Image Sequence Compression Using A Motion-compensated Technique

by

Robert James Moorhead II

Center for Communications and Signal Processing

Department of Electrical and Computer Engineering
North Carolina State University

July 1985

CCSP-TR-85/12

ABSTRACT

MOORHEAD II, ROBERT JAMES. Image Sequence Compression Using a Motion-Compensated Technique. (Under the direction of S. A. Rajala.)

A pel-recursive motion-compensated algorithm is developed and analyzed. First, the criteria for convergence and the converge rate of the motion estimate in a pel-recursive algorithm are derived. Secondly, a new motion prediction scheme called "projection-along-the-motion-trajectory" (PAMT) is developed and analytically shown to be an improvement over the previous motion prediction schemes. Simulations run on synthetic and actual image sequences to verify the analytical results indicate three improvements of the compression algorithm. First, implementing the analytical model as opposed to the generally used heuristic technique yields a small, but significant, decrease in the information rate and the computational requirements. Secondly, the PAMT motion prediction scheme is shown to offer the potential for increased information compression. Thirdly, zero-entropy encoding is shown to reduce the information transmission rate on the order of 20% and to reduce the mean square error in the reconstructed images on the order of 60% when compared to first-order entropy encoding. Although the compression ratio is only on the order of 10:1, the reconstructed image sequences in all simulations are of very high quality.

BIOGRAPHY

Robert James Moorhead II was born in Atlanta, Georgia on February 12, 1958. He grew up in Yazoo City and Clarksdale, Mississippi. He received the B.S.E.E. summa cum laude and with research honors from Geneva College in 1980 and the M.S. in Electrical Engineering from North Carolina State University in 1982.

Since 1980 the author has been a teaching and research assistant at North Carolina State University, Raleigh, North Carolina while working toward the Ph.D. degree in Electrical and Computer Engineering. During the summer of 1981 he worked at the IBM facility in Research Triangle Park, North Carolina in a components-testing laboratory. He has been the recipient of a Dean's Fellowship (1980-1981), a Microelectronics Center of North Carolina Fellowship (1981-1982), and an IBM VSLI Fellowship (1982-1983) during his graduate studies.

ACKNOWLEDGEMENTS

The author acknowledges the support of many people during his graduate studies: 1) all the members of his Ph.D. advisory committee and the chairman of his master's advisory committee, Dr. J. W. Gault, for their intellectual support, 2) Roger Greenhaugh, John Hong, and the other members of the Advanced Technologies Group at the IBM facility in RTP, NC, for their financial and technical support, and 3) Picture Element Limited, for the development and support of the Video Sequence Processor, a machine which proved extremely valuable for the research reported herein.

Some people are due a special word of thanks: 1) my family, for never ceasing in their financial and moral support, 2) Dr. Tom Alexander, for always being willing to advise and for proding me to my goal, and 3) my wife, Jane, for loving me enough to leave me alone when it became time to grind the chapters out.

The author acknowledges that in the end it is the Lord who has provided the gifts which have made the attainment of this doctorate possible. To Him be the glory.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER ONE

INTRODUCTION

Image sequence compression attempts to minimize the amount of information that must be transmitted or stored to obtain a certain level of picture quality. In a single image or an image sequence, the amount of information is typically compressed in two ways. First, the redundancy in the original signal is reduced. Secondly, some of the unessential information in the original signal is deleted. This deletion of information is not required, but frequently allows for a much greater compression. Thus the image compression techniques are not necessarily lossless coding and a tradeoff exists between picture quality and information bandwidth.

Image sequence compression is an outgrowth of single-frame image compression. Many of the techniques developed for compressing the information in a single-frame image have been extended to image sequence compression. Since computation costs have decreased much more rapidly than communication costs, image sequence compression has become a cost effective technique. However, much work remains to be done in developing high-compression techniques which can be implemented in real-time.

This dissertation seeks to solve some of the existing problems of real-time image-sequence compression in two ways. First, the convergence requirements and

the convergence rates of one class of image-sequence compression algorithms will be thoroughly analyzed. Secondly, a new motion prediction scheme will be presented which will increase the validity of the assumptions made in analyzing the convergence properties and which will decrease the total prediction error. The results will indicate that the convergence properties of the algorithm are enhanced and that the potential exists for greater information compression with the new motion prediction scheme.

Chapter 2 reviews the fundamentals of image compression. In Chapter 3 a method of image sequence compression which supersedes both spatial and transform domain techniques -- namely motion-compensation techniques -- is presented and discussed. The basic algorithm upon which a new motion prediction technique is predicated is developed in Chapter 4. In Chapter 5 convergence analysis of the pel-recursive gradient technique is performed. Chapter 6 presents the new motion prediction technique. The advantages and disadvantages of the technique are discussed conceptually and an analytical measure of improvement is formulated. Chapter 7 contains simulations on synthetic and real image sequences verifying the increased compression resulting from the new motion prediction technique. Conclusions, a summary, and a potential architecture for real-time implementation are presented in Chapter 8.

CHAPTER TWO

IMAGE COMPRESSION: A REVIEW

2.1 Introduction

A review of the literature on image compression will be helpful to fully understand the problems to be solved and the applicable solution techniques. Jain [1] summarized the mathematical models used in image processing algorithms. He discussed causal prediction, semicausal prediction, and noncausal prediction models, emphasizing minimum variance prediction. This paper provides a thorough presentation of the mathematical models used in image processing, especially those suited to image compression analysis.

A number of summaries and surveys of image compression techniques have been compiled in the past five years, the most referenced ones are by Nagel [2], Dubois et al. [3], Jain [4], and Netravali and Limb [5]. The various approaches to image compression and image sequence compression can be partitioned into two groups: spatial domain techniques and transform domain techniques. The foundations of these techniques will be discussed in the remainder of this chapter.

2.2 Spatial Domain Techniques

Spatial domain techniques are those techniques which operate directly on the intensity values. These were the first approaches to image compression, and many of the algorithms were subsequently prototyped in hardware. Although spatial domain codecs (coder/decoder pairs) are more easily implemented than transform domain codecs, they typically have a lower bandwidth compression. Haskell [6] presents a good survey of the spatial domain techniques.

2.2.1 Prediction Schemes

The first spatial domain codecs exploited the intrafield/intraframe redundancy in television images, i.e. the spatial redundancy within a single frame. This redundancy can be used to predict the intensity of a pel based on the intensity of previously transmitted pels. By sending only the prediction error, the bit rate is generally reduced. The actual value of the pel can then be reconstructed at the receiver, since the receiver has already received the pels upon which the prediction is based. O'Neal [7] was one of the first to investigate this area. He implemented several intrafield DPCM coders and found that little was gained by basing the prediction on more than the previous horizontal pel and one adjacent pel from the previous line.

The next major advancement in the research of spatial domain coders was an interfield/interframe coder, which exploits the temporal redundancy present in an

image sequence. In this technique the prediction is based on pel values in the same neighborhood in the previous field or frame.

The information compressibility of the interfield/interframe predictor was increased further by thresholding the intensity prediction error and transmitting only those errors which were greater than the threshold value. Even though this meant addressing information had to be transmitted, the bit-rate was reduced. This is the concept behind conditional replenishment, which was first proposed by Mounts [8]. Candy et al. [9] investigated the area of addressing and found that for head-and-shoulder scenes, a cluster addressing approach worked well.

It has been shown that the human visual system is less sensitive to spatial resolution in areas of high motion. In fact the greater the motion velocity, the less spatial resolution required. It should be noted that images may already be somewhat blurred in high motion areas due to the integrating effect of the camera capture mechanism. Pease and Limb [10] studied ways to exploit this fact to reduce the bit rate in moving areas. They found that the data rate can be halved by horizontal subsampling (transmitting information about only every other pel) without much noticeable picture degradation. The receiver can reconstruct the skipped pels by interpolation and it was found that a simple linear interpolation usually sufficed. They [11] went on to develop a codec which utilized spatial subsampling and interpolation in moving areas to reduce the required transmission bandwidth.

Haskell [12] continued the work on predictors in an attempt to gain a unified approach and ascertain the tradeoffs. He investigated linear predictors with as many as 22 coefficients from the same field and the previous two fields. Note that these sample points are also from the same frame and the previous frame. He determined that a combination intrafield/interfield predictor performed the best overall for the sequences with which he experimented. Advancing the work of Candy et al. [9], Haskell [13] developed an addressing scheme which addressed a cluster position based on the address of previous clusters. This lowered the bit rate and showed that the cluster positions were highly correlated both temporally and spatially.

A group of researchers at Bell Labs led by Haskell investigated and prototyped a multimode codec that transmitted information at 1.5 Mbits/sec [14]. The codec operated in different modes to prevent transmission buffer overflow while still producing the best picture quality possible. The mode in which to operate was based on the fullness of the transmission buffer and the previous mode. By conditioning the next mode on the previous mode, they were able to put a hysteresis in the mode switching. This prevented rapid mode switching which appeared to the observer as a flashing condition. The mode switching algorithm reduced the data rate by reducing the spatial, temporal, and amplitude resolution. These reductions were obtained by subsampling, field repeating, and adaptive quantization respectively. Isolated values over the threshold were assumed to be due to noise and

were rejected; isolated values below the threshold were bridged. Both increased the addressing efficiency. Dubois et al. [3] summarized this work and analyzed the multimode approach from the point of view of a state-transition diagram.

2.2.2 Quantizers

Until 1977 most quantizers were based on the criteria developed by Max [15]. Netravali and Prasada [16] investigated quantizers which weighted quantization errors based on "visibility" as opposed to prediction error probability. They developed a visibility function based on the probability and the perceptibility of an error.

Sharma and Netravali [17] continued this work on quantizers and determined that the statistically based mean-square error (MSE) quantizer [15] poorly modeled the human psycho-visual system response. They found that it was more important to decrease the less frequent, but larger quantizing noise produced by a MSE-based quantizer. A large quantizing error, though it occurs less frequently, is much more visually objectionable than the more frequently occurring, but smaller quantization noise. They experimented with various quantizers and found that a 27-level quantizer based on a mean-square subjective error [18] was sufficient to reconstruct an image with no detectable degradations. However, using the Max quantizer [15], 35 levels were required for the same quality reconstruction. They estimated the reduced number of levels would save about 1 bit/pel.

2.2.3 *Video Compression Techniques*

Iinuma et al. [19] reported the first attempt at processing and transmitting a 4 MHz NTSC color television signal with an interframe coder. The signal was separated into components, the components were processed separately, and the information time-division multiplexed onto a 6.3 Mbit/sec line. They reported that acceptable picture quality was obtained and that the chrominance information occupied less than 10% of the bandwidth. Yasuda et al. [20] reported on some field trials of a 1.5 Mbit/sec interframe codec which was formulated in an attempt to determine the technical feasibility of the conditional replenishment algorithm, i.e. how hard is it to implement in real-time? Building on their previous work [20] and that of Iinuma et al. [19], Yasuda et al. [21] designed a commercial codec to transmit NTSC signals over a 6.3 Mbit/sec line. The codec used combinational differencing to exploit both the temporal and spatial redundancy, variable-length coding to reduce the transmission bit-rate further, and multimodes to prevent buffer overflow while giving the best picture quality obtainable.

The next major codec by NEC required only a 1.5 Mbit/sec channel [22]. Like its predecessor [21], it input an NTSC color signal, separated the luminance and chrominance components, processed them separately, and multiplexed the two signals onto a channel using time division multiplexing (TDM). The prediction method was a combinational differencing approach similar to the one described in [21]. It involved taking intraframe differences of interframe differences. One of

the major contributions of this work was the basis of the mode-selection algorithm, which switched between eight ordered modes. The mode in which to process a frame was determined by: 1) the buffer contents, 2) the mode in which the last frame was processed, and 3) an estimate of the amount of information the frame will generate. The first condition was the most classical one. The second condition is the hysteretic condition. The third condition requires the frame to be "preprocessed," but the added delay was acclaimed to make a significant improvement in the sequence quality. The codec was able to limit the degradations to the areas in which they were least detectable. The similarities between and advancements of the codec Haskell et al. prototyped [14] should be noted. The codec of Kuroda et al. had a 15 level nonlinear quantizer and the data was processed in 2 line by 8 pel blocks. Another contribution of this work was the use of fixed size word lengths in the transmission buffer. A coder is placed between the buffer and line to convert to variable word lengths. This is advantageous since fixed word-length memories are faster. Kuroda et al. [22] and Yasuda et al. [21] together provide a fairly good example of production spatial domain codecs.

Nicole et al. [23] reported on a 2 Mbit/sec 625 line monochrome video teleconferencing codec developed cooperatively by researchers in Italy, the UK, West Germany, and France. They addressed prediction, coding, subsampling, pre- and post-filters, and error correction. They concluded among other things that field subsampling was better than line subsampling and that variable length coding was

better than fixed length coding, even using a comma code for the variable length code.

Recently, Netravali and Bowen [24] have published some work on improved reconstruction of DPCM-coded pictures. The improvement requires no additional communication bandwidth, only more computation at the receiver. It uses the structure of the neighborhood to reconstruct the picture, instead of simply transmitting one representative level for each prediction error. It is acclaimed to improve the quality of reconstruction (as measured by mean-absolute reconstruction error) by up to 20% for the sequences studied. Alternately the communication bandwidth can be reduced and the picture quality be maintained by using a quantizer with fewer levels.

In summary there are four important parameters in designing a predictive spatial-domain coder:

1) the number of previous intensity values used,
2) the location of the previous pels which are used,
3) the coefficients on the previous intensity values, and
4) the quantizer in the DPCM loop.

Most interframe coders use only one previous pel, the one in the same location in the previous frame, and predict the pel under consideration to have the same intensity as that pel [21-23]. A consensus on the quantizer's characteristics has not emerged, although the work of Netravali et al [16-18] has influenced people away from the Max quantizer [15].

2.3 Transform Domain Techniques

In this section those video teleconferencing algorithms which utilize linear transformations such as the Fourier [25], Hadamard [26], slant [27], or cosine [28] transform will be discussed. These transforms differ in the amount of data compression and computational complexity. However, they all pack most of the signal energy in a few coefficients, which allows more data compression with less picture quality degradation than with the spatial domain coders. However, as great as the data compression of these algorithms is, equally as great is the difficulty of implementing them in real-time. This is at least implied by the existing equipment. In early 1982, the following three video teleconferencing systems were available under the Picturephone Meeting Service from AT&T [29]: an intraframe coder, an interframe coder, and a motion-compensated coder. No mention is made of a transform system.

Once again the techniques and advancements are presented in the order in which they were developed. Much similarity with the order of development of the spatial domain techniques will be evident. Transform techniques have been applied in three different ways. The first procedures involved using two-dimensional transforms in intraframe techniques. Blocks of pels were linearly transformed and the coefficients transmitted. At the receiver the image was reconstructed using an inverse transform. This research transpired before 1973. References [25,26,30-35] investigated this type of technique, which exploits only the spa-

tial correlation between neighboring pels in a frame. The next major method made use of the temporal correlation of the pels in a sequence by extending the two-dimensional transforms to three dimensions. References [36,37] present some of the three-dimensional techniques developed in the mid 1970's. Note that this approach was short lived and not well developed. Since the mid 1970's the predominate transform techniques have been the hybrid algorithms in which the transform coefficients are predicted from frame-to-frame or from field-to-field, much like conditional replenishment in the spatial domain codecs. References [37-42] presented some of these techniques. Throughout this whole period the addressing schemes, the quantizing schemes, and the transform algorithms themselves were constantly being refined and developed. Tescher [43] provides a good summary of the pure transform techniques developed before 1979; Roese [44] provides a summary of the hybrid algorithms.

The first work in transform coding of images was reported by Andrews and Pratt [25] in 1968. They took the Fourier transform of the whole image. Later, Pratt et al. [26] showed the advantages of using the Hadamard transform to code the information in a picture. They noted that one advantage of transmitting transform coefficients is a higher tolerance to channel errors since an error in a transform coefficient is averaged out over an area instead of affecting only a single pel. Another advantage of transform coding is that most of the energy in an NxN block is packed in the first M coefficients, leading to the possibility of bandwidth

reduction with less visual degradation. As for the Hadamard transform in particular, it is shown to have an order of magnitude speedup over the more familiar Fourier transform, since it involves only additions and subtractions. Apparently Pratt et al. transformed the whole 256x256 image as one block.

Habibi [30] and Habibi and Wintz [31] performed some of the first investigations of linear transforms and block quantization in intraframe coders in general. They noted that different coefficients had different variances and that the lower frequency coefficients had the greater variances. They determined that the bit rate can be reduced without degrading the image quality by using fewer bits for those coefficients which have a smaller variance. They made the number of bits assigned a coefficient proportional to the logarithm of its variance.

Continuing this work, Wintz [32] was able to show some interesting results about optimum block size. Wintz was one of the first to see an advantage to breaking an image into subblocks and taking the transform of each subblock. Although the total energy compaction for the whole image is reduced, the smaller block size reduces the computational complexity. The smaller blocks also increase the probability of the region being homogeneous and of the transform coefficients changing little between spatially and/or temporally adjacent blocks. This fact can be used to reduce the bit rate by predicting coefficients from block-to-block and/or frame-to-frame. Although the smaller blocks do not provide the maximum decorrelation, Wintz [32] demonstrated that using subblocks larger than 8x8 or

16x16 did not increase the data compression significantly.

Landau and Slepian [33] investigated the quantization of the coefficients resulting from applying the Hadamard transform to a 4x4 block. Picture quality was apparently acceptable in all but detailed areas with a two bits per pel (bpp) quantizer.

Anderson and Huang [34] studied the Fourier transform of 16x16 blocks. Their adaptive technique measured the spatial activity in a frame and from that measurement determined how many coefficients to transmit and how to quantize the coefficients. They claimed good picture quality was obtained with an average of 1.25 bpp. Note that the decrease in bit rate from the previous algorithm necessitated an increase in computational complexity.

Habibi [35] investigated a hybrid intraframe coder which used the transform coefficients of a block to predict the coefficients in the next horizontally adjacent block. Thus this coder used two-dimensional transforms and a DPCM prediction method. This method was later shown to be inferior to interframe approaches.

Knauer [36] studied three-dimensional adaptive transform coding. The coder detected temporal activity by evaluating particular coefficients. In the presence of large temporal activity, the spatial coefficients were quantized more finely and the temporal coefficients were quantized more coarsely. Roese et al. [37] investigated the tradeoffs of using a straight three-dimensional cosine transform versus using a hybrid approach where the coefficients in each block were predicted from frame-

to-frame. The three-dimensional cosine transform is a direct extension of the two-dimensional cosine transform. Although a three-dimensional transform is algorithmically simple, it is architecturally difficult. The amount of memory and processing power required is exorbitant. Since a number of frames must be captured before any information about the first one in the group can be sent, there is a large delay between a frame being captured at the transmitter and seen at the receiver. Roese et al. [37] developed an interframe hybrid approach that performed as well as a three-dimensional coder with several frames of memory. They used an adaptive coding scheme in which the variance for each coefficient was estimated by both the receiver and the transmitter to determine the number of bits for each coefficient.

Mounts et al. [39] used the Hadamard transform of 2x2 blocks to study quantization based on psycho-visual fidelity criteria. As was found for the spatial domain quantizers [17], basing the quantization and coding on subjective criteria significantly increased the efficiency. They reported that on the order of 2.0 bpp were required for good quality reconstruction. Since this was a non-adaptive algorithm, the hardware requirements were reduced (only one framestore of memory was required and the processing was minimal).

Chen and Smith [40] did some investigation of adaptive quantization strategies. They looked at both monochrome and color images. As implied in the discussion of Roese et al. [37], the findings of [30,31] for quantization of coefficients

carries over to the quantization of coefficient differences. Namely those coefficients differences with the lesser variance can be quantized with fewer levels than those with more variance. Fewer levels implies fewer bits in the code words. Chen and Smith [40] proposed a two pass scheme in which the variances are collected on the first pass and the information is actually encoded in the second pass.

In 1981 Chen [41] described a hybrid interframe coder which used the fast two-dimensional cosine transform that Chen et al [28] had developed in 1977. According to Szarek [38], this is the only real-time transform-based coder which has been presented in the literature even though Knauer [36] indicated the coder he developed operated in real-time.

Chen and de Figueiredo [42] presented a summary of the development of transform coding. They showed the similarity of taking a linear transform and convolving the image with a window. They showed that a transform coder's performance can be improved using an overlap-and-save (really overlap-and-delete) process. They discussed the effect of zonal thresholding on the results. They concluded that transforms which are suitable for coding can be designed from either of two viewpoints: statistical or approximation theory.

Szarek [38] simulated a hybrid interframe coder which reconstructed the uncovered background rapidly and precisely by using two 8 kbit memories to know which blocks had motion in the previous two frames. Once no motion was detected in a block which used to have motion, the coefficients were quantized

more precisely. Motion detection was done in the spatial domain. Many sequences with 10% motion were shown to produce less than 1.5 Mbits/sec using this algorithm.

In summary there are five important parameters in designing a transform coder:

1) transform type,
2) size and shape of transform blocks,
3) coefficient selection,
4) coefficient quantization, and
5) coefficient prediction

Jain [45] provided a comparison of transform type tradeoffs. In general, the more energy a transform packs into the first M coefficients, the more difficult its implementation in real-time. The two most often used transforms today are the cosine and Hadamard, with blocks usually containing 4 to 64 pels. A four-line-by-eight-pel block (32 pels) seems to produce the best tradeoff between data compression and adaptivity.

As for coefficient selection, two techniques have been tried: 1) zonal sampling, in which the last M coefficients are ignored, and 2) threshold sampling, in which coefficient differences below a certain threshold are ignored. Though threshold sampling is more accurate, zonal is more easily implemented and usually sufficiently accurate. Pratt [46] tabulates the error resulting from zonal sampling for various block sizes and transform types.

As with spatial domain quantizers, the Max quantizer [15] has given way to visually-based quantizers. Unlike the spatial domain quantizers, however, the coefficient difference values are usually quantized in a coefficient-dependent manner, since lower frequency coefficients typically have more variance.

Interframe coefficient prediction is now more frequently used than intraframe prediction. Both of these types of prediction have been superceded by motion-compensated prediction for sequences containing large amounts of interframe translation motion.

In concluding this section, it should be noted that transform coding does have advantages over spatial domain coding. The coding degradations are less objectionable visually. Transform coding is less sensitive to picture variations than spatial domain techniques. Two other advantages are the higher probability that channel noise will affect only a small area (one block) and that any degradations due to channel noise will be averaged out over that small area.

CHAPTER THREE

MOTION-COMPENSATED IMAGE SEQUENCE COMPRESSION

Motion-compensation techniques predict the frame-to-frame (or field-to-field) motion of a object point and then access the intensity values from the previous frame (or field). The assumption is that predicting the motion and accessing the intensity values from the previous frame (or field) results in a better prediction of the intensity values than trying to predict the intensity values directly. Previous work [3,47-55] has shown that motion-estimation techniques do improve the predictions of the intensity values in the images. Although some attempts have been made to apply motion-compensation techniques to transform-domain codecs [56,57], the results were not as encouraging.

There have been basically two approaches to motion estimation: block-matching and pel-recursion [51,54,55]. In block-matching a block of intensity values in a frame is compared with blocks of intensity values in the previous frame until a best-match is determined. From this an interframe displacement vector (how much the block has moved between frames) for the whole block can be estimated for the frame being transmitted. Poor estimates result if all sample points in the block do not move the same way. Using the pel-recursive approach a displacement is determined for each pel value. This technique allows for a more exact estimation of the intensity value and has the ability to handle scale changes

(zooming, dilatation, movement perpendicular to the image plane).

In both block-matching and pel-recursion the prediction can be backward or forward, i.e., the displacement can be determined from previously transmitted information only (backward) or from past values and the current value (forward). Forward prediction requires explicit transmission of information about the displacement value; backward does not. The advantage of the forward technique is that the presumably better estimate of the displacement vector reduces the error in the intensity prediction. The majority of the previous approaches have used backward prediction, implying backward prediction leads to: 1) reduced bit rates, 2) lower computational requirements, or 3) faster prediction/estimation techniques.

The pioneering work in detecting motion in interframe coders was done by Limb and Murphy [47,48]. They first devised a way to estimate speed, i.e. the magnitude, but not the direction [47]. They estimated the speed by dividing the sum of the frame differences in a moving area by the sum of the element differences in that moving area. They assumed that a speed of half a pel per frame was relatively slow, while a speed of four pels per frame was seldom exceeded. Their results were obtained using a fixed camera and a moving object, but they claimed the technique could also be applied to a panning camera and a moving object. Later they extended their technique to estimate velocity, i.e. determine the direction of motion [48]. They claimed they could estimate the speed to an accuracy of a quarter pel for the range of 0 to 4 pels per frame with no more than eight addi-

tions per pel and some processing at the frame rate. Their technique was able to estimate the motion velocity rather well for a variety of sequences when the motion was below 2.5 pels per frame, but the estimate began to diverge from the actual values above 2.5 pels per frame. This algorithm was also shown to be robust to the motion direction, i.e. it could determine the vertical and horizontal components regardless of the motion direction. It was determined that the background had a great influence on the estimate, but that the effect could be reduced by subtracting the uncovered background and edge noise.

Cafforio and Rocca [49,50] also did some pioneering work in the area of motion-compensated techniques. Their work was more theoretical, while the work of Limb and Murphy [47,48] was more experimental. One of the major contributions of Cafforio and Rocca was the conclusion that if a choice has to be made between imperfect segmenters, choose the one which labels part of the moving area as stationary as opposed to one which labels part of the background as moving. Their technique used dynamic programming principles (Viterbi algorithm) and operated similar to the Hough transform. Since translational motion is two-dimensional, at any one point the motion cannot be uniquely determined; it can only be determined to within a linear relation between orthogonal components. Their segmentation approach was based on a two-state Markov process. Without blurring the image, their results were slightly worse than those of Limb and Murphy [47,48], but by blurring the image their results varied linearly with the exact

values. Later they altered their technique to account for noise and experimently demonstrated that noise biased the displacement estimate [50]. Low-pass filtering was used to control this bias.

The techniques proposed by Limb and Murphy [47,48] and Cafforio and Rocca [49,50] both required an estimate of the motion velocity to be sent. Netravali and Robbins [51,52] developed a pel-recursive spatio-temporal gradient technique in which the displacement of a pel was predicted from previously transmitted information. Thus since both transmitter and receiver could predict the motion vector, it did not have to be sent. If an error-correction needed to be sent for the predicted brightness, then only an address and the difference value had to be transmitted. Netravali and Robbins experimented with different prediction configurations and determined that using three adjacent, previously transmitted pels gave the best results [52]. They investigated addressing and found that a run-length addressing scheme gave the best results, but produced only about a 10% improvement over absolute address for their sequences. They used a 35 level symmetric quantizer, but claimed that the coder performance was only slightly affected by the quantizer. Previous field intensities were used for interpolation. They found that a rather simple interpolator sufficed. Their algorithm was able to reduce the data transmission rate by up to 50%. In most cases techniques such as gap bridging, isolated point rejection, and intraframe differencing of prediction errors improved the motion-compensated algorithm to about the same extent as

they had improved the straight conditional replenishment algorithms.

Netravali and Robbins [53] later reported on the results of further attempts to extend and improve the algorithm. None of these decreased the bit rate by more than 10% and most required a significant increase in computational complexity.

Stuller et al. [58] developed an algorithm in which the model was capable of predicting the brightness changes due to variances in illumination and reflectance, i.e. it modeled shadows. They called this gain compensation. Separately displacement compensation and gain compensation reduced the bit rate; together they reduced it even more, especially for the cases in which separately they produced minimal reduction. This was the best possible result. They pointed out, however, that a significant portion of the bit rate reduction was probably due to the use of a different segmenter. In the straight conditional-replenishment algorithm used as the baseline, the addressing bits were reduced by relabeling isolated predictable pels as unpredictable; in the compensated conditional replenishment algorithms, isolated unpredictable pels were relabeled as predictable. Their preliminary analysis indicated the hardware to implement a gain-compensated algorithm would be more complex than that to implement a conditional-replenishment algorithm, but not as complex as that to implement a motion-compensated algorithm. It should be noted that gain compensation has some inherent motion tracking ability.

Next, some theoretical work was done on the implications and constraints of the assumptions which were being made in the motion-compensated algorithms. Snyder et al. [59] investigated the assumption that the frame differences can be expanded as a Taylor series. This assumption is valid only if the function is analytic in the region about which it is expanded [60]. Snyder et al. were one on the first groups to really question retaining only the first order term in the series expansion. They used a gradient approach to determine both velocity and segmentation. They showed that most algorithms determine the motion only from the edges. Two algorithms were discussed and the fundamental weakness of both noted. The major contribution of the paper was the conclusion that blurring an image makes the continuity assumption closer to valid, albeit introducing increased uncertainity in the object location. This explains to some extent the results of Cafforio and Rocca [49].

Horn and Schunck [61] pointed out that estimating two-dimensional motion required two independent measurements for an exact solution to be obtainable. Their search for a second constraint led them to model the velocity as if it varied smoothly over the whole image. Their solution required iteration as opposed to the recursion of Netravali and Robbins [51], with the number of required iterations being equal to the width of the largest moving region. Later [62] they showed that their scheme was valid in the presence of discontinuous image irradiance if all perceived image irradiance changes were due to motion and only a finite number of

lines of discontinuity existed. They also laid the basis for a scheme which simultaneously computed the optical flow and segmented the image into moving and stationary regions.

By building on the work of Horn and Schunck, Nagel [63] developed a motion estimation technique which can be seen [64] to do a good job of predicting the motion in a scene containing translational motion. The problem with using this second constraint [61-64] is that the resulting system of equations is very computationally expensive. No attempt has been made to apply this technique to information-bandwidth compression.

Thompson and Barnard [65] reported on ways of estimating and interpreting motion. They discussed spatio-temporal gradient techniques, feature point matching, and three-dimensional inference. With regards to temporal-spatial gradient techniques, they compared a pseudoinverse approach and a clustering approach, the clustering approach being like the approach of Cafforio and Rocca [49]. They noted that the problem with both approaches is their requirement that the image gradient vary slowly, which is not always the case. Feature point matching (pattern matching) was determined to be too computationally expensive. In discussing three-dimensional inference, they reported no new results, only reviewed the existing theories.

Robbins and Netravali [66] investigated spatial subsampling in motion-compensated coders. Spatial subsampling is a common way of preventing buffer

overflow in the presence of high or complex motion. Although motion estimation was degraded somewhat, the bit rate was reduced by 50%, the same percentage as in conditional-replenishment coders. They were able to confine the blurring inherent in subsampling to the moving areas by an adaptive interpolation technique. They claimed that although the reduction factor was the same, the motion-compensated algorithm produced better quality reconstruction than the conditional replenishment algorithm.

Prabhu and Netravali [67] developed a motion-compensated algorithm to transmit component color sequences. First they investigated predicting each component separately. They evaluated three predictor schemes: 1) use only the previous frame, 2) switch the predictor between previous frame and displaced previous frame, and 3) switch the predictor between previous frame, displaced previous frame, and an intraframe predictor. They ultimately concluded that one predictor (the third one) could be used to predict both the luminance and the chrominance components. The luminance information was used to switch the predictor.

In 1982, Prabhu and Netravali [68] presented an algorithm to compress and transmit composite (as opposed to component) color sequences. The experiments were similar to those performed in their component color work. They investigated four methods of displacement estimation and four prediction modes. Their results were also similar to those found for the composite color coding. They used the same three sequences for their simulations, although the data was not separated

into components.

Ishiguro and Iinuma [54] gave a brief overview of the existing motion-compensated bandwidth-compression techniques. They divided the techniques into pel-recursive and pattern-matching. Given the pattern matching approach, the choice of backward or forward detection must be made. Backward detection implies the transmitter and receiver both determine the motion prediction from common information (previously transmitted data). The transmitter then sends a block of error-correcting values if necessary. In forward detection, the block about to be transmitted is translated and a motion vector determined. This motion vector must be sent as well as the block of error-correcting values. The assumption in forward detection is that the error values are smaller and thus require less bandwidth to be transmitted, leaving room for the motion vector. Ishiguro and Iinuma described a pattern-matching type of technique which was actually implemented in a production system [69] by NEC. It is interesting to note that it uses a pattern-matching technique, since other researchers had stated that a pattern-matching technique would be too computationally intensive [51,55] and since the spatio-temporal gradient methods had received more favorable consideration in the literature [2,3,70,71].

All the algorithms discussed so far have in effect modeled the motion in the sequences as purely translational. Huang and Tsai [70] pointed out that if rotation of objects is to be considered, then the pattern matching technique requires a

three-dimensional data space with a concomitant increase in processing bandwidth, indicating a spatio-temporal gradient approach would be more feasible. However, the decrease in the bandwidth in adding rotational motion to the model is rather small, indicating that frame-to-frame motion is predominantly translational.

Paquin and Dubois [55] investigated a spatio-temporal gradient algorithm which employed motion-compensated prediction. Although they obtained an algorithm similar to that of Netravali and Robbins [51,52], they started from a slightly different perspective and with slightly different assumptions. The displacements were estimated on a field basis. The algorithm was simulated on eight different sequences, one that was artificially created by moving a still frame and seven that were obtained from broadcast television. They assured stability by limiting the maximum displacement and increased the convergence by resetting the displacement estimate to zero whenever the frame difference was less than the displaced frame difference. It should be noted that their maximum allowable displacement was 10 pels per field while Limb and Murphy [47] assumed 4 pels per frame would seldom be exceeded. In their experiments, they were primarily interested in determining tradeoffs between accuracy and computational complexity for the interpolator and the estimator. They tried three interpolators with a number of different volumes and found that the simplest one, a bilinear two-dimensional polynominal with a volume of 7 pels by 3 lines by 1 field, performed almost as well as the most

computationally intensive one. They tried three different estimators and found once again the simplest the best, although the change in estimators did have more effect on the prediction error than the interpolator had.

Although some of the aforementioned motion-estimation techniques have existed for over ten years [72], they have not exhibited superior motion (or intensity) prediction characteristics. In the following chapters, a way to improve the motion prediction and thus the data compression is developed. Although convergence of the basic algorithm has been proven [51], that proof of convergence and the resulting algorithms that have been implemented do not coincide.

To reiterate, the major contribution of this dissertation is twofold: 1) the image sequence data was investigated in an attempt to develop a more thorough analysis of the convergence requirements and the convergence rate, and 2) a new motion prediction technique was presented which increases the validity of the assumptions made in proving convergence and which decreases the total prediction error.

CHAPTER FOUR

THE BASIC ALGORITHM

Netravali and Robbins [51] were among the first to develop a pel-recursive technique to estimate the displacement for a moving object in an image sequence. The following development will be similar to their approach.

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. When no ambiguity occurs "frame t" is used in lieu of "the frame at time t." Although the $I(\mathbf{z},t)$ function is a sampled discrete-valued function, it is assumed that $\mathbf{z}$ and $I(\mathbf{z},t)$ can be viewed as continuous by interpolating.

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 $[t-1,t]$,

$$I(\mathbf{z},t) = I(\mathbf{z}-\mathbf{d},t-1). \tag{4.1}$$

Define a function called the displaced frame difference:

$$DFD(\mathbf{z},\hat{\mathbf{d}}^i) = I(\mathbf{z},t) - I(\mathbf{z}-\hat{\mathbf{d}}^i,t-1), \tag{4.2}$$

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. Thus what is

sought is an iterative algorithm of the form

$$\hat{\mathbf{d}}^{i+1} = \hat{\mathbf{d}}^{i} + \textit{update term}, \tag{4.3}$$

where for each step, the update term seeks to improve the estimate of $\mathbf{d}$. The ultimate goal is minimization of the magnitude of the prediction error, DFD. If a pel at location $\mathbf{z}_a$ is predicted with $\hat{\mathbf{d}}^{i}$ to have intensity $I(\mathbf{z}-\hat{\mathbf{d}}^{i},t-1)$, resulting in a prediction error of $DFD(\mathbf{z},\hat{\mathbf{d}}^{i})$, the predictor should attempt to create a new estimate, $\hat{\mathbf{d}}^{i+1}$ such that $|DFD(\mathbf{z},\hat{\mathbf{d}}^{i+1})| \leq |DFD(\mathbf{z},\hat{\mathbf{d}}^{i})|$. To apply minization techniques, a function is needed which when minimized implies the DFD is zero. This can be accomplished by minimizing $[DFD(\mathbf{z},\hat{\mathbf{d}}^{i})]^2$. The most common approach is a steepest descent or gradient method. Although other techniques may have faster rates of convergence, they require more computation and are more likely to diverge [73].

The iterative equation using the gradient method is [73]:

$$\hat{\mathbf{d}}^{i+1} = \hat{\mathbf{d}}^{i} - \frac{\epsilon}{2}\nabla_{\hat{\mathbf{d}}}[DFD(\mathbf{z}_a,\hat{\mathbf{d}})]^2\Big|_{\hat{\mathbf{d}}=\hat{\mathbf{d}}^{i}}, \tag{4.4}$$

where $\nabla_{\hat{\mathbf{d}}}$ is the two-dimensional gradient operator with respect to displacement $\hat{\mathbf{d}}$ and ϵ is a positive scalar constant called the convergence coefficient or coefficient on the update term. This can be seen to simplify to

$$\hat{\mathbf{d}}^{i+1} = \hat{\mathbf{d}}^{i} \quad \epsilon DFD(\mathbf{z}_a,\hat{\mathbf{d}}^{i})\nabla_{\mathbf{d}} DFD(\mathbf{z}_a,\hat{\mathbf{d}})\Big|_{\hat{\mathbf{d}}=\hat{\mathbf{d}}^{i}} . \tag{4.5}$$

$\nabla_{\mathbf{d}}$ can be evaluated by using the definition of DFD in eq. (4.2) and noting that

$$\nabla_{\mathbf{d}} DFD(\mathbf{z}_a,\hat{\mathbf{d}})\Big|_{\hat{\mathbf{d}}=\hat{\mathbf{d}}^{i}} = \nabla_{\mathbf{z}} I(\mathbf{z}-\hat{\mathbf{d}}^{i},t-1)\Big|_{\mathbf{z}=\mathbf{z}_a}, \tag{4.6}$$

where $\nabla_{\mathbf{z}}$ is the two-dimensional spatial gradient operator with respect to $\mathbf{z}$. Thus

$$\nabla_{\mathbf{d}}[DFD(\mathbf{z}_a,\hat{\mathbf{d}})]^2\Big|_{\hat{\mathbf{d}}=\hat{\mathbf{d}}^{i}} = 2\cdot DFD(\mathbf{z}_a,\hat{\mathbf{d}}^{i})\nabla_{\mathbf{z}} I(\mathbf{z}-\hat{\mathbf{d}}^{i},t-1)\Big|_{\mathbf{z}=\mathbf{z}_a}. \tag{4.7}$$

Substituting (4.7) into (4.4) yields

$$\hat{\mathbf{d}}^{i+1} = \hat{\mathbf{d}}^{i} - \epsilon DFD(\mathbf{z}_a,\hat{\mathbf{d}}^{i})\nabla_{\mathbf{z}} I(\mathbf{z}-\hat{\mathbf{d}}^{i},t-1)\Big|_{\mathbf{z}=\mathbf{z}_a}. \tag{4.8}$$

For noninteger $\hat{\mathbf{d}}^{i}$, DFD and $\nabla_{\mathbf{z}} I$ can be obtained by interpolation. When no ambiguity or confusion will occur, the argument $(\mathbf{z}_a-\hat{\mathbf{d}}^{i},t-1)$ will be simplified to (i).

Although (4.8) is a standard iterative technique, it is not guaranteed to converge to $\mathbf{d}$ in every case. Constraints may be required on the value of ϵ. These constraints are derived in the next chapter. It would also be advantageous to have a measure of how fast the displacement estimates converge to the true displacement value. This rate of convergence is likewise derived in Chapter 5.

CHAPTER FIVE

CONVERGENCE ANALYSIS

5.1 Introduction

Two of the important issues of an iterative algorithm such as the one presented in Chapter 4 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 [51-53,55,66,74,75]. The compression rate would be greater if the displacement-estimate algorithm in (4.8) 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 5.1 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 spa-

tial location estimate, $(x - \dot{d}^0)$. then the displacement estimation algorithm should converge to d. However, if $(x - \dot{d}^0)$ 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.

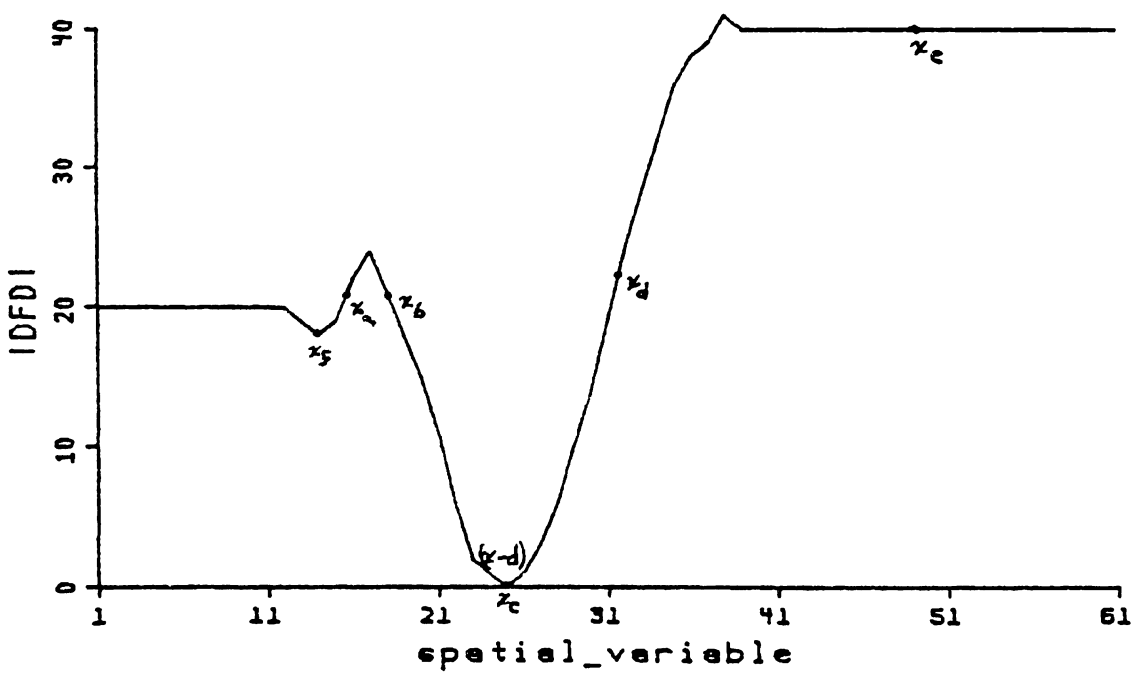


Figure 5.1: A Sample Plot of $|DFD|$

One problem with an algorithm which converges to the correct displacement at every pel is a large computational expense. Horn and Schunck [61,62] and Nagel [63] 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 [61,62] and Nagel [63] was that the displacement vectors had to vary smoothly.

In its initial passes over the image, Nagel's algorithm [63], which was an enhancement of Horn and Schunck's [61,62], ultimately had the same problem as an algorithm based only on minimizing $|DFD(\mathbf{z}_n,\vec{\mathbf{d}}^i)|^2$. 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 [76] which in effect finds the average gradient within an area. This is similar to the technique proposed by Huang [70] and does not necessarily lead to a well-conditioned problem. In fact, Netravali and Robbins [53] 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 dissertation, it should be reiterated that the techniques of Horn and Schunck [61,62] and Nagel [63,64] have only been applied to the motion-estimation/image-segmentation problem and have not been extended to the image 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 dissertation. 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.

5.2 First Proof of Convergence of the Displacement Vector Estimate

In this section, it will be proved that the displacement estimation algorithm defined by Equation (4.8) converges under certain conditions to the true displacement as a moving object is scanned. The proof procedure is similar to one used in [51]. The assumptions are that the motion is purely translational and that the uncovered background is neglected. Start by substituting Equation (4.2) into (4.8) to obtain

$$\hat{\mathbf{d}}^{i+1} = \hat{\mathbf{d}}^{i} - \epsilon\{I(\mathbf{z}_a,t) - I(\mathbf{z}_a - \hat{\mathbf{d}}^{i},t-1)\}\nabla_{\mathbf{z}} I(\mathbf{z} - \hat{\mathbf{d}}^{i},t-1)\Big|_{\mathbf{z}=\mathbf{z}_a}. \tag{5.1}$$

Substituting from (4.1) for $I(\mathbf{z}_a,t)$,

$$\hat{\mathbf{d}}^{i+1} = \hat{\mathbf{d}}^{i} - \epsilon\{I(\mathbf{z}_a - \mathbf{d},t-1) - I(\mathbf{z}_a - \hat{\mathbf{d}}^{i},t-1)\}\nabla_{\mathbf{z}} I(\mathbf{z} - \hat{\mathbf{d}}^{i},t-1)\Big|_{\mathbf{z}=\mathbf{z}_a}. \tag{5.2}$$

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

$$\begin{aligned} I(\mathbf{z}_a - \mathbf{d},t-1) - I(\mathbf{z}_a - \hat{\mathbf{d}}^{i},t-1) = {} & (\hat{\mathbf{d}}^{i} - \mathbf{d})^T \nabla_{\mathbf{z}} I(\mathbf{z} - \hat{\mathbf{d}}^{i},t-1)\Big|_{\mathbf{z}=\mathbf{z}_a} \\ & + \tfrac{1}{2}(\hat{\mathbf{d}}^{i} - \mathbf{d})^T \nabla_{\mathbf{z}}^2 I(\mathbf{z} - \hat{\mathbf{d}}^{i},t-1)(\hat{\mathbf{d}}^{i} - \mathbf{d})\Big|_{\mathbf{z}=\mathbf{z}_a} \\ & + O(\hat{\mathbf{d}}^{i} - \mathbf{d})^3, \end{aligned} \tag{5.3}$$

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

$$(\hat{\mathbf{d}}^{i} \quad \mathbf{d}) = \begin{bmatrix} \Delta d_x^i \\ \Delta d_y^i \end{bmatrix} \tag{5.4}$$

and

$$\nabla_{\mathbf{z}} I(i)\Big|_{\mathbf{z}=\mathbf{z}_a} = \begin{bmatrix} \dfrac{\partial I(i)}{\partial x} \\ \dfrac{\partial I(i)}{\partial y} \end{bmatrix}, \tag{5.5}$$

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

$$I(\mathbf{z}_a - \mathbf{d}, t-1) - I(\mathbf{z}_a - \hat{\mathbf{d}}^{i}, t-1) = \sum_{j=1}^{\infty} \frac{1}{j!} \left\{ \Delta d_x^i \left[\frac{\partial I(i)}{\partial x} \right] + \Delta d_y^i \left[\frac{\partial I(i)}{\partial y} \right] \right\}^{j}, \tag{5.6}$$

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 [77]. (See Appendix A.)

Substituting (5.6) into (5.2),

$$\hat{\mathbf{d}}^{i+1} = \hat{\mathbf{d}}^{i} - \epsilon \sum_{j=1}^{\infty} \frac{1}{j!} \left\{ \Delta d_x^i \left[\frac{\partial I(i)}{\partial x} \right] + \Delta d_y^i \left[\frac{\partial I(i)}{\partial y} \right] \right\}^j \nabla_z I(\mathbf{z} - \hat{\mathbf{d}}^i, t-1) \Big|_{\mathbf{z}=\mathbf{z}_a}, \tag{5.7}$$

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

$$\hat{\mathbf{d}}^{i+1} = \hat{\mathbf{d}}^{i} - \epsilon \{ \nabla_z I(\mathbf{z} - \hat{\mathbf{d}}^i, t-1) \mathbf{f}(\mathbf{z}_a - \hat{\mathbf{d}}^i, t-1) \} (\hat{\mathbf{d}}^i - \mathbf{d}) \Big|_{\mathbf{z}=\mathbf{z}_a} \tag{5.8}$$

or

$$\hat{\mathbf{d}}^{i+1} = \hat{\mathbf{d}}^{i} - \epsilon \{ \nabla_z I(i) \mathbf{f}(i) \} (\hat{\mathbf{d}}^i - \mathbf{d}), \Big|_{\mathbf{z}=\mathbf{z}_a} \tag{5.9}$$

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

$$\mathbf{f}(i) = \sum_{j=1}^{\infty} \frac{1}{j!} \left\{ \Delta d_x^i \left[\frac{\partial I(i)}{\partial x} \right] + \Delta d_y^i \left[\frac{\partial I(i)}{\partial y} \right] \right\}^{j-1} \left\{ \frac{\partial I(i)}{\partial x} \quad \frac{\partial I(i)}{\partial y} \right\} \tag{5.10}$$

The first two terms of $\mathbf{f}(i)$ are indicated in Figure 5.2.

Subtracting $\mathbf{d}$ from both sides of (5.9),

$$(\hat{\mathbf{d}}^{i+1} - \mathbf{d}) = (\hat{\mathbf{d}}^{i} - \mathbf{d}) - \epsilon \{ \nabla_z I(i) \mathbf{f}(i) \} (\hat{\mathbf{d}}^i - \mathbf{d}) \Big|_{\mathbf{z}=\mathbf{z}_a} \tag{5.11}$$

or

$$(\hat{\mathbf{d}}^{i+1} - \mathbf{d}) = [\mathbf{J} - \epsilon \{ \nabla_z I(i) \mathbf{f}(i) \}] (\hat{\mathbf{d}}^i - \mathbf{d}), \Big|_{\mathbf{z}=\mathbf{z}_a} \tag{5.12}$$

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

Equation (5.12) 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 $k \to \infty$, conver-

gence will be proved. Assume $\mathbf{B} = \prod_{i=0}^{k} \mathbf{A}_i$ as $k \to \infty$. $\mathbf{B}\mathbf{e}_0 = 0$ if $\mathbf{e}_0 \in N(\mathbf{B})$, the null space of $\mathbf{B}$. Alternatively it can be shown that $\mathbf{B}\mathbf{e}_0 = 0$ if the spectral radius of $\mathbf{B}$ is less than 1. Unfortunately, neither of these is necessarily the case.

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

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

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

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

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

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

Figure 5.2: Expansion of the First Two Terms of $\mathbf{f}(i)$

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 1 depending on ϵ. If the spectral radius of $\mathbf{A}$ is 1, 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 [78-81]. 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,

$$||E\{\hat{\mathbf{d}}^{i+1}\}-\mathbf{d}|| \leq ||\mathbf{J}-\epsilon E\{\nabla_{\mathbf{z}} I(i)\mathbf{f}(i)\}||\cdot||E\{\hat{\mathbf{d}}^{i}\}-\mathbf{d}||, \tag{5.13}$$

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

$$||E\{\hat{\mathbf{d}}^{i+1}\}-\mathbf{d}|| \leq |1-\epsilon\lambda_{\max}|\cdot||E\{\hat{\mathbf{d}}^{i}\}-\mathbf{d}||, \tag{5.14}$$

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

$$E\{\nabla_{\mathbf{z}} I(i)\mathbf{f}(i)\}. \tag{5.15}$$

For convergence of the algorithm, the following condition must be satisfied:

$$|1-\epsilon\lambda_{\max}| < 1, \tag{5.16}$$

which is equivalent to

$$\frac{2}{\lambda_{max}} > \epsilon > 0. \tag{5.17}$$

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:

$$\frac{4}{trE\{\nabla_z I(i)\mathbf{f}(i)\}} > \epsilon > 0, \tag{5.18}$$

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

If $\mathbf{f}_1(i) \approx \mathbf{f}(i)$ (see Figure 5.2 and Appendix A), then Equation (5.18) can be rewritten as

$$\frac{4M}{\sum_{i=1}^{M} [\{\nabla_z I(i)\}^2 + \{\nabla_y I(i)\}^2]} > \epsilon > 0, \tag{5.19}$$

where M iterations are performed within the moving area. The number of iterations per pel is not required to be constant. $\nabla_z I(i)$ 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 in $\{\nabla I(i)\}^2$ in Equation (5.19) and $\nabla^2 I(i)$ in Equation (5.3) and the equation in Figure 5.2.

Netravali and Robbins [51] derived an equation very similar to Equation (5.19). Their proof assumed convergence over time, i.e. within the moving area, with only one iteration per pel. They made no attempt to verify analytically that

the value of the convergence coefficient they used in their simulations (0.0625) met their required condition. In Chapter 7, 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.

5.3 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 (5.9) can be rewritten as

$$\hat{\mathbf{d}}^{i+1} = \hat{\mathbf{d}}^{i} - \epsilon\{\nabla_{\mathbf{z}} I(i)\mathbf{f}(i)\}\hat{\mathbf{d}}^{i}\Big|_{\mathbf{z}=\mathbf{z}_a} + \epsilon\{\nabla_{\mathbf{z}} I(i)\mathbf{f}(i)\}\mathbf{d}\Big|_{\mathbf{z}=\mathbf{z}_a} \tag{5.20}$$

or

$$\hat{\mathbf{d}}^{i+1} = [\mathbf{J} - \epsilon\{\nabla_{\mathbf{z}} I(i)\mathbf{f}(i)\}]\hat{\mathbf{d}}^{i}\Big|_{\mathbf{z}=\mathbf{z}_a} + \epsilon\{\nabla_{\mathbf{z}} I(i)\mathbf{f}(i)\}\mathbf{d}\Big|_{\mathbf{z}=\mathbf{z}_a}. \tag{5.21}$$

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

$$E\{\hat{\mathbf{d}}^{i+1}\} = [\mathbf{J} - \epsilon E\{\nabla_{\mathbf{z}} I(i)\mathbf{f}(i)\}]E\{\hat{\mathbf{d}}^{i}\} + \epsilon E\{\nabla_{\mathbf{z}} I(i)\mathbf{f}(i)\}\mathbf{d}. \tag{5.22}$$

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

$$E\{\nabla_{\mathbf{z}} I(i)\mathbf{f}(i)\} = \mathbf{U}(i)\mathbf{S}(i)\mathbf{U}^{T}(i), \tag{5.23}$$

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

$$\mathbf{U}^{T}(i)\mathbf{U}(i) = \mathbf{J}. \tag{5.24}$$

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" (5.21) for the expected displacement vector into a

coordinate system in which the expected displacement components are uncoupled. If one were to expand (4.21), 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 (5.23) provides a set of uncoupled 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

$$E\{\hat{\mathbf{d}}^i\} = \mathbf{U}(i)\mathbf{v}(i) \tag{5.25}$$

and

$$\mathbf{v}(i) = \mathbf{U}^T(i)E\{\hat{\mathbf{d}}^i\}. \tag{5.26}$$

Substituting (5.23) and (5.25) into (5.22),

$$\mathbf{v}(i+1)=\mathbf{G}(i+1)[\mathbf{J}-\epsilon\mathbf{S}(i)]\mathbf{v}(i) + \epsilon\mathbf{U}^T(i+1)\mathbf{S}(i)\mathbf{d}, \tag{5.27}$$

where

$$\mathbf{G}(i+1) = \mathbf{U}^T(i+1)\mathbf{U}(i). \tag{5.28}$$

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. (5.27) becomes

$$\mathbf{v}(i+1) = [\mathbf{J}-\epsilon\mathbf{S}]\mathbf{v}(i) + \epsilon\mathbf{U}^T\mathbf{S}\mathbf{d}, \tag{5.29}$$

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 necessary to solve (5.27) for uncoupled displacement vector, $\mathbf{v}(i+1)$, and utilize (5.25) for the transformation back to the $E\{\hat{\mathbf{d}}^{i+1}\}$.

To solve (5.29), it is instructive to expand a few terms in the $\mathbf{v}(i)$ sequence:

$$\begin{aligned}
i=0:\ \mathbf{v}(1) &= [\mathbf{J}-\epsilon\mathbf{S}]\mathbf{v}(0) + \epsilon\mathbf{U}^T\mathbf{S}\mathbf{d} \\
i=1:\ \mathbf{v}(2) &= [\mathbf{J}-\epsilon\mathbf{S}]\mathbf{v}(1) + \epsilon\mathbf{U}^T\mathbf{S}\mathbf{d} \\
&= [\mathbf{J}-\epsilon\mathbf{S}]\{[\mathbf{J}-\epsilon\mathbf{S}]\mathbf{v}(0) + \epsilon\mathbf{U}^T\mathbf{S}\mathbf{d}\} + \epsilon\mathbf{U}^T\mathbf{S}\mathbf{d} \\
&= [\mathbf{J}-\epsilon\mathbf{S}]^2\mathbf{v}(0) + [\mathbf{J}-\epsilon\mathbf{S}]\epsilon\mathbf{U}^T\mathbf{S}\mathbf{d} + \epsilon\mathbf{U}^T\mathbf{S}\mathbf{d} \\
i=2:\ \mathbf{v}(3) &= [\mathbf{J}-\epsilon\mathbf{S}]\mathbf{v}(2) + \epsilon\mathbf{U}^T\mathbf{S}\mathbf{d} \\
&= [\mathbf{J}-\epsilon\mathbf{S}]\{[\mathbf{J}-\epsilon\mathbf{S}]^2\mathbf{v}(0) + [\mathbf{J}-\epsilon\mathbf{S}]\epsilon\mathbf{U}^T S\mathbf{d} + \epsilon\mathbf{U}^T\mathbf{S}\mathbf{d}\} + \epsilon\mathbf{U}^T\mathbf{S}\mathbf{d}.
\end{aligned} \tag{5.30}$$

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

$$\mathbf{v}(i) = [\mathbf{J}-\epsilon\mathbf{S}]^i\mathbf{v}(0) + \sum_{n=0}^{i-1}\epsilon[\mathbf{J}-\epsilon\mathbf{S}]^n\mathbf{U}^T\mathbf{S}\mathbf{d}, \qquad i>0. \tag{5.31}$$

Since $[\mathbf{J}-\epsilon\mathbf{S}]$ is a diagonal matrix, the $v_j(i)$ in (5.31) are uncoupled. Thus, (5.31) written in terms of a single $v_j(i)$ becomes

$$v_j(i) = (1-\epsilon\lambda_j)^i v_j(0) + \epsilon c_j \sum_{n=0}^{i-1}(1-\epsilon\lambda_j)^n \tag{5.32}$$

where

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

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

Performing the summation in (5.32) produces

$$v_j(i) = (1 - \epsilon\lambda_j)^i v_j(0) + \epsilon c_j \left[\frac{1-(1-\epsilon\lambda_j)^i}{1-(1-\epsilon\lambda_j)} \right]. \tag{5.33}$$

Simplifying and using the substitution

$$\gamma_j = 1-\epsilon\lambda_j, \tag{5.34}$$

(5.33) becomes

$$v_j(i) = \gamma_j^i v_j(0) + \frac{c_j}{\lambda_j}(1-\gamma_j^i), \qquad i \geq 0. \tag{5.35}$$

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 (4.17) through (4.19).

(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 (5.29) to be within e^{-1} of its final value, $v_j(\infty)$. This produces the following equation:

$$\begin{aligned} v_j(\tau_j) &= v_j(\infty) + e^{-1}[v_j(0)-v_j(\infty)] \\ &= v_j(0) + (1-e^{-1})[v_j(\infty)-v_j(0)] \end{aligned} \tag{5.36}$$

which, with the use of (5.35), may be solved for τ_j:

$$\begin{aligned} \gamma_j^{\tau_j} v_j(0) + \frac{c_j}{\lambda_j}(1-\gamma_j^{\tau_j}) &= v_j(0)+(1-e^{-1})[\frac{c_j}{\lambda_j} - v_j(0)] \\ &= e^{-1}v_j(0)+\frac{c_j}{\lambda_j}(1-e^{-1}). \end{aligned} \tag{5.37}$$

This implies that

$$\begin{aligned} \gamma_j^{\tau_j} &= e^{-1} \\ (1-\epsilon\lambda_j)^{\tau_j} &= e^{-1} \\ \tau_j ln(1-\epsilon\lambda_j) &= -1 \\ \tau_j &= \frac{-1}{ln(1-\epsilon\lambda_j)}. \end{aligned} \tag{5.38}$$

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 5.1.

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 not always 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 (5.18), ϵ is constrained to be less than 0.0067. Arbitrarily choose $\epsilon = 0.0010$. Then

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

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

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

Example 5.1: Convergence Rate for Displacement Estimates

5.4 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 (5.17) 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, (4.4) can be rewritten as

$$\hat{\mathbf{d}}^{i+1} = \hat{\mathbf{d}}^{i} - \frac{\epsilon}{2}\{\nabla_{\mathbf{d}}[DFD(\mathbf{z}_a, \hat{\mathbf{d}}^{i})]^2 + \mathrm{N}(i)\}. \tag{5.39}$$

After working this change through the intermediate equations, (5.12) can be rewritten as

$$(\hat{\mathbf{d}}^{i+1} - \mathbf{d}) = [\mathbf{J} - \epsilon\{\nabla_{\mathbf{z}} I(i)\mathbf{f}(i)\}](\hat{\mathbf{d}}^{i} - \mathbf{d}) + \frac{\epsilon}{2}\{\mathrm{N}(i)\}. \tag{5.40}$$

The eigenvector matrix $\mathbf{U}$ is used to provide a transformation similar to Equations (5.25) and (5.26), which decouple the displacement vector equations.

Thus letting

$$\boldsymbol{\delta}(i) = \mathbf{U}^{T}(i)(\hat{\mathbf{d}}^{i} - \mathbf{d}) \tag{5.41}$$

and

$$(\hat{\mathbf{d}}^{i} \quad \mathbf{d}) = \mathbf{U}(i)\boldsymbol{\delta}(i) \tag{5.42}$$

and substituting (5.42) into (5.10), the following uncoupled difference vector iterative equation is obtained:

$$\boldsymbol{\delta}(i+1) = [\mathbf{J} - \epsilon\mathbf{S}(i)]\boldsymbol{\delta}(i) - \frac{\epsilon}{2}\mathbf{U}^{T}(i)\mathbf{N}(i). \tag{5.43}$$

Equation (5.43) 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 (5.43) go to zero as i increases. Taking the expected value of both sides of (5.43) and enforcing this requirement yields

$$E\{\mathbf{N}(i)\}=0. \tag{5.44}$$

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 (5.43):

$$\boldsymbol{\delta}(i)\boldsymbol{\delta}^{T}(i) = \{[\mathbf{J} - \epsilon\mathbf{S}]\boldsymbol{\delta}(i) - \frac{\epsilon}{2}\mathbf{U}^{T}\mathbf{N}(i)\}\{[\mathbf{J} - \epsilon\mathbf{S}]\boldsymbol{\delta}(i) - \frac{\epsilon}{2}\mathbf{U}^{T}\mathbf{N}(i)\}^{T}. \tag{5.45}$$

Performing the matrix multiplication and then simplifying produces:

$$\begin{aligned}\boldsymbol{\delta}(i)\boldsymbol{\delta}^{T}(i) = {} & [\mathbf{J} - \epsilon\mathbf{S}]\boldsymbol{\delta}(i)\boldsymbol{\delta}^{T}(i)[\mathbf{J} - \epsilon\mathbf{S}]^{T} \\ & + (\frac{\epsilon}{2})^{2}\mathbf{U}^{T}\mathbf{N}(i)\mathbf{N}^{T}(i)U \\ & - \frac{\epsilon}{2}\{[\mathbf{J} - \epsilon\mathbf{S}]\boldsymbol{\delta}(i)\mathbf{N}^{T}(i)\mathbf{U} + \mathbf{U}^{T}\mathbf{N}(i)\boldsymbol{\delta}(i)[\mathbf{J} - \epsilon\mathbf{S}]\}.\end{aligned} \tag{5.46}$$

Next take the expected value of both sides. Since $\boldsymbol{\delta}(i)$ is only affected by gradient estimation noise from the previous iterations, the vectors $\boldsymbol{\delta}(i)$ and $\mathbf{N}(i)$ are uncorrelated and the expected value of the term in brackets in (5.46) is zero. Using this result, the expectation of both sides of (5.46) becomes

$$E\{\boldsymbol{\delta}(i+1)\boldsymbol{\delta}^T(i+1)\} = [\mathbf{J}-\epsilon\mathbf{S}]E\{\boldsymbol{\delta}(i)\boldsymbol{\delta}^T(i)\}[\mathbf{J}-\epsilon\mathbf{S}] + \frac{\epsilon^2}{4}\mathbf{U}^T E\{\mathbf{N}(i)\mathbf{N}^T(i)\}\mathbf{U}. \tag{5.47}$$

To derive useful results from (5.47) some assumptions must be made. Recall that $\mathbf{N}(i)$ is the gradient estimation "noise" and is the difference between the true value of the gradient and the computed value. Denote $E\{\mathbf{N}(i)\mathbf{N}^T(i)\}$ as $\boldsymbol{\Sigma}^2_{gn}$, where $\boldsymbol{\Sigma}^2_{gn}$ is a 2x2 matrix. (See Appendix B for derivation of $\boldsymbol{\Sigma}^2_{gn}$.)

Substituting $\boldsymbol{\Sigma}^2_{gn}$ into (5.47) and recognizing that at convergence the expectation matrices are no longer time dependent, that is

$$E\{\boldsymbol{\delta}(i+1)\boldsymbol{\delta}^T(i+1)\} = E\{\boldsymbol{\delta}(i)\boldsymbol{\delta}^T(i)\} = E\{\boldsymbol{\delta}\boldsymbol{\delta}^T\}, \tag{5.48}$$

the following result is obtained:

$$E\{\boldsymbol{\delta}\boldsymbol{\delta}^T\} = [\mathbf{J}-\epsilon\mathbf{S}]E\{\boldsymbol{\delta}\boldsymbol{\delta}^T\}[\mathbf{J}-\epsilon\mathbf{S}] + \frac{\epsilon^2}{4}\mathbf{U}^T\boldsymbol{\Sigma}^2_{gn}U. \tag{5.49}$$

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

$$E\{\boldsymbol{\delta}\boldsymbol{\delta}^T\} = \frac{\epsilon^2}{4}[2\epsilon\mathbf{S}-\epsilon^2\mathbf{S}^2]^{-1}\mathbf{U}^T\boldsymbol{\Sigma}^2_{gn}\mathbf{U}. \tag{5.50}$$

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

$$E\{\delta\delta^T\} = \frac{\epsilon}{4}\mathbf{S}^{-1}\Lambda\mathbf{U}^T\Sigma_{gn}^2\mathbf{U}, \tag{5.51}$$

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

$$\Lambda_{jj} = (2 - \epsilon\lambda_j)^{-1} = (1+\gamma_j)^{-1} \tag{5.52}$$

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

Using (5.41) and the substitution $\Delta = E\{(\hat{\mathbf{d}}^i - \mathbf{d})(\hat{\mathbf{d}}^i - \mathbf{d})^T\}$, it can be derived that

$$\Delta = \frac{\epsilon}{4}\Sigma_{gn}^2\mathbf{S}^{-1}\Lambda. \tag{5.53}$$

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

$$Var[\hat{\mathbf{d}}_j^i - \mathbf{d}_j] = \Delta_{jj}. \tag{5.54}$$

See Example 5.2

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 always prudent. In the next chapter a new motion prediction technique is presented which does a better job of predicting the displacement vectors at the edges of moving objects and thereby leads to greater compression of the information in an image sequence.

Consider the edge in Example 5.1 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 $\dot{\mathbf{d}}$ at convergence, the noise variance of the image, σ_n^2, must be known. Assume $\sigma_n^2 = 0.0001$. Therefore from Appendix B,

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

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

$$\mathbf{S}^{-1} = \begin{bmatrix} \frac{1}{\lambda_1} & 0 \\ 0 & \frac{1}{\lambda_2} \end{bmatrix} \tag{5.23}$$

$$\Lambda = \begin{bmatrix} (2-\epsilon\lambda_1)^{-1} & 0 \\ 0 & (2-\epsilon\lambda_2)^{-1} \end{bmatrix}. \tag{5.52}$$

From (4.53)

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

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

Letting $\epsilon = 0.0010$ as in Example 5.1,

$$\Delta = 0.001 \begin{bmatrix} 0.08 & 0 \\ 0 & 0.01 \end{bmatrix} \begin{bmatrix} 0.0023 & 0 \\ 0 & 0.0063 \end{bmatrix} \begin{bmatrix} 0.642 & 0 \\ 0 & 0.543 \end{bmatrix}$$

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

Therefore the displacement variance at convergence is

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

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

Example 5.2: Displacement Estimate Variance at Convergence

CHAPTER SIX

AN IMPROVED MOTION PREDICTION TECHNIQUE

6.1 Introduction

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 [51-53,66,74,75]. Other researchers, mostly from Bell Northern Research [3,55], 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.

6.2 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 [59]. In Chapter 5 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 6.1 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 $\nabla_x I|_{t-1}$

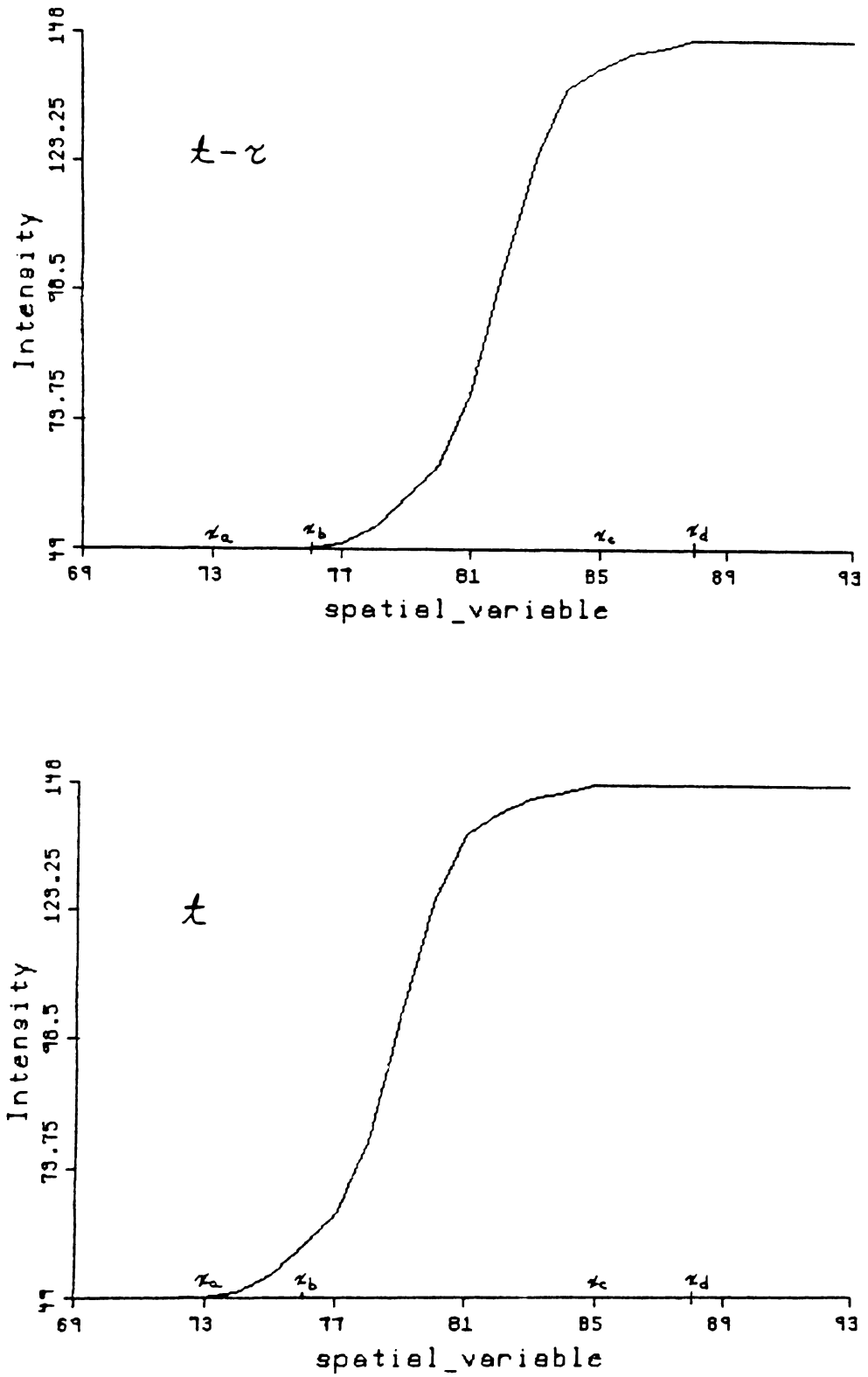


Figure 6.1: A Moving Edge in One Dimension

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 [3,55] 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.

As one example of the problem with temporal prediction, look again at Figure 6.1. 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 6.2. 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

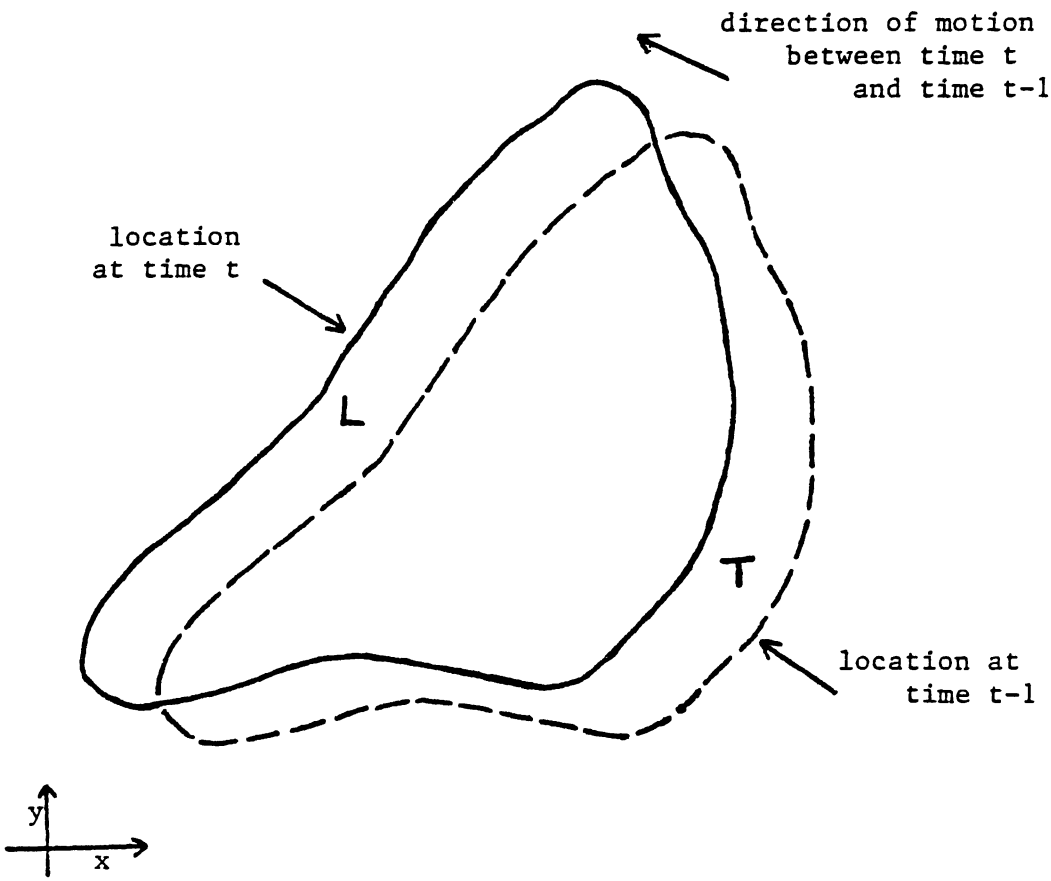


Figure 6.2: A Moving Object, Top View

leading edge (area L in Figure 6.2) 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 6.2), 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).

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.

6.3 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 [51]. Yet most algorithms [3,50-53,55,66,74] 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 [3,51]. As a side note, either constant interframe velocity or constant interframe acceleration can be assumed when using the PAMT prediction technique.

6.4 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 $\nabla_{\mathbf{z}} I(i) = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$. 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 6.2.) 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 $\mathbf{d}$.

CHAPTER SEVEN

SIMULATIONS

7.1 Introduction

In this chapter, simulations are presented to show the performance of the proposed motion-compensation image-sequence compression technique. Comparisons are made to existing techniques and to the analytical results in Chapters 5 and 6.

Three algorithms will be used for comparison purposes [82]:

1) A motion-compensated (MC) algorithm in which spatial prediction of the motion vectors is used. The motion vector at the pel which is one line and one pel previous to the pel under consideration is used.

2) A MC algorithm in which the motion vectors are predicted by projection along the motion trajectory (PAMT). A gap bridging technique is used on the predicted motion-vector field.

3) A MC algorithm in which the predicted motion vector ($\hat{\mathbf{d}}^0$) is the average of the motion vectors obtained using spatial and PAMT prediction. Although this prediction technique may increase the prediction error at the edges of moving objects, it will smooth the $\hat{\mathbf{d}}^0$ values as a moving object is traversed. The smaller the variance of the motion vector as it converges, the smaller the variance of the intensity error sequence. A small intensity-error variance

element-to-element can be exploited to further decrease the information rate by transmitting the difference in consecutive intensity errors rather than the intensity errors directly [12].

The total entropy bit rate (a measure to be discussed in Section 7.2) is the sum of the information bits and the addressing bits. The information bits are based on taking the difference of consecutive error values as indicated above; the addressing bits are obtained using a runlength encoding.

The following parameters are used in all three MC algorithms:

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 [14,22,51-53,66-68].

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 FD at those same pels. This is the technique proposed in [52].

5) As is usually done, the simulation starts with a previous frame at the receiver. The motion vectors for the first frame to be transmitted are always determined using spatial prediction.

In Section 7.3 the three motion-compensated techniques are simulated on six synthetic image sequences. In Section 7.4 the algorithms are simulated on three actual image sequences that are typical of those that might occur in a video teleconferencing environment. In both the synthetic and actual image sequences, the improvements in the intensity prediction and the motion prediction are shown by measuring the image compression, the image quality, and the motion predictability. These measures are discussed in the next section.

7.2 Measures of Image Quality and Compression

There are a two ways to improve image compression. One way is to maintain the image quality while increasing the amount of compression. The other is to to maintain the amount of compression while improving the visual fidelity (quality) of the image obtained from the compressed representation [81].

The average information rate is given by the entropy (measured in bits)

$$H(u) = -\sum_{i=1}^{L} p_i \log_2 p_i \tag{7.1}$$

where p_i is the probability that a quantized sample u takes the value r_i, for example, from a set of L unique symbols. This is called the zeroth-order entropy since no consideration is given to the fact that a given sample may be statistically dependent on its neighbors. For monochrome images with 256 levels (8 bit pels), the zeroth order entropy is usually around 4-6 bits [4].

The most common analytical (objective) measure of image fidelity is the average sample mean-square error [1,4,81], which for an NxM image is defined as

$$e_{ms}^2 \approx \frac{1}{N \cdot M} \sum_{i=1}^{N} \sum_{j=1}^{M} (u_{i,j} - \hat{u}_{i,j})^2 \tag{7.2}$$

where $\{u_{i,j}\}$ and $\{\hat{u}_{i,j}\}$ represent the NxM original and reconstructed images respectively. The average sample mean-square error of the reconstructed image will vary with the interframe motion. The interframe motion is defined as the percentage of pels which differ in value between two frames by greater than 1.5% of the peak-to-peak value [5,83]. It is determined by taking the difference between

each pair of corresponding pels in two frames, counting the number of differences greater than 3, and dividing by the total number of pels in a frame.

A viable measure used to ascertain motion-prediction quality is simply the number of pels for which $\hat{\mathbf{d}}^i$ is correctly predicted (i.e., no motion update is needed since the thresholded intensity error is zero).

7.3 Synthetic Image Sequences

7.3.1 Introduction

This section contains the results of simulating the motion-compensated image-sequence compression algorithms on synthetic image sequences. These simulations were done to show the convergence of the displacement estimation algorithm both as a moving object is scanned and as a sequence of frames is processed. A hemisphere was chosen as the object to be translated to create the synthetic image sequence. The hemisphere was generated using

$$I(\mathbf{z},t) = \begin{cases} 127 & \text{if } ||\mathbf{r}|| > 16 \\ 127(1+\cos(\frac{\pi}{32}\cdot||\mathbf{r}||) & \text{if } ||\mathbf{r}|| \leq 16 \end{cases}$$

where $\mathbf{r} = \mathbf{z} - (\mathbf{z}_0 + \mathbf{d}t)$, 127 is the background intensity (on a scale of 0 to 255), $\mathbf{z}_0$ is the location of the center of the moving hemisphere at t = 0, $\mathbf{d}$ is the frame-to-frame displacement, and $||\cdot||$ is the Euclidean norm. Note that all the pels have an intensity of 127 or greater. Ten frames were generated and there was no field interlace. One frame is shown in Figure 7.1. The hemisphere was chosen since it has nonzero gradients over the whole moving object and since it is a reasonable model of cheeks, hands, and faces. The translation was varied from 0 to 7 pels per frame in both horizontal directions.

Rather than present the data for all possible velocities with magnitude less than 7, six integer values were selected: ± 2, ± 4, and ± 7. Due to the switched

predictor, all algorithms should produce the same results for no interframe motion.

Upward motion within the field of view should give results similar to leftward motion and downward motion should give results similar to rightward motion, since a frame is scanned left-to-right and top-to-bottom and since the direction of spatial prediction is from the top left to the bottom right of the frame.

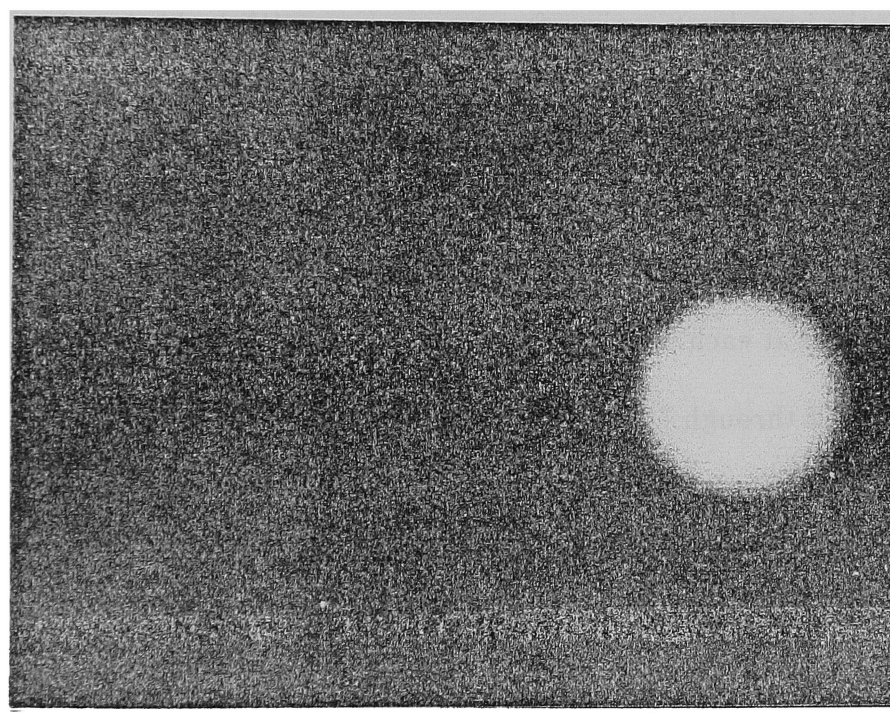


Figure 7.1: One Frame of Synthetic Sequence

Certain advantages occur with a synthetic image sequence. $E\{\nabla_s I\}$ is known a priori, thus the theoretical upper limit on ϵ can be determined. The velocity is known so that deviations from it can be determined.

For the given moving object,

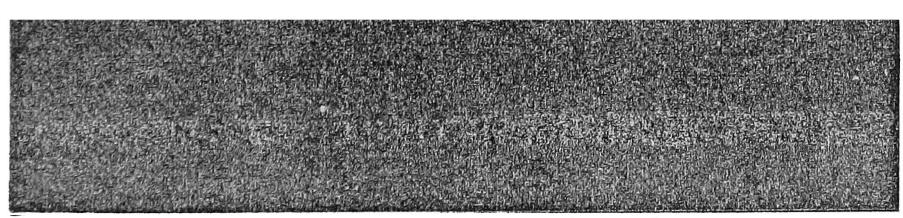


Figure 7.1: One Frame of Synthetic Sequence

maximum gradient components occur at the edges and, using a central difference evaluation, are approximately 13 or 14.

The analytical results do not indicate what value of ϵ will lead to the minimum average mse. The analysis only constrains ϵ to be positive and less than some predeterminable value. It also indicates that the largest allowable ϵ is not necessarily the best due to noise after convergence. With this in mind initial simulations were run to ascertain what values of ϵ gave the minimum entropy and error. It was determined that for all velocities, $0.010 \geq \epsilon \geq 0.005$ resulted in the minimum values. Therefore $\epsilon = 0.010$ and $\epsilon = 0.005$ were used for each of the three algorithms at each of the six velocities. The 36 simulation runs are presented in Sections 7.3.2 through 7.3.7.

The error in the average displacement estimate within the moving area in each frame is calculated as:

$$||\mathbf{d}_\epsilon|| = \frac{||\hat{\mathbf{d}} - \mathbf{d}||}{||\mathbf{d}||}$$

The displacement and the variance in the displacement will be denoted by a 2x1 matrix where the first entry corresponds to the vertical component and the second to the horizontal component. E.g., the $\mathbf{d}$ for the six sequences, in the order they are presented, are $\begin{bmatrix} 0 \\ -2 \end{bmatrix}$, $\begin{bmatrix} 0 \\ -4 \end{bmatrix}$, $\begin{bmatrix} 0 \\ -7 \end{bmatrix}$, $\begin{bmatrix} 0 \\ 7 \end{bmatrix}$, $\begin{bmatrix} 0 \\ 4 \end{bmatrix}$, and $\begin{bmatrix} 0 \\ 2 \end{bmatrix}$. In the first three sequences the hemisphere moves right to left as the viewer observes motion; in the last three the motion is in the opposite direction.

In the curves that will be plotted, spatial prediction results are indicated by circles, PAMT prediction results are indicated by triangles, and mixed prediction results are indicated by pluses.

7.3.2 Velocity of Two Leftward

7.3.2.1 $\epsilon = 0.010$

The average $||\mathbf{d}_e||$ and the average mse in each frame using each of the three prediction techniques are plotted in Figures 7.2a and 7.2b. Using spatial prediction, the variance in the displacement estimates over the moving area was about $\begin{bmatrix}0.72\\0.60\end{bmatrix}$ in all nine frames. Using PAMT prediction, the variance dropped from $\begin{bmatrix}0.68\\0.54\end{bmatrix}$ in the first frame to $\begin{bmatrix}0.54\\0.48\end{bmatrix}$ in the ninth frame. Using mixed prediction, the variance dropped from $\begin{bmatrix}0.68\\0.54\end{bmatrix}$ in the first frame to $\begin{bmatrix}0.46\\0.41\end{bmatrix}$ in the ninth frame.

The entropy bits required to transmit the nine frames were:

spatial:	23.5 kbits
PAMT:	22.9 kbits
mixed:	19.0 kbits

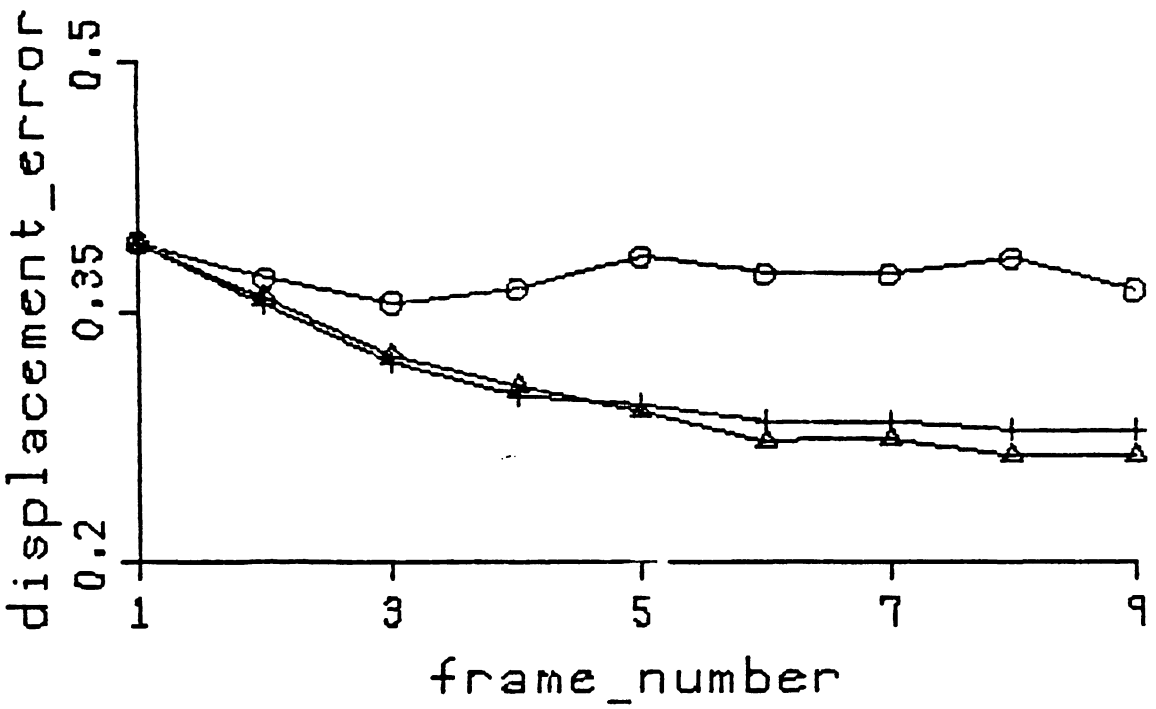


Figure 7.2a: Average $||\mathbf{d}_e||$, velocity = -2, ϵ = 0.010

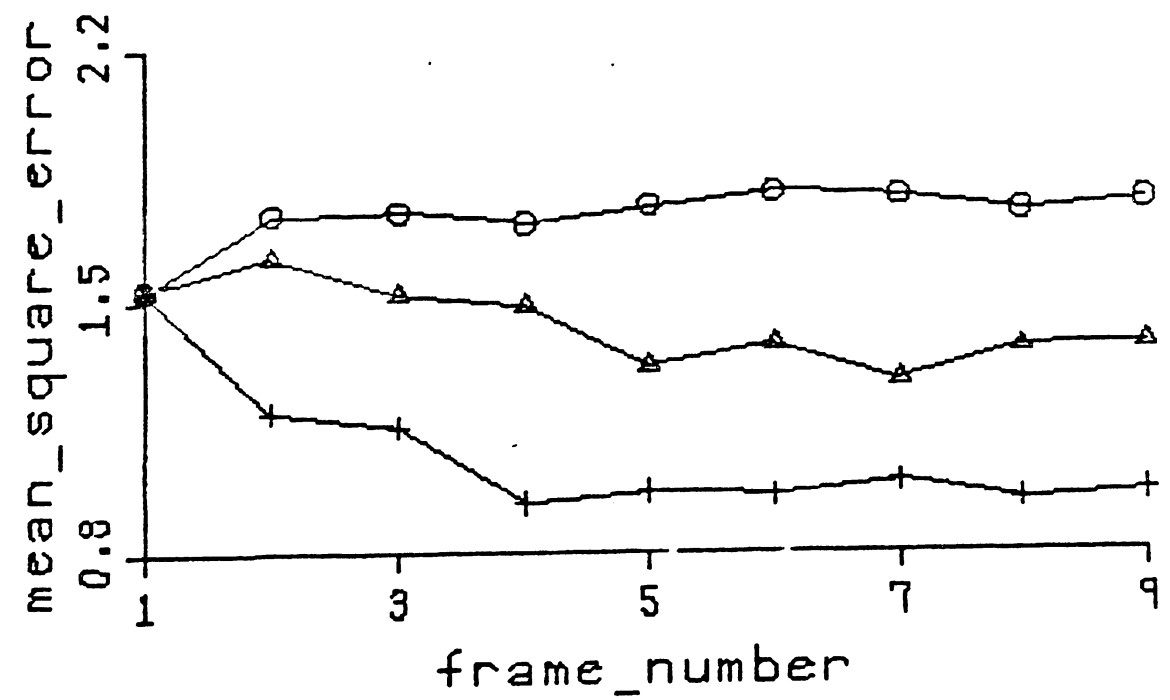


Figure 7.2b: Average mse, velocity = -2, ϵ = 0.010

7.3.2.2 $\epsilon = 0.005$

The average $||\mathbf{d}_e||$ and the average mse in each frame using each of the three prediction techniques are plotted in Figures 7.3a and 7.3b. Using spatial prediction, the variance in the displacement estimates over the moving area was about $\begin{bmatrix} 0.53 \\ 0.46 \end{bmatrix}$ in all nine frames. Using PAMT prediction, the variance dropped from $\begin{bmatrix} 0.53 \\ 0.45 \end{bmatrix}$ in the first frame to $\begin{bmatrix} 0.33 \\ 0.34 \end{bmatrix}$ in the ninth frame. Using mixed prediction, the variance dropped from $\begin{bmatrix} 0.53 \\ 0.45 \end{bmatrix}$ in the first frame to $\begin{bmatrix} 0.34 \\ 0.38 \end{bmatrix}$ in the ninth frame.

The entropy bits required to transmit the nine frames were:

spatial:	21.2 kbits
PAMT:	21.0 kbits
mixed:	19.0 kbits

In concluding this section, one fact is clear: the lowest average mse and bit rate are obtained using mixed prediction, although pure PAMT prediction has the lowest $||\mathbf{d}_e||$. The smallest displacement estimate errors and the smallest average mse are obtained using $\epsilon = 0.010$, but the bit rate is lower using $\epsilon = 0.005$.

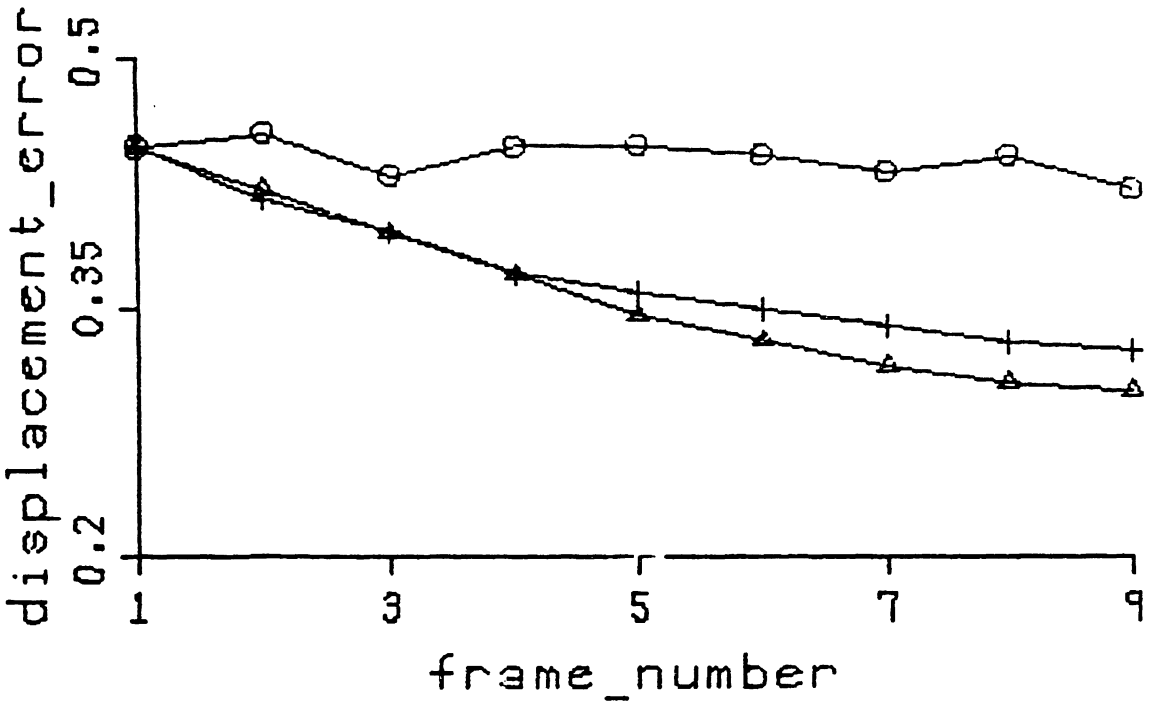


Figure 7.3a: Average $||\mathbf{d}_e||$, velocity = -2, ϵ = 0.005

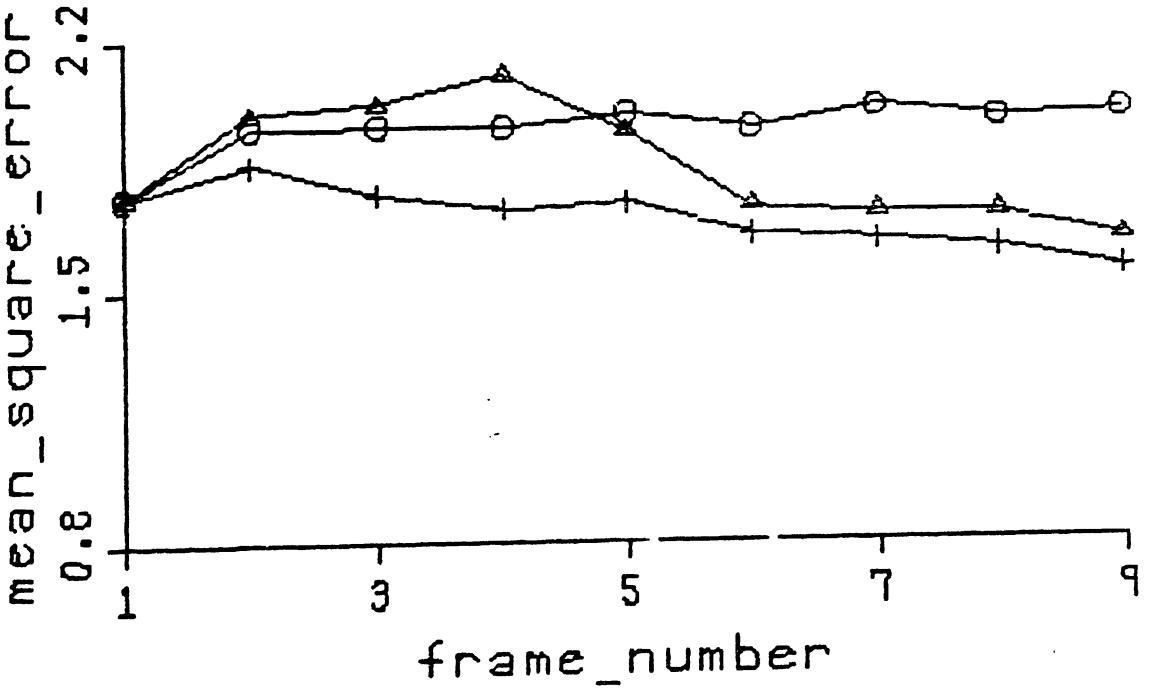


Figure 7.3b: Average mse, velocity = -2, ϵ = 0.005

7.3.3 Velocity of Four Leftward

7.3.3.1 $\epsilon = 0.010$

The average $||\mathbf{d}_\epsilon||$ and the average mse in each frame using each of the three prediction techniques are plotted in Figures 7.4a and 7.4b. Using spatial prediction, the variance in the displacement estimates over the moving area was about $\begin{bmatrix} 3.10 \\ 2.60 \end{bmatrix}$ in all nine frames. Using PAMT prediction, the variance dropped from $\begin{bmatrix} 2.91 \\ 2.55 \end{bmatrix}$ in the first frame to $\begin{bmatrix} 1.03 \\ 2.49 \end{bmatrix}$ in the ninth frame. Using mixed prediction, the variance dropped from $\begin{bmatrix} 2.91 \\ 2.55 \end{bmatrix}$ in the first frame to $\begin{bmatrix} 1.40 \\ 1.79 \end{bmatrix}$ in the ninth frame.

The entropy bits required to transmit the nine frames were:

spatial:	28.1 kbits
PAMT:	32.0 kbits
mixed:	27.5 kbits

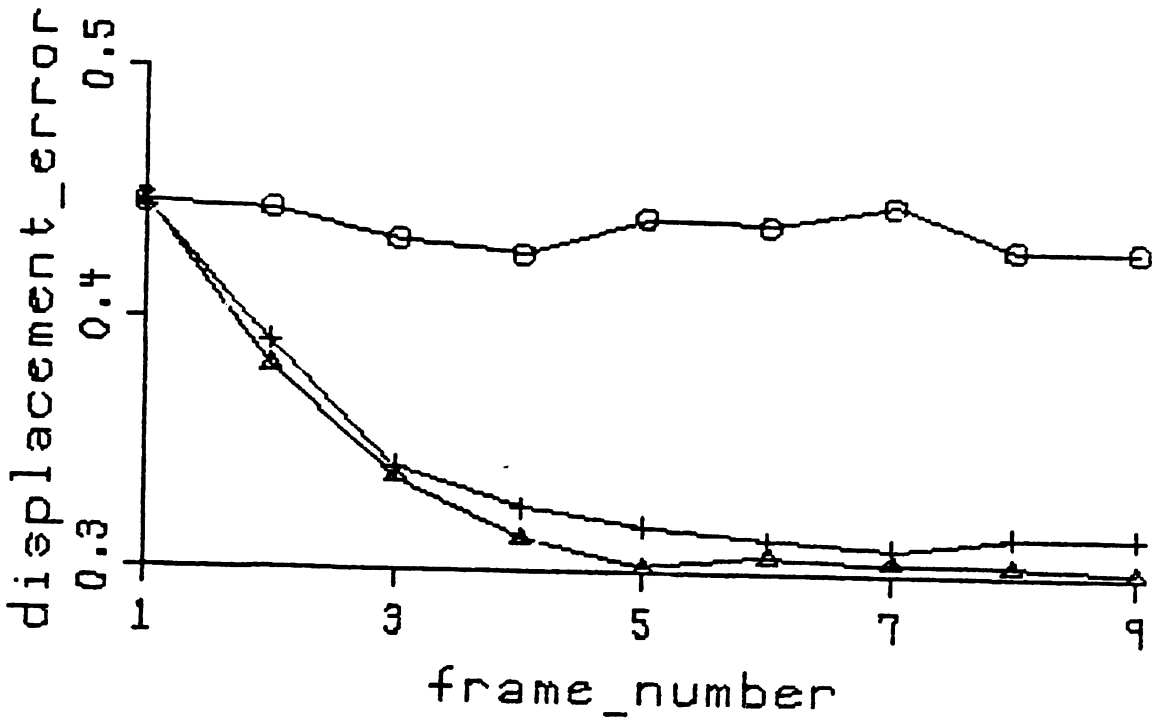


Figure 7.4a: Average $||d_e||$, velocity = -4, ϵ = 0.010

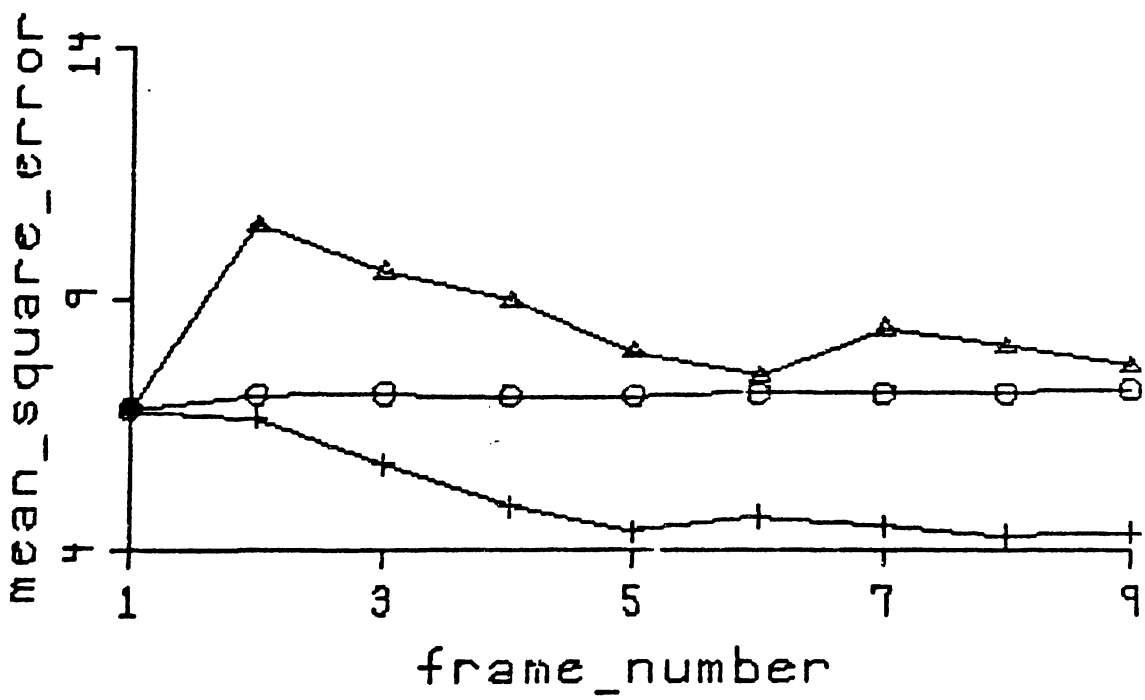


Figure 7.4b: Average mse, velocity = -4, ϵ = 0.010

7.3.3.2 $\epsilon = 0.005$

The average $||\mathbf{d}_e||$ and the average mse in each frame using each of the three prediction techniques are plotted in Figures 7.5a and 7.5b. Using spatial prediction, the variance in the displacement estimates over the moving area was about $\begin{bmatrix} 2.03 \\ 2.13 \end{bmatrix}$ in all nine frames. Using PAMT prediction, the variance dropped from $\begin{bmatrix} 2.03 \\ 2.06 \end{bmatrix}$ in the first frame to $\begin{bmatrix} 0.43 \\ 2.15 \end{bmatrix}$ in the ninth frame. Using mixed prediction, the variance dropped from $\begin{bmatrix} 2.03 \\ 2.06 \end{bmatrix}$ in the first frame to $\begin{bmatrix} 0.83 \\ 1.81 \end{bmatrix}$ in the ninth frame.

The entropy bits required to transmit the nine frames were:

spatial:	27.9 kbits
PAMT:	30.2 kbits
mixed:	28.2 kbits

In concluding the presentation of the results from this set of six simulation runs, it is not clear which ϵ yields the lowest bit rates, but $\epsilon = 0.010$ clearly yields the lowest average mse and lowest average displacement errors. Pure PAMT prediction again yields the lowest average displacement error.

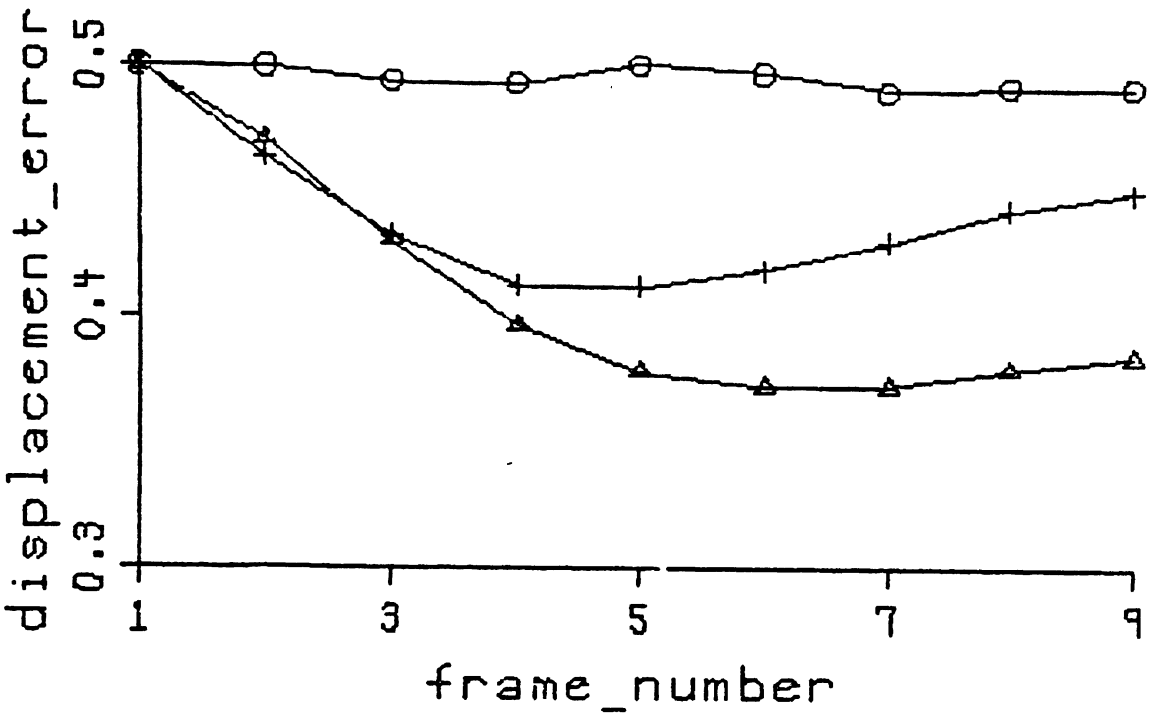


Figure 7.5a: Average $||d_e||$, velocity = -4, ϵ = 0.005

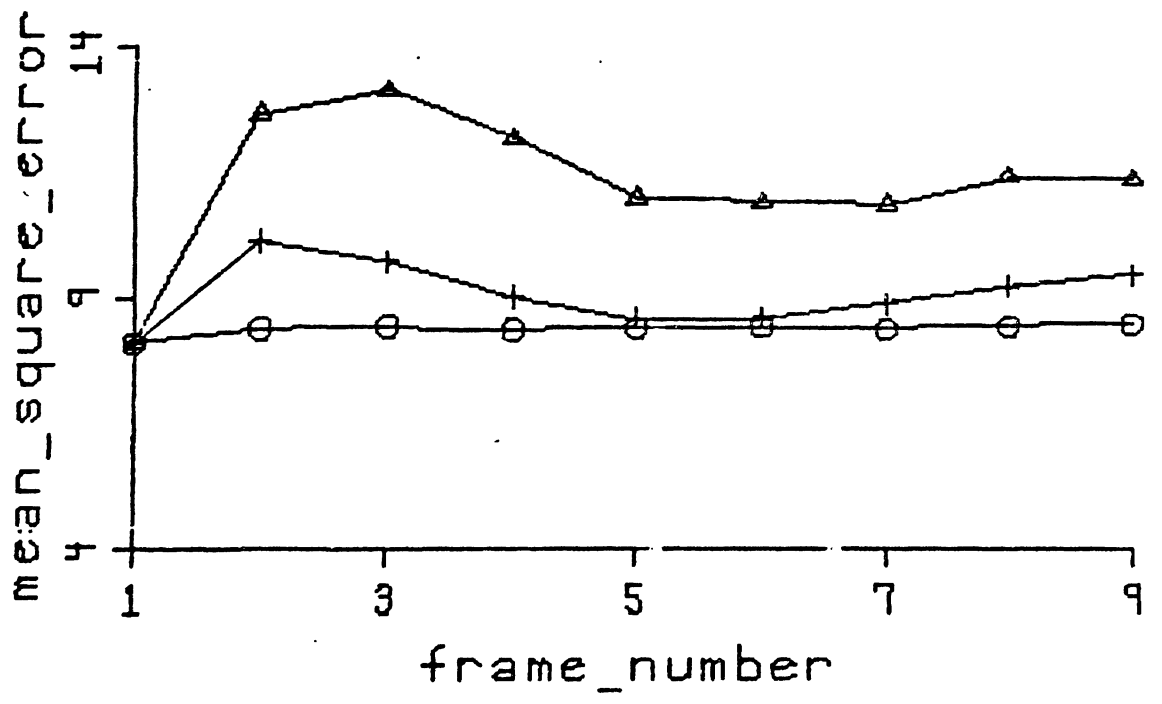


Figure 7.5b: Average mse, velocity = -4, ϵ = 0.005

7.3.5.2 $\epsilon = 0.005$

The average $||\mathbf{d}_e||$ and the average mse in each frame using each of the three prediction techniques are plotted in Figures 7.9a and 7.9b. Using spatial prediction, the variance in the displacement estimates over the moving area was about $\begin{bmatrix} 5.77 \\ 7.96 \end{bmatrix}$ in all nine frames. Using PAMT prediction, the variance dropped from $\begin{bmatrix} 5.82 \\ 7.88 \end{bmatrix}$ in the first frame to $\begin{bmatrix} 0.74 \\ 5.16 \end{bmatrix}$ in the ninth frame. Using mixed prediction, the variance dropped from $\begin{bmatrix} 5.82 \\ 7.88 \end{bmatrix}$ in the first frame to $\begin{bmatrix} 1.99 \\ 3.50 \end{bmatrix}$ in the ninth frame.

The entropy bits required to transmit the nine frames were:

spatial:	28.3 kbits
PAMT:	38.4 kbits
mixed:	31.8 kbits

In concluding this section, it should be noted how much better results are obtained using the larger ϵ. The average $||\mathbf{d}_e||$ and the average mse are much lower with $\epsilon = 0.010$; also two of the three algorithms yield their lower entropy with $\epsilon = 0.010$.

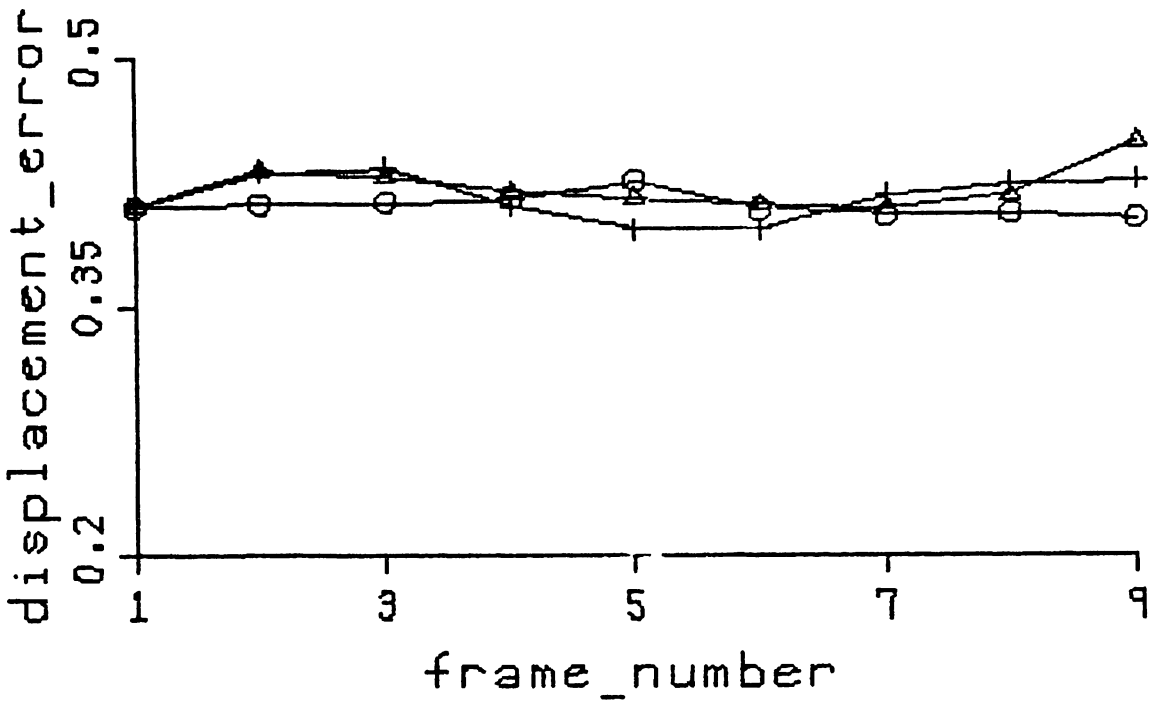


Figure 7.9a: Average $||d_e||$, velocity $= 7$, $\epsilon = 0.005$

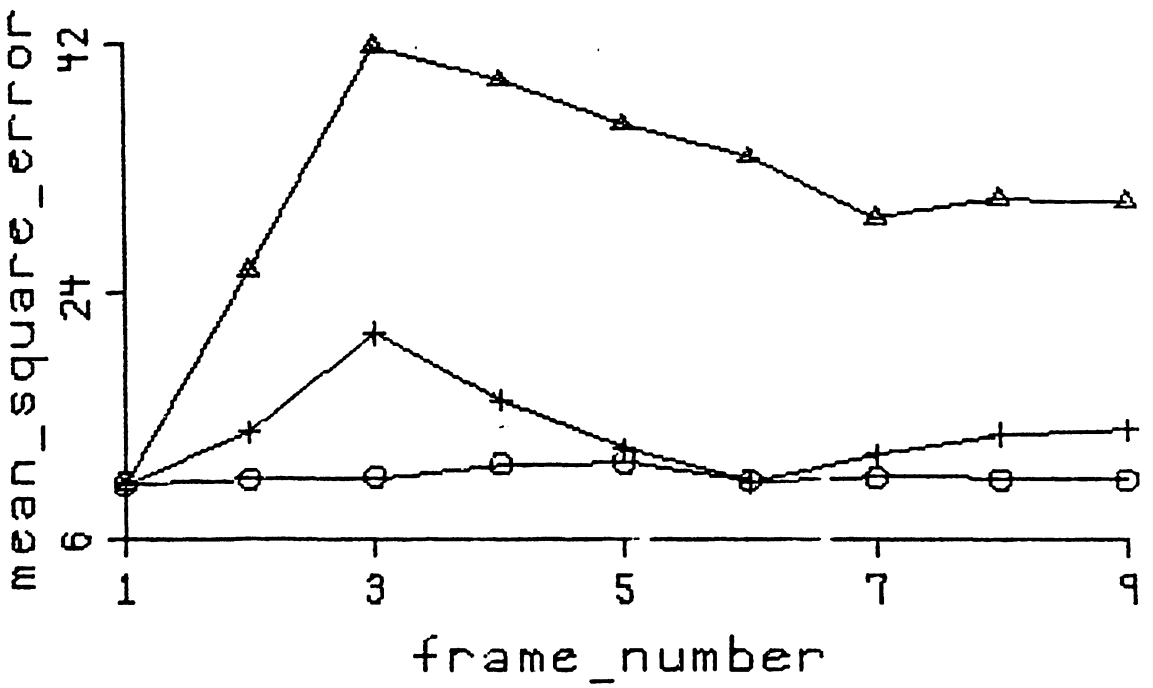


Figure 7.9b: Average mse, velocity $= 7$, $\epsilon = 0.005$

7.3.6 Velocity of Four Rightward

7.3.6.1 $\epsilon = 0.010$

The average $||\mathbf{d}_e||$ and the average mse in each frame using each of the three prediction techniques are plotted in Figures 7.10a and 7.10b. Using spatial prediction, the variance in the displacement estimates over the moving area was about $\begin{bmatrix} 2.51 \\ 2.57 \end{bmatrix}$ in all nine frames. Using PAMT prediction, the variance dropped from $\begin{bmatrix} 2.61 \\ 2.56 \end{bmatrix}$ in the first frame to $\begin{bmatrix} 1.22 \\ 1.35 \end{bmatrix}$ in the ninth frame. Using mixed prediction, the variance dropped from $\begin{bmatrix} 2.61 \\ 2.56 \end{bmatrix}$ in the first frame to $\begin{bmatrix} 2.06 \\ 2.03 \end{bmatrix}$ in the ninth frame.

The entropy bits required to transmit the nine frames were:

spatial:	24.4 kbits
PAMT:	24.6 kbits
mixed:	20.0 kbits

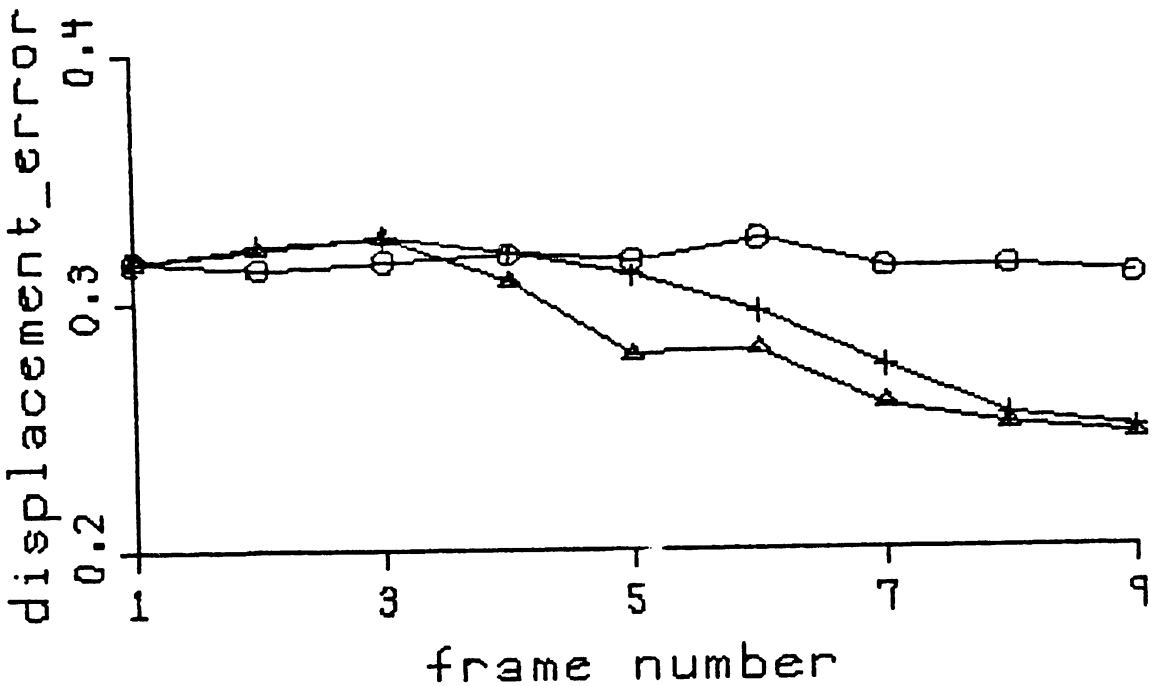


Figure 7.10a: Average $||\mathbf{d}_e||$, velocity $= 4$, $\epsilon = 0.010$

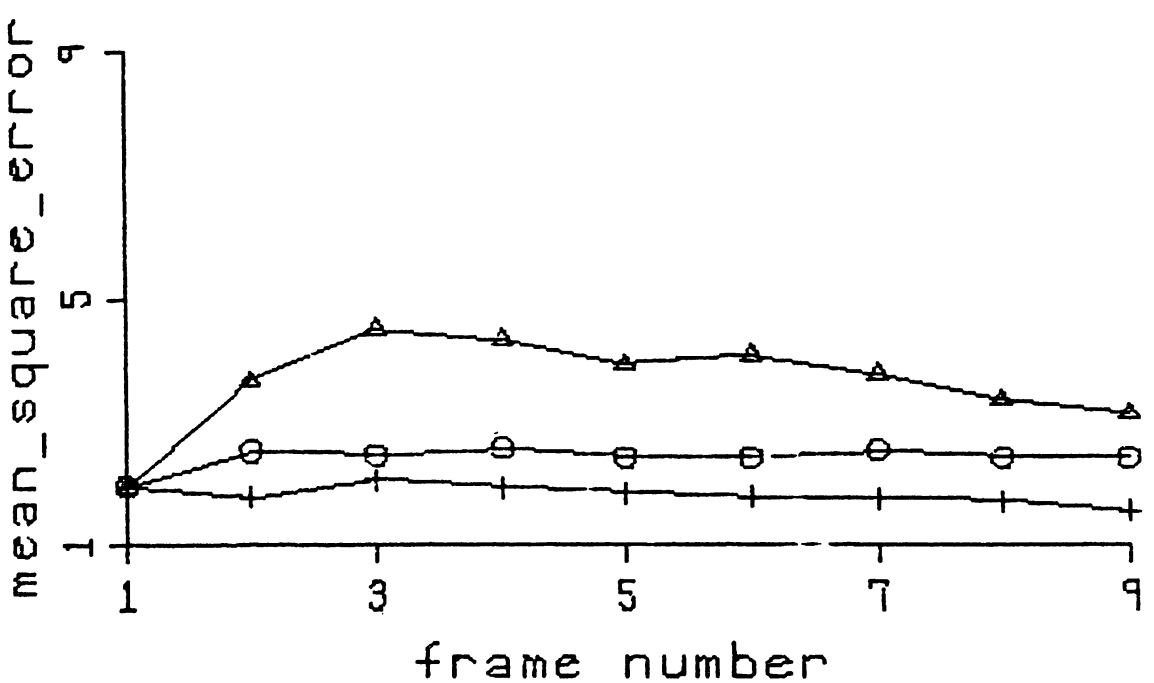


Figure 7.10b: Average mse, velocity $= 4$, $\epsilon = 0.010$

7.3.6.2 $\epsilon = 0.005$

The average $||\mathbf{d}_\epsilon||$ and the average mse in each frame using each of the three prediction techniques are plotted in Figures 7.11a and 7.11b. Using spatial prediction, the variance in the displacement estimates over the moving area was about $\begin{bmatrix} 2.04 \\ 2.11 \end{bmatrix}$ in all nine frames. Using PAMT prediction, the variance dropped from $\begin{bmatrix} 2.08 \\ 2.15 \end{bmatrix}$ in the first frame to $\begin{bmatrix} 0.54 \\ 1.26 \end{bmatrix}$ in the ninth frame. Using mixed prediction, the variance dropped from $\begin{bmatrix} 2.08 \\ 2.15 \end{bmatrix}$ in the first frame to $\begin{bmatrix} 1.48 \\ 1.34 \end{bmatrix}$ in the ninth frame.

The entropy bits required to transmit the nine frames were:

spatial:	22.7 kbits
PAMT:	26.3 kbits
mixed:	21.1 kbits

In concluding this section, it is noteworthy that mixed prediction consistently yielded the lowest average mse and the lowest entropy. Pure PAMT prediction yielded the lowest variances in the displacement estimates.

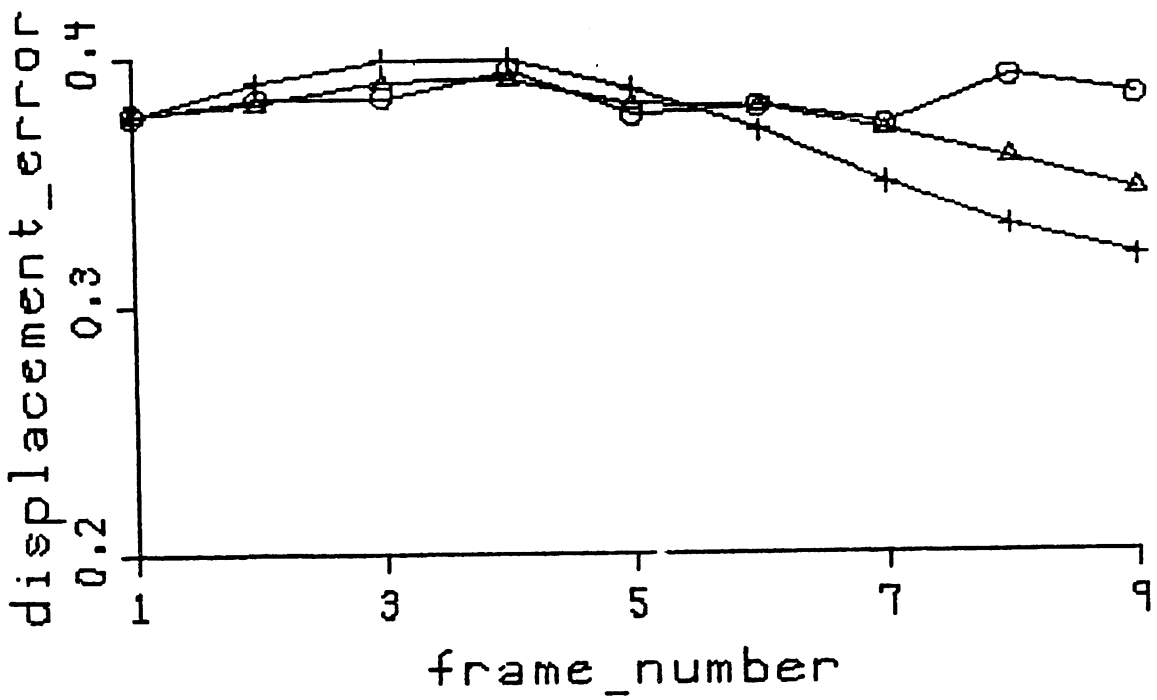


Figure 7.11a: Average $||\mathbf{d}_e||$, velocity $= 4$, $\epsilon = 0.005$

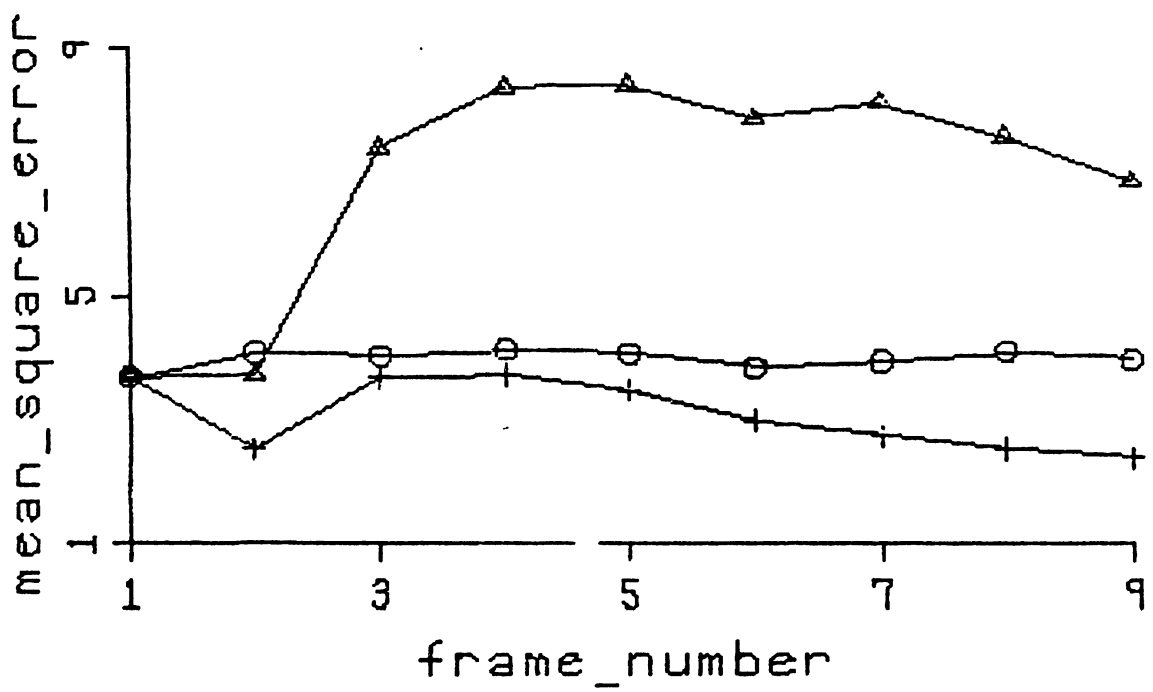


Figure 7.11b: Average mse, velocity $= 4$, $\epsilon = 0.005$

7.3.7 Velocity of Two Rightward

7.3.7.1 $\epsilon = 0.010$

The average $||\mathbf{d}_e||$ and the average mse in each frame using each of the three prediction techniques are plotted in Figures 7.12a and 7.12b. Using spatial prediction, the variance in the displacement estimates over the moving area was about $\begin{bmatrix}0.71\\0.55\end{bmatrix}$ in all nine frames. Using PAMT prediction, the variance dropped from $\begin{bmatrix}0.71\\0.55\end{bmatrix}$ in the first frame to $\begin{bmatrix}0.53\\0.35\end{bmatrix}$ in the ninth frame. Using mixed prediction, the variance dropped from $\begin{bmatrix}0.71\\0.55\end{bmatrix}$ in the first frame to $\begin{bmatrix}0.58\\0.52\end{bmatrix}$ in the ninth frame.

The entropy bits required to transmit the nine frames were:

spatial:	20.6 kbits
PAMT:	18.4 kbits
mixed:	15.3 kbits

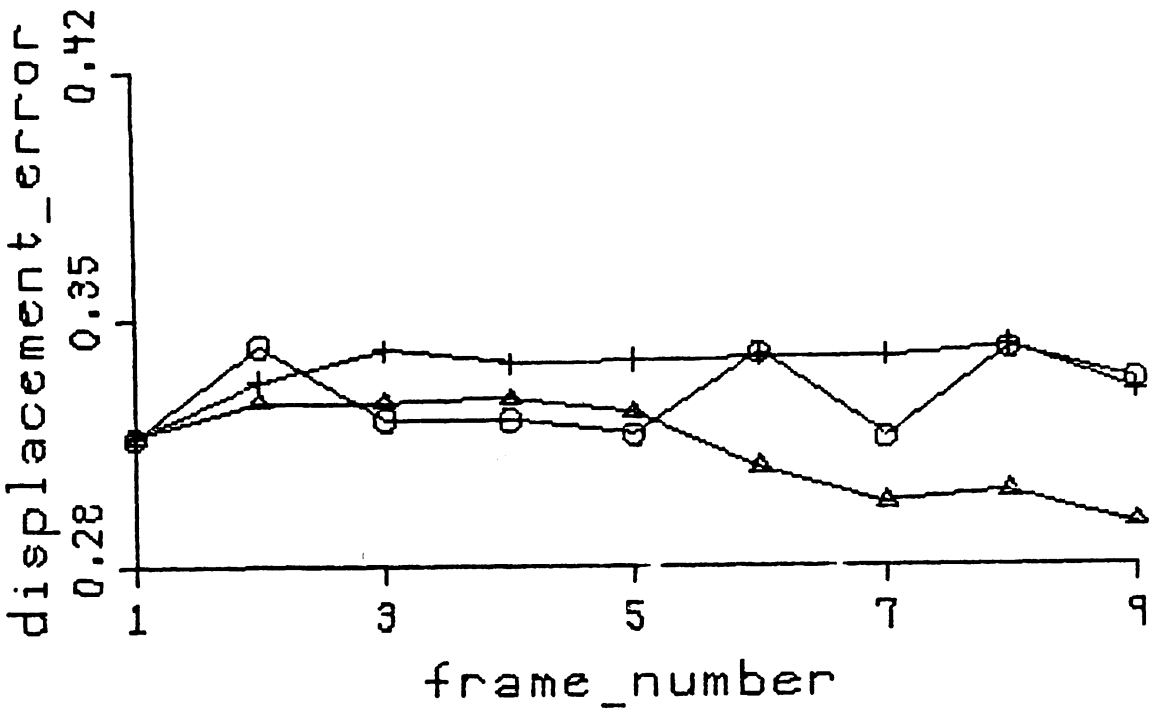


Figure 7.12a: Average $||\mathbf{d}_e||$, velocity $= 2$, $\epsilon = 0.010$

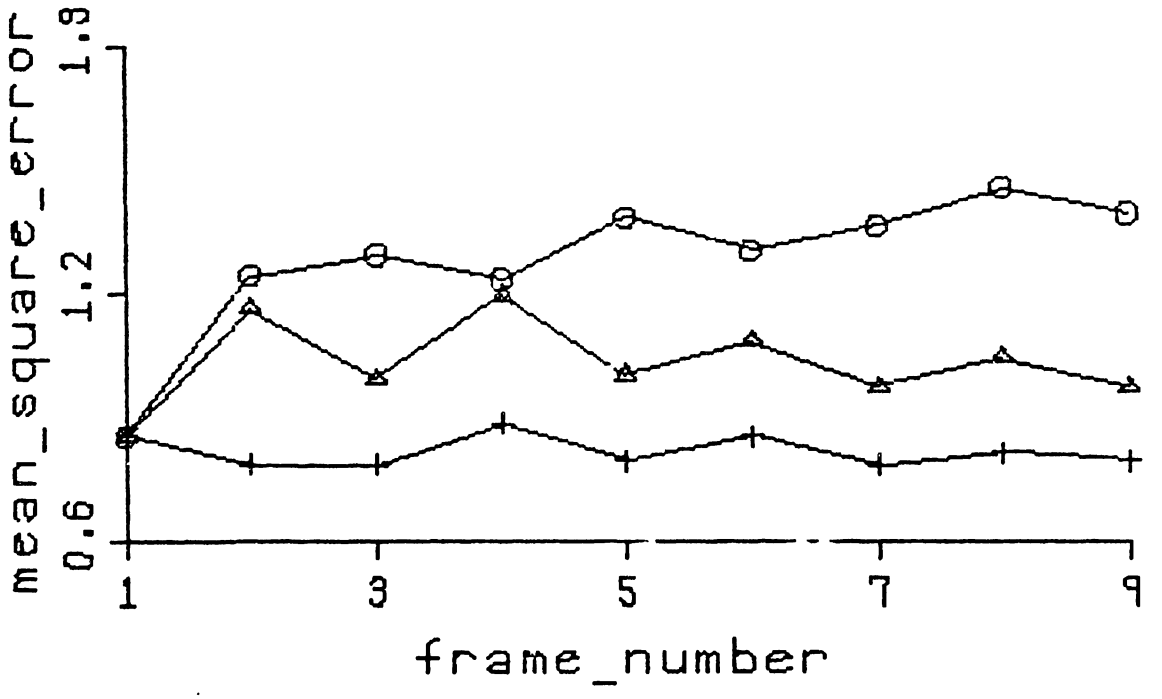


Figure 7.12b: Average mse, velocity $= 2$, $\epsilon = 0.010$

7.3.7.2 $\epsilon = 0.005$

The average $||\mathbf{d}_e||$ and the average mse in each frame using each of the three prediction techniques are plotted in figures 7.12a and 7.12b. Using spatial prediction, the variance in the displacement estimates over the moving area was about $\begin{bmatrix}0.57\\0.44\end{bmatrix}$ in all nine frames. Using PAMT prediction, the variance dropped from $\begin{bmatrix}0.57\\0.44\end{bmatrix}$ in the first frame to $\begin{bmatrix}0.34\\0.26\end{bmatrix}$ in the ninth frame. Using mixed prediction, the variance dropped from $\begin{bmatrix}0.57\\0.44\end{bmatrix}$ in the first frame to $\begin{bmatrix}0.44\\0.41\end{bmatrix}$ in the ninth frame.

The entropy bits required to transmit the nine frames were:

spatial:	17.7 kbits
PAMT:	15.6 kbits
mixed:	14.1 kbits

In concluding this section, note that displacement estimate errors and the average mean square errors are less using the larger ϵ, but the variances in the displacement errors and the entropy are less with the smaller ϵ. In comparing prediction schemes, the mixed consistently gives the lowest mse and the lowest entropy of the three prediction schemes. The pure PAMT scheme gives the next lowest mse and entropy and the lowest average $||\mathbf{d}_e||$.

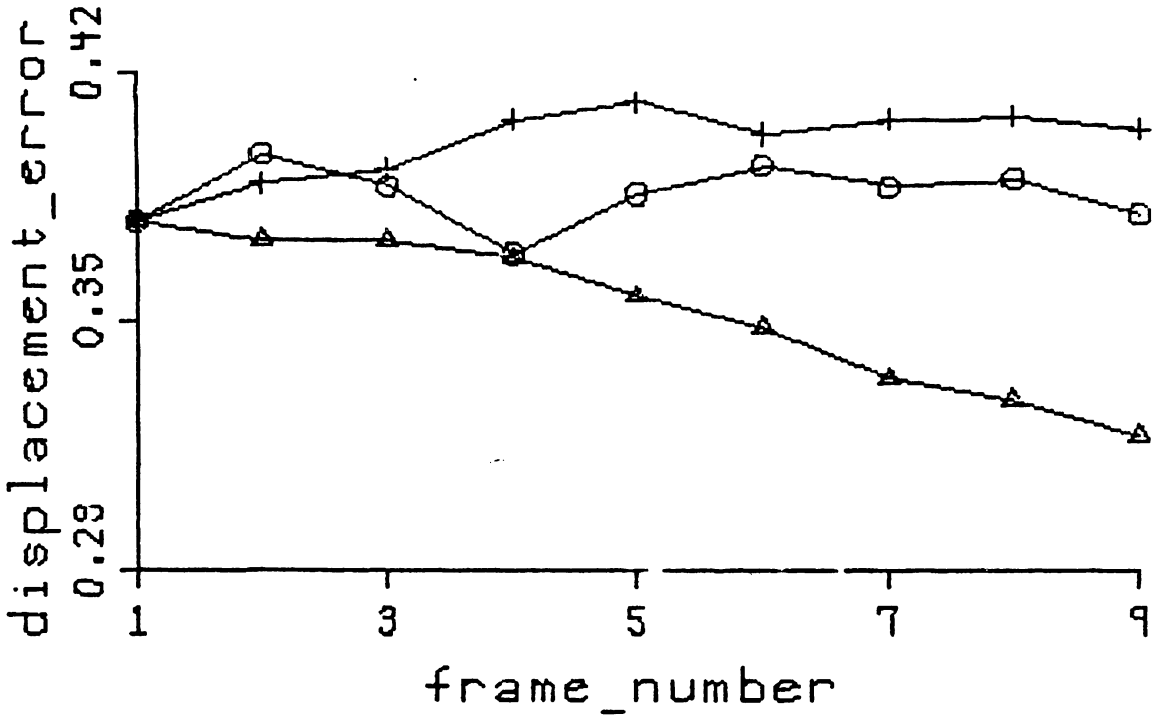


Figure 7.13a: Average $||\mathbf{d}_e||$, velocity $= 2$, $\epsilon = 0.005$

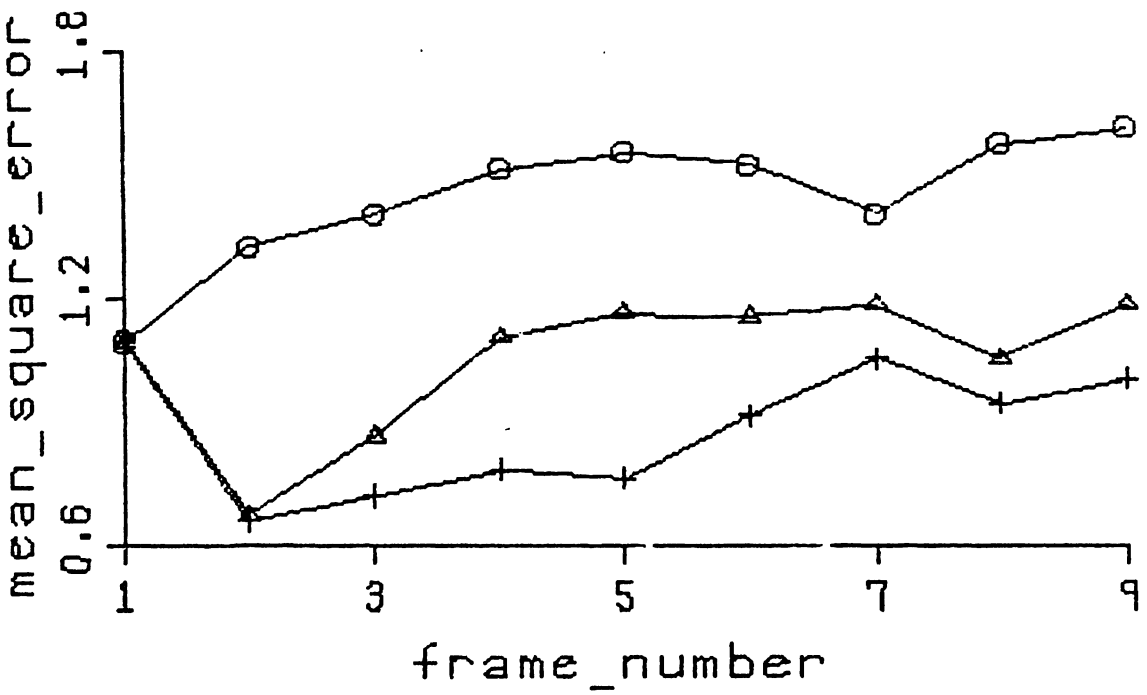


Figure 7.13b: Average mse, velocity $= 2$, $\epsilon = 0.005$

7.3.8 Conclusions and Summary of Synthetic Simulation Results

Before concluding this section, a summary of how the entropy, the average mse, and the average $||\mathbf{d}_e||$ vary with the prediction technique, the value of ϵ, the magnitude of the motion, and the direction of the motion is in order. The entropy values for each ϵ, prediction scheme, and velocity are tabulated in Table 7.1. The number of bits (in kbits) are entered at the ϵ value which gave the lower entropy for each motion prediction scheme at each velocity. The asterisked numbers indicate which prediction technique and ϵ value gave the lowest entropy for each velocity.

Entropy Minimums							
ϵ	prediction scheme	velocity					
		-2	-4	-7	7	4	2
0.010	spatial						
	PAMT				37.9	24.6	
	mixed		*27.5		30.9	*20.0	
0.005	spatial	21.2	27.9	*34.5	*28.3	22.7	17.7
	PAMT	21.0	30.2	40.6			15.6
	mixed	*19.0		38.1			*14.1

Table 7.1: Entropy Minimums for Synthetic Sequences

The results are:

1) Mixed prediction gives the lowest entropy at the lower two velocities, while spatial prediction yields the lowest entropy at the largest velocity.

2) The smaller ϵ value gives the lowest entropy for $||\mathbf{v}||$ of 2 and 7, while the larger ϵ gives the lowest entropy for $||\mathbf{v}||$ of 4.

3) The total entropy is proportional to the magnitude of the motion; i.e., the larger the motion, the larger the entropy.

4) The entropy is lower for rightward motion than leftward motion.

Result (4) is expected since the gradients are evaluated in the previous frame when $\hat{\mathbf{d}}$ must be corrected. For rightward motion, the location of the left edge of the moving object in frame t is "above" the moving object in frame t-1. Thus the gradient is non-zero and the value of $\hat{\mathbf{d}}$ can start to be "corrected" when an incorrect $\hat{\mathbf{d}}$ is first encountered. This is not the case for leftward motion.

To summarize the effect of the parameters on the mse:

1) As with the entropy results, mixed prediction gives the best results at the two lower velocities and spatial prediction gives the best results at the highest velocity.

2) The average mse is lower for $\epsilon = 0.010$ than for $\epsilon = 0.005$.

3) The average mse is directly proportional to the magnitude of the motion.

4) The average mse is lower for rightward motion.

In other words, the minimum average mse's occur with velocity of $\begin{bmatrix} 0 \\ 2 \end{bmatrix}$ and

$\epsilon = 0.010$.

With respect to $||\mathbf{d}_\epsilon||$, the average $||\mathbf{d}_\epsilon||$ within the moving object is:

1) lowest for PAMT prediction.

2) lower for $\epsilon = 0.010$ than $\epsilon = 0.005$.

3) not proportional to the magnitude of the motion.

4) lower for rightward motion than leftward motion, as is to be expected.

Some facts that should be noted in evaluating these results are:

1) The moving object has smooth, rounded edges and varying intensity over its interior. The results may differ for objects with sharp, straight edges and/or a constant intensity interior.

2) The difference of consecutive intensity errors is transmitted. Transmitting the intensity errors directly may affect the relative results (see Section 7.4.4).

3) The number of frames over which the velocity is assumed constant was arbitrarily chosen to be ten (1/3 of a second). This is probably a large value; five frames might be more reasonable. However, the relative results would be almost the same as can be seen by considering only the first half of the plots in Figures 7.2 - 7.13. Processing N frames yields the results for any interval N frames or less.

7.4 Real Image Sequences

7.4.1 Introduction

This section contains the results of simulating the MC image sequence compression algorithms on real image sequences. These simulations were performed to show

1) the information compression of all three MC algorithms,
2) the increased compression obtained by using the convergence analysis,
3) the increased compression of the proposed algorithm, and
4) the increased compression with zeroth-order entropy information.

Three 60-frame (two seconds) sequences were used. A pair of pictures for each sequence is shown in Figure 7.14. The percent of interframe motion for each sequence is plotted in Figure 7.15. The $E\{\nabla_z I\}$ must be determined experimently since the motion-correction algorithm is not used at every pel and is ϵ-dependent. The minimum and maximum values of $E\{\nabla_z I\}$ (where the expected value is taken within each frame) will be noted for each run. In all simulation runs the displacement correction algorithm was allowed to iterate at most once at each pel.

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.

In the following tables, which summarize the simulation results for real image sequences, there are six columns. The first column indicates the ϵ and clip values

used. The second column indicates the maximum entropy bits per frame and the average entropy bits per frame. 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. 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 is a strong measure of the ability to implement the algorithm in real-time, since the motion-correction equation (Equation (4.8)) 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. The sixth column indicates the maximum and minimum mse for the corrected frame, that is the picture which is actually viewed at the receiver.

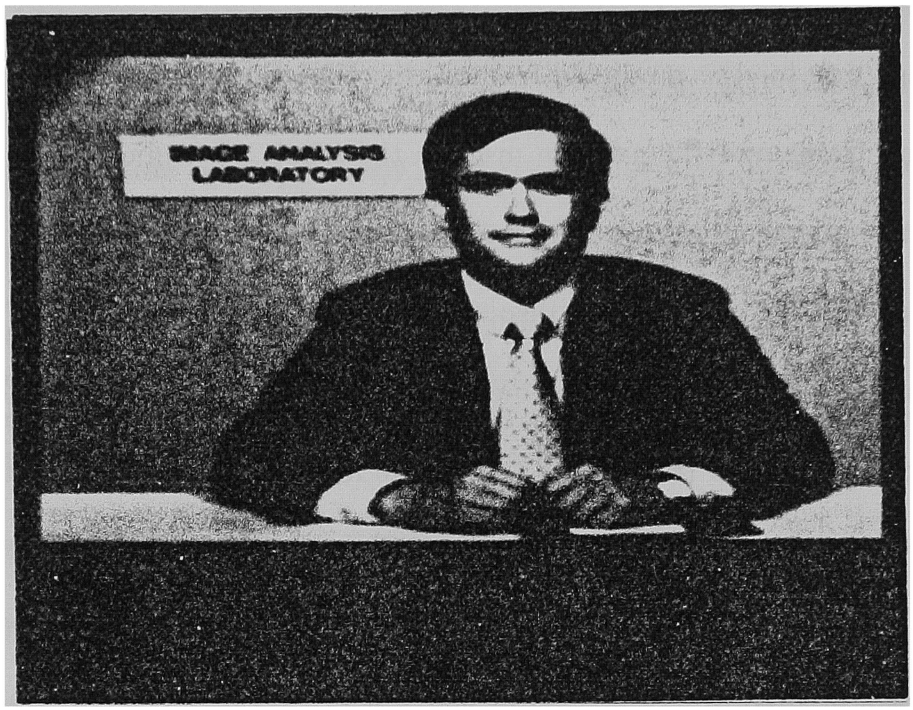


Figure 7.14a: Bobsjob, frame 0

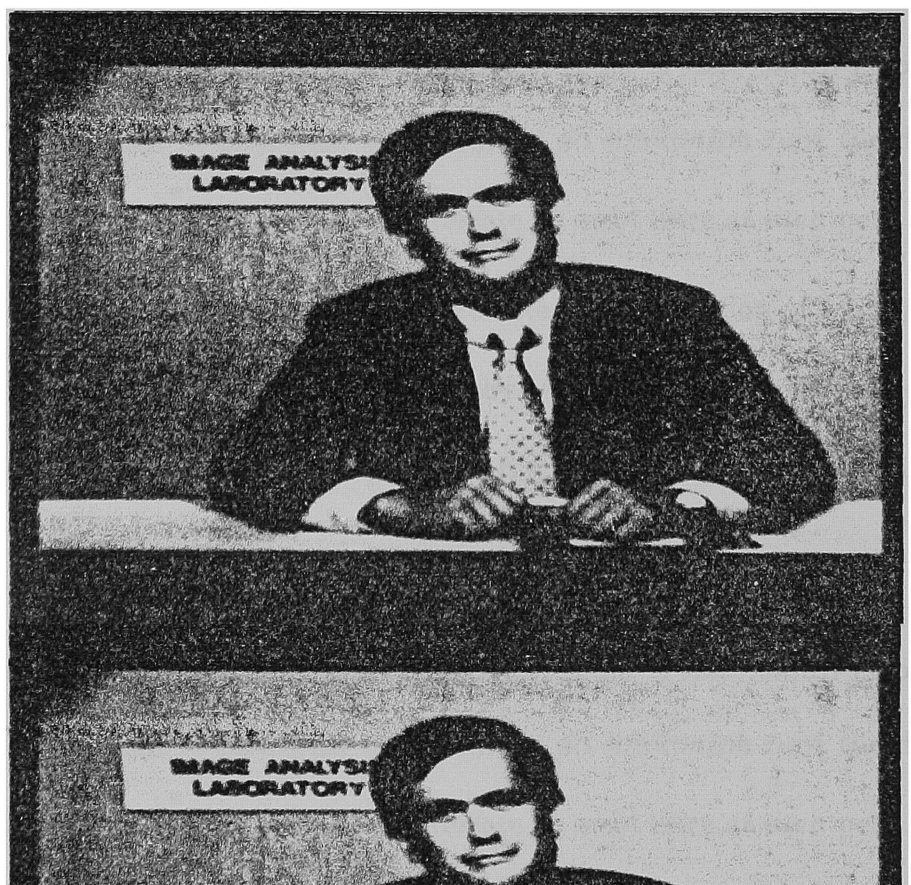


Figure 7.14c: Map, frame 0

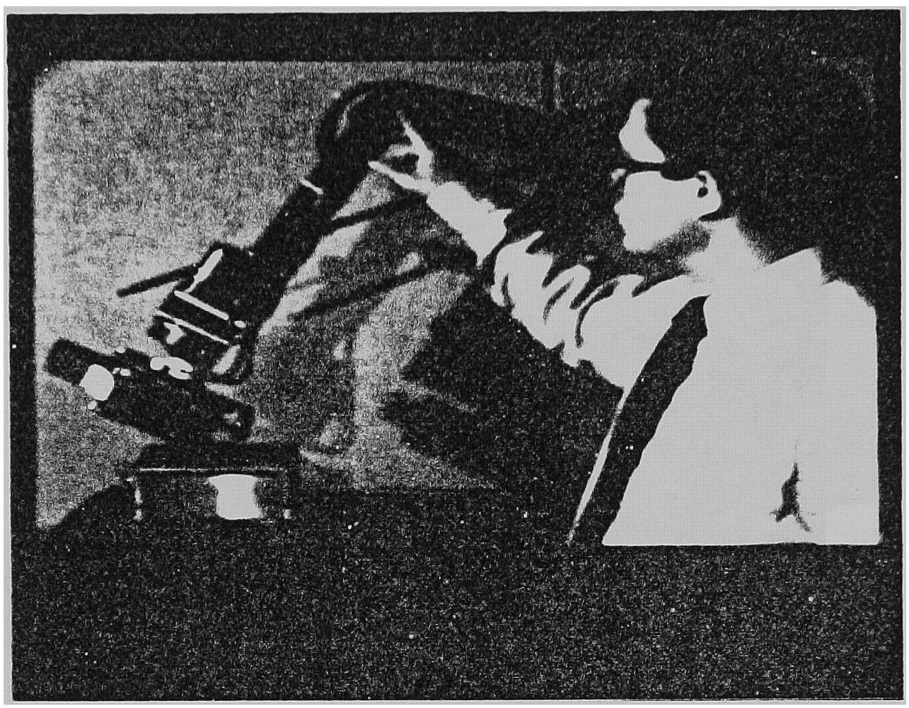
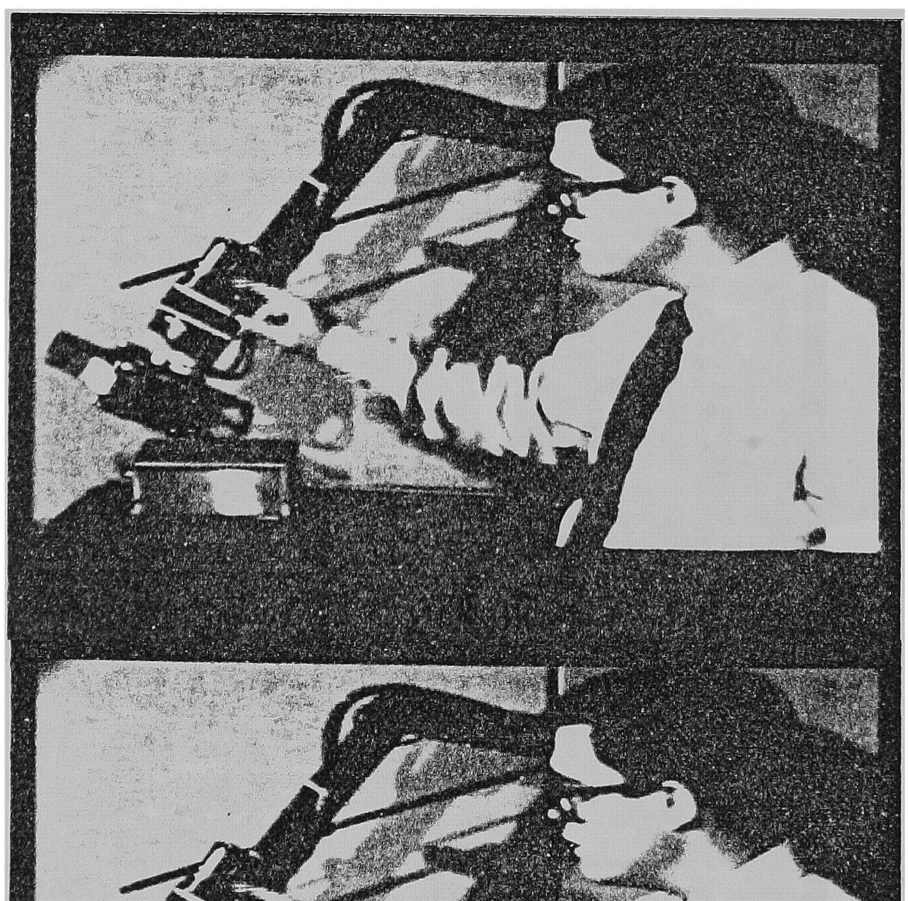


Figure 7.14e: Robot, frame 0



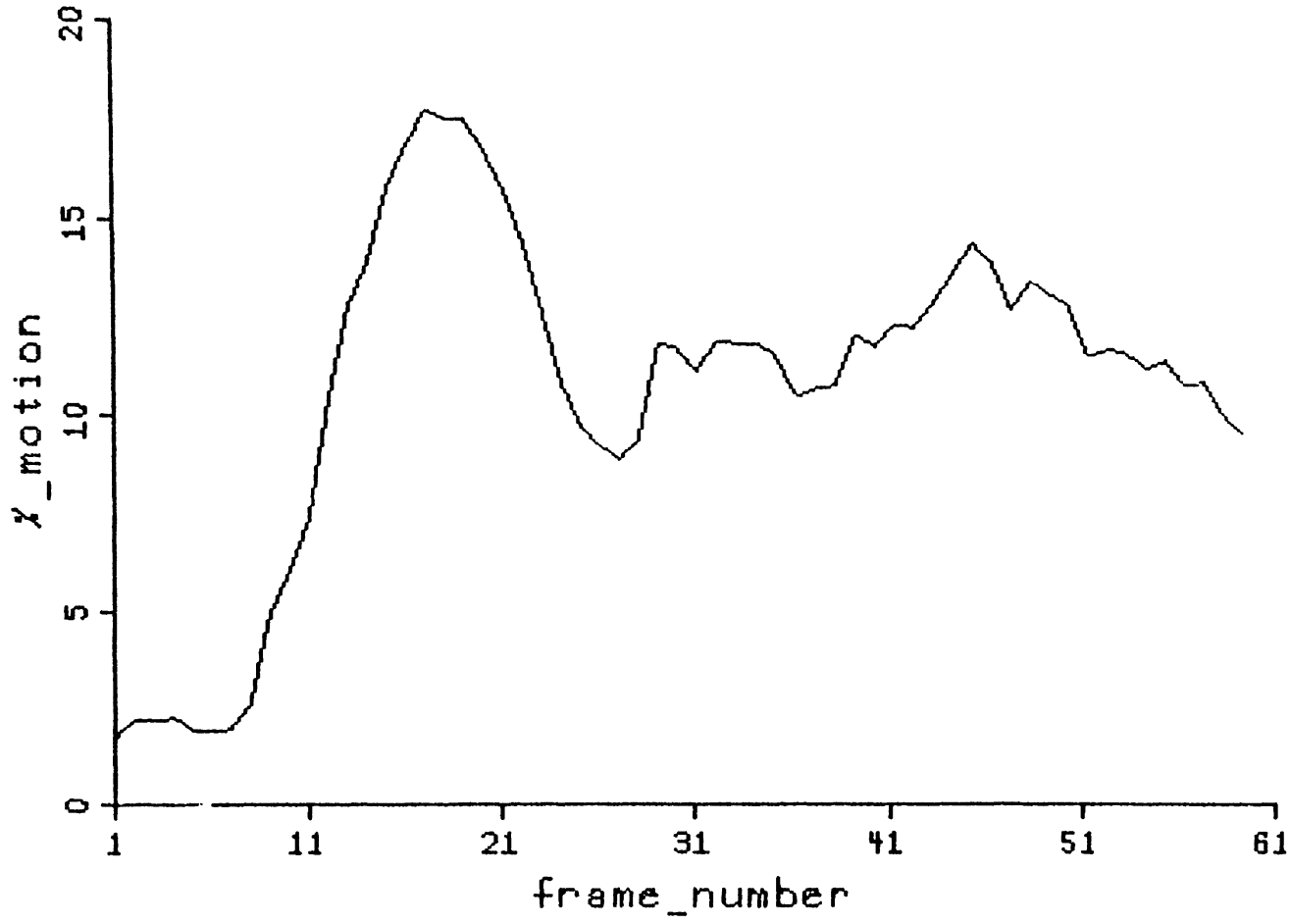


Figure 7.15a: Percent Interframe Motion, bobsjob sequence

Figure 7.15b: Percent Interframe Motion, map sequence

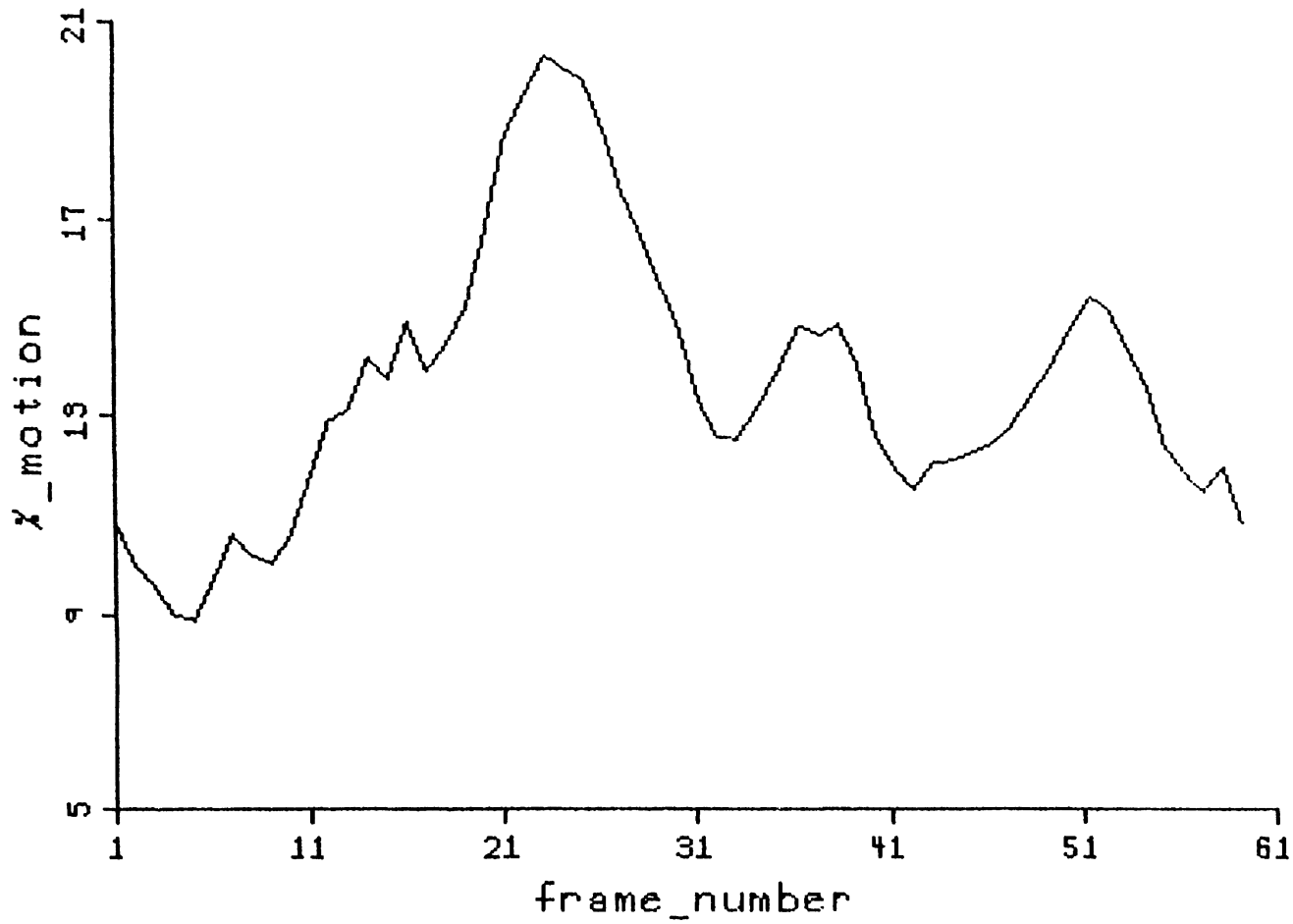


Figure 7.15c: Percent Interframe Motion, robot sequence

7.4.2 Convergence Analysis Simulations

The spatial prediction technique and PAMT prediction technique were both applied to the bobsjob sequence using various ϵ values and various clip values. The simulation results using spatial prediction are tabulated in Table 7.2. The simulation results using PAMT prediction are tabulated in Table 7.3.

An ϵ value of 0.0010 and a clip value of 0.0625 were used in earlier pel-recursive research [51]. Other ϵ/clip values that have been used in reported simulations are 0.0078/0.2 and 0.5/0.08 [53]. There is, however, no analytical reason for these values; they were determined to be the best for particular sequences by trial and error. The analysis in Chapter 5 indicated that $\hat{\mathbf{d}}$ should converge to $\mathbf{d}$ within the moving area without clipping the update values (see Equation (5.3)) if the assumptions are valid and the constraints on ϵ are met. Some researchers have tried to justify clipping the update term [84,85]. 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 Tables 7.2 and 7.3 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 in Chapter 5), 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).

Table 7.2: Spatial Prediction of Bobsjob Sequence

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 7.3: PAMT Prediction of Bobsjob Sequence

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	121345/ 90064	103.94/31.48	23983/15823	219.4/ 840.0	0.27/1.37
0.00200/0.0625	120259/ 90208	104.26/31.59	23840/15845	214.9/ 787.3	0.27/1.37
0.00200/1.0000	124979/ 93785	93.53/26.12	23698/16522	241.4/ 829.1	0.34/1.44
0.01000/1.0000	133985/ 99482	93.82/26.69	24358/18054	273.4/ 597.9	0.28/1.39
0.00200/ n.c.	130769/ 98208	105.77/33.50	24655/17703	275.1/ 549.0	0.32/1.49
0.00100/ n.c.	122626/ 92470	92.15/27.97	23594/15995	227.3/1004.1	0.28/1.41
0.00050/ n.c.	119588/ 90233	95.52/28.63	23489/15751	194.7/ 724.9	0.28/1.38
0.00025/ n.c.	121470/ 89640	101.77/30.85	24010/15794	210.3/ 699.9	0.26/1.33
0.00010/ n.c.	123703/ 90182	106.57/32.79	24428/15894	208.9/ 602.1	0.26/1.28

n.c. -- no clip

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.)

For PAMT prediction and the ϵ values used, the minimum total bit rate was obtained with $\epsilon = 2.5 \times 10^{-4}$, the minimum average predicted mse was obtained with $\epsilon = 10^{-5}$, and the minimum total number of unpredicted pels was obtained when $\epsilon = 5 \times 10^{-4}$. Note that these three measures were all minimized with ϵ values within a factor of four of each other with PAMT prediction.

The sum of the average values of $(\nabla_x I)^2$ and $(\nabla_y I)^2$ varied from 194.7 in frame 21 when using PAMT prediction to 1004.1 in frame 1 when using either technique. The average value of $\{(\nabla_x I)^2 + (\nabla_y I)^2\}$ was about 300 in all runs. For $\hat{\mathbf{d}}$ 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 (5.19). Equation (5.53) 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 variable 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.

7.4.3 Comparison of Prediction Techniques

In this section, the performance of the three different prediction schemes are indicated. All three prediction schemes (spatial, PAMT, and mixed) were applied to all three sequences (bobsjob, map, and robot). The results for the bobsjob sequence are in Tables 7.2, 7.3, and 7.4; the results for the map sequence are in Table 7.5, and the results for the robot sequence are in Table 7.6.

Instead of developing a single cost function involving the bit-rate, the picture quality, and the implementation complexity, the ϵ value was varied and the minimum value for the various measures were determined.

For the bobsjob sequence, the minimum bit rate for all three prediction schemes was found. These minimum bit rates occur at different ϵ values and differ by about 2%. The minimum number of unpredicted pels for each prediction scheme was also found. These minimum values once again vary by about 2%, but occur at different ϵ values than the entropy minimums. The results are similar for average value of the gradient and the corrected mse: the minimum values for each prediction scheme vary only slightly and occur at different ϵ values.

For the simulation runs on the map sequence only two ϵ values were used. As with the bobsjob sequence all three prediction schemes yield similar results. The largest difference is in the average value of the gradient within a frame. Using the values obtained with spatial prediction as a benchmark, the average gradient is 11-14% less using PAMT prediction and 6-12% less using mixed prediction.

For the simulation runs on the robot sequence only one ϵ value was used. No significant differences occurred in the results for the three prediction schemes.

The failure of the PAMT (and mixed) prediction scheme to show much improvement over spatial prediction is probably most attributable to the crude motion correction scheme. In the present motion correction scheme, $\hat{\mathbf{d}}$, may not converge to $\mathbf{d}$ until most of the moving object has been processed. The $\hat{\mathbf{d}}^i$ values in the first part of the moving object may be greatly in error. Since the improvement of the PAMT scheme occurs by utilizing the $\hat{\mathbf{d}}$ information in the previous frame, if the $\hat{\mathbf{d}}^N$ values in the previous frame are erronous, little (if any) advantage is gained by using them. In other words, with a better motion-correction scheme, better performance would be obtained with the PAMT technique.

Table 7.4: Mixed Prediction of Bobsjob Sequence

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	118537/88916	86.87/22.81	23350/15835	220.2/1004.1	0.32/1.42
0.00050	117120/87990	91.43/24.73	23743/15791	206.0/ 724.9	0.31/1.38
0.00025	116263/87423	98.08/27.15	23773/15880	214.1/ 699.9	0.26/1.20

Table 7.5: Map Sequence Simulation Results

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
spatial					
0.00100	162282/121997	67.95/17.41	29469/18726	91.2/210.2	1.14/1.70
0.00050	160078/119766	72.58/18.95	28898/18296	89.1/214.8	1.09/1.65
PAMT					
0.00100	160085/121466	72.64/19.71	27930/18649	78.7/210.2	1.16/1.70
0.00050	159632/120288	72.74/20.07	28298/18420	79.7/214.8	1.11/1.65
0.00025	161713/120307	74.01/20.55	28616/18422	86.5/230.5	1.09/1.55
mixed					
0.00100	155707/120494	73.03/19.74	27572/19156	80.7/210.3	1.17/1.70
0.00050	155304/119037	72.78/19.95	28006/18486	84.5/214.8	1.09/1.65

Table 7.6: Robot Sequence Simulation Results

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
spatial					
0.0001	139470/110565	49.27/21.58	25953/19061	189.4/439.3	0.84/1.31
PAMT					
0.0001	143474/111397	53.20/23.00	27063/19358	187.1/439.3	0.85/1.33
mixed					
0.0001	141994/110772	51.90/22.49	26807/19211	190.4/439.3	0.86/1.34

Table 7.6: Robot Sequence Simulation Results

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
spatial					
0.0001	139470/110565	49.27/21.58	25953/19061	189.4/439.3	0.84/1.31
PAMT					
0.0001	143474/111397	53.20/23.00	27063/19358	187.1/439.3	0.85/1.33
mixed					
0.0001	141994/110772	51.90/22.49	26807/19211	190.4/439.3	0.86/1.34

7.4.4 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% [51] 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 spatial and PAMT simulations were rerun transmitting the intensity prediction errors directly. The results are in Table 7.7 and 7.8. 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.

Table 7.7: Zeroth-Order Spatial Prediction of Bobsjob Sequence

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 7.8: Zeroth-Order PAMT Prediction of Bobsjob Sequence

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
0.0020	124247/80926	104.69/32.00	21877/14600	309.4/ 836.1	0.00/0.57
0.0010	123015/80119	90.72/27.06	21028/13162	250.1/1268.5	0.00/0.52
0.0005	124700/81267	95.14/27.83	21362/13156	214.3/1124.5	0.00/0.50

7.4.5 Conclusions from Real Sequence Simulations

Application of the analytical model used to prove convergence of the pel-recursive technique gives better results than using the heuristic technique of large ϵ values and clipping the update term in Equation (5.8). Transmitting zeroth-order error information also appears to be advantageous to transmitting first-order error information. Both of these "new" approaches actually reduce the computational requirements and increase the performance.

The simulations using the PAMT motion prediction scheme (and the ones using the mixed scheme) did not give appreciable improvement over spatial prediction. However, there may be some things that could be done to change that. The displacement estimate, $\hat{\mathbf{d}}^i$, converges over the moving area in each frame. If these "correct" estimates could be passed back to the previously-scanned part of the moving object, the motion prediction values projected forward into the next frame would be better estimates. Although this would take more computation in one frame, the improved predictions for the next frame might reduce the average computational requirements over a sequence of frames. Another way to improve the performance of the PAMT scheme would be to develop a better gap bridging scheme. Further results from the simulation runs on the synthetic sequences showed the $\hat{\mathbf{d}}$ convergence when using PAMT prediction to be noisy due to gaps in the predicted motion field.

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 all three sequences used for the simulations herein is lower than the motion in the sequences used in most other motion-compensated codec simulations [51-53,56-58,66-68,74-75,86-88].

2) The frames were not low-pass filtered to blur the edges. The sequences are very high quality sequences with sharp edges (see the examples of edge values in Appendix A).

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 in Section 7.1.

CHAPTER EIGHT

CONCLUSIONS, A SUMMARY, AND A POTENTIAL CIRCUIT DIAGRAM

This dissertation has presented a thorough analysis and simulation of the pel-recursive motion-compensated algorithm. In this chapter the results are summarized and areas needing further work are noted.

(1) An analytical model of the pel-recursive motion prediction technique has been developed. 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.

(2) A realistic model for the edges in captured images was developed and used to show the negligibility of the higher order terms in the Taylor series expansion of the intensity function along the edges of moving objects.

(3) The new motion-prediction technique, PAMT, was shown to offer the potential for greater information compression as well as the potential for easier implementation of a real-time codec.

(4) 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.

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 (5.6).) 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 [63,64]?

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) [51-53,55]. 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.

One way to determine the effect of the interlace would be to interlace the images in the synthetic sequence [53]. The algorithm should also be run on other synthetic images -- such as smoothed disks, rings, and squares -- to ascertain the effect of having moving objects with constant intensity interiors. These types of objects do occur in real sequences (dark suits, white shirts, robot arms, etc.).

A potential solution to the inability to correct the displacement vector when $\nabla_z I|_{t-1}=0$ is averaging the spatial gradients in the two frames with a 2:1 weighting. (Less weight being given to the gradient values obtained from the present frame since a backward difference must be used.) At least this would yield a nonzero spatial gradient in the correct direction without putting too much weight on a noisy estimate. Alternatively one could first see if the $\nabla_z I|_{t-1}=0$ when $|\mathrm{DFD}| > T$ and then (and only then) use $\nabla_z I|_t$.

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

The temporal prediction scheme was not simulated herein. However, it appears that it is inferior to the spatial prediction scheme (either in performance or implementation complexity) since the researchers who proposed temporal prediction [3,55] have since started using a perturbation of spatial prediction [86-88].

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 [74,75]. This could be viewed as a switched corrector.

A potential circuit diagram is in Figure 8.1. The codec diagrammed transmits zeroth-order entropy information. A few terms need to be defined:

1) $I(\mathbf{z},t)$ is the actual intensity captured by the camera.

2) $\hat{I}(\mathbf{z},t)$ is the predicted intensity, which contains prediction error.

3) $\tilde{I}(\mathbf{z},t)$ is the reconstructed intensity, which contains quantizer error.

4) R/W is read/write. The line is either high (indicating read, for example) or low (indicating write, for example).

The interpolation portion of the algorithm is not indicated. The condition queue is used to indicate whether the $\hat{\mathbf{d}}$ value needs to be corrected or not.

In the correction circuit there are two framestore pairs. The addressing and clocking to each pair is alternated each frame. During one frame-processing time, one pair (one intensity framestore and one displacement-estimate framestore) are clocked by CLK1, the same clock that is used in the prediction circuit. Intensity data is stored and displacement estimates are output. This pair is addressed by an

incrementer. The second intensity framestore and the second displacement-estimate framestore are clocked by CLK2, an asynchronous clock, and are addressed by the motion corrector using $\hat{\mathbf{d}}^1$ and an incrementer. For this pair, intensity data are output and displacement estimates are stored. CLK2 must toggle the same number of times in one frame interval as CLK1 does. Most of the time CLK2 can be faster than CLK1 since in the correction circuit $\hat{\mathbf{d}}^0$ is simply copied from one displacement estimate memory into the other with no intervening computation. When motion correction is being done, CLK2 cannot toggle as fast as CLK1 since more computation is being done in the correction circuit loop than in the prediction circuit loop.

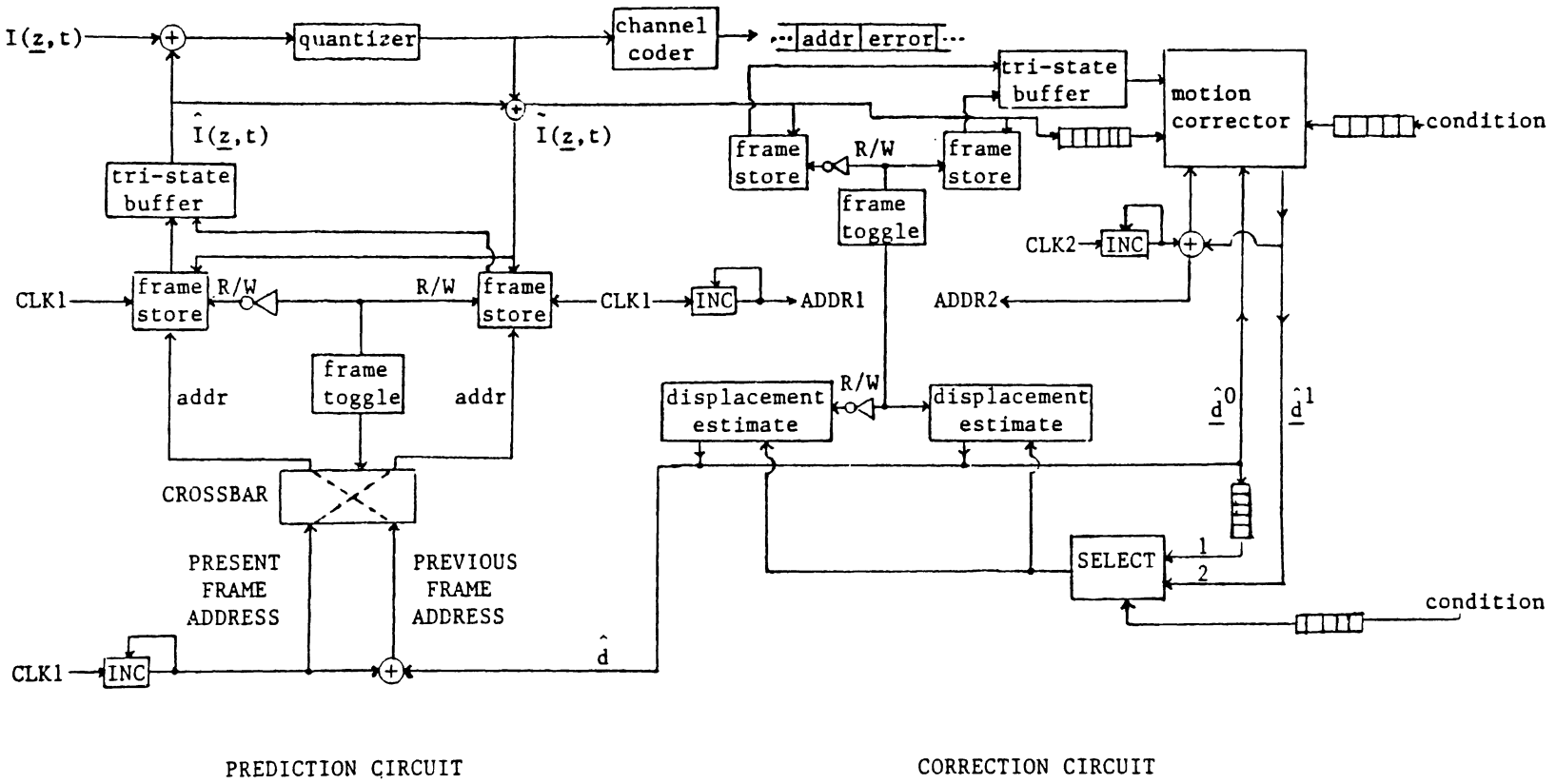


Figure 8.1: A Potential Circuit Diagram

CHAPTER NINE
REFERENCES

[1] A. K. Jain, "Advances in Mathematical Models for Image Processing," *Proc. IEEE*, Vol. 69, No. 5, pp. 502-528, May 1981.

[2] H. H. Nagel, "Image Sequence Analysis: What Can We Learn from Applications?", in *Image Sequence Analysis*, (T. S. Huang, Ed.), pp. 19-228, Springer-Verlag, Berlin, 1981.

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

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

[5] A. N. Netravali and J. O. Limb, "Picture Coding: A Review," *Proc. IEEE*, Vol. 68, No. 3, pp. 366-406, March 1980.

[6] B. G. Haskell, "Frame Replenishment Coding of Television," in *Image Transmission Techniques*, (W. K. Pratt, Ed.), pp. 189-217, Academic Press, New York, 1979.

[7] J. B. O'Neal, Jr., "Predictive Quantizing Systems (Differential Pulse Code Modulation) for the Transmission of Television Signals," *BSTJ*, Vol. 45, No. 5, pp. 689-722, May-June 1966.

[8] F. W. Mounts, "A Video Encoding System with Conditional Picture Element Replenishment," *BSTJ*, Vol. 48, No. 7, pp.2545-2554, Sept. 1969.

[9] J. C. Candy, M. A. Franke, B. G. Haskell, and F. W. Mounts, "Transmitting Television as Clusters of Frame to Frame Differences," *BSTJ*, Vol. 50, No. 6, pp. 1889-1917, July-Aug. 1971.

[10] R. F. W. Pease and J. O. Limb, "Exchange of Spatial and Temporal Resolution in Television Coding," *BSTJ*, Vol. 50, No. 1, pp. 191-200, Jan. 1971.

[11] J. O. Limb and R. F. W. Pease, "Simple Interframe Coder for Video Telephony," *BSTJ*, Vol. 50, No. 6, pp. 1877-1888, July-Aug. 1971.

[12] B. G. Haskell, "Entropy Measurements for Non-Adaptive and Adaptive, Frame to Frame Linear Predictive Coding of Video Telephone Signals," *BSTJ*, Vol. 54, No. 6, pp. 1155-1174, July-Aug. 1975.

[13] B. G. Haskell, "Differential Addressing of Clusters of Changed Picture Elements for Interframe Coding of Video Telephone Signals," *IEEE Trans. on Communications*, Vol. COM-24, No. 1, pp. 140-144, Jan. 1976.

[14] 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.

[15] J. Max, "Quantizing for Minimum Distortion," *IEEE Trans. on Inform. Theory*, Vol. IT-6, No. 2, pp. 7-12, Mar. 1960.

[16] A. N. Netravali and B. Prasada, "Adaptive Quantization of Picture Signals using Spatial Masking," *Proc. IEEE*, Vol.-65, No. 4, pp. 536-548, April 1977.

[17] D. K. Sharma and A. N. Netravali, "Design of Quantizers for DPCM Coding of Picture Signals," *IEEE Trans. on Communications*, Vol. COM-25, No. 11, pp. 1267-1274, Nov. 1977.

[18] A. N. Netravali, "On Quantizers for DPCM Coding of Picture Signals," *IEEE Trans. on Inform. Theory*, Vol. IT-23, No. 3, pp. 360-370, May 1977.

[19] K. Iinuma, Y. Iijima, T. Ishiguro, H. Kaneko, and S. Shigaki, "Interframe Coding for 4Mhz Color Television Signals," *IEEE Trans. on Communications*, Vol. COM-23, No. 12, pp. 1461-1466, Dec. 1975.

[20] H. Yasuda, F. Kanaya, and H. Kawanishi, "1.544-Mbits/s Transmission of TV Signals by Interframe Coding System," *IEEE Trans. on Communications*, Vol. COM-23, No. 10, pp. 1175-1180, Oct. 1976.

[21] H. Yasuda, H. Kuroda, H. Kawanishi, F. Kanaya, and H. Hashimoto, "Transmitting 4-MHz TV signals by Combinational Difference Coding," *IEEE Trans. on Communications*, Vol. COM-25, No. 5, pp. 508-516, May 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] R. C. Nicole, L. Chiariglione, and P. Schaeffer, "The Development of the

European Video Teleconferencing Codec," *1982 IEEE Globecom*, pp. D4.3.1-D4.3.5, Nov. 1982.

[24] A. N. Netravali and E. G. Bowen, "Improved Reconstruction of DPCM-Coded Pictures," *BSTJ*, Vol. 61, No. 6, pp. 969-979, July-Aug. 1982.

[25] H. C. Andrews and W. K. Pratt, "Fourier Transform Coding of Images," in *Proc. Hawaii Int. Conf. System Science*, pp. 677-678, Jan. 1968.

[26] W. K. Pratt, J. Kane, and H. C. Andrews," Hadamard Transform Image Coding," *Proc. IEEE*, Vol. 57, No. 1, pp. 58-68, Jan. 1969.

[27] W. K. Pratt, W. H. Chen, and L. R. Welch, "Slant Transform Image Coding," *IEEE Trans. on Communications Technology*, Vol. COM-22, No. 4, pp. 1075-1093, Aug. 1974.

[28] W. H. Chen, C. H. Smith, and S. Fralick, "A Fast Computational Algorithm for the Discreet Cosine Transform," *IEEE Trans. on Communications*, Vol. COM-25, No. 9, pp. 1004-1009, Sept. 1977.

[29] "Whatever Happened to the Picturephone?", *IEEE Spectrum*, Vol. 19, No. 2, p. 24, Feb. 1982.

[30] A. Habibi, "Comparison of N-th-order Encoders with Linear Transformations and Block Quantization Techniques," *IEEE Trans. on Communications Technology*, Vol COM-19, No. 6, pp. 948-956, Dec. 1971.

[31] A. Habibi and P. A. Wintz, "Image Coding by Linear Transformation and Block Quantization," *IEEE Trans. on Communications Technology*, Vol COM-19, No. 1, pp. 50-62, Feb. 1971.

[32] P. A. Wintz, "Transform Picture Coding," *Proc. IEEE*, Vol. 60, No. 7, pp. 809-820, July, 1972.

[33] H. J. Landau and D. Slepian, "Some Computer Experiments in Picture Processing for Bandwidth Reduction," *BSTJ*, Vol. 50, No. 5, pp. 1525-1540, May-June 1971.

[34] G. B. Anderson and T. S. Huang, "Piecewise Fourier Transformation for Picture Bandwidth Compression," *IEEE Trans. on Communications Technology*, Vol. COM-19, No. 2, pp. 133-140, April 1971.

[35] A. Habibi, "Hybrid Coding of Pictorial Data," *IEEE Trans. on Communications Technology*, Vol. COM-22, No. 5, pp. 614-623, May 1974.

[36] S. C. Knauer, "Real-time Video Compression Algorithm for Hadamard Transform Processing," *IEEE Trans. Electromagnetic Computation*, Vol. EMC-18, pp. 28-36, Feb. 1976.

[37] J. A. Roese, W. K. Pratt, and G. S. Robinson, "Interframe Cosine Transform Image Coding," *IEEE Trans. on Communications*, Vol. COM-25, No. 11, pp. 1329-1338, Nov. 1977.

[38] J. J. Szarek, "A Hybrid Digital Image Bandwidth Reduction Algorithm," Master's Thesis, North Carolina State University, Aug. 1983.

[39] F. W. Mounts, A. N. Netravali, and B. Prasada, "Design of Quantizers for Real-Time Hadamard Transform Coding of Pictures," *BSTJ*, Vol. 56, No. 1, pp. 21-48, Jan. 1977.

[40] W. H. Chen and C. H. Smith, "Adaptive Coding of Monochrome and Color Images," *IEEE Trans. on Communications*, Vol. COM-25, No. 11, pp. 1285-1292, Nov. 1977.

[41] W. H. Chen, "Scene Adaptive Coder," *Proc. Int. Conf. Communications*, pp. 22.5.1-22.5.6, June 1981.

[42] T. C. Chen and R. J. P. de Figueiredo, "An Image Transform Coding Scheme Based on Spatial Domain Considerations," *IEEE Trans. on PAMI*, Vol. PAMI-5, No. 3, pp. 332-337, May 1983.

[43] A. G. Tescher, "Transform Image Coding," in *Image Transmission Techniques*, (W. K. Pratt, Ed.), pp. 113-155, Academic Press, New York, 1979.

[44] J. A. Roese, "Hybrid Transform / Predictive Image Coding," in *Image Transmission Techniques*, (W. K. Pratt, Ed.), pp. 157-187, Academic Press, New York, 1979.

[45] A. K. Jain, "A Sinusoidal Family of Unitary Transforms," *IEEE Trans. on PAMI*, Vol. PAMI-1, No. 4, pp. 356-365, Oct. 1979.

[46] W. K. Pratt, *Digital Image Processing*, John Wiley & Sons, Inc., 1978.

[47] J. O. Limb and J. A. Murphy, "Measuring the Speed of Moving Objects from

Television Signals," *IEEE Trans. on Communications*, Vol. COM-23, No. 4, pp. 174-178, April 1975.

[48] J. O. Limb and J. A. Murphy, "Estimating the Velocity of Moving Images in Television Signals," *Computer Graphics and Image Processing*, Vol. 4, pp. 311-327, 1975.

[49] C. Cafforio and F. Rocca, "Methods for Measuring Small Displacements of Television Images," *IEEE Trans. on Inform. Theory*, Vol. IT-22, No. 5, pp. 573-579, Sept. 1976.

[50] C. Cafforio and F. Rocca, "Tracking Moving Objects in Television Images," *Signal Processing*, Vol. 1, pp. 133-140, 1979.

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

[52] 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.

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

[54] T. Ishiguro and K. Iinuma, "Television Bandwidth Compression Transmission by Motion-Compensated Coding," *Communications*, Vol. 20, No. 6, pp. 24-30, Nov. 1982.

[55] 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.

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

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

[58] 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.

[59] W. E. Snyder, S. A. Rajala, and G. Hirzinger, "Image Modeling: The

Continuity Assumption and Tracking," *Proc. 1980 Int. Joint Conf. Pattern Recognition*, pp. 1111-1114, 1980.

[60] R. W. Hornbeck, *Numerical Methods*, Quantam Publishers, New York, 1975.

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

[62] 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.

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

[64] 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.

[65] W. B. Thompson and S. T. Barnard, "Lower-Level Estimation and Interpretation of Visual Motion," *Computer*, Vol. 14, No. 8, pp. 20-28, Aug. 1981.

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

[67] 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.

[68] 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.

[69] Technical Description of NETEC-X1(MC) TV CODEC, DEX-5547, Issue 1, NEC America, Inc., NEC Corporation, Tokyo, Japan, March 1983.

[70] 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.

[71] T. S. Huang and Y. P. Hsu, "Image Sequence Enhancement," in *Image Sequence Analysis*, (T. S. Huang, Ed.), pp. 289-309, Springer-Verlag, Berlin, 1981.

[72] B. G. Haskell and J. O. Limb, "Predictive Video Encoding Using Subject Velocity," U.S. Patent 3,632,865, Jan. 1972.

[73] D. G. Luenberger, *Introduction to Linear and Nonlinear Programming*, Addison-Wesley, Reading, MA, 1973.

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

[75] 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.

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

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

[78] 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.

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

[80] 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.

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

[82] R. J. Moorhead and S. A. Rajala, "C Programs for Motion-Compensated Image Sequence Compression," CCSP-TR-85/13, North Carolina State University, July 1985.

[83] B. G. Haskell, F. W. Mounts, and J. C. Candy, "Interframe Coding of Video-Telephone Pictures," *Proc. IEEE*, Vol. 60, No. 7, pp 792-800, July 1972.

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

[85] C. Cafforio, "Remarks on the differential method for the estimation of movement in television images," *Signal Processing*, Vol. 4, pp. 45-52, 1982.

[86] S. Sabri and B. Prasada, "Coding of Broadcast TV Signals for Transmission over Satellite Channels," *IEEE Trans. on Communications*, Vol. COM-32, No. 12, pp. 1323-1330, Dec. 1984.

[87] S. Sabri, "Movement Compensated Interframe Prediction for NTSC Color TV Signals," *IEEE Trans. on Communications*, Vol. COM-32, No. 8, pp. 954-968, Aug. 1984.

[88] E. Dubois and S. Sabri, "Noise Reduction in Image Sequences Using Motion-Compensated Temporal Filtering," *IEEE Trans. on Communications*, Vol. COM-32, No. 7, pp. 826-831, July 1984.

CHAPTER TEN

APPENDICES

10.1 Appendix A: Model of the Edges of Moving Objects

This appendix contains an analysis of the edges found in some actual sequences of moving objects. A model for those edges is developed and the model's effect on the number of terms in a Taylor series expansion which must be retained for an accurate estimate is investigated.

Two consecutive frames of the sequence bobsjob are shown in Figure A.1. Each frame is 282 lines by 448 pels/line. Four blocks of data taken from moving edges in Figure A.1a are shown in Figure A.2. The value of the spatial gradient as the edges are traversed in either the x or y direction can readily be seen to vary. It would be advantageous for the analysis done in Chapter 5 if the edges could be modeled by either a linear or parabolic function. However, an arctan function appears to be a much better model. In calculating $\nabla_z I$ we actually calculate the components $\nabla_x I$ and $\nabla_y I$. Thus it is those values that should be modeled. The intensity function in one dimension and the derivatives are:

$$I = (A)\arctan(Bx + C) + D$$

$$\frac{dI}{dx} = \frac{AB}{1 + (Bx + C)^2}$$

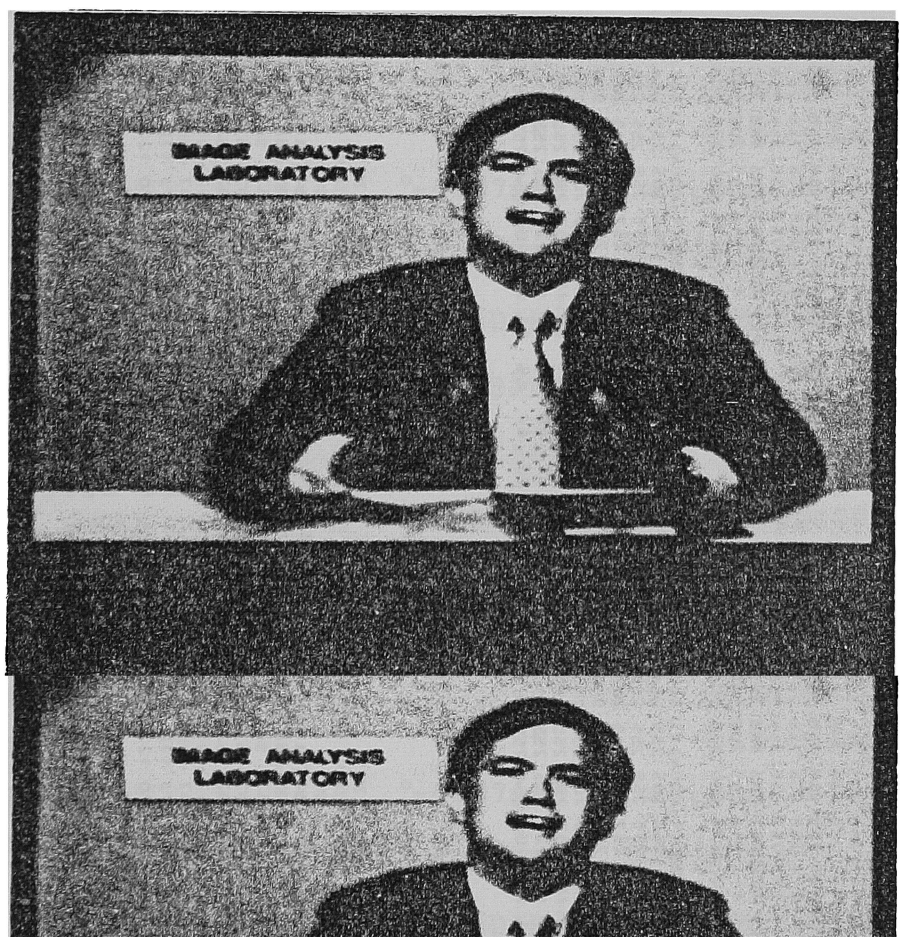
$$\frac{d^2I}{dx^2} = \frac{-2AB^2(Bx + C)}{[1 + (Bx + C)^2]^2}$$

$$\frac{d^3I}{dx^3} = -2AB^3\frac{1 - 3(Bx + C)^2}{[1 + (Bx + C)^2]^3}$$

The x for which Bx+C equals zero is the center of the edge. Table A.1 shows the values of the derivatives and the ratio of derivatives for various values of u, where u=Bx+C is a measure of the distance from the center of the edge. Note that the magnitudes of the higher order derivatives decrease faster than dI/dx as |u| increases for B≈1. However, the magnitudes of the higher order derivatives are significant for values of u close to zero.

Note also that $E\{\nabla_x I \ \nabla_y I\}$ is zero when evaluated over the entirety of an edge. Therefore the assumption made in going from equation (5.18) to equation (5.19), namely that $\mathbf{f}(i) \approx \mathbf{f}_1(i)$, is reasonable.

Figure A.1a: Bobsjob, frame 20



|170,230| - |185,245| (coat/shirt edge on rt. chest) first field

61	59	60	64	68	72	82	117	185	247	255	255	255	251	210
60	59	60	60	63	72	79	107	172	236	255	255	255	255	245
60	60	61	61	61	68	75	96	155	226	255	255	255	255	255
59	59	61	63	62	65	72	88	140	214	255	255	255	255	255
58	63	61	60	61	64	70	81	124	199	251	255	255	255	255
61	60	61	59	61	64	68	77	113	187	247	255	255	255	255
60	60	60	60	61	61	66	77	107	173	241	255	255	255	245
60	62	63	59	61	62	63	72	97	159	230	255	255	255	239

|170,230| - |185,245| (coat/shirt edge on rt. chest) second field

62	61	61	62	65	72	78	99	157	226	255	255	255	255	238
62	61	59	61	62	67	75	89	140	213	255	255	255	255	255
62	59	60	62	61	66	72	84	126	201	252	255	255	255	255
62	60	60	62	61	64	72	80	112	186	246	255	255	255	255
59	62	61	61	62	63	65	76	105	170	238	255	255	255	255
61	60	59	62	62	63	66	74	98	158	233	255	255	255	255
60	59	62	62	60	62	66	72	91	147	220	255	255	255	252
61	60	62	61	58	60	64	69	86	133	206	254	255	255	244

|185,215| - |200,230| (right thumb) first field

64	64	64	63	63	62	63	64	64	63	62	62	61	61	62	61
63	63	63	60	62	64	64	68	72	78	81	77	70	67	63	61
63	63	61	63	66	73	83	96	111	122	132	133	126	116	98	74
64	64	65	70	83	96	111	123	133	139	140	144	142	134	121	103
65	69	83	92	106	120	127	136	136	138	139	136	134	124	112	101
76	88	100	114	125	129	133	133	132	132	130	125	117	108	95	86
86	103	116	124	133	135	133	132	129	121	119	113	102	94	83	78
89	101	110	120	127	128	129	125	123	120	107	101	93	80	76	72

|185,215| - |200,230| (right thumb) second field

69	68	66	67	64	64	66	63	63	64	63	63	62	61	61	62
66	67	64	62	63	62	63	63	61	64	66	66	67	66	64	63
66	64	64	61	62	63	61	64	67	75	87	98	107	111	110	100
64	63	62	60	63	65	70	77	88	104	117	128	136	139	140	138
64	62	62	64	69	76	89	102	112	122	131	135	138	139	137	133
61	62	67	73	86	100	113	121	125	130	132	133	133	131	126	118
63	66	72	88	104	115	126	129	130	131	129	127	121	120	112	102
66	70	77	92	102	112	121	126	130	127	124	122	115	108	98	90

|218,138| - |233,153| (coat/shirt sleeve, rt. wrist) first field

50	49	50	50	53	50	48	51	48	49	48	45	50	49	46	47
52	50	50	51	51	52	52	50	51	49	50	53	50	50	52	50
51	49	50	52	50	52	51	51	52	51	52	52	52	52	55	61
49	50	49	50	52	50	51	51	52	53	53	57	63	74	98	124
50	50	49	51	52	51	52	52	57	83	83	112	136	157	172	180
49	51	51	50	52	52	56	66	85	118	148	167	180	190	199	201
50	50	50	52	53	58	75	107	137	156	166	172	176	184	197	203
51	52	52	54	63	85	115	141	159	166	168	170	171	174	177	183

|218,138| - |233,153| (coat/shirt sleeve, rt. wrist) second field

52	53	51	54	53	51	53	50	50	51	50	52	51	48	51	52
51	51	51	51	53	54	51	52	52	51	52	52	55	55	60	72
52	51	49	51	51	50	53	54	52	56	58	62	76	98	129	155
50	50	50	52	53	52	53	58	64	84	113	138	159	172	180	186
49	51	51	52	52	56	68	91	125	153	174	187	194	199	200	200
49	51	51	54	62	84	119	147	181	187	175	183	192	201	201	201
52	51	56	66	93	124	148	163	167	168	171	173	175	180	189	191
54	60	77	109	136	153	166	169	171	172	172	172	171	171	176	180

|210,400| - |225,415| (left elbow/bkg edge) first field

53	54	54	56	63	75	108	136	143	144	143	144	142	143	146	145
52	51	52	55	53	54	61	79	113	135	142	143	139	146	145	144
54	53	53	55	53	54	55	55	60	87	124	138	142	145	144	147
53	53	51	51	53	52	53	53	54	58	73	109	136	140	141	144
52	53	53	51	50	52	50	51	54	53	54	66	97	130	140	140
51	52	54	54	51	52	50	49	51	50	51	54	61	84	121	142
49	48	50	51	52	53	50	49	50	48	47	50	53	60	81	120
50	48	48	49	49	50	50	50	49	47	49	46	44	52	59	81

|210,400| - |225,415| (left elbow/bkg edge) second field

52	52	55	54	55	60	70	98	131	143	143	142	142	144	146	145
52	52	53	53	53	54	57	59	74	107	134	143	144	145	146	145
54	53	54	55	55	54	53	55	56	61	88	124	141	141	144	145
56	53	53	51	49	54	53	53	54	54	58	74	110	137	139	143
52	53	54	53	54	51	50	53	51	51	52	55	65	96	130	141
50	49	52	53	52	52	51	50	50	49	50	52	55	63	88	126
51	49	49	49	51	52	51	50	50	49	47	47	49	55	61	86
51	49	50	50	49	51	51	51	49	49	47	47	46	46	53	80

Figure A.2: Examples of Moving Edges

TABLE A.1

Value of Intensity Derivatives along Image Edges

u	$\frac{dI}{dx}$	$\frac{d^2I}{dx^2}$	$\frac{d^3I}{dx^3}$	$\frac{d^2I}{dx^2}\Big/\frac{dI}{dx}$	$\frac{d^3I}{dx^3}\Big/\frac{dI}{dx}$
-4	$0.06AB$	$0.03AB^2$	$0.02AB^3$	$0.47B$	$0.33B^2$
-3	$0.10AB$	$0.06AB^2$	$0.05AB^3$	$0.60B$	$0.52B^2$
-2	$0.20AB$	$0.16AB^2$	$0.18AB^3$	$0.80B$	$0.88B^2$
-1	$0.50AB$	$0.50AB^2$	$0.50AB^3$	$1.00B$	$1.00B^2$
-0.50	$0.80AB$	$0.64AB^2$	$-0.26AB^3$	$0.80B$	$-0.33B^2$
-0.33	$0.90AB$	$0.54AB^2$	$-0.97AB^3$	$0.60B$	$-1.08B^2$
-0.25	$0.94AB$	$0.44AB^2$	$-1.35AB^3$	$0.47B$	$-1.44B^2$
0	$1.00AB$	0	$-2.00AB^3$	0	$-2.00B^2$
0.25	$0.94AB$	$-0.44AB^2$	$-1.35AB^3$	$-0.47B$	$-1.44B^2$
0.33	$0.90AB$	$-0.54AB^2$	$-0.97AB^3$	$-0.60B$	$-1.08B^2$
0.50	$0.80AB$	$-0.64AB^2$	$-0.26AB^3$	$-0.80B$	$-0.33B^2$
1	$0.50AB$	$-0.50AB^2$	$0.50AB^3$	$-1.00B$	$1.00B^2$
2	$0.20AB$	$-0.16AB^2$	$0.18AB^3$	$-0.80B$	$0.88B^2$
3	$0.10AB$	$-0.06AB^2$	$0.05AB^3$	$-0.60B$	$0.52B^2$
4	$0.06AB$	$-0.03AB^2$	$0.02AB^3$	$-0.47B$	$0.33B^2$

10.2 Appendix B: Derivation of the Gradient Estimation Noise Covariance, Σ_{gn}^2

This appendix contains the derivation of the noise covariance in the gradient $\nabla_{\dot{\mathbf{d}}}[DFD(\mathbf{z}_a,\hat{\mathbf{d}})]^2\Big|_{\hat{\mathbf{d}}=\hat{\mathbf{d}}'}$. 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:

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

where $\hat{\nabla}_{\dot{\mathbf{d}}}$ is the computed value of the gradient and $\nabla_{\dot{\mathbf{d}}}$ is the true value. $\nabla_{\dot{\mathbf{d}}}$ was evaluated in equation (4.7) to be

$$\nabla_{\dot{\mathbf{d}}}[DFD(\mathbf{z}_a,\hat{\mathbf{d}})]^2\Big|_{\hat{\mathbf{d}}=\hat{\mathbf{d}}'} = 2DFD(\mathbf{z}_a,\hat{\mathbf{d}}^t)\nabla_{\mathbf{z}}I(\mathbf{z}-\hat{\mathbf{d}}^t,t-1)\Big|_{\mathbf{z}=\mathbf{z}_a}. \tag{4.7}$$

Substituting (4.7) into (B.1) produces:

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

where

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

e_{DFD} *is the DFD error*,

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

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

Let

$$I_1 = I(\mathbf{z}_a, t), \quad \text{and} \tag{B.3}$$

$$I_2 = I(\mathbf{z}_a - \hat{\mathbf{d}}^i, t-1). \tag{B.4}$$

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,

$$\begin{aligned}
\mathbf{N}(i) &= 2[(I_1+e_1) - (I_2+e_2)]\begin{bmatrix}\nabla_x I+e_x\\ \nabla_y I+e_y\end{bmatrix} - 2(I_1-I_2)\begin{bmatrix}\nabla_x I\\ \nabla_y I\end{bmatrix}\\
&= 2[(I_1-I_2) + (e_1-e_2)]\begin{bmatrix}\nabla_x I+e_x\\ \nabla_y I+e_y\end{bmatrix} - 2(I_1-I_2)\begin{bmatrix}\nabla_x I\\ \nabla_y I\end{bmatrix}\\
&= 2\begin{bmatrix}\{(I_1-I_2)+(e_1-e_2)\}(\nabla_x I+e_x)\\ \{(I_1-I_2)+(e_1-e_2)\}(\nabla_y I+e_y)\end{bmatrix} - 2\begin{bmatrix}(I_1-I_2)\nabla_x I\\ (I_1-I_2)\nabla_y I\end{bmatrix}\\
&= 2\begin{bmatrix}(I_1-I_2)e_x + (e_1-e_2)\nabla_x I + (e_1-e_2)e_x\\ (I_1-I_2)e_y + (e_1-e_2)\nabla_y I + (e_1-e_2)e_y\end{bmatrix}\\
&= 2\begin{bmatrix}a\\ b\end{bmatrix}.
\end{aligned} \tag{B.5}$$

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

$$\Sigma_{gn}^2 = E\{\mathbf{N}(i)\mathbf{N}^T(i)\} = 4E\left\{\begin{bmatrix}a^2 & ab\\ ab & b^2\end{bmatrix}\right\} = 4\begin{bmatrix}E\{a^2\} & E\{ab\}\\ E\{ab\} & E\{b^2\}\end{bmatrix}. \tag{B.6}$$

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

$$E\{a^2\} = E\{(I_1e_x - I_2e_x + e_1\nabla_x I - e_2\nabla_x I + e_1e_x - e_2e_x)^2\}. \tag{B.7}$$

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

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

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

$$I_3 = I(\mathbf{z}_a - \hat{\mathbf{d}}^t - 1, t), \tag{B.8}$$

$$I_4 = I(\mathbf{z}_a - \hat{\mathbf{d}}^t + 1, t - 1), \tag{B.9}$$

$$I_5 = I(\mathbf{z}_a - \hat{\mathbf{d}}^t - N, t - 1), \quad \text{and} \tag{B.10}$$

$$I_6 = I(\mathbf{z}_a - \hat{\mathbf{d}}^t + N, t - 1). \tag{B.11}$$

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:

$$
\begin{aligned}
e_x &= \tfrac{1}{2}[(I_4 + e_4) - (I_3 + e_3)] - \tfrac{1}{2}[I_4 - I_3] \\
&= \tfrac{1}{2} e_4 - \tfrac{1}{2} e_3 \\
&= \tfrac{1}{2}(e_4 - e_3).
\end{aligned} \tag{B.12}
$$

Likewise e_y is:

$$e_y = \tfrac{1}{2}(e_6 \quad e_5). \tag{B.13}$$

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 [46]. 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:

$$\begin{aligned} E\{a^2\} = E\{&I_1 e_x I_1 e_x - 2 I_1 e_x I_2 e_x + I_2 e_x I_2 e_x \\ &+ e_1 \nabla_x I e_1 \nabla_x I + e_2 \nabla_x I e_2 \nabla_x I + e_1 e_x e_1 e_x + e_2 e_x e_2 e_x\}. \end{aligned} \tag{B.14}$$

Factoring out common factors as prefixes,

$$\begin{aligned} E\{a^2\} = {} & E\{e_x^2\}[E\{I_1 I_1\} - 2E\{I_1 I_2\} + E\{I_2 I_2\}] \\ & + E\{\nabla_x^2 I\}[E\{e_1^2\} + E\{e_2^2\}] \\ & + E\{e_x^2\}[E\{e_1^2\} + E\{e_2^2\}]. \end{aligned} \tag{B.15}$$

The first term in (B.15) is zero since

$$E\{I_1 I_1\} = E\{I_2 I_2\} = E\{I_1 I_2\}. \tag{B.16}$$

Note that I() is not a random field. Since $E\{e_1^2\} = E\{e_2^2\}$, the last two terms in (B.15) can be rewritten as:

$$2E\{e_1^2\}[E\{\nabla_x^2 I\} + E\{e_x^2\}]. \tag{B.17}$$

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

$$E\{a^2\} = 2\sigma_n^2[E\{\nabla_x^2 I\} + E\{e_x^2\}]. \tag{B.18}$$

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

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

$$\begin{aligned} E\{e_x^2\} &= E\{(\tfrac{1}{2}e_4 - \tfrac{1}{2}e_3)^2\} \\ &= E\{\tfrac{1}{4}e_4e_4 - \tfrac{1}{2}e_4e_3 + \tfrac{1}{4}e_3e_3\} \\ &= \tfrac{1}{4}E\{e_4e_4\} - \tfrac{1}{2}E\{e_4e_3\} + \tfrac{1}{4}E\{e_3e_3\} \\ &= \tfrac{1}{4}\sigma_n^2 + 0 + \tfrac{1}{4}\sigma_n^2 \\ &= \tfrac{1}{2}\sigma_n^2. \end{aligned} \tag{B.19}$$

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

$$E\{a^2\} = 2\sigma_n^2 E\{\nabla_x^2 I\} + (\sigma_n^2)^2. \tag{B.20}$$

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

$$E\{b^2\} = 2\sigma_n^2 E\{\nabla_y^2 I\} + (\sigma_n^2)^2. \tag{B.21}$$

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

$$\begin{aligned} E\{a^2\} = E\{&(I_1e_x - I_2e_x + e_1\nabla_x I \quad e_2\nabla_x I + e_1e_x - e_2e_x)\cdot \\ &(I_1e_y \quad I_2e_y + e_1\nabla_y I - e_2\nabla_y I + e_1e_y \quad e_2e_y)\}. \end{aligned} \tag{B.22}$$

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

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

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

$$E\{ab\} = 0. \tag{B.23}$$

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

$$\Sigma_{gn}^2 = 4\begin{bmatrix} 2\sigma_n^2 E\{\nabla_x^2 I\} + (\sigma_n^2)^2 & 0 \\ 0 & 2\sigma_n^2 E\{\nabla_y^2 I\} + (\sigma_n^2)^2 \end{bmatrix}. \tag{B.24}$$

7.3.3.3 $\epsilon = 0.020$

The three MC algorithms were run using $\epsilon = 0.020$ to see the effect. Although the $||\mathbf{d}_e||$ was reduced by about 10%, it was the only advantage of using the larger convergence coefficient ($\epsilon = 0.020$). A much larger average mse was obtained: on the order of 12 using spatial prediction and as high as 24.88 using PAMT, although using mixed prediction the mse dropped from 11.39 in the first frame to 4.52 in the ninth frame. A larger variance in the displacement estimates occurred in all cases. A larger bit rate was required in all cases: 38.1 kbits using spatial prediction, 41.5 kbits using PAMT prediction, and 35.2 kbits using mixed prediction. This is about a 25% increase in bandwidth in all 3 cases.

7.3.4 Velocity of Seven Leftward

7.3.4.1 $\epsilon = 0.010$

The average $||\mathbf{d}_\epsilon||$ and the average mse in each frame using each of the three prediction techniques are plotted in Figures 7.6a and 7.6b. Using spatial prediction, the variance in the displacement estimates over the moving area was about $\begin{bmatrix} 9.10 \\ 8.10 \end{bmatrix}$ in all nine frames. Using PAMT prediction, the variance dropped from $\begin{bmatrix} 8.87 \\ 7.89 \end{bmatrix}$ in the first frame to $\begin{bmatrix} 1.59 \\ 7.73 \end{bmatrix}$ in the ninth frame. Using mixed prediction, the variance dropped from $\begin{bmatrix} 8.87 \\ 7.89 \end{bmatrix}$ in the first frame to $\begin{bmatrix} 2.77 \\ 5.89 \end{bmatrix}$ in the ninth frame.

The entropy bits required to transmit the nine frames were:

spatial:	34.9 kbits
PAMT:	45.0 kbits
mixed:	39.5 kbits

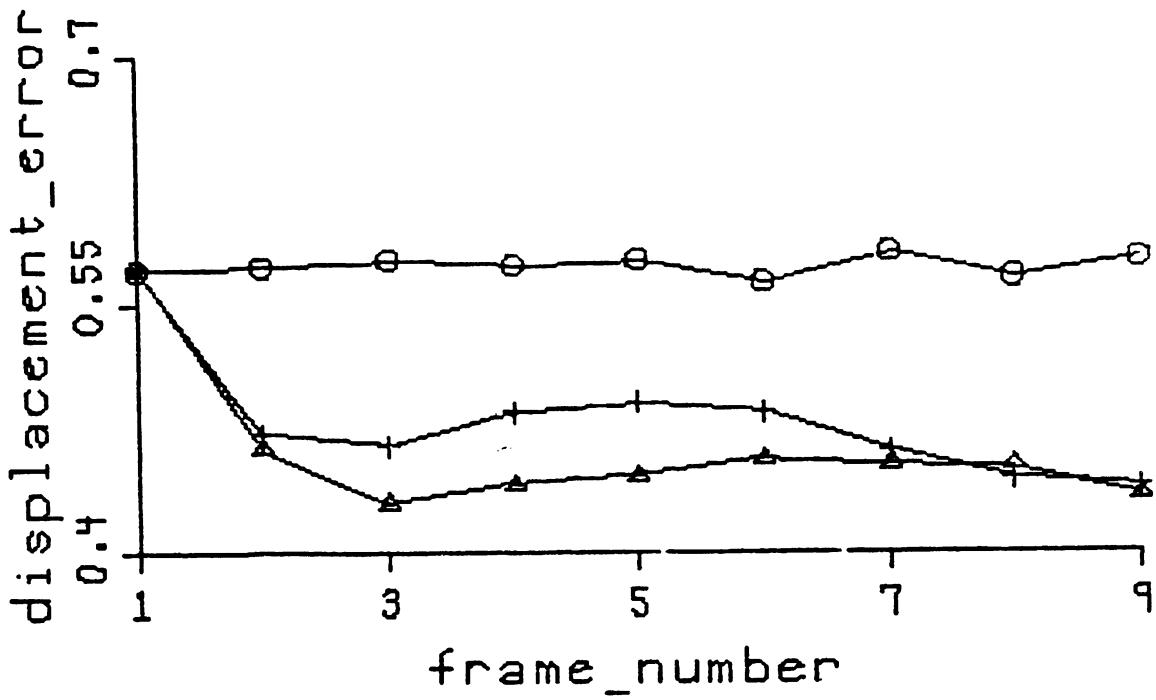


Figure 7.6a: Average $||\mathbf{d}_e||$, velocity = -7, ϵ = 0.010

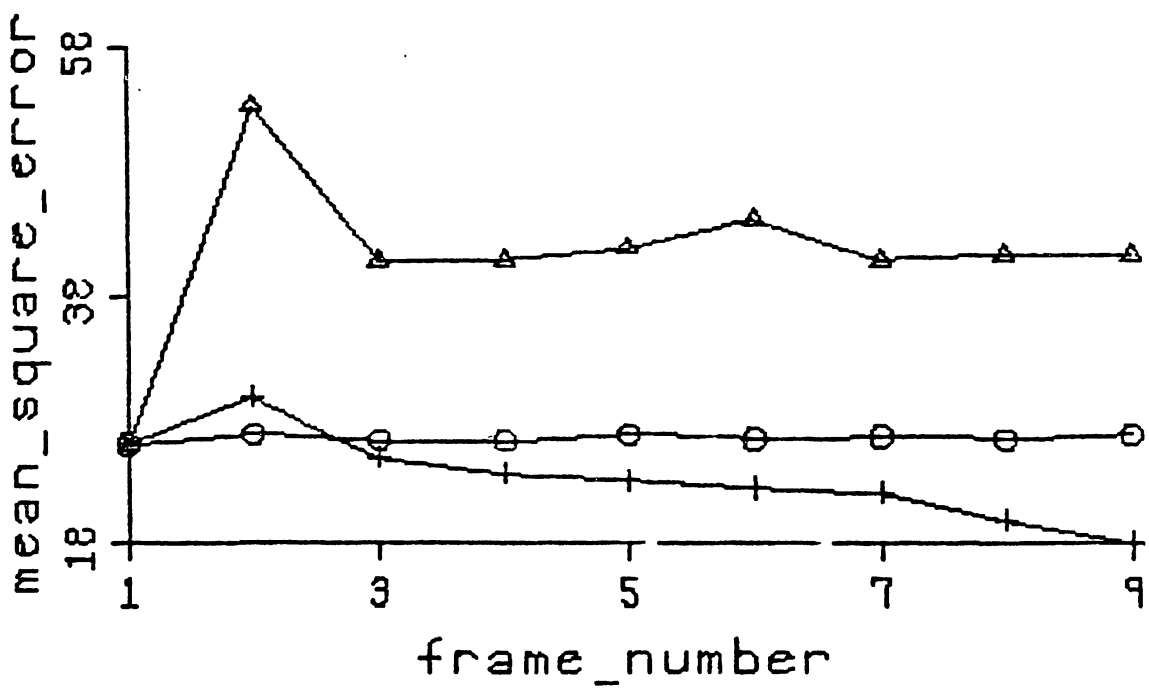


Figure 7.6b: Average mse, velocity = -7, ϵ = 0.010

7.3.4.2 $\epsilon = 0.005$

The average $||\mathbf{d}_e||$ and the average mse in each frame using each of the three prediction techniques are plotted in Figures 7.7a and 7.7b. Using spatial prediction, the variance in the displacement estimates over the moving area was about $\begin{bmatrix}5.26\\6.20\end{bmatrix}$ in all nine frames. Using PAMT prediction, the variance dropped from $\begin{bmatrix}5.16\\6.31\end{bmatrix}$ in the first frame to $\begin{bmatrix}0.75\\5.96\end{bmatrix}$ in the ninth frame. Using mixed prediction, the variance dropped from $\begin{bmatrix}5.16\\6.31\end{bmatrix}$ in the first frame to $\begin{bmatrix}1.09\\4.56\end{bmatrix}$ in the ninth frame.

The entropy bits required to transmit the nine frames were:

spatial:	34.5 kbits
PAMT:	40.6 kbits
mixed:	38.1 kbits

In concluding this section, the lowest average $||\mathbf{d}_e||$ and the lowest mse are obtained with the larger convergence coefficient, but the least variance and entropy are obtained with the smaller ϵ. PAMT prediction produces the lowest average $||\mathbf{d}_e||$, mixed prediction produces the lowest mse, and spatial prediction produces the lowest entropy.

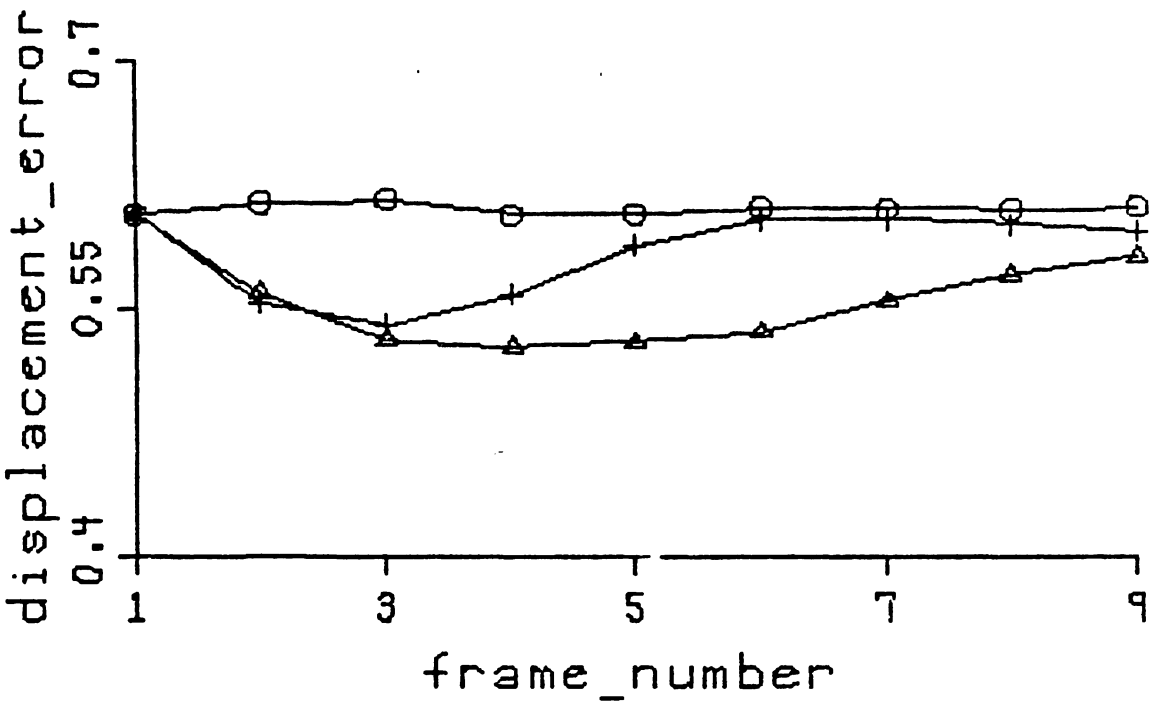


Figure 7.7a: Average $||d_e||$, velocity = -7, ϵ = 0.005

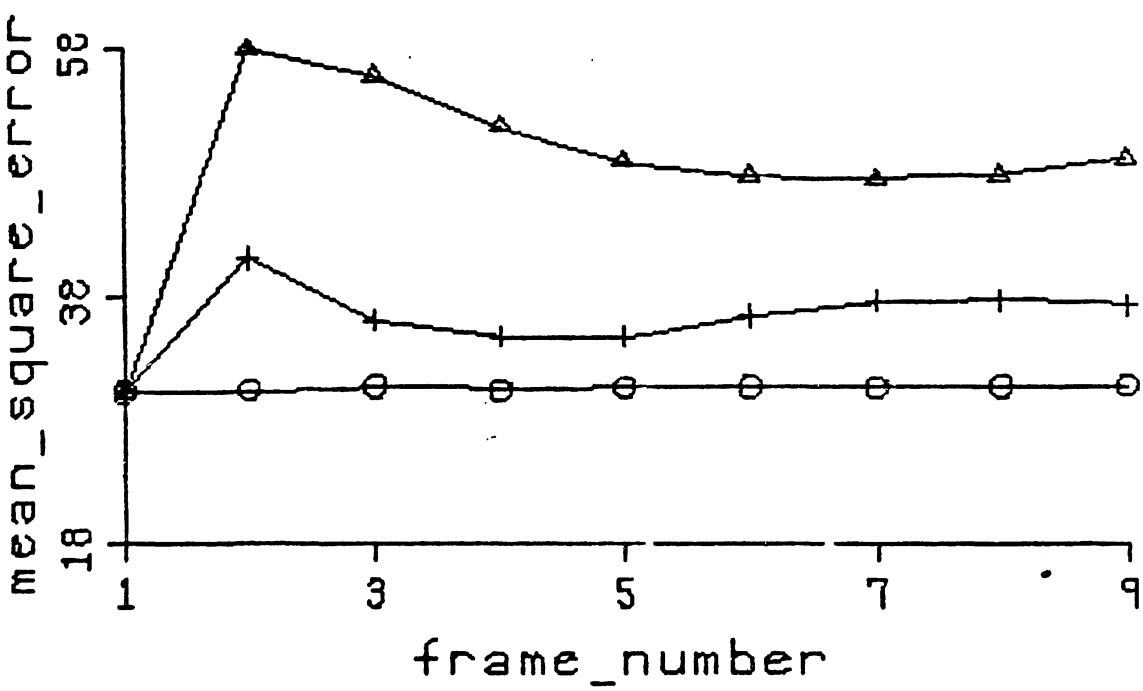


Figure 7.7b: Average mse, velocity = -7, ϵ = 0.005

7.3.5 Velocity of Seven Rightward

7.3.5.1 $\epsilon = 0.010$

The average $||\mathbf{d}_r||$ and the average mse in each frame using each of the three prediction techniques are plotted in Figures 7.8a and 7.8b. Using spatial prediction, the variance in the displacement estimates over the moving area was about $\begin{bmatrix} 6.96 \\ 10.17 \end{bmatrix}$ in all nine frames. Using PAMT prediction, the variance dropped from $\begin{bmatrix} 7.06 \\ 9.18 \end{bmatrix}$ in the first frame to $\begin{bmatrix} 1.85 \\ 4.57 \end{bmatrix}$ in the ninth frame. Using mixed prediction, the variance dropped from $\begin{bmatrix} 7.06 \\ 9.18 \end{bmatrix}$ in the first frame to $\begin{bmatrix} 3.61 \\ 3.80 \end{bmatrix}$ in the ninth frame.

The entropy bits required to transmit the nine frames were:

spatial:	29.3 kbits
PAMT:	37.9 kbits
mixed:	30.9 kbits

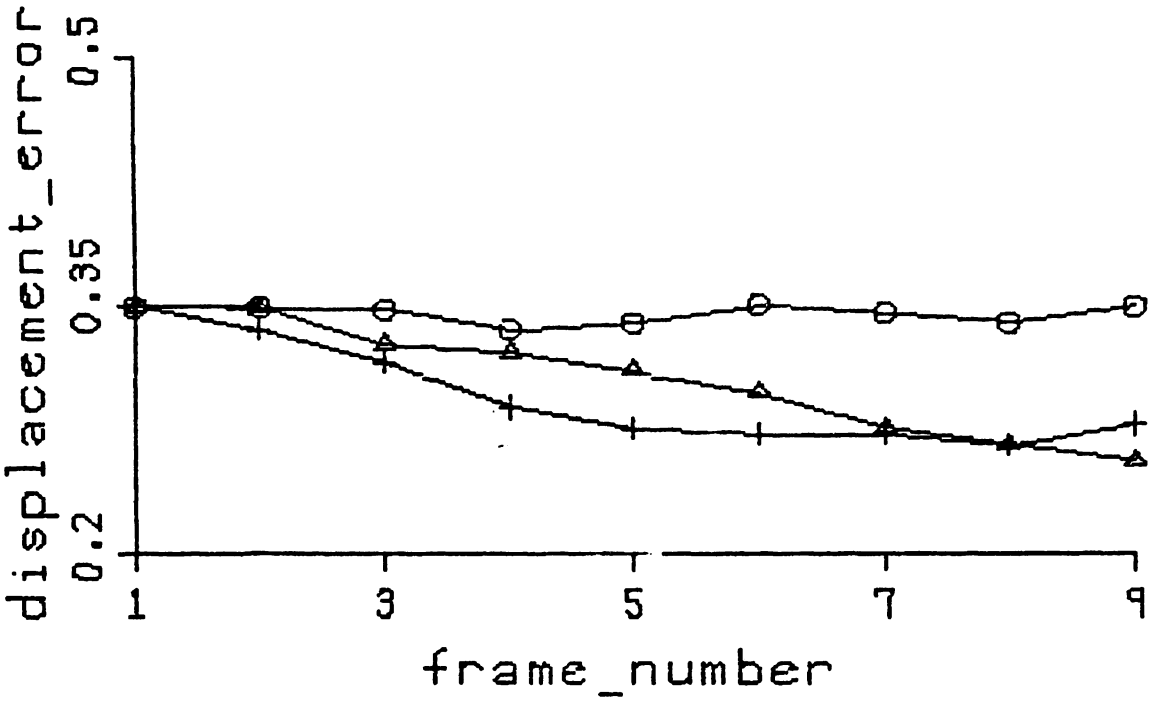


Figure 7.8a: Average $||\mathbf{d}_e||$, velocity $= 7$, $\epsilon = 0.010$

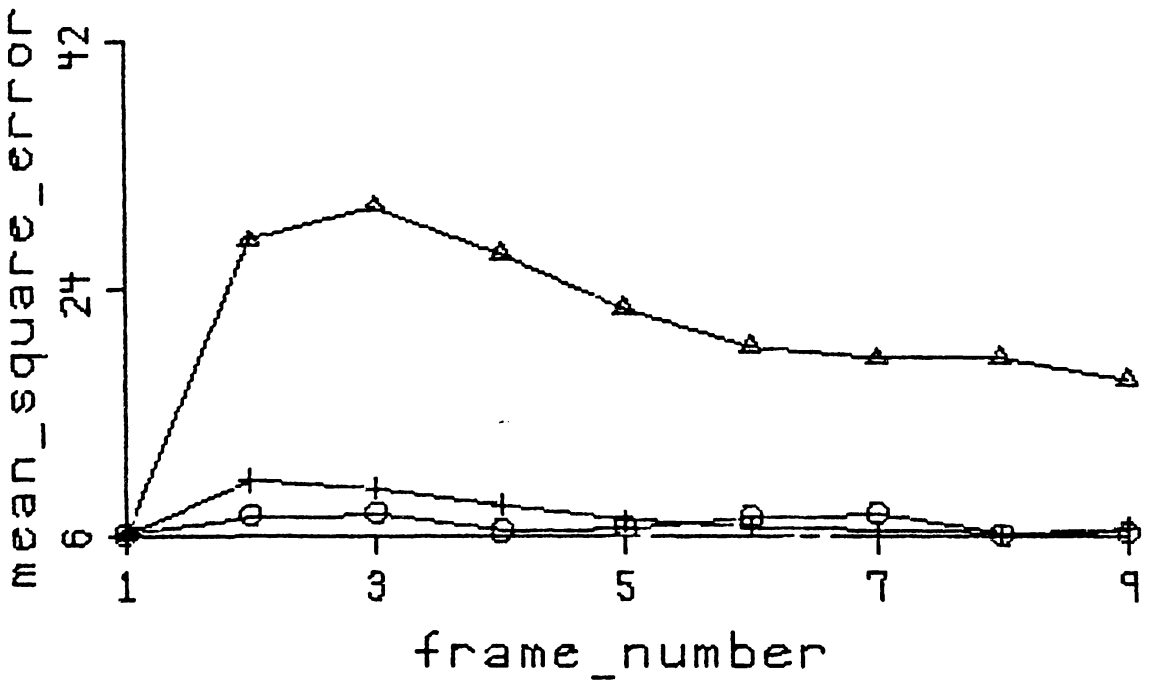


Figure 7.8b: Average mse, velocity $= 7$, $\epsilon = 0.010$