The cost of obtaining this improvement is an extra frame buffer to store the $\hat{\mathbf{d}}^0$ from frame to frame.

As a second example, consider the object moving to the left in the plane of view with a constant translational velocity in Figure III.4. If the displacement vectors are projected forward parallel to the temporal axis, then there will be errors associated with both the leading and the trailing edge. The intensities along the leading edge (area L in Figure III.4) will not be predicted correctly since in the previous frame (at time $t-1$), nothing was moving in those pel locations into which the leading edge has now moved. The trailing edge (area T in Figure III.4), on the other hand, has left some pel locations between time $t-1$ and t. The intensities at these pel locations at time t constitute newly uncovered background. The algorithm will try to predict the intensities for these pels from displaced intensities in the previous frame. The accuracy of this prediction will depend on the correlation between the intensity values in the displaced region in the previous frame (frame $t-1$) and the intensity values in the newly uncovered background region in the present frame (frame t).

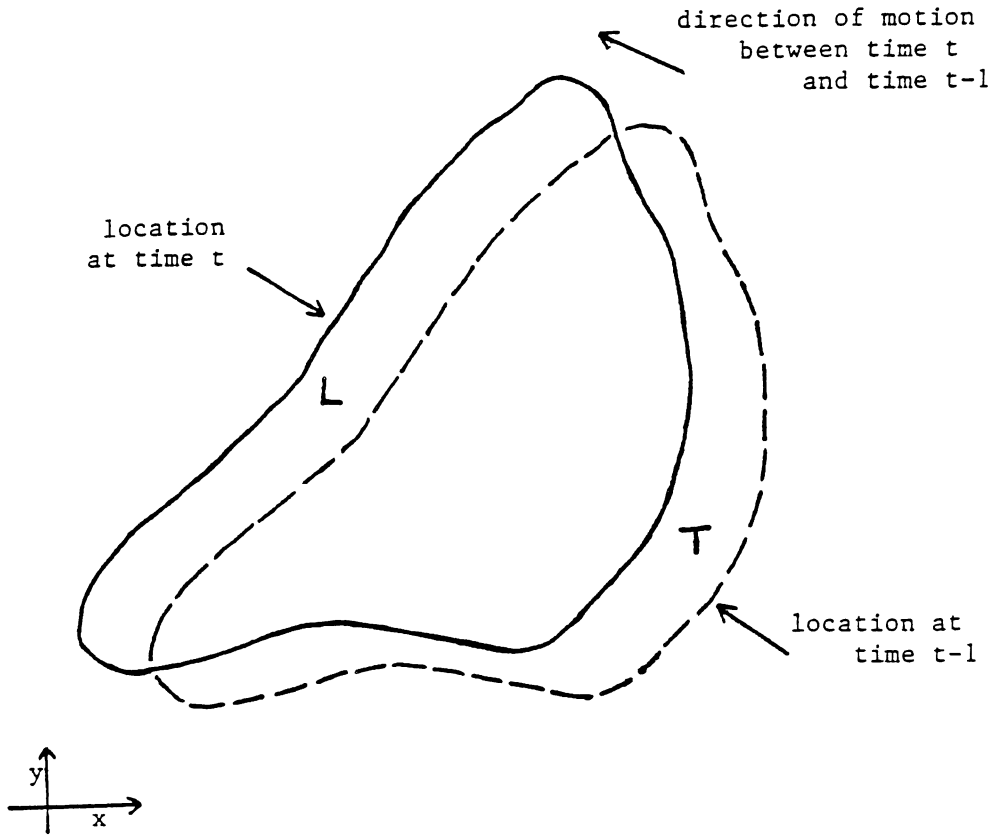

Figure III.4: A Moving Object, Top View

A better prediction scheme would be to assume the motion, not the object location, remained the same. Instead of projecting the motion estimations forward parallel to the temporal axis, project them forward along the motion trajectory. This is the improved motion prediction technique.

III.4.2 The Improved Motion Prediction Technique

If the object has a constant velocity frame-to-frame, projecting the displacement vectors forward in the direction of motion will correctly predict the leading edge values. Also, those areas of the image which contain newly uncovered background can be correctly detected.

By projecting the motion vectors forward in the direction of motion, a problem that has existed in the implementation of the algorithm is solved. In proving convergence the uncovered background was neglected [1]. Yet most algorithms [1-6,19] attempt to determine the intensity values for the newly uncovered background at time t using intensities in the frame at time $t-1$. The structure of the algorithm is at fault. By obtaining the initial estimates for the displacement vector from spatially or temporally adjacent pels there is no way to detect what regions are newly uncovered background. By predicting the motion vectors forward in the direction of motion, the uncovered background will have no displacement values predicted for it. The uncovered background is then easily detected, allowing a better predictor to be used for it and allowing the implementation to be a true implementation of the algorithm which was proved to converge.

To reiterate and summarize, by projecting the displacement estimates forward along the motion trajectory four improvements are obtained:

1) With respect to spatial prediction, sharp discontinuities can exist at the boundaries between moving objects and the background.

2) With respect to temporal prediction, the actual displacement of the object point can be found more often since the motion, not the location, of the moving area is assumed constant.

3) The number of iterations required for convergence will be decreased due to better initial estimates. Also a smaller displacement prediction error allows a larger ϵ which increases the convergence rate.

4) A substantial portion of the uncovered background is detectable and can be segmented out.

The computation requirements for PAMT prediction are only slightly greater than those for temporal prediction. The addressing for the framestore into which the motion prediction field is loaded is random; in temporal prediction it is sequential. When round(d_x) or round(d_y) changes, a gap is left in the predicted motion vector field for the next frame when using the PAMT prediction scheme. However, this problem can be at least partially resolved by using a gap-bridging filter [1,19]. As a side note, either constant interframe velocity or constant interframe acceleration can be assumed when using the PAMT prediction technique.

III.4.3 Analytical Measure of Improvement

A quantitative comparison of the three displacement prediction schemes (spatial, temporal, and PAMT) is difficult. To say anything substantive, some assumptions must be made. It will be assumed that the correct displacement was found for all pels along the leading edge of the moving area in the previous frame and that the interframe motion is constant. Note that these assumptions are rather stringent, but must be made to be able to isolate the improvements of the proposed motion prediction technique. There are two ways to measure the improvement. First, the number of pels whose displacement is correctly predicted with the PAMT technique but not with spatial prediction is the number of pels in the leading edge of the moving area whose $|\mathrm{DFD}| > \mathrm{T}$ and $\nabla_x I(i) = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$. The size of this area can be estimated by the product of $\mathbf{d}$ and the "length" of the leading edge. (See area L in Figure III.4) The same improvement is obtained over temporal prediction. The increase in the number of pels whose intensity is "correctly" predicted with PAMT prediction, but not with spatial (or temporal) prediction can be determined likewise.

The relative improvement in being able to identify the uncovered background portion of a frame is much more difficult to ascertain. It is probably most dependent on the similarity between the newly uncovered background and the nearby background which was visible in the previous frame.

Note that even if the true interframe motion is not constant, constant motion is a better estimate of the true motion than zero motion, since by the third law of thermodynamics (the law of entropy), bodies in motion tend to remain in motion. Thus the PAMT motion prediction technique is intuitively better than temporal prediction, since temporal prediction in effect assumes the moving body has not moved in the time interval $[t-1,t]$.

The validity of the second assumption (that the "correct" displacement was found in the previous frame for all pels along the leading edge of the moving area in the previous frame) is most dependent on whether the leading edge is one of the last edges scanned or one of the first. If the leading edge is one of the last, $\hat{\mathbf{d}}^t$ will probably have converged to $\mathbf{d}$; if it is one of the first, $\hat{\mathbf{d}}^t$ will probably not have converged to $\mathbf{d}$.

III.5 Simulations

In this section simulation results are presented which show the increased compression obtained by using the convergence analysis results and the increased compression obtained by transmitting zeroth-order entropy information. The total entropy bit rate is the sum of the information bits and addressing bits. The addressing bits in all the simulation results herein are calculated using a runlength encoding. In the first few simulation runs the information bits are based on taking the difference of consecutive non-zero error values (first-order entropy); in the later

runs the bits are calculated based on the error values directly (zeroth-order entropy).

The following parameters are used in the algorithm:

1) The motion vector is corrected if the DFD is greater than 3 out of 255.

2) A noise-suppression pre-filter is used. It does two thresholding operations. First it zeroes any frame differences less than four in magnitude. Secondly, if the four closest pels have a thresholded frame difference of zero, the intensity of the pel under consideration is set to the intensity at the same location in the previous frame (i.e., the frame difference is made zero). This pre-filter has been used in previous investigations of video teleconferencing by other researchers [1-3,5,21-24].

3) The intensity difference-of-errors sequence is quantized with a 33-level symmetric quantizer whose positive representative values are: 0, 4, 7, 11, 16, 21, 28, 35, 44, 53, 64, 77, 92, 109, 128, 149, 178.

4) The motion-predictor switches between straight interframe prediction and motion-compensated interframe prediction. Motion-compensated prediction is used when the sum of the DFD at the three closest pels on the previous line is less than the frame difference (FD) at those same pels. This is the technique proposed in [2].

5) As is usually done, the simulation starts with a previous frame at the receiver.

6) Spatial prediction of $\hat{\mathbf{d}}$ is used; specifically the $\hat{\mathbf{d}}$ at the location one line and one pel previous is used.

7) The displacement correction algorithm iterates at most once at each pel.

The algorithm is simulated on an actual image sequence that is typical of one that might occur in a video teleconferencing environment. A pair of pictures from the 60-frame (two second) sequence is shown in Figure II.1. The percent of interframe motion is plotted in Figure II.4. The sequence has a spatial resolution of 282 lines/frame by 448 pels/line and an intensity resolution of 8 bits/pel for an uncompressed bit rate of 1.01 Mbits/frame. The frames are interlaced.

The improvements in the intensity prediction and the motion prediction are shown by measuring the image compression, the image quality, and the motion predictability. The following tables summarize the simulation results. The first column indicates the ϵ and clip value (if applicable). The second column indicates the maximum entropy bits per frame and the average entropy bits per frame. This measures the image (intensity information) compression. The third column indicates the maximum and average mse for the predicted frames. In other words, the maximum average mse for any frame and the average mse for the whole 60-frame sequence. This is one measure of the image quality. The fourth column indicates the maximum and average number of unpredicted pels per frame (or the number of times the displacement estimate is updated). This measures the motion predictability and is a strong measure of the ability to implement the algorithm in real-time, since the motion-correction equation (equation (III-4)) is the most complex requiring an integer subtraction, a shift 10-12 bits right, a floating-point multiplication, and a floating point addition for each vector component at each iteration. The fifth

column contains the maximum and minimum value of the average gradient ($E\{\nabla_z I\}$) squared per frame. This determines the limits on ϵ. The sixth column indicates the maximum and minimum mse for the corrected frame, that is the picture quality which is actually viewed at the receiver.

III.5.1 Convergence Analysis Simulations

The motion-compensated (MC) algorithm was applied to the bobsjob sequence using various ϵ values and various clip values. The simulation results are tabulated in Table III.1.

ϵ/clip	entropy bits max/aver	predicted mse max/aver	number of unpredicted pels max/aver	average value of gradient squared min/max	corrected mse min/max
0.06250/0.0625	121081/ 90910	97.43/25.75	25242/16960	227.5/ 840.0	0.33/1.35
0.00200/1.0000	133962/100474	85.94/22.83	26918/19674	233.9/ 829.1	0.35/1.46
0.00100/ n.c.	130328/ 97571	88.24/23.43	26486/18545	221.8/1004.1	0.37/1.43
0.00050/ n.c.	123381/ 92351	87.33/22.98	25751/17309	219.3/ 724.9	0.38/1.38
0.00025/ n.c.	120663/ 89508	89.68/24.93	25182/16922	218.6/ 699.9	0.33/1.36
0.00010/ n.c.	118330/ 87888	100.71/28.13	24383/16298	227.7/ 602.1	0.26/1.28
0.00001/ n.c.	123801/ 90393	107.41/33.20	24650/15978	209.4/ 552.8	0.26/1.26

n.c. -- no clip

Table III.1: Simulation Results of Bobsjob Sequence

An ϵ value of 0.0010 and a clip value of 0.0625 were used in earlier pel-recursive research [1]. Other ϵ/clip values that have been used in reported simulations are 0.0078/0.2 and 0.5/0.08 [3]. There is, however, no analytical reason for these values; they were determined to be the best for a particular sequence by trial and error. The analysis in Section III indicated that $\hat{\mathbf{d}}$ should converge to $\mathbf{d}$ within the moving area without clipping the update values (see equations (III-3) and (III-4)) if the assumptions are valid and the constraints on ϵ are met. Some researchers have tried to justify clipping the update term [25,26]. One might think that by using a large value of ϵ and clipping the update term $\hat{\mathbf{d}}$ would converge to $\mathbf{d}$ faster. The simulation results in Table III.1 do not verify this fact. Granted all possible pairs of ϵ and clip values were not tried (that is what is trying to be avoided by doing the analysis), but the heuristic technique does not seem to offer any performance advantages over implementing the analytical technique, i.e., relatively small ϵ values and no clip. Also note that implementing the analytical model

reduces the computational requirements (no clipping or hardlimiting).

For spatial prediction with the ϵ values tried, the minimum total bit rate was obtained with $\epsilon = 10^{-4}$, the minimum average predicted mse was obtained with $\epsilon = 5 \times 10^{-4}$, and the minimum total number of unpredicted pels was obtained when $\epsilon = 10^{-5}$. (The number of unpredicted pels may be reducible further by reducing the ϵ value.)

The sum of the average values of $(\nabla_x I)^2$ and $(\nabla_y I)^2$ per frame varied from 209.4 (high motion) to 1004.1 (low motion). The average value of $\{(\nabla_x I)^2 + (\nabla_y I)^2\}$ over the whole sequence was about 300. For $\dot{\mathbf{d}}$ to converge to $\mathbf{d}$ in every frame with a fixed ϵ value, the largest value of $\{(\nabla_x I)^2 + (\nabla_y I)^2\}$ should be used to determine ϵ. For example, if $(\nabla_x I)^2 + (\nabla_y I)^2 = 1000$, ϵ is constrained to be less than 0.004 by equation (III-18). Equation (III-52) indicates that a smaller value of ϵ may be the optimum value. The simulations indicate such to be the case. Note that as a general rule the smaller ϵ values yielded a smaller maximum average gradient value.

With respect to the received picture quality, the smaller the value of ϵ, the smaller the average mse in the frames as actually viewed at the receiver. However, the maximum mse in the corrected frames for all runs was less than 1.50. Thus there was little visual degradations in any of the frames in any of the simulation runs on the bobsjob sequence.

Note that the optimum ϵ value was not determined analytically; only a range was obtained. Thus the optimum ϵ must be found by trial and error. However, only one vari-

able needs to be found, not two, and using a relatively small value of ϵ (10^{-3} to 10^{-5}) and no clip appears to reduce all the applicable measures from the results obtained by using a relatively large ϵ (0.0625 to 0.5000) and a clip (0.0625 to 1). In effect, better results can be obtained by doing less computation.

III.5.2 Zeroth-Order Entropy Information Transmission

In all the simulation runs reported so far the difference of consecutive intensity prediction errors was transmitted. This had been shown to reduce the bit rate in pel-recursive motion-compensated coders by 5-15% [1] over transmitting the intensity prediction errors directly. The only perceptual picture quality degradation in any of the previous simulation runs herein was streaking. In an attempt to remove these degradations, some of the simulations were rerun transmitting the intensity prediction errors directly. The results are in Table III.2. They were surprising. The bit rate was 12% lower and was minimized at a larger ϵ value. The number of unpredicted pels dropped by 17% and the average mse in the reconstructed frames dropped by 60%! All this with less computation.

Note that a lot of that mse in the reconstructed frames may not really be noise, but rather the absence of the noise which was in the original frame. Remember that a noise prefilter was used in the simulations and that the corrected mse is the error between the original picture and the picture viewed at the receiver.

ϵ	entropy bits max/aver	predicted mse max/aver	number of unpredicted pels max/aver	average value of gradient squared min/max	corrected mse min/max
0.00100	117476/77269	88.37/21.81	21507/14612	263.0/1043.7	0.00/0.58
0.00050	117819/76855	86.25/21.51	21197/13746	257.4/1158.6	0.00/0.59
0.00025	120287/78184	89.17/23.53	21448/13741	255.2/1053.1	0.00/0.57

Table III.2: Zeroth-Order Entropy Transmission of Bobsjob Sequence

IV. SUBSAMPLING AND INTERPOLATION IN VIDEO SEQUENCE CODING

IV.1 Introduction

This section addresses the subsampling issue as used in frame replenishment coding to reduce the bit rate requirement of the communication system.

Frame replenishment coding [30] is a special case of interframe coding [31] which takes advantage of the considerable similarity between successive frames (for example in TV sequences) in two ways:

(1) The parts of the picture which have not changed from frame to frame are not transmitted. At the receiver, they are reconstructed by repeating from the previous frame.

(2) The changing parts are transmitted with varying resolution depending on the subjective requirement of the quality and the bit rate of the transmission facility.

The changing parts of the picture are determined by subtracting an estimate of the current frame from it. In its simplest form a frame replenishment coder uses the previous frame as the estimate of the current frame. This scheme produces unacceptable results in lower bit rates and more sophisticated methods such as motion compensation [1,32] must be used to obtain a better estimate of the current frame.

One method for reducing the bit rate in a frame replenishment coder is to subsample in the spatial or temporal domain. Filtering along the time axis reduces the temporal resolution, the rapidity with which a given picture element can change. According to the sampling theory, the lower the temporal resolution, the lower the required sampling rate would be. The effectiveness of temporal filtering in frame replenishment coders is limited

by the fact that each stage of delay in the filter requires a frame memory. The bit rate is reduced more effectively by spatial subsampling [5]. Horizontal 2:1 subsampling like vertical subsampling gives appropriately a 2:1 data rate reduction and results in little degradation for moderate speeds of movements. Subsampling with larger ratios will increase the degradation above "acceptable" levels. Conventional uniform subsampling strategy and the simple averaging (over two nearest neighbors) interpolation method suffer in two ways; suboptimal subsampling strategies based on uniform grid requirements and suboptimal reconstruction algorithm.

The suboptimal subsampling strategy will result in the elimination of some "subjectively important" samples and hence introduce annoying effects in the reconstructed picture. Furthermore, the simple nearest neighbor averaging interpolation method does not take advantage of all the data available (other samples) at the receiver in the reconstruction process. In this work, a subsampling method is introduced in which the subsampling is performed in another coordinate system, in which more energy can be transmitted in fewer samples. The specific transformation used is the discrete cosine transform, however, other transforms such as Fourier, Hadamard, etc. may be used. The question of best transform is not considered. Second, the reconstruction problem is shown to be that of solving a large system of linear equations. An iterative method of solving the above system of equation, based on the method of successive projection onto convex sets, is proposed. A complete analysis of the solution and the iterative method is presented.

In Section IV.2 two restoration digital image methods are reviewed, Minimum Mean Square (Wiener) filtering and the method of successive projections onto convex sets.

These two methods are the basis of two interpolation techniques analyzed in sections IV.3 and IV.4. In Section IV.3 two different linear interpolation methods are analyzed. One is the conventional subsampling and interpolation scheme outlined above. The other one is based on uniform subsampling in the spatial (v.s. transform) domain but with Minimum Mean Square Error interpolation filter included at the receiver. Section IV.4 contains the analysis of the transform domain subsampling method and the corresponding reconstruction method. Finally, Section IV.5 presents some preliminary simulation results.

IV.2 A Review of Two Digital Image Restoration Technique

Image restoration is the discipline which one attempts to undo the degradation an image undergoes in processing and/or transmission. This section briefly reviews two methods of digital image restoration in such a way that seems appropriate for application to the bandwidth compression problem of interest to us.

IV.2.1 Minimum Mean Square Error (MMSE) Filter [33]

A simple signal formation model is given in equation (1).

$$g = Hf + n \tag{IV-1}$$

Where, g (in the Euclidean N-space,E^N) is the degraded signal; f (in E^N) is the original signal; H is a linear operator modeling the degradation process; and n (in E^N) is signal independent noise.

If the autocovariance matrices of the noise and the signal are known, a MMSE filter (Wiener filter) may be used to minimize the mean square error between the original and the reconstructed signal,i.e.

$$minimize\ E\{||f \quad f||^2\}. \tag{IV-2}$$

Where, $\hat{f} = Lg$ is a linear estimate and E{} is the mathematical expectation operator. The solution to this problem is given by,

$$\hat{f} = \{R_f H_t [HR_f H^t + R_n]^{-1}\} g, \tag{IV-3}$$

where, R_f and R_n are the autocorrelation matrices of the signal and the noise respectively. It is shown that the MMSE filter tends to attenuate the noise outside the significant band of the signal. This may cause the loss of high frequency information and result in a smooth solution. The MMSE estimate controls the ill-conditioning problem of H in a fashion that is determined by the signal to noise ratio as a function of frequency. Thus in low SNR cases the results do not appear as good usually as those provided by other techniques.

IV.2.2 Method of Successive Projection Onto Convex Sets (POCS)

L. M. Bergman [34] and later L.G. Gabin et. al. [35] described the method of successive projections for finding a point in the intersection of a number of convex sets. Youla and Webb [36] used successive projection onto closed convex sets as a signal reconstruction technique. The primary application of this work has been in extrapolation of band-limited signals [37,38], and the reconstruction of tomographic images [39]. A brief description of this method follows: Let C_i , i=1...M, be closed and convex sets in H with C representing their nonempty intersection. If P_i is the projection operator into the set C_i, then starting from any point f(0)$\in$H the iteration given by

$$f(k+1) = P_1 \quad P_2 \ldots P_m \cdot f(k) \tag{IV-4}$$

strongly converges to a point in C if H is finite dimensional (convergence is weak otherwise). Many a priori knowledge about the signal can be used to define appropriate convex sets [36], for example, positivity (in images), band-limitedness, and partial knowledge about signal. If f has to satisfy each of the constraints, it must belong to each of the convex sets, then it must also belong to the intersection, C. Therefore starting from any point in H, the iterations (IV-4) will converge to a point in the C called a feasible solution. Furthermore, since the projection operators are nonexpansive, the distance between the points in the sequence and the original signal is non-increasing.

IV.3 Linear Interpolation

In this section, two linear interpolation methods will be presented. First, uniform subsampling with nearest neighbor averaging, conventionally used in reducing the bandwidth requirement in frame replenishment techniques, will be analyzed. Second, a technique utilizing minimum mean square error interpolation of the subsampled data is presented and analyzed.

IV.3.1 Nearest Neighbor Interpolation

It is assumed that the samples taken from a line of the difference picture (of length N) are zero outside some region [a,b]. Let x denote the Px1 ($P \leq N$) vector of nonzero samples. The vector x is operated upon by a subsampling operator, [D], which may be presented by a PxP diagonal matrix with,

$$d_{ii} = \begin{cases} 0 & i = 0,2,4,\ldots \\ 1 & i = 1,3,5,\ldots \end{cases} \qquad \text{(IV-5)}$$

In equation (IV-1). $d_{ii} = 1$ correspond to the samples maintained and transmitted while $d_{ii} = 0$ correspond to the discarded samples. At the receiver the discarded samples are replaced by the arithmetic average of the two neighboring samples, i.e.

$$\hat{x}_i = \frac{g_{i-1} + g_{i-1}}{2} \qquad \text{(IV-6)}$$

where g is the contaminated (quantized) version of the vector x given by,

$$g_i = x_i + n_i \qquad \text{(IV-7)}$$

The error measured between the discarded and reconstructed samples is given by,

$$d_i = x_i - \hat{x}_i \quad \text{for all } i's \text{ such that } d_{ii} = 0. \qquad \text{(IV-8)}$$

Based on equation (IV-8), the mean square error per sample is found to be,

$$Ed_i^2 = \frac{3}{2}C_x(0) + \frac{1}{2}C_n(0) + \frac{1}{2}C_x(2) - 2C_x(1) \qquad \text{(IV-9)}$$

where $C_x(i)$ and $C_n(i)$ are the ith lag covariance coefficients of the signal, x, and noise, n, respectively.

IV.3.2 Minimum Mean Square Error Interpolation

Minimum Mean Square Error (MMSE) interpolation of the subsampled data is presented in this section. The analysis of this section will be used in a section IV.4 to compare the performance of this filter with that of the conventional interpolation scheme (see Sect. IV.3.1) and the method of signal reconstruction from partial knowledge in two coordinate systems.

In order to define the problem, let $x = (x_1, x_2, \ldots, x_N)$ be the original vector. Let [S] be the PxN subsampling matrix whose elements are all zero except one per row. The nonzero element in each row is equal to one. When a vector is multiplied by [S], those elements of x corresponding to the 1's on the diagonal of [S] will be maintained and the rest will be eliminated.

For example, a 2:1 uniform subsampling operator operating on a seven-dimensional vector may be represented by the 4x7 matrix,

$$[S] = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \tag{IV-10}$$

The subsampled data is then represented by the Px1 vector [S]x. In this section no assumption will be made with respect to the form of the matrix [S] and therefore the results are applicable to nonuniform sampling as well. Furthermore, no mention is made of the domain of the signal so that subsampling and interpolation may be performed in any domain. The only requirement is knowing the covariance matrix of the data, $[C_x]$, and noise $[C_n]$.

If the subsampled vector is contaminated with additive noise it can be represented by

$$g = [S]x + n \tag{IV-11}$$

where n is a Px1 vector of samples taken from uncorrelated noise with diagonal autocovariance matrix $[C_n]$. It is also assumed that the noise process is independent of the signal. This assumption may not be true in the case of quantization noise, however together

with the additive assumption it will simplify the problem considerably.

Define an (N-P)xN matrix, [D], such that [D]x be equal to the (N-P)x1 vector of eliminated samples. The matrix [D] has the same form as [S] with the nonzero elements corresponding to the discarded elements of x. For the example given in equation (IV-10) the matrix [D] is given by,

$$[D] = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}. \tag{IV-12}$$

The vectors of subsampled and eliminated elements of x are then,

$$\begin{aligned} [S]x &= (x_1, x_3, x_5, x_7)^T \\ [D]x &= (x_2, x_4, x_6)^T \end{aligned} \tag{IV-13}$$

The vector of contaminated subsampled elements in the example given in equation (IV-10) is,

$$g = (x_1+n_1, x_3+n_3, x_5+n_5, x_7+n_7)^T = (g_1, g_3, g_5, g_7)^T \tag{IV-14}$$

The MMSE interpolation filter minimizes the mean square error, $E\{||[D]x - g'||^2\}$, where $g' = [L]g$, is the output of the linear MMSE filter. The minimization is taken over the space of all linear operators [L] : $E^P \rightarrow E^P$.

To derive the interpolation filter, denote [D]x by y, then

$$\begin{aligned} & E\{||[D]x - [L]g||^2\} \\ &= E\{||y - [L]g||^2\} \\ &= E\{y^T y - y^T[L]g - g^T[L]^T y + g^T[L]^T[L]g\}. \end{aligned} \tag{IV-15}$$

Taking derivative with respect to [L], equating to zero and solving for [L] results in the MMSE interpolation filter as follows,

$$\frac{\delta E}{\delta [L]} = E\{-2yg^T + 2[L]gg^T\} = 0$$
$$E\{(y + [L]g)g^T\} = 0 \qquad \text{(IV-16)}$$

$$[\hat{L}] = E\{yg^T\}(E\{gg^T\})^{-1}$$
$$= [D][C_x][S]^T\{[S][C_x][S]^T + [C_n]\}^{-1} \qquad \text{(IV-17)}$$

The matrix $\{[S][C_x][S]^T + [C_n]\}$ is invertible and therefore $[\hat{L}]$ can be realized.

The resulting minimum mean square error is found as follow.

$$E\{||[D]x - [L]g||^2\}$$
$$= tr\ E\{([D]x - [\hat{L}]g)([D]x - [\hat{L}]g)^T\}$$
$$= tr\ E\{([D]x - [\hat{L}]g)x^T[D]^T\} \qquad \text{(IV-18)}$$
$$= tr\ \{[D]R_x[D]^T - [\hat{L}][S][R_x][D]^T\}$$
$$= tr\ \{([D] - [\hat{L}][S])[R_x][D]^T\}$$

The filter of equation (IV-17) is complex in its general form. However, if the covariance functions of signal and noise have simple structures, they can be reduced to simpler forms.

IV.4 Transform Domain Subsampling

There are a number of problems in digital signal processing that can be formulated as " Signal Reconstruction From Partial Knowledge in Two Coordinate Systems." Two examples of such problems are bandlimited extrapolation [40-42] and video sequence coding [43,44].

In bandlimited extrapolation problem, the signal is known in some finite region of the time (or space) coordinate system (domain) and it is known to be a sample version of some bandlimited process. The assumption (a priori information) that the signal is bandlimited is equivalent to assuming that the signal is equal to zero outside the passband of some ideal lowpass filter. If the sampling frequency is high enough so that the errors associated with the sampling process can be ignored, we can restrict our attention to the signals in N-dimensional Euclidean space. Then, the bandlimited extrapolation problem is equivalent to solving,

$$\begin{aligned} [D]f &= f_1 \\ [B]F &= 0 \end{aligned} \tag{IV-19}$$

where the matrices [B] and [D] in above equations are NxN diagonal matrices of the form,

$$b_{ii} = \begin{cases} 1 \text{ for } i = 0, 1, .., M \text{ and } i = N-M, N-M+1, .. .N-1 \\ 0 \text{ for } i = M+1, M+2, .., N-M-1. \end{cases} \tag{IV-20}$$

$$d_{ii} = \begin{cases} 1 \text{ for } i = P+1, P+2, .., N-P-1 \\ 0 \text{ for } i = 0, 1, .., P \text{ and } i = N-P, N-P+1, .. ,N-1. \end{cases}$$

In equation (IV-19) the letter F refer to the discrete Fourier transform (DFT) of the vector f.

In some applications of video sequence coding such as video teleconferencing, a large portion of the signal remains constant from frame to frame. This high degree of redundancy may be exploited by subtracting an estimate of the current frame from it. This action results in a frame difference which is zero outside the motion area. Conventionally spatial subsampling is used in order to reduce the bit rate required for transmission of the nonzero section of the frame difference. This type of subsampling suffers from fixed subsampling grid effects. That is, some large difference samples may be dropped and

therefore produce annoying effects in the reconstructed picture.

A solution to this problem is to perform the subsampling operation in another coordinate system where some useful criteria may be used for selecting the samples to be transmitted. For example, the subsampling operation may be performed after transforming the data using Fourier, cosine or Hadamard transform. Then, a simple criteria for selecting the samples to be transmitted is the magnitude of the transformed coefficients. This will result in more consistent performance of the subsampling and interpolation process. Figure IV.1 compares the percentage of energy transmitted when 2:1 subsampling is performed in the original and in the transform (cosine) domain. The signal used is a truncated first order Markov process. By this we mean the samples in the nonzero region are taken from a first order Markov process.

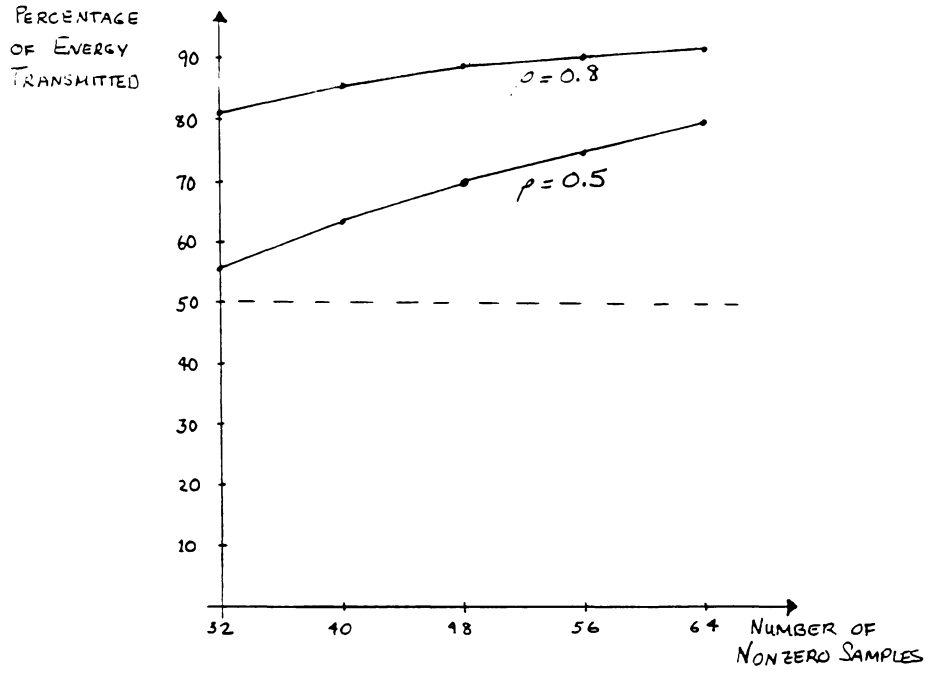

Figure IV.1: Percentage of energy transmitted when 2:1 subsampling

The problem with such a procedure, however, is the complexity involved with transforming sequences (nonzero segments) of arbitrary and changing length. This problem may be overcome by increasing the length of the nonzero sequence appropriately by adding enough zeros to the end of the sequence. Fast transform algorithms may then be used to transform the new sequence. Because of the zeros appended to the sequence, the energy compaction property of the transform operator will be reduced and a simple inverse transformation may not result in an "acceptable" reconstruction. This effect is shown in Figure IV.1.

From the above discussion, we conclude that a method is desired to exploit the spatial domain information in the reconstruction process. This can be done by solving the

system of linear equations.

$$\begin{aligned} [D]f &= 0 \\ [B]F &= F_0 \end{aligned} \tag{IV-21}$$

where, now, [B] and [D] are any diagonal matrix with diagonal elements equal to either one or zero, and F_0 present the selected transform domain samples of f. Furthermore, the transformation [W] : E^N–E^N defined by,

$$\begin{aligned} F &= [W]f \\ [W][W]^T &= I \end{aligned} \tag{IV-22}$$

may now be any orthogonal (as well as unitary) transformation and not just DFT.

An important source of degradation in video teleconferencing techniques is the effect of the edge of the moving object. Although the transform domain subsampling technique may preserve some of the edge information, it will not be able to minimize the degradation effects unless large number of samples are transmitted. This problem can be solved by modifying the system of equations (IV-21) to,

$$\begin{aligned} [D]f &= f_1 \\ [B]F &= F_0 \end{aligned} \tag{IV-23}$$

The extrapolation problem in equation (IV-19) is now just a special case of the new problem given by equation (IV-23). Now if some spatial domain information about some areas of interest are also transmitted, they can be incorporated in the reconstruction process as shown in equation (IV-23).

The solution of the system of equations given in equation (IV-23) is the objective of this work. In general, because of the nature of the digital signal processing problems the system of equations given by equation (IV-23) does not have a unique solution. In such

cases, a solution may be obtained by introducing additional constraints such as minimum norm.

There are many methods that can be used to solve a system of linear equations. However, because of the simple structure of the matrices [B] and [D] and the existence of a fast transform algorithm for the [W] transformation, we shall use an iterative method based on the method of successive Projection Onto Convex Sets (POCS) [45,46]. It was shown by Bergman [45] that starting from an arbitrary point in E^N the sequence generated by this method approaches a point which satisfies all the equations in equation (IV-23).

In the remainder of this section we will:

* determine a closed form for the solution of the system of equations (IV-23) and elaborate on the properties of the solution;
* determine the asymptotic rate of convergence;
* investigate the effects of the quantization noise on the solution; and
* look for means to reduce the effects of the quantization noise.
* derive the mean square error expression.

IV.4.1 Signal Reconstruction From Partial Knowledge In Two Coordinate Systems

Let E^N denote the N-dimensional Euclidean space with the usual inner product and norm defined as,

$$
\begin{aligned}
&<x,y> = x^T y = \sum_{i=1}^{N} x_i y_i \\
&||x|| = <x,x>^{1/2}
\end{aligned}
\tag{IV-24}
$$

Let [W] be the matrix presentation of a linear orthogonal transformation for which it is assumed that a fast transformation algorithm exists. Examples of such transformations are, discrete Fourier, discrete cosine and Hadamard transform. Then,

$$
\begin{aligned}
[W][W]^T &= [W]^T[W] = [I] \\
F &= [W]f \\
f &= [W]^T F
\end{aligned}
\tag{IV-25}
$$

where, F is the representation of f in the transformed (rotated) coordinate system.

Define the matrices [D] and [B] to be NxN matrices with diagonal elements one or zero, i.e.

$$
\begin{aligned}
d_{ii} &= 1 \text{ or } 0 \\
b_{ii} &= 1 \text{ or } 0
\end{aligned}
\tag{IV-26}
$$

The objective is to find an estimate of the vector f in E^N given some of its elements in the two coordinate systems, namely the original and the rotated coordinate systems. It is also assumed that none of the axes of the two systems are colinear. This assumption, plus the orthogonality of [W] guarantee that every N axes out of 2N are linearly independent and therefore constitute a basis for E^N.

The problem of finding (or estimating) f is equivalent to solving the following system of linear equations,

$$
[D]f = f_1 \tag{IV-27a}
$$

$$[B]F = F_0 \tag{IV-27b}$$

Equation (IV-27a) represents what is known about f in the original coordinate system and equation (IV-27b) represents what is known about f in the transformed coordinate system. The above system of equations when mapped to the transformed coordinate systems is given by.

$$[W][D][W]^T F = F_1 \tag{IV-28a}$$

$$[B]F = F_0 \tag{IV-28b}$$

where, $F_1 = [W]f_1$. The exact solution of the above system of equations may be found if,

$$\text{rank}([D]) + \text{rank}([B]) = N.$$

If, however,

$$\text{rank}([D]) + \text{rank}([B]) < N$$

Then there are infinitely many solutions from which one must be chosen as an estimate of F.

Conventional elimination and iterative methods are time consuming and require storage area for all the equations [47]. Therefore, in the remainder of this section an iterative method is described based on the method of Successive Projections Onto Convex Sets [45,46]. This method takes advantage of, the simple structure of the matrices [D] and [B], and a fast transform algorithm for [W].

Several simple definitions are needed before we can proceed with the description of the iterative method that is used to solve the system of equations given by (IV-23). Let C_D and C_B be the set of all vectors in E^N which satisfies equations (IV-28a) and (IV-28b)

respectively. That is,

$$C_D = \{X \in E^N : [W][D][W]^T X = F_1\}$$
$$C_B = \{X \in E^N : [B]X = F_0\}. \tag{IV-29}$$

Definition: A set C is said to be convex if, for every $\alpha \in [0,1]$

$$x \in C$$
$$y \in C$$

implies

$$\alpha x + (1 - \alpha)y = z \in C.$$

Using this definition it is easy to see that both C_D and C_B are convex.

A linear variety is composed of all vectors x of the form $x = g + y$ where $g \in E^N$ is fixed and y ranges throughout some subspace of E^N. Thus, a linear variety is any closed convex set that is produced by a translation of a subspace away from the origin. Consequently, the sets C_D and C_B are linear varieties produced by translation of the subspaces

$$S_D = \{X \in E^N : [W][D][W]^T X = 0\}$$
$$S_B = \{X \in E^N : [B]X = 0\} \tag{IV-30}$$

The fact that C_D and C_B are translations of the above two planes will be used in Corollary 1 to characterize the solution.

The projection operators onto C_D and C_B are rules which assign to every $X \in E^N$ its nearest neighbor in C_D and C_B, respectively. Denote these projection operators by $P_D : E^N \to C_D$ and $P_B : E^N \to C_B$, then

$$P_D(X) = [T]X + F_1 \tag{IV-32}$$

$$P_B(X) = ([I] - [B])X + F_0 \tag{IV-33}$$

where, $[T] = [W]([I] - [D])[W]^T$.

Algorithm

A solution to the system of the equations in (IV-23) may be found as follows. Starting from an arbitrary point, $X_0 \in E^N$, project successively onto C_D and C_B as shown in Figure IV.2.

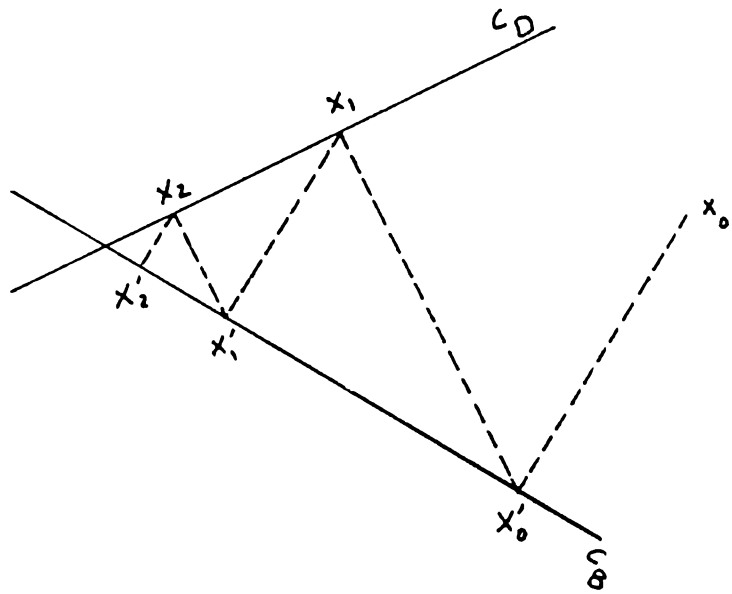

Figure IV.2 Simple demonstration of the projection method.

These points are given by,

$$\begin{aligned} X'_0 &= P_B(X_0) \\ X_1 &= P_D(X'_0) \\ X'_1 &= P_B(X_1) \\ X_2 &= P_D(X'_1) \\ &\textit{etc.} \end{aligned} \tag{IV-34}$$

The next theorem and the associated corollaries establish the convergence and characterize the solution as the projection of X_0 onto the intersection of C_D and C_B.

Theorem 1: [45,46]- The sequence $\{X_n\}$, as defined above, converges to a point on the intersection of C_D and C_B as $n \to \infty$.

This theorem is a special case of the theorem proved in [45] and [46]. The proof of this theorem is lengthy and hence omitted here. However, an alternative proof is presented later when a closed form relation is derived for the solution.

In general the limit point is not the projection of the starting point onto the intersection. However, as stated in the following corollary, this is the case when the sets are linear varieties . A general proof of Corollary 1 may be found in [46]. However, an alternative proof is given to provide more insight into the properties of the sequence generated by the projection method.

Corollary 1 [46]- The limit of the sequence $\{X_n\}$ is the projection of X_0 onto the intersection of C_D and C_B.

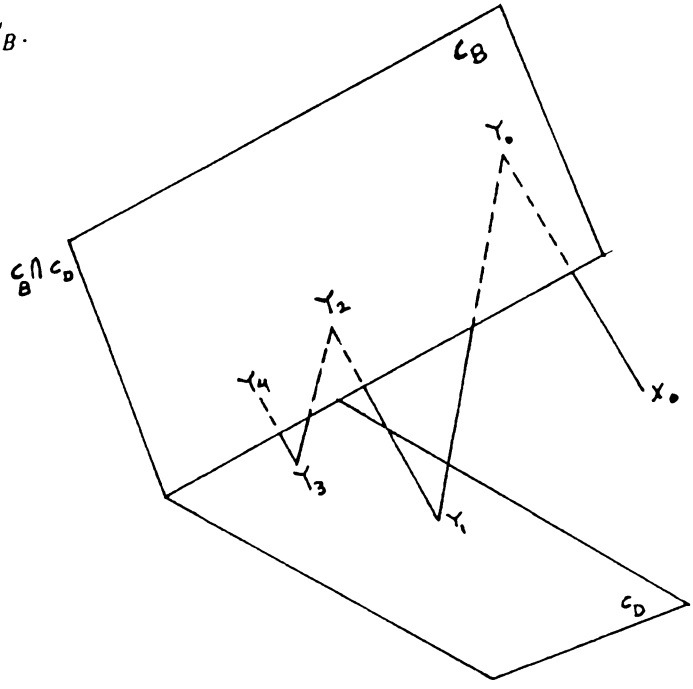

Figure IV.3 Three dimensional example of the projection method.

Proof - Figure IV.3 depicts a three-dimensional example of the projection method.

Denote by $C_B{}^p$ and $C_D{}^p$ the perpendicular complement of C_B and C_D, respectively. That is,

$$C_D{}^p = \{X \in E^N : \langle X, Y \rangle = 0 \text{ for } all \ Y \in C_D\}$$

$$C_B{}^p = \{X \in E^N : \langle X, Y \rangle = 0 \text{ for } all \ Y \in C_B\}$$

Although C_D and C_B may not be subspaces, their perpendicular complements will be.

From the description of the projection method and Figure IV.3 it is concluded that,

$$\begin{aligned} Y_{2k+1} - Y_{2k} &\in C_D{}^p \\ Y_{2k+2} - Y_{2k+1} &\in C_B{}^p \end{aligned} \tag{IV-35}$$

for all k= 0,1,2,.... Therefore, $Y_{j+1} - Y_j$ belong to $C_D{}^p + C_B{}^p$ for all j=0,1,.... It is easily shown that if $X \in C_D{}^p + C_B{}^p$ then $X \in (C_D \,.\, C_B)^p$, the intersection of the two sets. Hence,

$$Y_{j+1} - Y_j \in (C_D \,.\, C_B)^p. \tag{IV-36}$$

Since $(C_D \,.\, C_B)^p$ is a subspace of E^N, for every $Y_1, Y_2 \in (C_D \,.\, C_B)^p$ and scalars α and β,

$$\alpha Y_1 + \beta Y_2 \in (C_D \,.\, C_B)^p. \tag{IV-37}$$

Furthermore, since E_N is finite dimensional, $(C_D \,.\, C_B)^p$ is closed and

$$\lim_{j \to \infty} (Y_{j+1} - Y_j) \in (C_D \,.\, C_B)^p. \tag{IV-38}$$

Finally, since $(C_D \,.\, C_B)^p$ is a closed subspace it is concluded that,

$$\hat{X} - X_0 = \sum_{i=0}^{\infty} (X_{i+1} - X_i) \in (C_D \,.\, C_B)^p. \tag{IV-39}$$

This concludes the proof of Corollary 1. Since every X_n can be assumed a starting point of its own, it is also concluded that $\hat{X}$ is also the projection of each of the X_n's onto

the intersection of C_D and C_B. This in turn means that the norm square of the residual after n iterations can be written as,

$$||F - X_n||^2 = ||F - \hat{X}||^2 + ||\hat{X} - X_n||^2. \quad \text{(IV-40)}$$

That is, the error norm can be broken into two parts. One is the error between the signal and the limiting solution obtained by this method, while the second one is the distance remained to the solution.

Properties of [A]=[Λ][T] matrix

Before proceeding with the rest of the analysis, it is useful to consider some of the properties of the singular values of the matrices

$$[A] = [\Lambda][T] \quad \text{(IV-41)}$$

where,

$$[T] = [W][\Delta][W]^T \quad \text{(IV-42)}$$

and [Λ] and [Δ] are diagonal matrices with diagonal elements one or zero. As above,[W] is an orthogonal matrix, i.e.

$$[W][W]^T = [I].$$

All matrices are assumed to be NxN and real, extension to complex matrices ([W] unitary) is straight forward. The singular values of the matrix [A] are the square root of the eigenvalues of the matrix $[A][A]^T = [\Lambda][T][\Lambda]$ or equivalently $[A]^T[A] = [T][\Lambda][T]$. That is, the singular values of [A], $\lambda_k^{\frac{1}{2}}$, are given by,

$$[A][A]^T\phi_k = \lambda_k\phi_k \tag{IV-43}$$

$$[A]^T[A]\psi_k = \lambda_k\psi_k \tag{IV-44}$$

where to each λ_k corresponds an eigenvector ϕ_k *of* $[A][A]^T$ and ψ_k *of* $[A]^T[A]$.

Property 1 - $\lambda_k \in [0,1]$.

Proof: The matrix $[A][A]^T$ is symmetric and positive semi-definite, i.e.

$$([A][A]^T)^T = [A][A]^T$$
$$x^T[A][A]^T x = <x^T[A],(x^T[A])^T> \geq 0.$$

Therefore,

$$\lambda_k \geq 0 \qquad \text{for} \quad k = 1...N$$

Furthermore,

$$\begin{aligned}\lambda_{max} &= \max_x \frac{x^T[A][A]^T x}{x^T x} \\ &= \max_{|x|=1} x^T[A][A]^T x \\ &= \max_{|x|=1} <[A]^T x,[A]^T> \\ &= \max_{|x|=1} ||[A]x|| \leq \max_{|x|=1} ||[A]|| \leq 1.\end{aligned}$$

Hence, $-\leq \lambda \leq 1$ for $k = 1...N$.

Property 2 - ϕ_k is in the range of Λ satisfying,

$$[\Lambda]\phi_k = \phi_k \tag{IV-45}$$

Proof: This is seen from the definition of λ_k and ϕ_k and that $[\Lambda]$ is a projection operator.

Property 3 - λ_k is a nonzero (zero) eigenvalue of $[A][A]^T$ and ϕ_k its associated

eigenvector, if and only if λ_k and ϕ_k are a nonzero (zero) eigenvalue and its corresponding eigenvector of $[A]$ ($[A]^T$ in $\lambda_k = 0$ case).

Proof: Let

$$[A][A]^T\phi_k = \lambda_k\phi_k \tag{IV-46}$$

that is,

$$[A][T][A]\phi_k = \lambda_k\phi_k. \tag{IV-47}$$

Using Property 2, we obtain

$$[A][T]\phi_k = \lambda_k\phi_k. \tag{IV-47}$$

To prove in the other direction just reverse the order of the above equations. The statement for the zero eigenvalues is true because for any matrix [C],

$$[C][C]^T\phi = 0 \text{ if and only if } [C]^T\phi = 0. \tag{IV-48}$$

Property 4 - ψ_k is in the range of $[T] = [W][\Delta][W]^T$ satisfying, $[T]\psi_k = \psi_k$. This is seen from the definition of λ_k and ψ_k and that [T] is a projection operator.

Property 5 - λ_k is a nonzero (zero) eigenvalue of $[A]^T[A]$ and ψ_k its associated eigenvector if and only if λ_k and ψ_k are a nonzero (zero) eigenvalue and its corresponding eigenvector of $[A]^T$ ([A] in $\lambda_k = 0$ case).

Proof: As in Property 3.

Property 6 - For every ψ_k there exist a ϕ_k such that,

$$\psi_k = \frac{1}{\lambda_k^{1/2}}[T]\phi_k. \tag{IV-49}$$

Proof: Start by showing that $[T]\phi_k$ is an eigenvector of $[A]^T[A]$,

$$\begin{aligned}[A]^T[A][T]\phi_k &= [T][\Lambda][T][T]\phi_k \\ &= [T][\Lambda][T]\phi_k \\ &= [T][\Lambda][T][\Lambda]\phi_k \\ &= \lambda_k[T]\phi_k.\end{aligned}$$

Therefore, there exists a ψ_k corresponding to λ_k for which,

$$\psi_k = c\ [T]\phi_k. \tag{IV-50}$$

In order to determine c, we use the orthogonality of $\psi_k's$, i.e.

$$\psi_k{}^T\psi_k = 1, \tag{IV-51}$$

Then,

$$\begin{aligned}c^2\ \phi_k{}^T[T]\phi_k &= 1, \\ c^2\ \phi_k{}^T[\Lambda][T][\Lambda]\phi_k &= 1, \\ c^2\ \lambda_k\phi_k{}^T\phi_k &= 1, \\ c &= \frac{1}{\lambda_k{}^{1/2}}.\end{aligned} \tag{IV-52}$$

Property 7 - For every ϕ_k there exist a ψ_k such that,

$$\phi_k = \frac{1}{\lambda_k{}^{1/2}}[\Lambda]\psi_k. \tag{IV-53}$$

Proof: Similar to the proof of Property 6.

Definition [48] - A matrix [T] is said to be semiconvergent if the limit,

$$\lim_{j\to\infty}[T]^j \tag{IV-54}$$

exist.

Property 8 The matrix [A] is a semiconvergent matrix.

Proof: From Properties 3 and/or 5 it is seen that to each nonzero eigenvalue correspond a distinct eigenvector. Therefore, there exists one eigenvector corresponding to each zero eigenvalue too. In another words, the matrix x can be diagonalized by a similarity tansformation,

$$[A] = [P][\Lambda][P]^{-1}. \tag{IV-55}$$

Where the diagonal elements of [Λ] are the eigenvalues of [A] and [P] is the matrix of corresponding eigenvectors. Furthermore, the nonzero eigenvalues of [A] are equal to the nonzero eigenvalues of $[A][A]^T$ and hence they belong to the interval [0,1]. Now, it is easy to see that if the spectral radius of [A] is equal to 1 then [A] is a semiconvergent matrix. If the spectral radius of [A] is smaller than 1 then [A] is a convergent matrix.

We now proceed with the analysis of the iterative method outlined above. We start by presenting an alternative proof of the convergence which result in a closed form formula for the solution obtained by the above iterative method.

Corollary 2 - The limit of the sequence $\{X_n\}$, $\hat{F}$, is given by,

$$\hat{F} = \lim_{n\to\infty} X_n = ([I]-[H][H]^D)X_0 + [A]^{\dagger}F_0 + ([I]-[A]^{\dagger})F_1 \tag{IV-56}$$

where,

$$[H] = [I] - [T]([I]-[B]),$$
$$[A] = [B][T], \tag{IV-57}$$

and,

$$[T] = [W]([I]-[D])[W]^T. \tag{IV-58}$$

where $[A]^+$ denotes the pseudo inverse of [A].

Proof: The sequence $\{X_n\}$ can be generated by,

$$X_n = [T]\{([I]-[B])X_{n-1} + F_0\} + F_1 \tag{IV-59}$$

which may be expanded to,

$$X_n = [K]^n X_0 + \{\sum_{i=0}^{n-1}[K]^i\}[T]F_0 + \{\sum_{i=0}^{n-1}[K]^i\}F_1 \tag{IV-60}$$

where,

$$[K] = [T]([I]-[B]) \tag{IV-61}$$

and,

$$[K]^0 = [I]. \tag{IV-62}$$

The matrix [K] is of the form of matrix [A] whose properties were derived in the previous section. Therefore Properties 1 through 8 also apply to the matrix [K]. In particular, [K] is a semiconvergent matrix and by Lemma 2, Appendix B,

$$\lim_{n\to\infty}[K]^n = [I]-[E], \tag{IV-63}$$

where,

$$[E] = ([I]-[K])([I]-[K])^{D} \tag{IV-64}$$

Therefore, the first term on the right hand side converges to

$$\lim_{n\to\infty}[K]^{n} = ([I]-[K]([I]-[K])^{D}X_0 \tag{IV-65}$$

The second term in the right hand side of equation (IV-60), after some algebraic manipulation, can be reduced to,

$$\{\sum_{i=0}^{n-1}[K]^{i}\}[T]F_0 = \tag{IV-66}$$

$$\{\sum_{i=0}^{n-1}([B][T])^{T}\{[I]-([B][T])([B][T])^{T}\}^{i}\}F_0.$$

The above equation is in the form given in Lemma 1, Appendix B, and therefore,

$$\lim_{n\to\infty}\{\sum_{i=0}^{n-1}[K]^{i}\}[T]F_0 = ([B][T])^{\dagger}F_0. \tag{IV-67}$$

Finally the last term in eq.(IV-60) may be written as,

$$\{\sum_{i=0}^{n-1}[K]^{i}\}F_1 = \{\sum_{i=0}^{n-1}[K]^{i}\}([I]-[T])F_1, \tag{IV-68}$$

where,

$$[I]-[T] = [W][D][W]^{T} \tag{IV-69}$$

$$([I]-[T])F_1 = F_1. \tag{IV-70}$$

Again, as in the case of the second term, after some algebraic manipulations,

$$\{\sum_{i=0}^{n-1}[K]^i\}F_1 = \{[I] - \sum_{i=0}^{n-1}[T][B]([I] \quad [B][T][B])^i\}F_1 \tag{IV-71}$$

which by Lemma 1, Appendix 1, has the limit given by,

$$\lim_{n\to\infty}\{\sum_{i=0}^{n-1}[K]^i\}F_1 = \{[I]-([B][T])^{\dagger}\}F_1. \tag{IV-72}$$

Replacing the above results in equation (IV-60) completes the proof of the corollary.

IV-4.2 Asymptotic Convergence Rate

Define the error after n iteration as,

$$E_n = F - X_n. \tag{IV-73}$$

The above equation may be written as,

$$E_n = (F - \hat{F}) - (\hat{F} - X_n) \tag{IV-74}$$

and as was shown in corollary 1, the error norm square is given by,

$$||E_n||^2 = ||F - \hat{F}||^2 + ||\hat{F} - X_n||^2. \tag{IV-75}$$

The first term in the right hand side is not a function of n and is not equal to zero in general. The second term, however, approaches zero as n goes to infinity. Therefore it is the second term which determines the rate of convergence. Proceed by looking at the term

$\hat{F}-X_n$,

$$\begin{aligned}
\hat{F} - X_n &= \hat{F} - [T]\{([I]-[B])X_{n-1} + F_0\} - F_1 \\
&= \{[T]([I]-[B])\}(\hat{F} - X_{n-1}) \\
&= . \\
&= . \\
&= \{[T]([I]-[B])\}^n (\hat{F} - X_0).
\end{aligned} \tag{IV-76}$$

Note that although the limit of $\{[T]([I]-[B])\}^n$ *as* $n \to \infty$ is not zero, in general, the right hand side of equation (IV.46) does approach zero. In another words, $\hat{F} - X_n$ will lie in the null space of the limit of $\{[T]([I]-[B])\}^n$ as $n \to \infty$. The matrix $[T]([I]-[B])$ is a semi-convergent (convergent if $0 < \lambda < 1$) matrix for which the asymptotic rate of convergence is given by [48],

$$R_\infty = \delta \tag{IV-77}$$

where δ is the magnitude of the largest eigenvalue of $[T]([I]-[B])$ which is smaller than one, i.e.

$$\delta = \max\{\delta : |\delta| < 1\}. \tag{IV-78}$$

IV.4.3 Quantization Noise Effect

Assume that the given data is contaminated with quantization noise. Then the the method described in the previous sections will result in a solution of the following system of equations,

$$\begin{aligned}
[T]F' &= F_1 + \Delta F_1 \\
[B]F' &= F_0 + \Delta F_0
\end{aligned} \tag{IV-79}$$

As before, the sequence $\{X_n\}$, generated by the projection method may be written as,

$$
\begin{aligned}
X_n &= [T]\{([I]-[B])X_{n-1} + F_0 + \Delta F_0\} + F_1 + \Delta F_1 \\
&= [K]^n X_0 + \{\sum_{i=0}^{n-1}[K]^i\}[T](F_0 + \Delta F_0) \\
&+ \{\sum_{i=0}^{n-1}[K]^i\}(F_1 + \Delta F_1).
\end{aligned}
\tag{IV-80}
$$

Assuming that X_0 is not contaminated (is determined at the receiver), then $X_0 = 0$.

In this case, the sequence X_n approaches the solution $\hat{F}'$ given by,

$$
\begin{aligned}
&\hat{F}' = \lim_{n\to\infty} X_n = \\
&[A]^{\dagger}(F_0 + \Delta F_0) + F_1 + \Delta F_1 - [A]^{\dagger}(F_1 + \Delta F_1).
\end{aligned}
\tag{IV-81}
$$

If the signal and noise are statistically independent, the quantization noise effect can then be extracted, by comparing equations (IV-61) and (IV-81),

$$
\Delta F = \Delta F_1 + [A]^{\dagger}(\Delta F_0 - \Delta F_1), \tag{IV-82}
$$

where, $[A] = [B][T]$.

Now consider the singular value decomposition of $[A]^{\dagger}$. It is well known that, if

$$
\begin{aligned}
[A][A]^T\phi_k &= \lambda_k\phi_k \\
[A]^T[A]\psi_k &= \lambda_k\psi_k,
\end{aligned}
$$

then,

$$
[A]^{\dagger} = \sum_{k=1}^{p} \frac{1}{\lambda_k^{1/2}} \psi_k \phi^T{}_k. \tag{IV-83}
$$

It was shown in that (Property 7 above),

$$
\phi_k = \frac{1}{\lambda_k^{1/2}}[B]\psi_k. \tag{IV-84}
$$

eigenvalues of the matrix, $\sum_{i=0}^{n-1}[U]^i$ are given by,

$$\lambda\left(\sum_{i=0}^{n-1}[U]^i\right) = \begin{cases} 0 & u_{ii}=0 \\ n & u_{ii}=1 \\ \dfrac{1-u_{ii}}{1-u_{ii}} & u_{ii}\leq 1 \end{cases} \qquad \text{(IV-91)}$$

As is seen from equations (IV-80), (IV-90) and (IV-91), the larger the eigenvalues of the matrix [K] the more will be the quantization noise amplification. Also obvious from the above equation is that one easy way to control the quantization effect is to stop the iteration as soon as possible.

A simple lower and upper bound on the norm of the quantization noise can be shown to be given by,

$$\frac{1-u_{\min}^{n}}{1-u_{\min}}\,||[T]\Delta F_0+\Delta F_1|| \leq \Delta X_n|| \leq n||[T]\Delta F_0+\Delta F_1|| \qquad \text{(IV-92)}$$

where, $u_{\min}$ is the minimum nonzero eigenvalue of the matrix [K].

A special case of the above analysis is when $\Delta F_1=0$. In this event, equation (IV-88) is reduced to,

$$\begin{aligned} \Delta X_n &= \{\sum_{i=0}^{n-1}[K]^i\}[T]\Delta F_0 \\ &= \{\sum_{i=0}^{n-1}([T]([I]-[B])[T])^i\}\Delta F_0 \end{aligned} \qquad \text{(IV-93)}$$

The matrix [T]([I]-[B])[T] is symmetric and therefore it can be diagonalized by mean of a similar transformation,

$$[Q][T]([I]-[B])[T][Q]^T = [Z]. \tag{IV-94}$$

Where, [Q] is an orthagonal matrix and [Z] is diagonal with its diagonal elements equal to the eigenvalues of the matrix [T]([I]-[B])[T]. As before, these eigenvalues are in [0,1]. Following the same argument as for the general case ($\Delta F_1 \neq 0$), it can be shown that the same results can be obtained except for the eigenvalues of [T]([I]-[B])[T] replacing those of [K]=[T]([I]-[B]).

IV.4.4 Mean Square Error

It was shown in Corollary 2 that the solution obtained by the projection method, when starting from the origin, is given by,

$$\hat{F} = [A]^{\dagger}(F_0+\Delta F_0) + ([I]-[A]^{\dagger})(F_1+\Delta F_1). \tag{IV-95}$$

The purpose of this section is to derive an expression for the mean square error (MSE) defined as,

$$\epsilon = E\{||F - \hat{F}||^2\}. \tag{IV-96}$$

The error between F and $\hat{F}$ may be written as follow,

$$\begin{aligned} F - \hat{F} &= F - [A]^{\dagger}F_0 - ([I]-[A]^{\dagger})F_1 \\ &\quad - [A]^{\dagger}\Delta F_0 - ([I]-[A]^{\dagger})\Delta F_1 \\ &= [G][W]f - [A]^{\dagger}\Delta F_0 - ([I]-[A]^{\dagger})[W]\Delta f_1, \end{aligned} \tag{IV-97}$$

where, $[G] = [I] - [A]^{\dagger}[B] - [T] + [A]^{\dagger}[T]$.

The overall mean square error is then found to be,

$$\begin{aligned}\epsilon &= E\{(F - \hat{F})^T(F - \hat{F})\} \\ &= tr\ E\{(F - \hat{F})(F - \hat{F})^T\} \\ &= tr\ \{[G][W][C_f][W]^T[G]^T\} + tr\ \{[A]^\dagger[C_{\Delta F_0}][A]^{\dagger T}\} \\ &+ tr\ \{([I]-[A]^\dagger)[W][C_{\Delta f_1}][W]^T([I]-[A]^\dagger)^T\},\end{aligned} \tag{IV-98}$$

where, tr {[X]} means the trace of the matrix [X]. A simpler form for equation (IV-98) may be obtained using,

$$tr\ \{[P][\Delta][P]^T\} = tr\ \{[\Delta][P][P]^T\}. \tag{IV-99}$$

Simple algebraic manipulation will result in,

$$\begin{aligned}\epsilon &= tr\ \{[\Delta][G][G]^T\} \\ &+ tr\ \{[\Delta_{F_0}][A]^\dagger[A]^{\dagger T}\} \\ &+ tr\ \{[\Delta_{f_1}]([I]-[A]^\dagger)([I]-[A]^\dagger)^T\}.\end{aligned} \tag{IV-100}$$

In equation (IV-100), $[\Delta]$, $[\Delta_{F_0}]$ and $[\Delta_{f_1}]$ are the diagonal matrices of eigenvalues of the covariance matrices of the signal, the quantization noise in the transform domain and the quantization noise in the original domain respectively. This equation represents the mean square error as defined by equation (IV-98), with the first term represents the mean square error due to subsampling operation and the later two representing the quantization noise effects in the transform and the original domain respectively.

IV.5 Simulation Results

This section presents preliminary results obtained from the computer simulations. Comparisons are made between the new transform domain subsampling technique and the conventional subsampling technique. A truncated first order Markov process with

correlation coefficient ρ, unit variance, and zero mean is used as synthetic data. This implies an N dimensional vector which is zero outside [0,p], $p \leq N$ and a nonzero segment which is modeled with a first order Markov process. In addition, several examples of the application of the transform domain subsampling technique to lines extracted from real data are presented. The lines are extracted from frames containing different amounts of motion (18% and 5%), but both are taken from the same head and shoulder sequence, bobsjob (see Figure II.1).

The error measure used in this section is the Mean Square Error (MSE). However, where there is an attempt to compare the result of the transform domain subsampling method with the conventional subsampling method, mean square error may not yield the best representation of the error present. The reason is that the error is measured over the eliminated samples in conventional subsampling method, while in the transform domain technique it is spread over all nonzero segments. Therefore it seems reasonable to divide the MSE by the number of samples affected by the error. This new measure will be referred to as MSE per sample.

Effect of Nonzero Segment

A comparison of the performance of the conventional and transform domain subsampling techniques, as a function of the length of the nonzero segment, is presented in Figure IV.4. The extent of the nonzero segment depends on the amount of motion and the size of the object. In this experiment the length of the vector is 64 and the measurements are repeated for 32, 40, 48, 56 and 64 nonzero samples. The correlation coefficient of the Markov process is 0.7. In Figure IV.4, the solid line corresponds to the result of the

iterative method after three iterations. The dashed line present the performance of the conventional method. As seen from Figure IV.4. the performance of the conventional subsampling method does not depend on the length of the nonzero segment. The reason is that the reconstruction process only uses the samples of the two nearest neighbors.

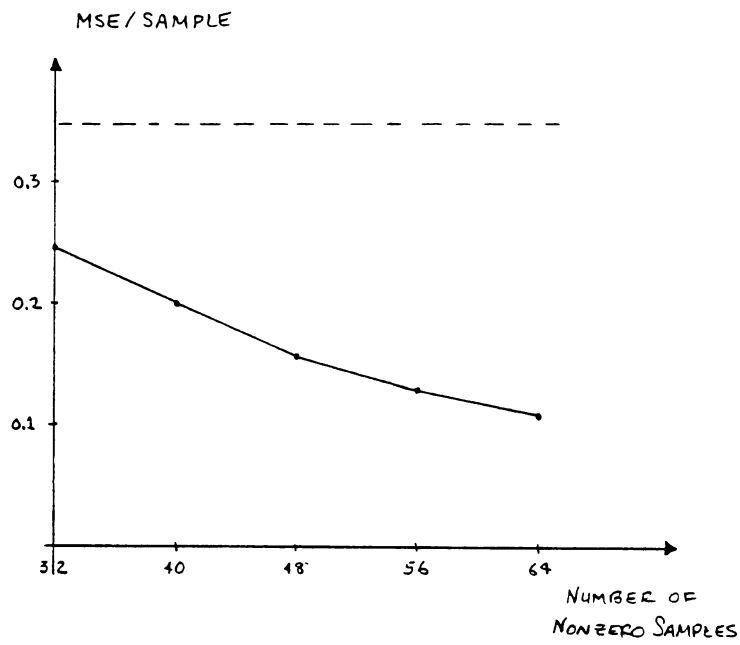

Figure IV.4 MSE per sample vs number of nonzero samples.
- - - - - Conventional method.
_____ Transform domain method.

On the other hand, in transform domain subsampling method and its iterative reconstruction method the MSE is reduced as the number of nonzero samples increases. This occurs for two reasons; the energy compaction property of the transform kernel (see Figure IV.1); and the fact that the reconstruction algorithm takes advantage of all the information at its disposal.

Speed of Convergence

Figure IV.5 presents the results of estimating the speed of convergence measure using a Monte Carlo Method. The measurement is calculated using 400 lines of the synthetic data with correlation coefficient equal to 0.7. The length of the zero segment cosen to be 48. Also shown in this Figure IV.5 is the standard deviation of the measured value. The relatively slow speed of the convergence is a characteristic of the method of Succesive Projection onto Convex Sets which is the basis of our iterative method. However, it is important to note that the largest improvements are made in the first few iterations. In this figure the the step size reduced from 2% to 0.7%.

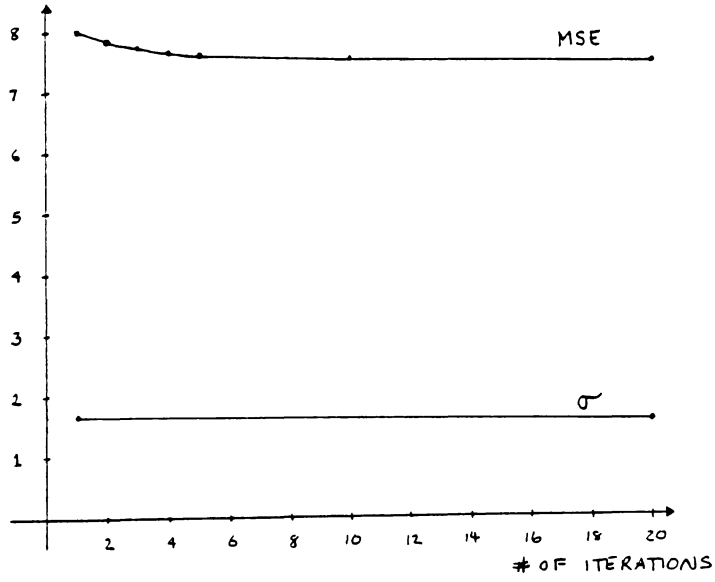

Figure IV.5: Speed of convergence.

Effect of the correlation coefficient

Figure IV.6 compares the performance of the conventional and the transform domain subsampling method as a function of the correlation coefficient of the Markov process. The length of the nonzero segment is fixed at 48. In both cases the performance is deteriorated by reducing the sample to sample correlation. However, the rate of increase in the MSE is much faster in the conventional method. This phenomena may be explained as follows. For large correlation coefficient, the knowledge of the two nearest neighboring samples will convey a large amount of information to the reconstruction algorithm. Therefore, the conventional method is capable of a "good" reconstruction. The transform domain technique, however, can pack more energy in the transmitted samples and at the same time uses the information about all those samples to reconstruct the signal, thus performing better than the conventional method.

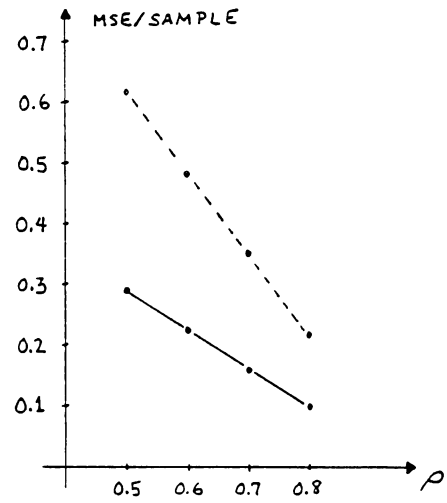

Figure IV.6 Effect of the correlation coefficient.

When the correlation coefficient is reduced, the conventional method loses more information than the transform domain technique because the latter does not rely only on the two neighboring samples.

Examples with real data

Two lines where extracted from the bobsjob sequence. Each line is 256 pixels long. One corresponds to an area of 18% motion and the other to and area of 5% motion. The length of the nonzero segment is 100 pixels for the 18% motion and 72 for the 5% motion line. Figure IV.7 shows the original and the reconstructed version of each of the lines using 25 FFT points after four iterations.

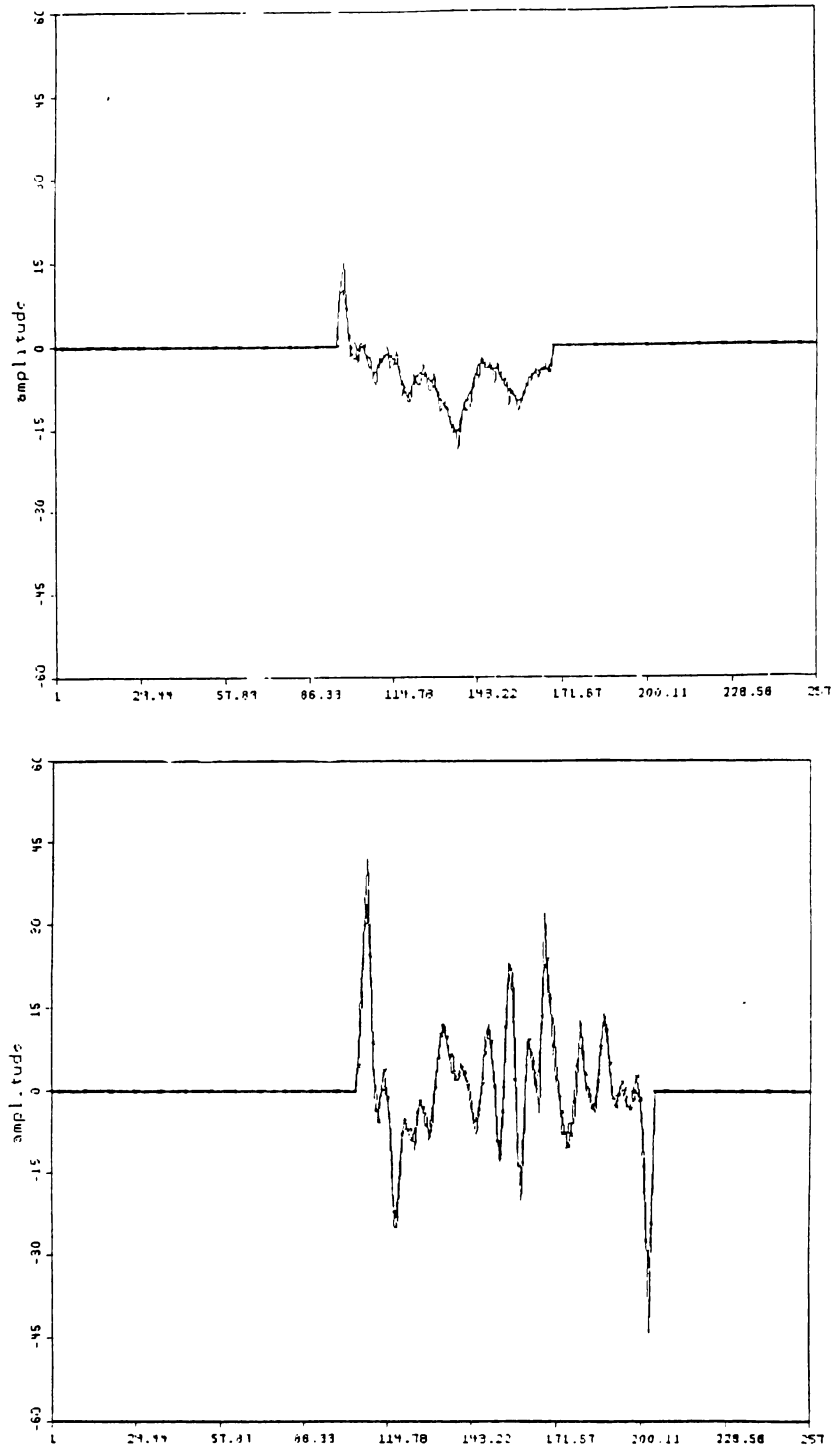

Figure IV.7 Original (plane line) and the reconstructed version (crossed line) from 25 FFT coefficients after four iterations. a-18% motion; b-5% motion.

The edge of the moving object is the source of one or more large spikes as seen in Figure IV.7. In the spatial domain subsampling techniques, it is possible that one or more of these spikes may be eliminated, therefore creating objectionable effects on in the reconstructed images at the receiver. This effect is more apparent at lower bit rates. In the transform domain subsampling technique, they are not eliminated however, they are not reconstructed exactly (see Figure IV.7). Again as the bit rate is reduced, the degradations become more apparent.

Transmitting some information (residual) about the spikes can be used to improve the picture quality in their neighborhood. Figure IV.8 is an example of such a case, where the 22 largest FFT points are used with two spatial samples representing the two spikes on either sides of the line.

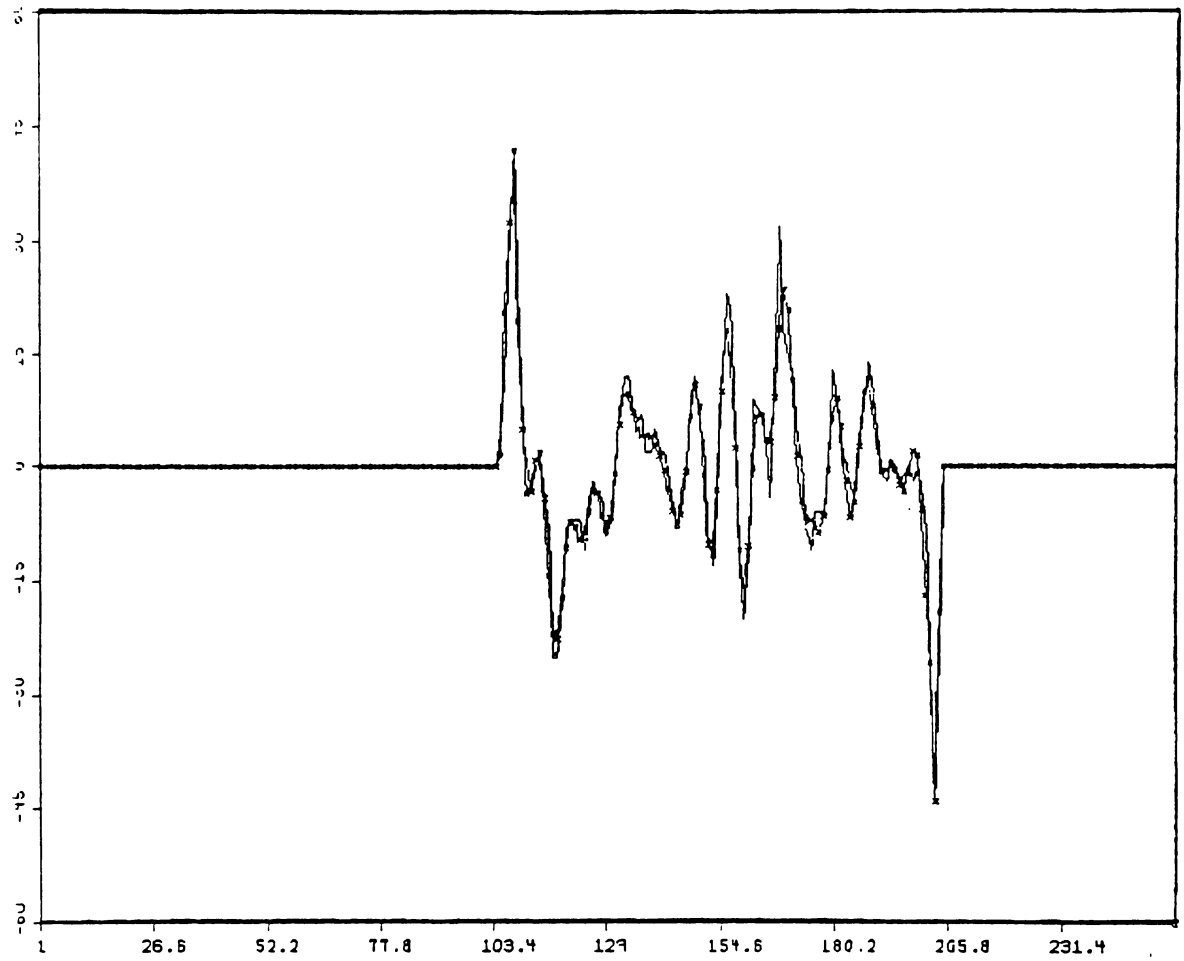

Figure IV.8: Original v.s. 22 best FFT + 2 spatial domain sample.

V. CONCLUSIONS

This report presents results of the research for the "Video Teleconferencing Project" in CCSP for the period July 1, 1984 - June 30, 1985. Additional CCSP documents associated with this project include: CCSP-WP-84/11, CCSP-TR-84/12, CCSP-TR-85/11, and CCSP-TR-85/12.

V.1 Image Statistics

A presentation of the statistical characteristics of image sequences is included in Section II. It includes information on the percent motion, the noise statistics, the image statistics, and the motion statistics of three different image sequences.

The statistics generated are in reasonable agreement with those used for modeling and analysis, although the correlation values calculated for these three sample processes may be a little high in general. Also it showed that the variance for the noise is for the difference sequence and that the variance for the actual sequence would be one half of this value. Further, the motion percentage is probably over estimated. This problem is being studied.

These statistics provide valuable information when analyzing algorithms and generating image sequence models. They can be used to confirm any assumptions used which before were only educated guesses or based on other data.

V.2 Motion-Compensated Video Teleconferencing

Five accomplishments have been reported in Section III:

(1) The pel-recursive motion-compensated algorithm has been thoroughly analyzed. The displacement estimate is seen, at convergence, to consist of two components: a mean component which converges toward the true displacement at a rate dependent upon the convergence coefficient, ϵ, and a random, noise-like component whose variance is directly proportional to ϵ. The benefit of a more rapid convergence to the true displacement with larger ϵ values is offset by the increased estimate variance. The analytical model is shown by simulation to yield better results than the heuristic technique of large ϵ values and clipping the update term in equation (III-4).

(2) A measure of the gradient estimation noise has been developed.

(3) A measure of the variance of the motion estimate at convergence has been derived.

(4) A new pel-recursive motion-compensate algorithm has been presented.

(5) The previously shown fact of greater information compression by coding the difference of consecutive intensity errors was shown to be false for at least one sequence. A 10% decrease was obtained by coding the consecutive intensity errors directly.

Some facts should be noted about these image sequences and the simulation parameters when comparing these simulation results and the simulation results of other researchers.

1) The motion in the sequence used for the simulations herein is lower than the motion in the sequences used in most other motion-compensated codec simulations [1-3,5-7,23-24,27-29].

2) The frames were not low-pass filtered to blur the edges. The sequences are very high quality sequences with sharp edges.

3) The results are affected by the endpoints of the quantizer bins and the threshold values in the codec. The values used herein are noted at the beginning of Section IV.

V.3 Subsampling and Interpolation in Image Sequence Coding

A thorough analysis of the problem of subsampling in image sequence coding has been presented. A new technique for performing the subsampling in the transform domain was developed and it is shown that the reconstruction problem at the receiver is equivalent solving a large system of linear equations. An iterative method is used to solve the system of equations. The analysis of this reconstruction method includes a closed form formula for the solution, derivation of the assymptotic rate of convergence, derivation of the mean square error, and analysis of the effect of the quantization noise. It is shown that by stopping the iteration in the first few steps, adequate reconstruction is achieved while controlling the effect of the additive noise. Finally, initial simulation results were presented for truncated first order Markov processes and some examples are given for real data.

VI. AREAS FOR FURTHER RESEARCH

VI.1 Image Statistics

Two additional pieces of statistical information that would be useful are a complete correlation matrix and tighter statistics of the motion information. The only thing precluding the completion of the correlation matrix is computer time. It simply requires a lot of calculations and these did not seem warranted at this point. The calculation of the statistics of the motion information is more difficult. In order to calculate such statistics, one would need a good segmentation of the motion region and a way to deal with irregularly shaped regions.

VI.2 Motion-Compensated Video Teleconferencing

Two questions remain about the analysis:

(1) Can it be assumed that the mixed partial derivatives with respect to the intensity function are zero? (See equation (III-9).) If so the analysis of ϵ for convergence is much easier.

(2) Is the convergence analysis approach used herein the best approach? It is **the** approach which has been used when only one iteration is performed at each sample point and convergence over time (or space) is sought. Can a tighter approach be had without requiring all the computation of Nagel's approach [10,13]?

A number of things have already been mentioned which could improve the performance of the pel-recursive motion-compensated compression algorithm.

A technique that could reduce unnecessary iterations is a post-processing filter to smooth the displacements within the moving areas and within the background. Predicting field-to-field motion instead of frame-to-frame motion also might help. Most researchers who have simulated a switched predictor have used field-to-field intensity prediction with motion-compensated (MC) and frame-to-frame intensity prediction with conditional replenishment (CR) [1-4]. This tactic results in a precise prediction of the non-moving background by using frame-to-frame prediction, but smaller displacement vectors by using field-to-field prediction.

Something that should be done is investigate the effect of signal noise. Although an analytical measure of the effect was derived herein and the sequences do contain noise, no simulations were run on image sequences to which noise had been intentionally added.

Some advantages might accrue by using a fixed predictor for the intensity instead of switching between conditional replenishment (CR) and motion-compensated (MC). When using a fixed predictor, some researchers have advocated resetting $\hat{\mathbf{d}}^t$ when $\sum FD < \sum DFD$ within a previously scanned area [6-7]. This could be viewed as a switched corrector.

VI.3 Subsampling and Interpolation in Video Sequence Coding

More quantitative information on the new subsampling technique would be desirable. This can be obtained by performing further simulations on synthetic and real data.

A second issue that has not been addressed in this portion of the research is the appropriateness of the new subsampling technique if real-time constraints are imposed.

Further research could include an evaluation of the algorithm performance in a real-time environment.

VII. REFERENCES

[1] A. N. Netravali and J. D. Robbins, "Motion Compensated Television Coding, Part I," *BSTJ,* Vol. 58, No. 3, pp. 631-670, March 1979.

[2] J. D. Robbins and A. N. Netravali, "Interframe Television Coding using Movement Compensation," *Proc. Int. Conf. Communications,* pp. 23.4.1-23.4.5, 1979.

[3] A. N. Netravali and J. D. Robbins, "Motion-Compensated Coding: Some New Results," *BSTJ,* Vol. 59, No. 9, pp. 1735-1745, Nov. 1980.

[4] R. Paquin and E. Dubois, "A Spatio-Temporal Gradient Method for Estimating the Displacement Field in Time-Varying Imagery," *Computer Vision, Graphics, and Image Processing,* Vol. 21, pp. 205-221, 1983.

[5] J. D. Robbins and A. N. Netravali, "Spatial Subsampling in Motion-Compensated Television Coders," *BSTJ,* Vol. 61, No. 8, pp. 1895-1910, Oct. 1982.

[6] D. R. Walker and K. R. Rao, "New Techniques in Pel-Recursive Motion Compensation," *Proc. Int. Conf. Communications,* pp. 703-706, May 1984.

[7] D. R. Walker and K. R. Rao, "Improved Pel Recursive Motion Compensation," *IEEE Trans. on Communications,* Vol. COM-32, No. 10, pp. 1128-1134, Oct. 1984.

[8] B. K. P. Horn and B. G. Schunck, "Determining Optical Flow," *Artificial Intelligence,* Vol. 17, No. 1-3, pp. 185-203, Aug. 1981.

[9] B. G. Schunck and B. K. P. Horn, "Constraints on Optical Flow Computation," *Proc. IEEE Computer Society Conf. Pattern Recognition and Image Processing,* pp. 205-210, Aug. 1981.

[10] H.-H. Nagel, "Constraints for the Estimation of Displacement Vector Fields from Image Sequences," *IJCAI,* pp. 945-951, Aug. 1983.

[11] P. R. Beaudet, "Rotationally Invariant Image Operators," *Proc. Int. Conf. on Pattern Recognition,* pp. 579-583, 1978.

[12] T. S. Huang and R. Y. Tsai, "Image Sequence Analysis: Motion Estimation," in *Image Sequence Analysis,* (T. S. Huang, Ed.), pp. 1-18, Springer-Verlag, Berlin, 1981.

[13] H.-H. Nagel and W. Enkelmann, "Towards The Estimation of Displacement Vector Fields by 'Oriented Smoothness' Constraints," *Proc. Int. Conf. on Pattern Recognition,*

pp. 6-8, Aug. 1984.

[14] R. J. Moorhead and S. A. Rajala, "Motion-Compensated Interframe Coding," *Proc. Int. Conf. Acoustics, Speech, and Signal Processing,* March 1985.

[15] B. Widrow, J. Glover, J. McCool, et. al., "Adaptive Noise Canceling: Principles and Applications," *Proc. IEEE,* Vol. 63, No. 12, pp. 1692-1716, Dec. 1975.

[16] L. Ljung, "Analysis of Recursive Stochastic Algorithms," *IEEE Trans. on Automatic Control,* Vol. AC-22, No. 4, pp. 551-575, Aug. 1977.

[17] B. Widrow and J. M. McCool, "A Comparison of Adaptive Algorithms Based on the Methods of Steepest Descent and Random Search," *IEEE Trans. Antennas and Propagat.,* Vol. AP-24, pp. 615-637, Sept. 1976.

[18] S. T. Alexander, "Adaptive Image Compression Using the Least Mean Square (LMS) Algorithm," Ph.D. Dissertation, North Carolina State University, 1982.

[19] E. Dubois, B. Prasada, and M.S. Sabri, "Image Sequence Coding," in *Image Sequence Analysis* (T.S. Huange, Ed.), pp. 229-287, Springer-Verlag, Berlin, 1981.

[20] W.E. Snyder, S.A. Rajala, and G. Hirzinger, "Image Modeling: The Contoinuity Assumption and Trackings," *Proc. 1980 Int. Joint Conf. on Pattern Recognition,* pp. 1111-1114, 1980

[21] B. G. Haskell, P. L. Gordon, R. L. Schmidt, and J. V. Scattaglia, "Interframe Coding of 525 line, Monochrome Television at 1.5 Mbits/s," *IEEE Trans. on Communications,* Vol. COM-25, No. 11, pp. 1339-1348, Nov. 1977.

[22] H. Kuroda, N. Mukawa, T. Matsuoka, and S. Okudo, "1.5 Mbit/s Interframe Codec for Video Teleconferencing Signals," *1982 IEEE Globecom,* pp. E2.5.1-E2.5.5, Nov. 1982.

[23] K. A. Prabhu and A. N. Netravali, "Motion Compensated Component Color Coding," *IEEE Trans. on Communications,* Vol. COM-30, No. 12, pp. 2519-2527, Dec. 1982.

[24] K. A. Prabhu and A. N. Netravali, "Motion Compensated Composite Color Coding," *IEEE Trans. on Communications,* Vol. COM-31, No. 2, pp. 216-223, Feb. 1983.

[25] C. Cafforio and F. Rocca, "The Differential method for Motion Estimation," *NATO ASI Series, Image Sequence Processing and Dynamic Scene Analysis,* (T.S. Huange, Ed.), Vol. 2, pp. 104-124, Springer-Verlag, 1983.

[26] C. Cafforio, "Remarks on the Differential Method for the Estimation of Movement in Television Images," *Signal Processing,* Vol. 4, pp. 45-52, 1982.

[27] J. A. Stuller and A. N. Netravali, "Transform Domain Motion Estimation," *BSTJ*, Vol. 58, No. 7, pp. 1673-1702, Sept. 1979.

[28] A. N. Netravali and J. A. Stuller, "Motion Compensated Transform Coding," *BSTJ*, Vol. 58, No. 7, pp. 1703-1708, Sept. 1979.

[29] J. A. Stuller, A. N. Netravali, and J. D. Robbins, "Interframe Television Coding Using Gain and Displacement Compensation," *BSTJ*, Vol. 59, No. 7, pp. 1227-1240, Sept. 1980.

[30] B. G. Haskell, "Frame Replenishment Coding of Television," in *Image Transmission Techniques* (W. K. Pratt editor), Academic Press, 1979.

[31] J.A. Roese, W.K. Pratt and G.S. Robinson, "Interframe Cosine Transform Image Coding," *IEEE Trans. Commun.*, Nov. 1977, pp. 1329-1339.

[32] J. R. Jain and A. K. Jain, "Displacement Measurement and its Application to Interframe Coding," *IEEE Trans. Commun., COM-29,* Dec. 1981, pp. 1799-1808.

[33] H. C. Andrews, B. R. Hunt, *Digital Image Restoration,* Prentice-Hall Signal Processing series, 1977.

[34] L. M. Bergman, "The Method of Successive Projection for Finding The Commom Point of Convex Sets," *Dokl. Akad. Nauk. SSSR 162, no.3,* 1965, pp. 688-692.

[35] L. G. Gubin, B. T. Polyak and E. V. Paik, "The Method of Projection for Finding The Common Point of Convex Sets," *U.S.S.R Computational Math. amd Math. Phys. 7,* no. 6, 1967, pp. 1-24.

[36] D. C. Youla and H. Webb, "Image Reconstruction by the Method of Convex Projections, Part 1- Theory," *IEEE Trans. on Medical Imaging,* vol. MI-1, no. 2, Oct. 1982, pp. 81-94.

[37] A. Papoulis, "A New Algorithm is Spectral Analysis and Bandlimited Extrapolation," *IEEE Trans. Circ. and Sys.*, vol. CAS-22, no. 9, Sept. 1975.

[38] R. W. Gerchberg, "Super Resolution Through Error Energy Reduction," *Opt. Acta.*, vol. 14, no. 9, Sept.1979, pp. 704-720.

[39] M. I. Sezan and H. Stark, "Image Restoration by the Method of Convex Projections: Part 2- Application and Numerical Results," *IEEE Trans. Medical Imaging,* vol. MI-1, no. 2, Oct. 1982.

[40] H. C. Andrews and W. K. Pratt, "Fourier Transform Coding of Images," *Hawaii Int.*

Conf. Sys. Sci., Western Periodicals Co., North Hollywood, CA, pp. 677-679.

[41] W. K. Pratt, J. Kane, H. C. Andrews, *Pro. IEEE 57, pp.58-68,1969.*

[42] A. Habibi, P. Wintz, *IEEE Trans. Commun. Technol., vol. 19, 1971, pp. 957-972.*

[43] N. Ahmed, T. Natrajan, K. Rao, "Discrete Cosine Transform," *IEEE Trans.* on Comput., Jan. 1974, pp. 90-93.

[44] A. G. Tescher, "Transform Image Coding," in *Image Transmission* Techniques (W. A. Pratt editor), Academic Press, 1979.

[45] L. M. Bergman, "The Method of Successive Projection for Finding The Commom Point of Convex Sets," *Dokl. Akad. Nauk. SSSR 162,* no.3, 1965, pp. 688-692.

[46] D. C. Youla and H. Webb, "Image Reconstruction by the Method of Convex Projections, Part 1- Theory," *IEEE Trans. on Medical Imaging,* vol. MI-1, no. 2, Oct. 1982, pp. 81-94.

[47] Ralston, A., *First Course in Numerical Analysis, Mc. Graw Hill book* co., 1965. [48] Abraham Berman and Robert J. Pelmmons, "Nonnegative Matrices in the Mathematical Sciences, Computer Sciences and Applied Mathematics," Academic Press, 1979, p.179.

Appendix A

Derivation of the Gradient Estimation Noise Covariance, Σ^2_{gn}

This appendix contains the derivation of the noise covariance in the gradient $\nabla_{\dot{\mathbf{d}}}[DFD(\mathbf{z}_a,\dot{\mathbf{d}})]^2\Big|_{\dot{\mathbf{d}}=\dot{\mathbf{d}}'}$. The gradient estimation noise covariance is evaluated in terms of the expected gradient values and the noise in the image. The gradient estimation noise, $\mathrm{N}(i)$, is the difference between the computed value of the gradient and its true value. Therefore it can be written as:

$$\mathrm{N}(i) = \hat{\nabla}_{\mathbf{d}} - \nabla_{\mathbf{d}}, \tag{A-1}$$

where $\hat{\nabla}_{\dot{\mathbf{d}}}$ is the computed value of the gradient and $\nabla_{\dot{\mathbf{d}}}$ is the true value. $\nabla_{\dot{\mathbf{d}}}$ can be evaluated from equation (III-2) to be

$$\nabla_{\dot{\mathbf{d}}}[DFD(\mathbf{z}_a,\dot{\mathbf{d}})]^2\Big|_{\dot{\mathbf{d}}=\dot{\mathbf{d}}'} = 2DFD(\mathbf{z}_a,\hat{\mathbf{d}}^i)\nabla_{\mathbf{z}}I(\mathbf{z}-\hat{\mathbf{d}}^i,t-1)\Big|_{\mathbf{z}=\mathbf{z}_a}. \tag{A-2}$$

Substituting (A-2) into (A-1) produces:

$$\mathrm{N}(i) = 2(DFD + e_{DFD})(\nabla_{\mathbf{z}}I + e_{\nabla_{\mathbf{z}}}) - 2(DFD)(\nabla_{\mathbf{z}}I). \tag{A-3}$$

where

$DFD = DFD(\mathbf{z}_a,\hat{\mathbf{d}}^i)$,

e_{DFD} *is the DFD error*,

$\nabla_{\mathbf{z}}I = \nabla_{\mathbf{z}}I(\mathbf{z}-\hat{\mathbf{d}}^i,t-1)\Big|_{\mathbf{z}=\mathbf{z}_a}$, and

$e_{\nabla_{\mathbf{z}}}$ *is the spatial gradient error.*

Let

$$I_1 = I(\mathbf{z}_a, t), \quad \text{and} \tag{A-4}$$

$$I_2 = I(\mathbf{z}_a - \hat{\mathbf{d}}^t, t-1). \tag{A-5}$$

Assume the measurement error in both I_1 and I_2 is due solely to noise. I.e., there is no error due to interpolation in the measurement of I_2. Denote the error in the two intensity measures as e_1 and e_2, respectively, and denote the noise error in the two components of the gradient as e_x and e_y. Evaluating the DFD and writing the vectors in component form,

$$\begin{aligned}
\mathrm{N}(i) &= 2[(I_1+e_1) - (I_2+e_2)]\begin{bmatrix}\nabla_x I + e_x \\ \nabla_y I + e_y\end{bmatrix} - 2(I_1-I_2)\begin{bmatrix}\nabla_x I \\ \nabla_y I\end{bmatrix} \\
&= 2[(I_1-I_2) + (e_1-e_2)]\begin{bmatrix}\nabla_x I + e_x \\ \nabla_y I + e_y\end{bmatrix} - 2(I_1-I_2)\begin{bmatrix}\nabla_x I \\ \nabla_y I\end{bmatrix} \\
&= 2\begin{bmatrix}\{(I_1-I_2)+(e_1-e_2)\}(\nabla_x I + e_x) \\ \{(I_1-I_2)+(e_1-e_2)\}(\nabla_y I + e_y)\end{bmatrix} - 2\begin{bmatrix}(I_1-I_2)\nabla_x I \\ (I_1-I_2)\nabla_y I\end{bmatrix} \\
&= 2\begin{bmatrix}(I_1-I_2)e_x + (e_1-e_2)\nabla_x I + (e_1-e_2)e_x \\ (I_1-I_2)e_y + (e_1-e_2)\nabla_y I + (e_1-e_2)e_y\end{bmatrix} \\
&= 2\begin{bmatrix}a \\ b\end{bmatrix}.
\end{aligned} \tag{A-6}$$

Σ_{gn}^2 can now be evaluated in terms of a and b as

$$\Sigma_{gn}^2 = E\{\mathrm{N}(i)\mathrm{N}^T(i)\} = 4E\left\{\begin{bmatrix}a^2 & ab \\ ab & b^2\end{bmatrix}\right\} = 4\begin{bmatrix}E\{a^2\} & E\{ab\} \\ E\{ab\} & E\{b^2\}\end{bmatrix}. \tag{A-7}$$

$E\{a^2\}$ can be evaluated as follows:

$$E\{a^2\} = E\{(I_1 e_x - I_2 e_x + e_1\nabla_x I - e_2\nabla_x I + e_1 e_x - e_2 e_x)^2\}. \tag{A-8}$$

$E\{a^2\}$ can be expanded as:

$$
\begin{aligned}
E\{&I_1 e_x I_1 e_x - I_1 e_x I_2 e_x + I_1 e_x e_1 \nabla_x I - I_1 e_x e_2 \nabla_x I + I_1 e_x e_1 e_x - I_1 e_x e_2 e_x \\
&- I_2 e_x I_1 e_x + I_2 e_x I_2 e_x - I_2 e_x e_1 \nabla_x I + I_2 e_x e_2 \nabla_x I - I_2 e_x e_1 e_x + I_2 e_x e_2 e_x \\
&+ e_1 \nabla_x I I_1 e_x - e_1 \nabla_x I I_2 e_x + e_1 \nabla_x I e_1 \nabla_x I - e_1 \nabla_x I e_2 \nabla_x I + e_1 \nabla_x I e_1 e_x - e_1 \nabla_x I e_2 e_x \\
&- e_2 \nabla_x I I_1 e_x + e_2 \nabla_x I I_2 e_x - e_2 \nabla_x I e_1 \nabla_x I + e_2 \nabla_x I e_2 \nabla_x I - e_2 \nabla_x I e_1 e_x + e_2 \nabla_x I e_2 e_x \\
&+ e_1 e_x I_1 e_x - e_1 e_x I_2 e_x + e_1 e_x e_1 \nabla_x I - e_1 e_x e_2 \nabla_x I + e_1 e_x e_1 e_x - e_1 e_x e_2 e_x \\
&- e_2 e_x I_1 e_x + e_2 e_x I_2 e_x - e_2 e_x e_1 \nabla_x I + e_2 e_x e_2 \nabla_x I - e_2 e_x e_1 e_x + e_2 e_x e_2 e_x \}.
\end{aligned}
$$

All thirty-six terms have four factors, at least one of which is an error factor. Let

$$I_3 = I(\mathbf{z}_a - \hat{\mathbf{d}}^i - 1, t), \tag{A-9}$$

$$I_4 = I(\mathbf{z}_a - \hat{\mathbf{d}}^i + 1, t - 1), \tag{A-10}$$

$$I_5 = I(\mathbf{z}_a - \hat{\mathbf{d}}^i - N, t - 1), \quad \text{and} \tag{A-11}$$

$$I_6 = I(\mathbf{z}_a - \hat{\mathbf{d}}^i + N, t - 1). \tag{A-12}$$

where N is the number of samples in one row.

Given that the gradient is estimated by a bilinear central difference equation, e_x can be evaluated as:

$$
\begin{aligned}
e_x &= \tfrac{1}{2}[(I_4 + e_4) - (I_3 + e_3)] - \tfrac{1}{2}[I_4 - I_3] \\
&= \tfrac{1}{2} e_4 - \tfrac{1}{2} e_3 \\
&= \tfrac{1}{2}(e_4 - e_3).
\end{aligned} \tag{A-13}
$$

Likewise e_y is:

$$e_y = \tfrac{1}{2}(e_6 - e_5). \tag{A-14}$$

Assume the noise is white and has a gaussian distribution and a zero mean. In other words, assume that each of the error terms is additive, that the true value of each measure is uncorrelated with its error (noise), that the errors are mutually uncorrelated, and that the expected value of each of the error terms is zero. These are standard assumptions. Given the assumptions, all terms which contain only a single occurrence of any of the error factors reduce to zero. Thus $E\{a^2\}$ reduces to:

$$E\{a^2\} = E\{I_1 e_x I_1 e_x - 2I_1 e_x I_2 e_x + I_2 e_x I_2 e_x + e_1 \nabla_x I e_1 \nabla_x I + e_2 \nabla_x I e_2 \nabla_x I + e_1 e_x e_1 e_x + e_2 e_x e_2 e_x\}. \quad \text{(A-15)}$$

Factoring out common factors as prefixes,

$$E\{a^2\} = E\{e_x^2\}[E\{I_1 I_1\} - 2E\{I_1 I_2\} + E\{I_2 I_2\}] + E\{\nabla_x^2 I\}[E\{e_1^2\} + E\{e_2^2\}] + E\{e_x^2\}[E\{e_1^2\} + E\{e_2^2\}]. \quad \text{(A-16)}$$

The first term in (A-16) is zero since

$$E\{I_1 I_1\} = E\{I_2 I_2\} = E\{I_1 I_2\}. \quad \text{(A-17)}$$

Note that I() is not a random field. Since $E\{e_1^2\} = E\{e_2^2\}$, the last two terms in (a16) can be rewritten as:

$$2E\{e_1^2\}[E\{\nabla_x^2 I\} + E\{e_x^2\}]. \quad \text{(A-18)}$$

$E\{e_1^2\}$ is simply the noise variance of the image. Therefore $E\{a^2\}$ can be written as

$$E\{a^2\} = 2\sigma_n^2[E\{\nabla_x^2 I\} + E\{e_x^2\}]. \quad \text{(A-19)}$$

where σ_n^2 is the noise variance of the image.

$E\{e_x^2\}$ can be evaluated as:

$$
\begin{aligned}
E\{e_x^2\} &= E\{(\tfrac{1}{2}e_4 \quad \tfrac{1}{2}e_3)^2\} \\
&= E\{\tfrac{1}{4}e_4e_4 \quad \tfrac{1}{2}e_4e_3 + \tfrac{1}{4}e_3e_3\} \\
&= \tfrac{1}{4}E\{e_4e_4\} - \tfrac{1}{2}E\{e_4e_3\} + \tfrac{1}{4}E\{e_3e_3\} \\
&= \tfrac{1}{4}\sigma_n^2 + 0 + \tfrac{1}{4}\sigma_n^2 \\
&= \tfrac{1}{2}\sigma_n^2.
\end{aligned} \tag{A-20}
$$

Thus $E\{a^2\}$ can be written as:

$$
E\{a^2\} = 2\sigma_n^2 E\{\nabla_x^2 I\} + (\sigma_n^2)^2. \tag{A-21}
$$

$E\{b^2\}$ is evaluated likewise and is:

$$
E\{b^2\} = 2\sigma_n^2 E\{\nabla_y^2 I\} + (\sigma_n^2)^2. \tag{A-22}
$$

$E\{ab\}$ can be evaluated as follows:

$$
\begin{aligned}
E\{a^2\} = E\{&(I_1e_x - I_2e_x + e_1\nabla_x I - e_2\nabla_x I + e_1e_x - e_2e_x)\cdot \\
&(I_1e_y - I_2e_y + e_1\nabla_y I - e_2\nabla_y I + e_1e_y - e_2e_y)\}.
\end{aligned} \tag{A-23}
$$

$E\{ab\}$ can be expanded to:

$$
\begin{aligned}
E\{&I_1e_xI_1e_y - I_1e_xI_2e_y + I_1e_xe_1\nabla_y I - I_1e_xe_2\nabla_y I + I_1e_xe_1e_y - I_1e_xe_2e_y \\
&- I_2e_xI_1e_y + I_2e_xI_2e_y - I_2e_xe_1\nabla_y I + I_2e_xe_2\nabla_y I - I_2e_xe_1e_y + I_2e_xe_2e_y \\
&+ e_1\nabla_x II_1e_y - e_1\nabla_x II_2e_y + e_1\nabla_x Ie_1\nabla_y I - e_1\nabla_x Ie_2\nabla_y I + e_1\nabla_x Ie_1e_y - e_1\nabla_x Ie_2e_y \\
&- e_2\nabla_x II_1e_y + e_2\nabla_x II_2e_y - e_2\nabla_x Ie_1\nabla_y I + e_2\nabla_x Ie_2\nabla_y I - e_2\nabla_x Ie_1e_y + e_2\nabla_x Ie_2e_y \\
&+ e_1e_xI_1e_y - e_1e_xI_2e_y + e_1e_xe_1\nabla_y I - e_1e_xe_2\nabla_y I + e_1e_xe_1e_y - e_1e_xe_2e_y \\
&- e_2e_xI_1e_y + e_2e_xI_2e_y - e_2e_xe_1\nabla_y I + e_2e_xe_2\nabla_y I - e_2e_xe_1e_y + e_2e_xe_2e_y\}.
\end{aligned}
$$

All thirty-six terms contain a single occurrence of an error factor and therefore evaluate to zero. Thus

$$
E\{ab\} = 0. \tag{A-24}
$$

Therefore Σ_{gn}^2, the gradient noise at convergence, is a diagonal matrix and maybe expressed as:

$$\Sigma_{gn}^{2} = 4\begin{bmatrix} 2\sigma_n^2 E\{\nabla_x^2 I\} + (\sigma_n^2)^2 & 0 \\ 0 & 2\sigma_n^2 E\{\nabla_y^2 I\} + (\sigma_n^2)^2 \end{bmatrix}. \tag{A-25}$$

Appendix B

Proof of Two Lemmas

The proof of the following two lemmas may be found in sighted references.

Lemma 1 [3] - Let $0<\alpha<2/c$ where, $c = \max(\alpha_1^2 , \alpha_2^2 , ..,\alpha_n^2)$ is the largest singular value of the NxN matrix [A]. Then,

$$\alpha \sum_{k=0}^{\infty} ([I]-\alpha[A]^*[A])^k[A]^* \tag{B-1}$$

and

$$\alpha \sum_{k=0}^{\infty} [A]^*([I]-\alpha[A][A]^*)^k \tag{B-2}$$

converges and is equal to $[A]^\dagger$.

Lemma 2 [4]- Let [K] be an NxN semi convergent matrix, then

$$\lim_{n\to\infty} [K]^n = [I]-[E] \tag{B-3}$$

where,

$$[E] = ([I]-[K])([I]-[K])^{\mathbf{d}} \tag{B-4}$$

In the above lemmas $[A]^\dagger$ and $[A]^{\mathbf{d}}$ are the Moore-Penrose and Drazin inverse of the matrix [A].

Replacing (IV-84) into (IV-83),

$$[A]^{\dagger} = \sum_{k=1}^{p} \frac{1}{\lambda_k} \psi_k \psi_k{}^T [B]. \tag{IV-85}$$

The matrices $\psi_k \psi_k{}^T$ are symmetric, positive semi definite and therefore their largest eigenvalue is equal to one. Therefore for any $X \in E^N$,

$$\begin{aligned} ||[A]^{\dagger} X|| &= ||\{ \sum_{k=1}^{p} \frac{1}{\lambda_k} \psi_k \psi_k{}^T [B] \} X|| \\ &\leq \sum_{k=1}^{p} \frac{1}{\lambda_k} ||\psi_k \psi_k{}^T [B] X|| \\ &\leq \sum_{k=1}^{p} \frac{1}{\lambda_k} ||X|| \\ &= ||x|| \sum_{k=1}^{p} \frac{1}{\lambda_k} \end{aligned} \tag{IV-86}$$

Hence,

$$\begin{aligned} ||\Delta F|| &= ||\Delta F_1 + [A]^{\dagger} (\Delta F_0 - \Delta F_1)|| \\ &\leq ||\Delta F_1|| + ||\Delta F_0 - \Delta F_1|| \sum_{k=1}^{p} \frac{1}{\lambda_k}. \end{aligned} \tag{IV-87}$$

The above expression is an upper bound on the quantization noise, demonstrating the asymptotic effect of the quantization noise. It shows that the quantization noise can be excessively amplified if the matrix [A] is ill-conditioned. The above noise amplification property is observed in every solution of system of linear equations obtained by calculation the pseudo-inverse of the coefficient matrix. Although the projection method described here does not find the pseudo inverse of the actual matrix of coefficients, it does find the pseudo inverse of [A].

In what follows it is shown that the quantization noise is accumulated as the iteration progresses and therefore can be controlled by terminating the process before the amplified noise destroys the constructive trend of the iteration.

From equation (IV-80), it is seen that the quantization noise after n iteration is given by,

$$\Delta X_n = \{\sum_{i=0}^{n-1} [K]^i\}([T]\Delta F_0 + \Delta F_1) \tag{IV-88}$$

where, signal independent noise is assumed as before. It is seen that the noise effect at n'th iteration is a function of $\{\sum_{i=o}^{n-1} [K]^i\}$ and $[T]\Delta F_0 + \Delta F_1$. As expected only the out of range (in null space of [D]) portion of F_0 contribute to the final quantization. This is because the samples in range of [D] are given by $F_1 + \Delta F_1$.

The matrix [K] given by equation (IV-62) is not symmetric. However, it is positive semi-definite and its eigenvalues are in [0,1]. This matrix can be transformed to a triangular matrix by means of an orthogonal transformation, i.e.

$$[P][K][P]^T = [U] \tag{IV-89}$$

where [U] is a triangular matrix whose diagonal elements are the eigenvalues of [K]. Therefore,

$$\sum_{i=0}^{n-1} [K]^i = [P]\{\sum_{i=0}^{n-1} [U]^i\}[P]^T \tag{IV-90}$$

It is easily shown that the diagonal elements of $[U]^i$ are equivalent to the diagonal elements of [U] (and therefore eigenvalues of [K]) raised to the ith power. Therefore, the