Motion Compensation for Image Compression

(Pel Recursive Motion Estimation Algorithm)

A doctoral thesis submitted in partial fulfilment of the requirements for the award of Doctor of Philosophy of Loughborough University

Department of Electronic and Electrical Engineering

Loughborough University

Hamid-Reza Reza-Alikhani.

April 2002.

Abstract

Pel-Recursive Motion Compensation Techniques for Video Compression

In motion pictures, there is a certain amount of redundancy between consecutive frames These redundancies can be exploited by using interframe prediction techniques To further enhance the efficiency of interframe prediction, motion estimation and compensation, various motion compensation techniques can be used There are two distinct techniques for motion estimation block matching and Pel-recursive Block matching has been widely used as it produces a better signal to noise ratio or a lower bit rate for transmission than the Pel-recursive method

In this thesis, various Pel-recursive motion estimation techniques such as steepest descent gradient algorithm have been considered and simulated Netravali's algorithm was one of the early algorithms which was implemented and simulated to evaluate the performance of the Pel-Recursive technique compared with the Block Matching approach The performance of the gradient method was further enhanced by adaptively selecting the convergence factor (modified gradient) A second algorithm was developed and simulated to produce further improvements

A hybrid system incorporating both the block matching and the Pel-recursive approaches was developed and simulated This combination exhibits even further improvement over existing techniques.

These methods were then applied to various hierarchical hybrid based video coding techniques such as the ITU-T H 263 standard The aim was to reduce the overall bit rate required to transmit video signals

ACKNOWLEDGEMENTS & DEDICATION

I am greatly indebted to my dearest wife Dr. Z. Rezapour, my daughter Melika and my son Maakaan for the steadfast support and encouragement they have given me over the years, without which this endeavour would not have been possible

It is a pleasure to thank all members of staff in the Department of Electronic and Electrical Engineering at the University of Loughborough for their assistance with this research project.

In particular, I wish to expressly convey my sincere gratitude, gratefulness, appreciation and thanks to my supervisors, Professor D J. Parish and Professor H. Gharavi. whom I can not thank adequately for their generous invaluable guidance, suggestions, patience, devotion, constructive criticisms and practical advice throughout the research period that stimulated my thoughts and provided impetus to my practical efforts

Finally, a special note of thanks goes to my friends and my colleagues, Dr. H Sari, Dr R Naimi-mohasses, Dr. K. Shahhamzei for all their valuable discussions, assistance, help and encouragement.

Above all I would like to thank my mother and father for their unconditional love and support throughout many years of life devotion and dedication. This work is a small token of my immense love and gratitude to them both, whom I *dedicate* this thesis to.

Special thanks are extended to Islam Rahematullah and Platinum Rahematullah for being a supportive friend for me in the course of my PhD.

Acronyms and Abbreviations

AC	Alternative Current.
ATD	Absolute Temporal Difference
BBMC	Block-Based Motion Compensation
BMMC	Block-Matching Motion Compensation
BT	British Telecommunications
CCIR	International Radio Consultative Committee
CCITT	International Telegraph and Telecommunication Consultative Committee (see ITU)
CD-ROM	Compact Disc Read-Only Memory
CIF	Common Intermediate Format
Codec	Coder-decoder
CRT	Cathode Ray Tube
DAT	Digital Audio Tape
DC	Direct Current
DCT	Discrete Cosine Transform
DPCM	Differential Pulse Code Modulation
HDTV	High Definition Television
IS	International Standard
ISDN	Integrated Systems Digital Network
ISO	International Standardisation Organisation
ITU	International Telecommunication Union
ITU-T	International Telecommunication Union Telecommunication Standardisation Sector
MC	Motion Compensation

MCP	Motion Compensated Prediction
ME	Motion Estimation
Modem	Modulator-demodulator
MPEG	Moving Picture coding Experts Group
NICAM	Near Instantaneous Companded Audio Multiplex
NTSC	National Television System Committee
PAL	Phase Alternating Line
PSNR	Peak Signal to Noise Ratio.
QCIF	Quarter Common Intermediate Format.
SUB-QCIF	Sub-Quarter Common Intermediate Format
MAD	Mean Absolute Difference.
MSD	Mean Squared Difference
PSNR	Peak Signal to Noise Ratio
PSTN	Public Switched Telephone Network
QCIF	Quarter Common Intermediate Format
SAC	Syntax-based Arithmetic Coding
Sub-QCIF	Sub-Quarter Common Intermediate Format
VLC	Variable Length Code or Variable Length Coding
VLSI	Very Large Scale Integration
VOD	Video on Demand
x, y	spatial co-ordinates in the pixel domain

AUTHOR'S PUBLICATON

[1] H Gharavi, and H Reza-Alikhani, " Pel-Recursive Motion Estimation for video compression", *IEE, Electronics Letters,* Vol. 37, No. 21, pp 1285 – 1286, October 2001.

A Pel-Recursive Motion Estimation Algorithm

H Gharavi and H Reza-Alikhani

Abstract: This paper presents a new pel recursive motion estimation algorithm for video coding applications. The derivation of the algorithm is based on Recursive Least-Squares (RLS) estimation that minimizes the mean square prediction error A comparison with the modified Steepest-descent gradient estimation technique algorithm shows significant improvement in terms of mean-square prediction error performance.

Introduction: Netravali and Robbins [1] developed a pel recursive spatio-temporal steepest-descent gradient technique in which the displacement of a pel (picture element) was predicted from previously transmitted information Since then various algorithms have been proposed to improve the performance of pel recursive motion estimation (PRME) techniques The most important contribution was the modification of the steepest-descent algorithm developed by Walker and Rao [2] In this paper we present a simple but very efficient PRME algorithm that significantly outperforms the modified steepest-descent technique

Proposed Algorithm: For the sake of our analysis, we assume the translational movement of an object is in a plane parallel to the camera and illumination is uniform We also assume the effect of uncovered background to be negligible Under these assumptions, let $S(x, y, t)$ denote the monochrome intensities at point (x, y) of a moving object in the image plane where its translational movement is at a constant velocity of v_x and v_y We can show that after Δt second (one frame period), the object moves to a new location where we can show,

$$S(x, y\ t+\Delta t) = S[(x + V_x \Delta t), (y + V_y \Delta t)\ t] \quad (1)$$

After expanding the field in a power series in Δt and neglecting the higher order terms, the frame difference can be shown as,

$$S(x, y: t + \Delta t) - S(x, y. t) = \frac{\partial}{\partial x} S(x, y\ t)\, dx + \frac{\partial}{\partial y} S(x, y\ t)\, dy \quad (2)$$

where d_x and d_y correspond to the horizontal and vertical components of the motion vector D Assuming $\frac{\partial}{\partial x} S(x, y\ t)$ and $\frac{\partial}{\partial y} S(x, y\ t)$ are known for each x, y, t, and defining ED, LD, and FD as the magnitude of the element, line, and frame difference at point *n*, from (3), we can write,

$$FD = \Phi_n^T D \quad (3)$$

$$\text{where } \Phi n = \begin{bmatrix} \frac{\partial}{\partial x} S(xn, yn\ t) \\ \frac{\partial}{\partial y} S(xn, yn\ t) \end{bmatrix} = \begin{bmatrix} ED \\ LD \end{bmatrix} \quad (4)$$

From (4) the frame difference (FD) measurement is,

$$\xi_n = \Phi_n^T \overline{D} + \text{noise} \quad (5)$$

where $\overline{D} = [\bar{d}(x), \bar{d}(y)]^T$ is the motion vector estimate

For a cluster of M moving pels, the least-squares estimate of D, after carrying out the minimization, can be shown as,

$$\sum_{n=1}^{m} \Phi_n \xi_n = \overline{D} \sum_{n=1}^{m} \Phi_n \Phi_n^T \quad (6)$$

$$\text{For, } \eta = \frac{1}{M} \sum_{n=1}^{M} \Phi_n \xi_n \quad \text{and } R = \frac{1}{M} \sum_{n=1}^{M} \Phi_n \Phi_n^T \quad (7)$$

the estimated motion vector from (6) is obtained as,

$$\bar{D} = R^{-1}\eta \qquad (8)$$

For recursive estimation of η and R, we can write

$$\begin{aligned} \eta_i &= \eta_{i-1} + \Phi_n \xi_n \\ R_i &= R_{i-1} + \Phi_n\Phi_n^T \end{aligned} \qquad (9)$$

Based on the so-called matrix inversion lemma, the inverse of R_i can be obtained

$$R_i^{-1} = R_{i-1}^{-1} - \frac{R_{i-1}^{-1}\Phi_n\Phi_n^T R_{i-1}^{-1}}{1+ \Phi_n^T R_{i-1}^{-1}\Phi_n} \qquad (10)$$

as,

From (8), (9), and (10),

$$\bar{D}_i = \bar{D}_{i-1} - \frac{R_{i-1}^{-1}\Phi_n}{1+ \Phi_n^T R_{i-1}^{-1}\Phi_n}\left(\Phi^T\bar{D}_{i-1} - \xi \right) \qquad (11)$$

In the above equation, the term in the right hand side bracket can be replaced by what is known as the Displaced Frame Difference, DFD Thus,

$$\bar{D}_i = \bar{D}_{i-1} - \frac{R_{i-1}^{-1}\Phi_n}{1+ \Phi_n^T R_{i-1}^{-1}\Phi_n}\left[DFD(x, y, \bar{D}_{i-1}) \right] \qquad (12)$$

To avoid matrix inversion at each iteration, (12) can be simplified by ignoring the x and y cross terms in calculating ϕ_n and R Thus, from (4) and (7),

$$\begin{aligned} &\Phi_n(x) = ED \quad \text{and} \quad \Phi_n(y) = LD \\ &R(x) = \frac{1}{M}\sum_m ED_m^2 \ \text{and} \ R(y) = \frac{1}{M}\sum_m LD_m^2 \end{aligned} \qquad (13)$$

Applying (13) to (12), the components of the motion displacement estimates are,

$$\begin{aligned} \bar{d}_i(x) &= \bar{d}_{i-1}(x) - \frac{ED}{\frac{1}{M}\sum ED^2 + ED^2}\left\{ DFD[x, y, \bar{d}_{i-1}(x)\right\} \\ \bar{d}_i(y) &= \bar{d}_{i-1}(y) - \frac{LD}{\frac{1}{M}\sum LD^2 + LD^2}\left\{ DFD[x, y, \bar{d}_{i-1}(x)]\right\} \end{aligned} \qquad (14)$$

Simulation Results: The computation involved in (14) is performed recursively. At each iteration the estimated motion displacement is applied to measure a new DFD This would first require obtaining the location of the displaced pel on the previous frame, based on the estimated components of motion displacement Since the motion estimates are expected to be non-integer, the luminance value of the displaced pel is predicted by a two dimensional interpolator which uses the four corners of the surrounding pels in a two dimensional grid. In our experiments, the DFD is measured at two locations with reference to the current pel, the pel above (i e , previous line), and the previous pel along the same line The average of the two DFD's (with equal weightings) is then used to update the displacement estimates

The ED and LD in (14) were also measured using the interpolated luminance values from the displaced previous frame For ΣED^2 and ΣLD^2 the summation includes the luminance values of five interpolated neighboring pels from the previous frame

Two video sequences, known as "Salesman" and "Suzie," were used to evaluate the performance of the proposed algorithm The format of both sequences was based on the CIF (Common Intermediate Format 352-pels by 288-lines and 30 frames/s) In addition, for the sake of comparison, we have simulated the Walker-Rao algorithm [2] The simulation results of both schemes, in terms of mean square prediction error (in dB), are shown in Figures 1 and 2 for the "Salesman" and "Suzie" sequences, respectively In these figures, we have also included the results of interframe prediction without motion compensation (i e , frame difference) The number of iterations for both schemes was 3 The above algorithm was applied to those pels whose frame difference exceeds a predefined threshold (i e $|FD| > 9$) In addition, these results were obtained using the second previous frame for prediction (i e , skipping one frame). Looking at these figures, it is clear that the proposed scheme significantly reduces the motion compensated prediction error. In terms of subjective comparisons, Figure 2 presents the motion compensated prediction error images between frames 49 and 51 of the "Suzie" sequence In these images, relatively darker or lighter patches represent the degree of inaccuracies in estimating the components of the motion displacement Comparing the two images confirms the superior performance of the proposed scheme over the modified steepest-descent algorithm, particularly in regions where the motion activities are relatively high

Conclusion: This paper proposes an efficient pel-recursive estimation technique for motion tracking and coding of moving images. The proposed algorithm has been compared with the modified steepest-descent gradient algorithm The results indicate a considerable reduction in the prediction error, particularly in regions where the motion activities are relatively high

References

[1] A N. Netravali and J D Robbins, "Motion compensated television coding- part I", Bell Syst Tech J , Vol 58, pp 631-670, Mar 1979

[2] D R Walker and K r Rao, "Improved Pel-recursive motion-estimation", IEEE Trans Commun, Vol COM-32, No 10, 1984, pp 1128-1134

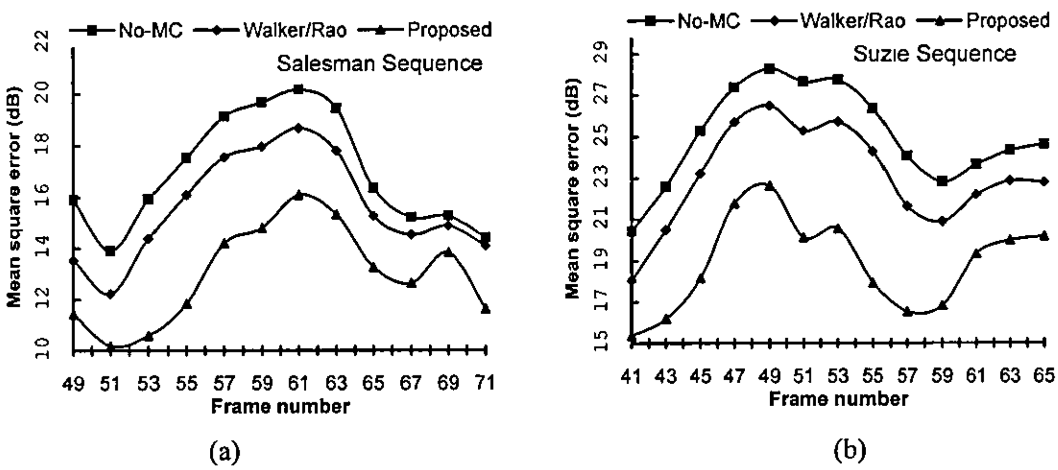


Figure 1 Mean square error performance using the second previous frame for prediction (a) Salesman sequence, (b) Suzie sequence

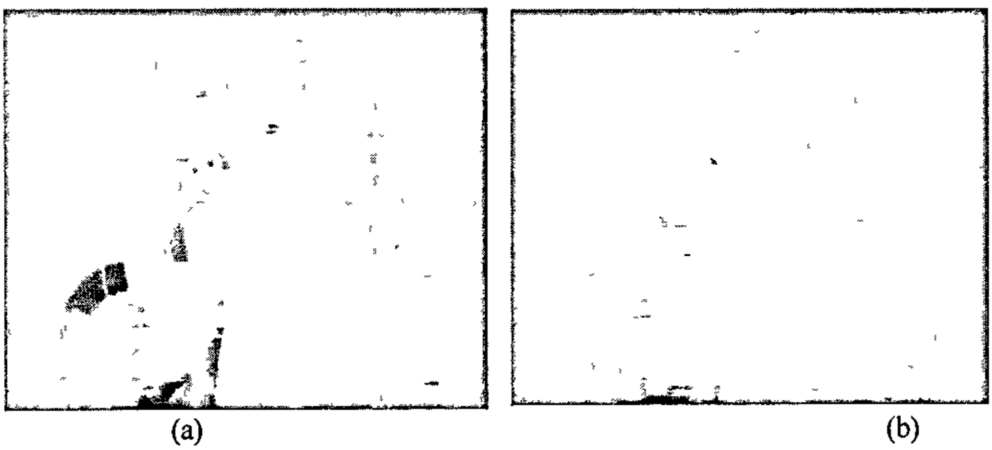


Figure 2: Motion compensated prediction error images for Suzie sequence (a): Walker & Rao scheme (b)· Proposed scheme.

Pel-recursive motion estimation algorithm

H Gharavi and H Reza-Alikhani

A new pel-recursive motion estimation algorithm for video coding applications is presented The derivation of the algorithm is based on recursive least-squares estimation that minimises the mean-square prediction error A comparison with the modified steepest-descent gradient estimation technique algorithm shows significant improvement in terms of mean-square prediction error performance

ntroduction Netravali and Robbins [1] developed a pel-recursive patio-temporal steepest-descent gradient technique in which the isplacement of a pel (picture element) was predicted from previ- usly transmitted information Since then various algorithms have een proposed to improve the performance of pel-recursive notion estimation (PRME) techniques The most important con- ribution was the modification of the steepest-descent algorithm leveloped by Walker and Rao [2] In this Letter we present a sim- le but very efficient PRME algorithm that significantly outper- orms the modified steepest-descent technique

roposed algorithm For the sake of our analysis, we assume the anslational movement of an object is in a plane parallel to the amera and illumination is uniform We also assume the effect of ncovered background to be negligible Under these assumptions, t $S(x, y, t)$ denote the monochrome intensities at point (x, y) of a noving object in the image plane where its translational move- ent is at a constant velocity of v_x and v_y We can show that after t second (one frame period), the object moves to a new location here we can show

$$S(x,y \quad t+\Delta t) = S[(x+v_x\Delta t),(y+v_y\Delta t) \quad t] \qquad (1)$$

fter expanding the field in a power series in Δt and neglecting the igher-order terms, the frame difference can be shown as

$$S(x,y \quad t+\Delta t)-S(x,y \quad t) = \frac{\partial}{\partial x}S(x,y \quad t)d_x+\frac{\partial}{\partial y}S(x,y \quad t)d_y \qquad (2)$$

here d_x and d_y correspond to the horizontal and vertical compo- ents of the motion vector D Assuming $\partial/\partial x\, S(x, y \quad t)$ and $\partial/\partial v$ $(x, y \quad t)$ are known for each x, y, t, and defining ED, LD, and D as the magnitude of the element, line, and frame difference at oint n, from eqn 3, we can write

$$FD = \Phi_n^T D \qquad (3)$$

here

$$\Phi_n = \begin{bmatrix} \frac{\partial}{\partial x}S(x_n,y_n \quad t) \\ \frac{\partial}{\partial y}S(x_n,y_n \quad t) \end{bmatrix} = \begin{bmatrix} ED \\ LD \end{bmatrix} \qquad (4)$$

rom eqn 4 the frame difference (FD) measurement is

$$\xi_n = \Phi_n^T \bar{D} + \text{noise} \qquad (5)$$

here $\bar{D} = [\bar{d}(x), \bar{d}(y)]^T$ is the motion vector estimate

For a cluster of M moving pels, the least-squares estimate of D, ter carrying out the minimisation, can be shown as

$$\sum_{n=1}^{m}\Phi_n\xi_n = \bar{D}\sum_{n=1}^{m}\Phi_n\Phi_n^T \qquad (6)$$

r,

$$\eta = \frac{1}{M}\sum_{n=1}^{M}\Phi_n\xi_n \quad \text{and} \quad R = \frac{1}{M}\sum_{n=1}^{M}\Phi_n\Phi_n^T \qquad (7)$$

e estimated motion vector from eqn 6 is obtained as

$$\bar{D} = R^{-1}\eta \qquad (8)$$

r recursive estimation of η and R, we can write

$$\eta_i = \eta_{i-1} + \Phi_n\xi_n$$
$$R_i = R_{i-1} + \Phi_n\Phi_n^T \qquad (9)$$

Based on the so-called matrix inversion lemma, the inverse of R_i n be obtained as

$$R_i^{-1} = R_{i-1}^{-1} - \frac{R_{i-1}^{-1}\Phi_n\Phi_n^T R_{i-1}^{-1}}{1+\Phi_n^T R_{i-1}^{-1}\Phi_n} \qquad (10)$$

From eqns 8 – 10

$$\bar{D}_i = \bar{D}_{i-1} - \frac{R_{i-1}^{-1}\Phi_n}{1+\Phi_n^T R_{i-1}^{-1}\Phi_n}(\Phi^T\bar{D}_{i-1}-\xi) \qquad (11)$$

In the above equation, the term in the right-hand-side bracket can be replaced by what is known as the displaced frame difference (DFD) Thus,

$$\bar{D}_i = \bar{D}_{i-1} - \frac{R_{i-1}^{-1}\Phi_n}{1+\Phi_n^T R_{i-1}^{-1}\Phi_n}[DFD(x,y,\bar{D}_{i-1})] \qquad (12)$$

To avoid matrix inversion at each iteration, eqn 12 can be simplified by ignoring the x and y cross terms in calculating ϕ_n and R Thus, from eqns 4 and 7,

$$\Phi_n(x) = ED \quad \text{and} \quad \Phi_n(y) = LD$$
$$R(x) = \frac{1}{M}\sum_m ED_m^2 \quad \text{and} \quad R(y) = \frac{1}{M}\sum_m LD_m^2 \qquad (13)$$

Applying eqn 13 to eqn 12, the components of the motion displacement estimates are

$$\bar{d}_i(x) = \bar{d}_{i-1}(x) - \frac{ED}{\frac{1}{M}\sum ED^2+ED^2}\{DFD[x,y,\bar{d}_{i-1}(x)]\}$$
$$\bar{d}_i(y) = \bar{d}_{i-1}(y) - \frac{LD}{\frac{1}{M}\sum LD^2+LD^2}\{DFD[x,y,\bar{d}_{i-1}(x)]\} \qquad (14)$$

Simulation results The computation involved in eqn 14 is performed recursively At each iteration the estimated motion displacement is applied to measure a new DFD This would first require obtaining the location of the displaced pel on the previous frame, based on the estimated components of motion displacement Since the motion estimates are expected to be non-integer, the luminance value of the displaced pel is predicted by a two-dimensional interpolator which uses the four corners of the surrounding pels in a two-dimensional grid In our experiments, the DFD is measured at two locations with reference to the current pel, the pel above (i e previous line), and the previous pel along the same line The average of the two DFDs (with equal weightings) is then used to update the displacement estimates

The ED and LD in eqn 14 were also measured using the interpolated luminance values from the displaced previous frame For ΣED^2 and ΣLD^2 the summation includes the luminance values of five interpolated neighbouring pels from the previous frame

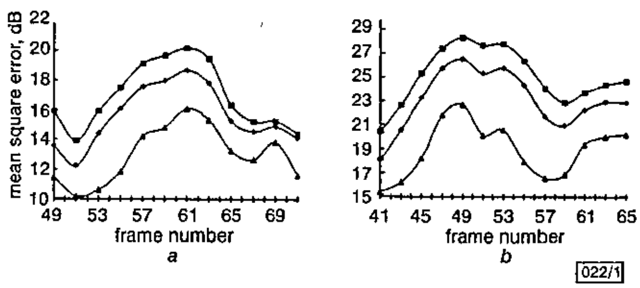
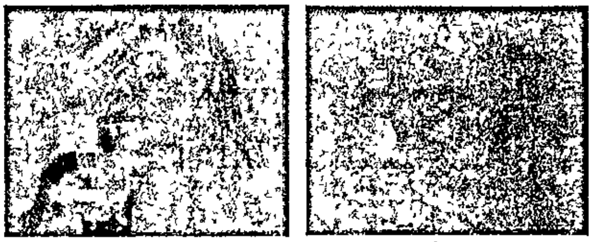


Fig 1 *Mean square error performance using second previous frame for prediction*

a Salesman sequence
b Suzie sequence
—■— no motion compensation
—◆— Walker-Rao algorithm
—▲— proposed algorithm

Two video sequences, known as 'Salesman' and 'Suzie', were used to evaluate the performance of the proposed algorithm The format of both sequences was based on the common intermediate format (CIF) 352 pels by 288 lines and 30 frames/s) In addition, for the sake of comparison, we have simulated the Walker-Rao algorithm [2] The simulation results of both schemes, in terms of mean-square prediction error (in dB), are shown in Figs 1*a* and *b* for the 'Salesman' and 'Suzie' sequences, respectively We have also included the results of interframe prediction without motion compensation (i.e. frame difference). The number of iterations for both schemes was three. The above algorithm was applied to those pels the frame difference of which exceeds a predefined threshold (i.e. $|FD| > 9$). In addition, these results were obtained using the second previous frame for prediction (i.e. skipping one frame). It is clear that the proposed scheme significantly reduces the motion compensated prediction error. In terms of subjective comparisons, Fig. 2 presents the motion compensated prediction error images between frames 49 and 51 of the 'Suzie' sequence. In these images, relatively darker or lighter patches represent the degree of inaccuracies in estimating the components of the motion displacement. Comparing the two images confirms the superior performance of the proposed scheme over the modified steepest-descent algorithm, particularly in regions where the motion activities are relatively high.



a b

022/2

Fig. 2 *Motion compensated prediction error images for Suzie sequence*

a Walker-Rao algorithm
b Proposed algorithm

Conclusion: We have proposed an efficient pel-recursive estimation technique for motion tracking and coding of moving images. The proposed algorithm has been compared with the modified steepest-descent gradient algorithm. The results indicate a considerable reduction in the prediction error, particularly in regions where the motion activities are relatively high.

10 April 2001
Electronics Letters Online No: 20010879
DOI: 10.1049/el:20010879

H. Gharavi (*National Institute of Standards & Technology (NIST), US Department of Commerce, 100 Bureau DR Stop8920, Gaithersburg, MD 20899-8920, USA*)

E-mail: Gharavi@nist.gov

H. Reza-Alikhani (*Loughborough University, Loughborough, Leics LE11 3TU, United Kingdom*)

References

1 NETRAVALI, A.N., and ROBBINS, J.D.: 'Motion compensated television coding – part I', *Bell Syst. Tech. J.*, 1979, **58**, pp. 631–670

2 WALKER, D.R., and RAO, K.R.: 'Improved pel-recursive motion-estimation', *IEEE Trans. Commun.*, 1984, **COM-32**, (10), pp. 1128–1134

Montgomery residue number systems

B.J. Phillips

The Montgomery residue number system (MRNS) for long word-length arithmetic is introduced. MRNS, a modification of the residue number system (RNS), represents a long integer as a set of smaller Montgomery residues. Long integer addition, subtraction and multiplication can then be performed using hardware-efficient Montgomery operations applied independently to each of the residues. An MRNS hardware architecture suitable for incorporation on a microprocessor data path is also proposed.

Background: Residue number systems (RNS) have long been considered an efficient means of performing long word-length addition, subtraction and multiplication [1]. Recent efforts have succeeded in reducing the cost of RNS modular multiplication and reignited interest in RNS, especially for the implementation of public-key cryptosystems [2]. The Montgomery residue number system (MRNS) described in this Letter is a modification to RNS that permits the use of hardware-efficient Montgomery multiplication and reduction [3].

Residue number systems: In RNS a number X is represented by its residues modulo a set of co-prime moduli $\{m_{k-1}, \ldots, m_1, m_0\}$. We write $X = (x_{k-1}| \ldots |x_1|x_0)_{RNS(m_{k-1}| \ldots |m_1|m_0)}$ where $x_i = X \bmod m_i = \langle X \rangle_{m_i}$. The dynamic range of the RNS (the number of different values that can be represented) is given by $M = \prod_{i=0}^{k-1} m_i$.

Addition, subtraction and multiplication can be performed within RNS by operating on each of the k residues independently:

$$X + Y = \big(\langle x_{k-1} + y_{k-1}\rangle_{m_{k-1}}| \ldots |\langle x_1 + y_1\rangle_{m_1}|\langle x_0 + y_0\rangle_{m_0}\big)_{RNS(m_{k-1}| \ldots |m_1|m_0)}$$

$$X - Y = \big(\langle x_{k-1} - y_{k-1}\rangle_{m_{k-1}}| \ldots |\langle x_1 - y_1\rangle_{m_1}|\langle x_0 - y_0\rangle_{m_0}\big)_{RNS(m_{k-1}| \ldots |m_1|m_0)}$$

$$X \times Y = \big(\langle x_{k-1} \times y_{k-1}\rangle_{m_{k-1}}| \ldots |\langle x_1 \times y_1\rangle_{m_1}|\langle x_0 \times y_0\rangle_{m_0}\big)_{RNS(m_{k-1}| \ldots |m_1|m_0)}$$

Montgomery residues: As discussed in subsequent Sections, Montgomery's reduction method [3] is an alternative to full modular reduction with advantages for hardware implementations. For now, let us concentrate on the mathematical formulation of Montgomery reduction and begin by defining a Montgomery residue $\overline{x_i}$ thus: $\overline{x_i} = x_i r_i \bmod m_i$. Montgomery reduction is the function $MR_{m_i,r_i}(x_i) = x_i r_i^{-1} \bmod m_i$ so that $MR_{m_i,r_i}(\overline{x_i}) = x_i \bmod m_i$. The Montgomery residue $\overline{x_i}$ is unique for each residue x_i provided $r_i > m_i$ and m_i and r_i are co-prime numbers [3]. Therefore, for every representation within a residue number system, there is an equivalent representation in the Montgomery residue number system thus:

$$X = (x_{k-1}| \ldots |x_1|x_0)_{RNS(m_{k-1}| \ldots |m_1|m_0)}$$
$$= (\overline{x_{k-1}}| \ldots |\overline{x_1}|\overline{x_0})_{MRNS(m_{k-1}| \ldots |m_1|m_0)}$$

MRNS operations: The Montgomery sum, difference and product functions can be defined as:

$$MS_{m_i,r_i}(\overline{x_i}, \overline{y_i}) = \overline{x_i} + \overline{y_i} \bmod m_i$$
$$MD_{m_i,r_i}(\overline{x_i}, \overline{y_i}) = \overline{x_i} - \overline{y_i} \bmod m_i$$
$$MP_{m_i,r_i}(\overline{x_i}, \overline{y_i}) = MR_{m_i,r_i}(\overline{x_i} \times \overline{y_i}) = x_i y_i r_i \bmod m_i$$

Note that if $z_i = x_i + y_i \bmod m_i$ then $\overline{z_i} = MS_{m_i,r_i}(\overline{x_i}, \overline{y_i})$, if $z_i = x_i - y_i \bmod m_i$ then $\overline{z_i} = MD_{m_i,r_i}(\overline{x_i}, \overline{y_i})$, and if $z_i = x_i \times y_i \bmod m_i$ then $\overline{z_i} = MP_{m_i,r_i}(\overline{x_i}, \overline{y_i})$. Also note that the Montgomery sum and difference functions are identical to full modular addition and subtraction but that the product function makes use of Montgomery reduction.

Using these functions addition, subtraction and multiplication can be performed directly on numbers in MRNS representation:

$$X + Y = \big(MS_{m_{k-1},r_{k-1}}(\overline{x_{k-1}}, \overline{y_{k-1}})| \ldots |MS_{m_1,r_1}(\overline{x_1}, \overline{y_1})|MS_{m_0,r_0}(\overline{x_0}, \overline{y_0})\big)_{MRNS(m_{k-1}| \ldots |m_1|m_0)}$$

$$X - Y = \big(MD_{m_{k-1},r_{k-1}}(\overline{x_{k-1}}, \overline{y_{k-1}})| \ldots |MD_{m_1,r_1}(\overline{x_1}, \overline{y_1})|MD_{m_0,r_0}(\overline{x_0}, \overline{y_0})\big)_{MRNS(m_{k-1}| \ldots |m_1|m_0)}$$

$$X \times Y = \big(MP_{m_{k-1},r_{k-1}}(\overline{x_{k-1}}, \overline{y_{k-1}})| \ldots |MP_{m_1,r_1}(\overline{x_1}, \overline{y_1})|MP_{m_0,r_0}(\overline{x_0}, \overline{y_0})\big)_{MRNS(m_{k-1}| \ldots |m_1|m_0)}$$

Converting to and from MRNS: Conversion between MRNS and RNS can be straightforwardly accomplished by converting each of the k residues using $x_i = MR_{m_i,r_i}(\overline{x_i})$ or $\overline{x_i} = MP_{m_i,r_i}(x_i, r_i^2 \bmod m_i)$. Note that in the latter equation $r_i^2 \bmod m_i$ may be pre-computed.

It is also possible to convert directly to MRNS using a sum of pre-computed residues approach. If we take an n-bit number X in a multi-precision form as w-bit words

$$X = \sum_{j=0}^{n/w-1} X_j 2^{j \times w}$$

Table of contents

CHAPTER THREE

A PEL-RECURSIVE TECHNIQUES

CHAPTER FOUR

NEW PEL-RECURSIVE TECHNIQUE (Simplıfied Kalman Technique)

CHAPTER FIVE

COMBINED BLOCK MATCHING AND PEL-RECURSIVE TECHNIQUES

CHAPTER SIX

APPLICATION TO A HIERARCHICAL CODING SYSTEM

CHAPTER SEVEN

CONCLUSIONS AND FUTURE WORK

Chapter 1

Introduction

1.1 General Overview and Background

Of all the technological achievements in the 20th century, broadcast television has assumed a dominant role and has shown a great usage and effect in our everyday life to such an extent that today in the U.S. there are more homes that contain a television set than have telephone service

Television has perhaps had the greatest effect on our everyday lives. For many people, a television set is an obscure box in the corner of their living room - providing education, entertainment, and etc. Children are now said to be addicted to it and there is no doubt that the nature of leisure time activities has radically changed over the past thirty years to accommodate television. Telecommunications systems have also invaded our home People can now hold a telephone conversation as comfortably as they would do face to face Going further it became possible to combine pictures and sound and transmit them via telephone lines for video conferencing.

The telecommunications system has been able to take advantage of new technology enabling modern digital network to become available to everyone Recent advances in mobile communications have shown high potential for telephones to be associated with individuals rather than their home and offices.

But the evolution in television and telecommunications systems has followed different paths Since the introduction of color television in the 1950's, there have been no significant changes to the mechanism of picture transmission and display. The difficulty in modifying the television signal that is broadcasted for local distribution is that the television receiver would almost certainly need to be modified or replaced. The difficulty of achieving this with an invested base of over $10 billion is staggering

In this country, the 625 line format has been with us for a long time and for many people, their perception of improvements in the quality of television has been assisted by advances in associated audio reproduction, particularly since the advent of NICAM digital stereo The telecommunications system, on the other hand, has been able to take advantage of new technology to provide a modern, digital network, available to the users. So it is natural that in thinking of television transmission we immediately think of the signal that is broadcast into the home. More efficient encoding of this signal would free valuable spectrum space.

There is a large amount of point-to-point transmission of picture material taking place today in addition to UHF/VHF broadcasting. For example, each of the four U.S television networks has a distribution system spanning the whole of the continental United States, international satellite links transmit live programs around the world. Video-conferencing services are receiving increasing attention. Satellites are transmitting to earth a continuous stream of weather photographs and earth-resource pictures, and there are a number of important military applications such as the control of remotely piloted vehicles and so on

Taking account of this background, it is perhaps surprising that the concept of combining pictures and sound into a single PSTN channel for video conferencing has taken so long to evolve. The essential difficulty is that bandwidth is limited in the services provided by telephone companies on the basis that to transmit speech, only 4 kHz is required for acceptable quality. Broadcast quality digital television, on the other hand, in comparison with a digitized speech signal at 64 kb/s, requires over 100 Mbits/s to supply pictures Even existing terrestrial channels allocated for television cannot accommodate this amount of data. Consequently, video compression and coding appear to be the best approach to the problem, until someone provides a mass communications system in which bandwidth is not a limitation Further more efficient coding of picture material for these applications provides the opportunity for significantly decreasing transmission costs, these costs can be quite large. The aim of efficient coding is to reduce the required transmission rate of a given picture quality so as to yield a reduction in transmission costs.

Some early efforts in picture coding used analog coding techniques and attempted to reduce the required analog bandwidth, giving rise to the term "bandwidth compression". Complex manipulations of the signal are today much more easily done by first sampling and digitizing the signal and then processing the signal in the digital domain rather than using analog techniques

Ideally, one would like to take advantage of any structure (both geometric and statistical) in a picture signal to increase the efficiency of the encoding operation Also the coding process should take into consideration the resolution (amplitude, spatial, and temporal) requirements of the receiver, i e., the television display and very often the human viewer [1].

International co-operation has proved important in the development of video codec algorithms. Under the auspices of the CCITT, now known as the International Telecommunication Union (ITU), a recommendation was published in 1990, describing the framework of a video codec intended for use on the ISDN system on channels of 64kbits/s Its primary concern is the removal of redundancy, which occurs within and between picture frames Intraframe coding can be used to compress a single frame and redundancy is said to be present where the picture comprises of groups of adjacent equal value picture elements, or pixels. Similarly, where pixel values have not changed over time, interframe coding can remove temporal redundancy. Only changes in picture content need to be supplied to the decoder and, as a result, an efficient mechanism of picture coding is developed.

In most cases, however, video codecs are said to be lossy, since additional processing tends to lower the resolution and introduce errors This said, provided certain requirements of quality are kept, most users are unable to detect coding errors and those who do will probably be able to tolerate them.

The implementation of video codecs has also been limited by the technology available. Where real-time processing is required, compression and coding must be performed at high speed - a requirement that VLSI technology has recently appeared to be able to satisfy A new generation of software video codecs is being proposed in current ITU recommendations, to work on the growing number of personal computers connected to the PSTN by a modem As the processes are refined and the technology is improved, video conferencing codecs will become less expensive and more widely available. Whether they become more popular is, however, a different matter. It took many years for televisions and telephones to get into most homes and wariness about seeing the person the user is talking to may, for some while, make the videophone something the public feels it can do without

1.2 Aims, Motivations , Objectives and the Scope Of The Research

This thesis examines the current state of video technology and assesses different aspects of video compression. Further it goes into developing new ways of motion estimation The combined new proposed algorithm with block matching is to contribute a higher performance to the existing algorithm which in time could perhaps given rise to an alternative standard.

1.3 Structure and outline of this Thesis

Chapter two provides the reader with an insight into contemporary techniques of video compression. Although motion compensation is a very wide-ranging topic, chapter 2 concentrates on the principles of DPCM and block matching motion compensation This chapter also considers the ISO/MPEG standard and comes up to date with the latest H 263 recommendation for very low bitrate video codecs, using the framework of the H 263 algorithm.

Chapter three Analyses the state of the art techniques of another class of image compression known as pel-recursive motion compensation with focus on the pel-recursive Wiener-based displacement estimation algorithm.

Chapter four Investigates the novel techniques of displacement estimation algorithm in comparison to existing techniques.

Chapter five shows experimental results illustrating the performance of a few applications applying the proposed idea and method to some degree.

Chapter six examines the novel idea of combining the two different classes of image compression, the block matching motion estimation and pel-recursive motion estimation, into a Hybrid system.

Chapter seven concludes the thesis with a summary and provides conclusions drawn from this work. Also suggestions for further work are made, particularly in the area of image compression, expressing the trade-off between quality and compression complexity which could outline and open up further avenues of research

Chapter 2

Review of contemporary techniques

2.1 Introduction

Over fifty years have passed since the introduction of broadcast television in the United Kingdom. However, it is only recently that the concept of using moving pictures for interactive video and multimedia has received interest, as the costs of transmitting a television signal over anything other than short distances have proved prohibitive We have been limited to sending mainly still images over the public telephone network, mainly due to restriction in the bandwidth available to most users

It seems paradoxical that whilst the technology of digital television has advanced in remarkable leaps in recent years, we still have no efficient, widespread means of sending high quality video over the telephone network for the purposes of videotelephones One of the fundamental costs of colour television is the bandwidth required to transmit a channel of sound and pictures The five terrestrial channels allocated in the United Kingdom have equivalent digital bandwidths from 12 to 24Mbits/s, which would be insufficient to carry sound, chrominance (colour) and luminance (brightness) signals without any form of compression It is the scarcity of space in the radio-frequency spectrum that has limited the extent of broadcast television

In its uncompressed state, conventional broadcast-quality digital television requires bit rates of typically 166 megabits per second - well over that available for the 2 Mb/s link Integrated Services Digital Network (ISDN) [2] channel and it is not economical to use 155 0r 622 Mb/s links. Given this primary constraint, contemporary research has focused on the compression of video images, allowing transmission of low resolution images over digital networks. In most cases, compression is easy to achieve, removing spatial and temporal redundancies naturally occurring in sequences of images.

Figure 2.1.1 A frame of Suzie, demonstrating picture redundancies.

Consider the image of figure 2.1.1. This could be regarded as typical of a videoconferencing scene, where during the conversation, most of the picture will not change other than, say, the lips, eyes and occasional hand or head movements. This feature can be used to good effect, such that only information about differences that have occurred will need to be sent to the recipient. This process is called *interframe* coding and is ideal for the low level of temporal changes, associated with videoconferencing. Interframe coding is based on the fact that there exists a large amount of frame-to-frame correlation in moving images, which is also called temporal correlation. It is also possible to extract information about differences between spatially adjacent pixels at a given instant in time. This process is called *intraframe* coding and efficiently compresses large areas of consistent colour and shade (the plain background in this example). Boundaries are easy to detect, where significant changes in luminance and chrominance occur Interframe and intraframe coding are two methods of redundancy compression that have been used to good effect in the development of videoconferencing hardware for transmission over telecommunications channels

The intrinsic effect of redundancy coding is not to reduce picture quality significantly, or to affect spatial resolution However, subsequent processing of the difference information can take place, where useful information can be described as those aspects of the image that convey meaning to the human viewer, even if that is only a small proportion of the image content The contrast sensitivity function [3] allows understanding of the human ability to detect spatial and temporal detail Assuming the human eye can resolve down to two minutes of an arc, it can take in the equivalent of a million pixels of information without moving By moving the eye, but not the head, the field of view is at least an order of magnitude greater We know the head is likely to remain stationary whilst a person is doing something specific, but the eyes are moving continuously. If we assume that to represent the colour and luminance of a pixel, 12 bits are required, over 100 million bits of information are needed to represent the user's static scene

Consideration of these factors gives an understanding of the essential nature of video compression algorithms. It is necessary to take a picture, which under normal circumstances would require extensive data representation, and code it to the constraints of, telecommunications network, whilst maintaining an image satisfactory to the human perception

At an early stage, the international telecommunications community identified the need for close collaboration to ensure the adoption of a system which could be applied in all countries and make videotelephony available to a world market. Even though a European standard specification did emerge in the 1980's [4], for a 2Mbits/s, 625 line, 25 frames per second PAL system, demand in North America required plans using the 525 line, 30 frames per second NTSC system Subsequently, the conversion between these standards was regarded as the focal point of international co-operation and under the auspices of the Organization now known as the International Telecommunication

Union * (the ITU), a videophone algorithm was recommended, meeting the needs of the new ISDN systems and working for all bit rates between 64kbits/s and 2Mbits/s

The resulting ITU-T Recommendation, H 261 [5][6], forms the basis of the international development of videoconferencing systems using the new ISDN networks being installed throughout the world However, many concepts used are equally applicable to other areas of video codec design, such as high-definition digital television (HDTV), where an increased amount of picture data is to be transmitted within the constraints of existing terrestrial bandwidths

* The International Telecommunication Union was formed from an amalgamation of the CCITT and the CCIR

2.2 Review of Image Compression

In image transmission and storage, digital techniques instead of analog are increasingly used due to the rapid growth in the use of digital computers, and the declining cost of digital processing and transmitting equipments This is also because the digital transmission and storage system has many inherent advantages over the analog system, such as easy processing, processing flexibility, easy and random access in storage, higher signal-to-noise ratio (SNR), possibility of errorless transmission etc. However, images, whether digital or analog, contain a large amount of information and require wideband channels for transmission and big memory for storage, especially digital images. For example, a 4MHz television signal sampled at Nyquist rate with 8 bit samples could require a transmitting bandwidth of 64 MHz. Therefore it is highly desirable to compress image data for transmission and storage A lot of techniques for digital image compression [7] [8] have been developed.

The statistical properties of images are the main reasons that images can be compressed The statistical property upon which intraframe coding techniques rely is the high correlation between neighboring pixels This means that adjacent pixels are usually similar to one another and the magnitude of a pixel may be estimated from the values of the pixels around it Most images, even fairly active images which contain a large amount of spatial detail, have quite high values of correlation For example, in moving images, the background is likely to remain stationary in successive frames The correlation in one frame of an image or successive frames are image data redundancies which can be reduced without apparent degradation of image quality

Image compression techniques can be classified into two classes, namely information lossless and information lossy techniques The former is able to reconstruct the original image without any loss of information, whereas the later introduces some distortion in the reconstructed image and cannot recover the original image exactly which can not be perceived by human eyes. Lossless and lossy compression techniques are used in different applications For example, medical images often require completely lossless compression because any slight distortion may result in wrong diagnosis In other applications, such as entertainment, education etc , the reconstructed images need not necessarily be exactly the same as the original ones and lossy compression techniques are then widely used The lossless techniques normally reach lower compression ratio while the lossy techniques can reach higher compression ratio.

2.3 Transform Coding Technique

One of the most effective image compression techniques is transform coding The basic motivation and fundamental principle behind transform coding [9] [10] is to transform the image from the data domain to a frequency domain by an energy preserving unitary transform In the frequency domain, the image pixels are decorrelated and the energy is concentrated on a few coefficients so that the high frequency coefficients and the coefficients with less energy can be removed without any visual effect on the reconstructed image, since they play less important roles in the image reconstruction The transform could be applied to the entire image but implementation problems make this impractical First, the amount of the memory and the computation required increase proportionally to M^2, where M is the image dimension. Second, because of the elimination of unimportant coefficients, small transform size often leads to more significant degradation than a large size A typical approach is to divide the image into a number of rectangular blocks or sub-images, normally the input image is partitioned into N x N (e g 8 x 8 or 16 x 16) blocks (sub-images), and then an unitary transform is applied to each sub-image A block size 8 by 8 has been adopted for most video coding standards mainly to reduce the transformation complexity as well as better exploitation of image redundancies between the neighboring blocks

After the transformation, actual image compression is achieved by quantizing the transform coefficients. If all the coefficients are quantized and coded, the compression ratio is quite small. It has been pointed out that the important characteristic of the transform is that most of the energy of the image is packed into a small number of low frequency coefficients and the coefficients with less energy or the high frequency coefficients play less important roles in the image reconstruction To achieve higher compression, one possibility is to use a mask covering an area of low frequency coefficients and to discard the remaining coefficients, i e. set the remaining coefficients to zero Only those coefficients in the mask are quantized and coded Considerable compression can be achieved depending on the size of the mask used in this method This technique is known as zonal coding The only problem with the zonal coding approach is a blurring effect as a result of the elimination of higher frequency components Another possibility is to use a threshold on each transform coefficient and set the coefficients which are below the threshold value to zero The remaining non-zero coefficients together with their address information are quantized and finally entropy coded efficiently by coding schemes such as, Huffman coding [11], arithmetic coding [12] or combining Variable Length Coding (VLC) and runlength coding For better subjective image quality, the quantizer in all cases should be designed to optimize the reconstructed image quality for a given number of bits

The encoded image is transmitted through the channel (or stored) An inverse operation is performed at the receiver end. A number of orthogonal transforms can be used in the transform coding and most of them are linear transformations

Transform coding has a good immunity to channel noise. In transform coding, a code error in transmission only influences the corresponding block and has no effect on the succeeding blocks because this error is distributed by the reverse transform over the entire block. Visually, a code error in the transform coding is less visible than that in predictive coding However, the transform coding has some defects First, since the image is divided into blocks, block to block correlation is not employed in the transform Furthermore, artificial blocking segments the image arbitrarily without considering its contents Second, in transform coding at low bit rate, sometimes so called blocking effects are apparent in the reconstructed image Blocking effects are perceived in the reconstructed image as visible discontinuities between adjacent blocks This is especially visible around the boundaries of moving objects and, still background This is caused by the improper coding of the transform coefficients, such as eliminating too many coefficients or due to coarse quantization Finally, transform coding needs more operations and memory than predictive coding This is improved due to the rapidly decreasing cost of digital hardware and computer memory, and this may no longer be a disadvantage

2.3.1 The Karhunen-Loeve Transform (KLT) Technique

The Karhunen-Loeve transform [10] is an optimal linear transformation in the sense that it completely decorrelates the data and maximizes the amount of energy compacted into the lowest order coefficients However, it is not certain that the KLT is the absolute optimum transform since it does not consider other factors such as the human visual system Additionally, the transform matrix depends on the image data, i e the transform matrix is different for different image data Thus, the KLT transform matrices are also transmitted and stored along with the coded data. Furthermore, the amount of computation in the transform matrix generation is very large and the KLT has no fast transformation algorithm associated with it.

Because of the computation complexity, the large storage requirement and dependence on the input images, the KLT is seldom used in practice but it is employed in theoretical studies of image coding It gives an indication about the upper bound computational efficiency of what other transformations should attempt to reach for decorrelating data samples.

2.3.2 The Discrete Fourier Transform (DFT) Technique

The discrete Fourier transform [10] is naturally applied to image coding because of its widespread use in other signal processing fields and the fact that it has efficient computational algorithms and fast implementation It is the only complex transform used in data coding schemes The DFT is not convenient for general use due to the necessity to evaluate both real and imaginary components, which require a large number of operations and large storage

2.3.3 The Walsh-Hadamard Transform (WHT) Technique

The Walsh-Hadamard transform [10][13] is the simplest transform among various types of orthogonal transforms The elements in the transform matrix are either 1 and -1, and the only multiplication needed is that of the final scaling operation However, it is too simple to compact energy well

2.3.4 The Discrete Cosine Transform (DCT) Technique

The Discrete Cosine Transform (DCT), which is an information lossless technique [10][14][15] was proposed by Ahmed et al 1974 It is one of an extensive family of sinusoidal transforms At that time, there was increasing interests in the class of orthogonal transforms, such as the discrete Fourier transform, the Hadamard transform, in the general area of digital signal processing, such as image coding, pattern recognition etc. It is known that the KLT is the optimal transform with respect to performance measure, but it needs a large amount of computation and has no fast transform Compared with other orthogonal transforms, the DCT has the best all-around performance with respect to efficient computation and acceptable perceptual quality for a given compression rate. It also has correlation reduction capability, good energy compaction and fast computational properties [16] It is a widely used transformation for image compression for example in JPEG still-image compression standard Therefore, researchers tried to develop a transform which is close to the performance of the KLT and has fast algorithms To fill the role, the discrete cosine transform was proposed

The two-dimensional discrete cosine transform of a data sequence $X(x, y)$ is defined as

$$F(u,v) = \frac{1}{N}\sum_{x=0}^{N-1}\sum_{y=0}^{N-1} C(u)C(v)X(x,y)\cos\left[\frac{(2x+1)u\pi}{2N}\right]\cos\left[\frac{(2y+1)v\pi}{2N}\right]$$

Where $x, y = 0, 1, \ldots\ldots, N - 1$

The inverse two-dimensional discrete cosine transform is defined as.

$$X(x,y) = \frac{1}{N}\sum_{x=0}^{N-1}\sum_{y=0}^{N-1} C(u)C(v)F(u,v)\cos\left[\frac{(2x+1)u\pi}{2N}\right]\cos\left[\frac{(2y+1)v\pi}{2N}\right]$$

Where $u, v = 0, 1, \ldots, N - 1$

It has been shown that the performance of the DCT is nearly identical to the KLT transform for blocks of reasonably large size [17] Furthermore, the empirical evidence shows that even for blocks of small size the performances of the DCT and the KLT are close It also has correlation reduction capability, minimum block distortion, superb high energy compaction and fast computational properties [18] [19] DCT is widely used transformation for image compression for example in JPEG still-image compression standard

Since the computation effort for DCT is quite large for time critical applications, fast versions of the DCT[20] [21] have been proposed Though speed performance is improved by the fast algorithms, the fast algorithms still require a large amount of computation DCT can be hardware implemented by digital signal processor to achieve high speed at reasonable cost

2.3.5 Hybrid (Transform / DPCM) Coding system

Hybrid coding [23] – [25] is a kind of technique which combines transform coding and predictive coding together to generate a better coding scheme It takes the advantages of transform coding and predictive coding and overcomes their short comings to a certain degree Generally, hybrid coding performance lies between transform coding and predictive coding This technique removes the spatial redundancies, which normally exist between the neighbouring pixels within a two dimensional image array Hybrid coding is less sensitive to channel errors than predictive coding

Typically, in hybrid coding, a two-dimensional image is unitarily transformed to obtain a sequence of one dimensional sequences Each of these sequences is then coded independently by a one dimensional predictive coding technique, such as the DPCM

2.3.6 Differential Pulse Code Modulation (DPCM)

In PCM time discrete, amplitude discrete, representation of the sample is made without removing much statistical or perceptual redundancy from the signal The time discreteness is provided by sampling the signal generally at the Nyquist rate, amplitude discreteness is provided by using a sufficient number of quantization levels so that the degradation due to quantization is not easily visible In DPCM, the sample to be encoded is predicted from the encoded values of the previously transmitted samples and only the prediction error is quantized for transmission Such an approach can be made adaptive either by changing the prediction or quantization or by not transmitting the prediction error whenever it is below a certain threshold, as in conditional replenishment.

In basic predictive coding systems [26]-[28] (Fig. 2 3.6 1) in their simplest form, DPCM uses the coded value of the previously coded horizontal information (pel) that has been transmitted as the prediction However, more sophisticated predictors, use the previous line (two-dimensional Predictor) as well as previous frame of information (interframe predictor) The error resulting from the subtraction of the prediction from the actual value of the sample is quantised into a set of discrete amplitude levels. These levels are represented as binary words of either fixed or variable length and sent to the channel coder for transmission The predictive coder has three basic components. 1) predictor, 2) quantizer, 3) entropy coding

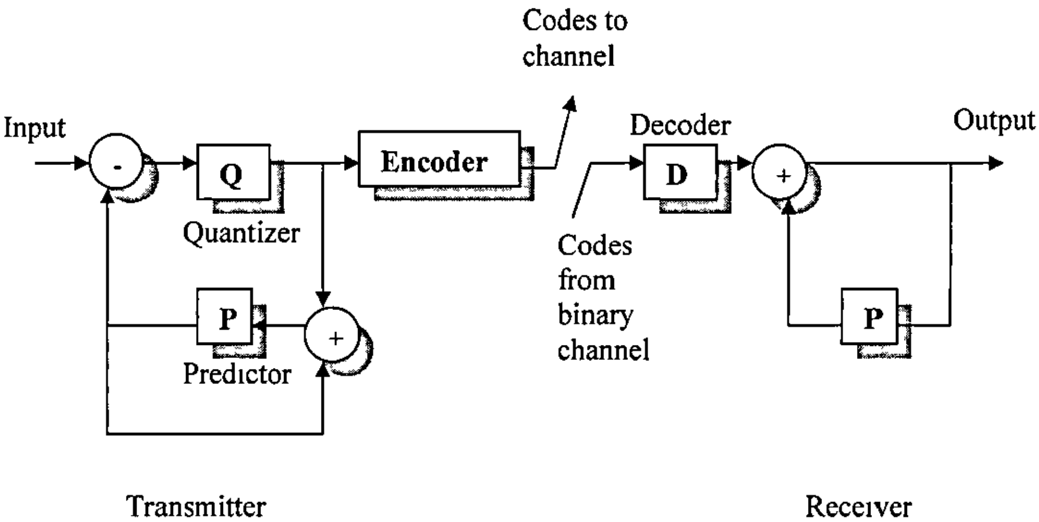


Figure 2.3.6.1 Block diagram of a DPCM transmitter and receiver.

1 - Predictors for DPCM coding can be classified as linear or nonlinear depending upon whether the prediction is a linear or a nonlinear function of the previously transmitted sample values. A further division can be made depending upon the location of the previous elements used: one-dimensional predictors use previous elements in the same line, two-dimensional predictors use elements in the previous lines as well, whereas interframe predictors use picture elements also from the previously transmitted frames Predictors can be fixed or adaptive Adaptive predictors change their characteristics as a function of the data, whereas fixed predictors maintain the same characteristics independent of the data. As an example of adaptive prediction, see Habibi [29] for predictors which use different numbers of picture elements within a frame.

The set of predictors from which a predictor is selected are usually linear and are chosen such that each one of them will give a small prediction error if the signal was correlated in a certain manner. In Graham's predictor [30]-[32], either the previous line or the previous element is used for prediction, and the switching is done by the surrounding line and element differences. Several extensions have been made to this basic philosophy However, the results have not been very encouraging in terms of the mean square error or the entropy of the prediction error In some cases the rendering of certain types of edges can be remarkably improved by these adaptive predictors Another variation [33] in adaptive prediction is to use a weighted sum of several predictors, where the weights are switched from element to element and are chosen by observing certain characteristics of already transmitted neighboring pels The same calculation can be performed at the receiver and, therefore, the predictor switching information does not need to be transmitted Such techniques have been considered for gray level signals [34].

The more successful adaptive predictors for frame-to-frame coding are the ones that take into account the motion of objects These are based on the notion that, if there are objects moving in the field of view of a television camera and if an estimate of their translation is available, then more efficient predictive coding can be performed by taking the differences of elements with respect to elements in the previous frame that are appropriately spatially translated. Such prediction has been called motion compensated prediction [35] [36] Its success obviously depends upon the amount of translational motion of objects in real television scenes and the ability of an algorithm to estimate translation with the accuracy that is desirable for good prediction One set of techniques developed [37] [38] obtain an estimate of translation in a block of pels, whereas techniques developed by Netravali et al [39]-[41], recursively adjust the translational estimate at every pel or at every small block of pels. Another approach to motion compensation is adaptive linear prediction by using elements in both the present and the previous frame (or field), which surround the element being encoded, and adapting the coefficients to minimize an intensity error function [42]. Such an approach is implementationally difficult and requires transmission of coefficients of the predictors

In scenes with higher detail and motion, field difference prediction does better than frame difference prediction [43] As the motion in the scene is increased further, intrafield predictors do better [44] This is largely because for higher motion, there is less correlation between the present pel and either the previous field or the frame pels

For the same reason, predictions such as element or line difference of frame or field differences perform better than frame or field difference for higher motion

2 - Quantization· DPCM schemes achieve compression, to a large extent, by not quantizing the prediction error as finely as the original signal itself. Several methods of optimizing quantizers have been studied Most of the work on systematic procedures for quantizer optimization were taken from studies of DPCM coding, in which the approximate horizontal slope of the input signal is quantized Three types of degradations can be seen due to the improper design of the quantizer of a DPCM coder These are referred to as granular noise, edge busyness and slope overload as shown in Fig 2 3 6 2 If the inner levels (for small magnitudes of differential signal) of the quantizer are too coarse, then the flat areas are coarsely quantized and have the appearance of random noise added to the picture On the other hand, if the dynamic range (i e , largest representative level) of the quantizer is small, then for every high contrast edge it takes several samples for the output to follow the input, resulting in slope overload, which appears similar to low-pass filtering of the image For edges whose contrast changes somewhat gradually, the quantizer output oscillates around the signal value and may change from line to line, or frame to frame, giving the appearance of a busy edge Quantizers can be designed purely on a statistical basis or by using certain psychovisual measures

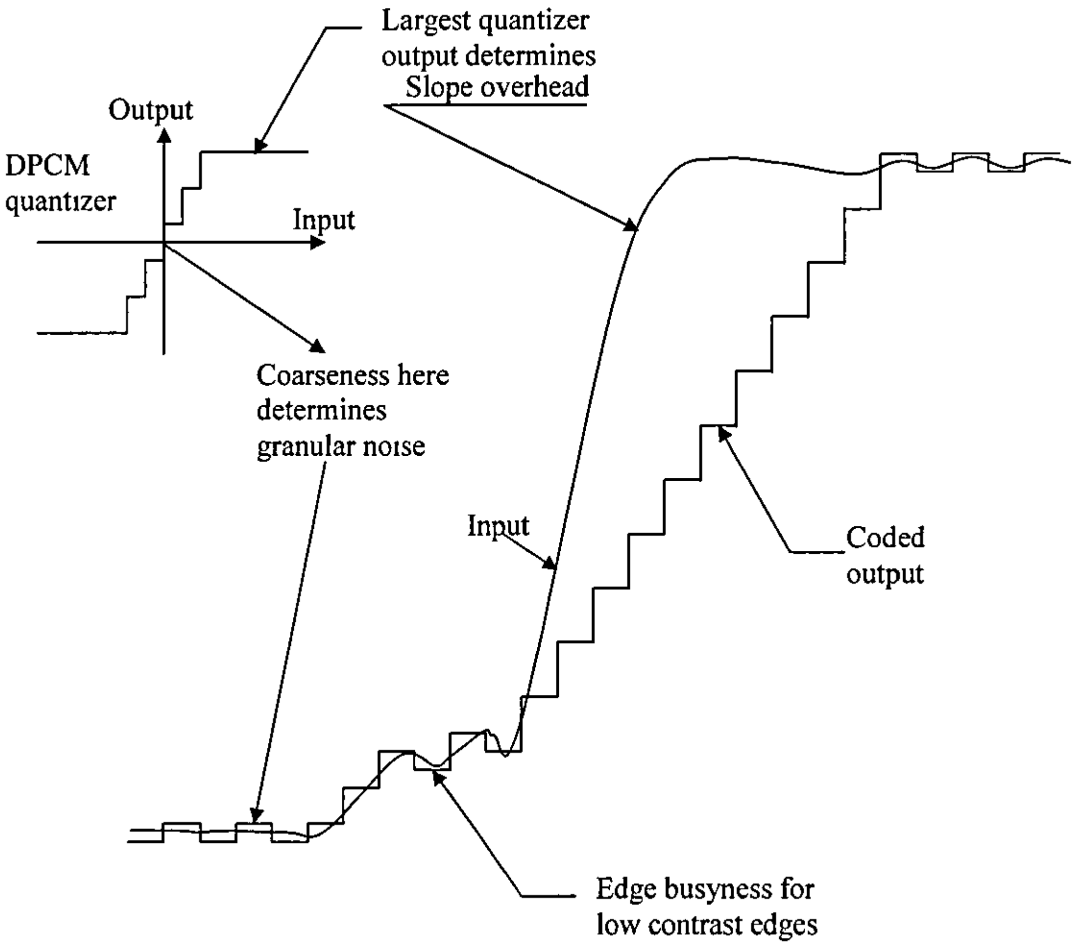


Figure 2.3.6.2 An intuitive classification distortion due to DPCM coding. (Adapted from digital picture)

It had been realized for some time that for a better picture quality, quantizers should be designed on the basis of psycho-visual criteria. However, the debate [45] [46] continues on what is a good criterion to use, and expectedly so, considering the complexities of the human visual system

3·- Entropy coding is the last stage in which shorter code word are assigned to the more frequent occurring symbols, therefore minimizing the average length of the binary representation of the input information [47] The average information rate is given by entropy (measured in bits) .-

$$H(S) = -\sum_{i=1}^{N} p(s_i)\log_2 p(s_i)$$ (Eqn. 2.3 6 1)

Where there are N input symbols $s_1, s_2, s_3, \quad , s_N$ with probabilities $p(s_1), p(s_2), p(s_3), \quad , p(s_N)$

And the average codeword length which is the average number of bits required is given by :-

$$R(S) = \sum_{i=1}^{N} l_i p(s_i)$$ (Eqn 2 3 6 2)

Where $l_1, l_2, l_3, \quad , l_N$ are the word length for the code words

Run Length coding (RLC) was first considered for black and white images The run length is found by counting the number of consecutive black and white pixels along each line, as an example for a horizontal line along an image as illustrated in figure 2.3 6 3 is 7 black-run, 3 white-run, 4 black-run, 4 white-run Where 0 and 1 represents black and white pixels respectively

000000011100001111

Figure 2.3.6.3 An example of runtength coding.

This runlength coding method has been further developed as two-dimensional Variable Length Coding (2D VLC) [48] so that colour images can be encoded The image is encoded as an EVENT. Each EVENT contains RUN and LEVEL.

EVENT = (RUN, LEVEL)

Where RUN is the number of successive zeros preceding the quantised coefficient

LEVEL is the non zero value for the quantised coefficient

Finally, 3D VLC [48] is developed to improve the coding efficiency. In this approach, each EVENT contains LAST, RUN, LEVEL The LAST event is represented by the

End of Block (EOB) which indicates that no more zero coefficients are encoded for this block

EVENT = (LAST, RUN, LEVEL)

Where LAST = 0	there are more non zero coefficients in this block.
LAST = 1	this is the last non zero coefficient in this block
RUN	is the number of successive zeros preceding the quantised coefficient
LEVEL	is the non zero value of the quantised coefficient

The limitation of this method is the complexity of constructing the codebook However, it is very efficient in terms of coding and has been adopted as part of the ITU-T H.263 Coding Standard [49].

Another problem with the use of variable length codes is that the output rate from the source coder changes with local picture content In order to send such a signal over a constant bit rate channel, the source coder output has to be held temporarily in a buffer which can accept inputs at a non uniform rate and can be read out to the channel at a uniform rate

2.4 Block Matching

In block matching motion estimation the coding (current) frame is partitioned into small non-overlapping blocks of size m x n (where often m = n), assuming that all the pixels within each of the non-overlapped block have the same displacement vector It is assumed that the motion is purely translational The motion vector for each block is estimated by searching through a larger block (search window of size m+2u x n+2v), centered at the same location on the previous frame, for the best matching block (figure 2 4 1). For the minimum error, set by a criteria, the motion vector is therefore taken from this location

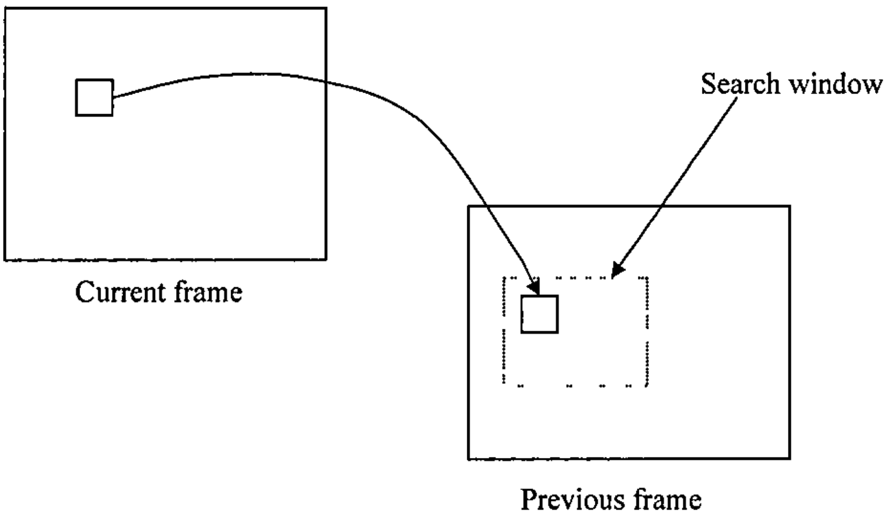


Figure 2.4.1 Block Matching search window.

The matching of the blocks can be quantified according to various criteria including Sum Absolute Difference (SAD), Sum Squared Difference (SSD), and Pel Difference Classification (PDC), etc

These criteria are outlined as followed:-

Sum Absolute Difference (SAD)

$$SAD(x, y) = \sum_{x=0}^{m-1}\sum_{y=0}^{n-1}\left|s(i,j,k) - s(i-x, j-y, k-1)\right| \qquad \text{(Eqn. 2.4.1)}$$

Sum Squared Difference (SSD)

$$SSD(x, y) = \sum_{x=0}^{m-1}\sum_{y=0}^{n-1}\left[s(i,j,k) - s(i-x, j-y, k-1\right]^2 \qquad \text{(Eqn. 2 4 2)}$$

Pel Difference Classification (PDC)

In the Pel Difference Classification method [50], each pixel in the block is classified as A matching or mismatching pixel

$$\begin{aligned} T(i, j, x, y) &= 1, \quad \text{if } \left|s(i,j,k) - s(i-x, j-y, k-1\right| \le t \qquad \text{(Eqn. 2.4.3)} \\ &= 0, \quad \text{otherwise} \end{aligned}$$

Where t is a selected threshold

T (i, j, x, y) is the binary representation of pixel difference and its value of either one or zero corresponds to a matching or mismatching pixel, respectively

The numbers of matching pixels are given by $G(x, y)$, which can be defined as follows.

$$G(x, y) = \sum_{x=0}^{m-1}\sum_{y=0}^{n-1} T(i, j, x, y) \qquad \text{(Eqn 2 4 4)}$$

Where $G(x, y)$ is the number of matching pixels that exist between the current block and the block on the previous reference frame that was shifted by i pixels and j lines

The largest value of $G(x, y)$ is found by searching through a search window. This gives the best match Thus

$$Gm\ (d_x, d_y) = max[G(x, y)] \qquad \text{(Eqn 2 4 5)}$$

Where i, j are the spatial coordinates,

x, y are the motion vector spatial coordinates,

d_x, d_y are the components of the best estimated displacement vector,

k is the time reference for the current frame,

$k\text{-}1$ is the time reference for the previous frames,

(i, j, k) is the intensity of the current frame,

$s(i, j, k\text{-} 1)$ is the intensity of the previous frame.

The performance of PDC from prediction matching point of view is better than the other methods i e. SAD, SSD . Etc In this method, the matching process is reduced to a binary level which consequently simplifies the computational complexity, as described by Gharavi [50] in 1990. However, the SAD method has been adopted as an international standard because of its simplicity

2.4.1 Half pixel interpolation

A half pixel searching window is created by a bilinear interpolation technique [51][52] (Figure 2.4.1.1). Matching is now done by first using the integer searching window and then using half pixel searching to find the best block match. This method has the advantage of producing more accurate prediction than the integer pixel block matching method However, this method requires extra computational complexity to create the half pixel searching window. Therefore, for each of the reference blocks the search begins with an integer pixel block first. Then the motion vector for the best match is used to carry out further half pixel searching. This searching will carry on until the best block match is found

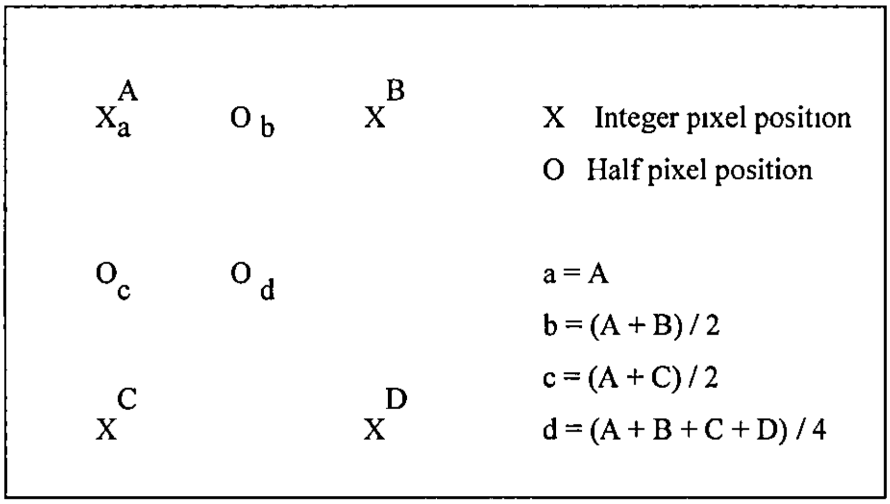
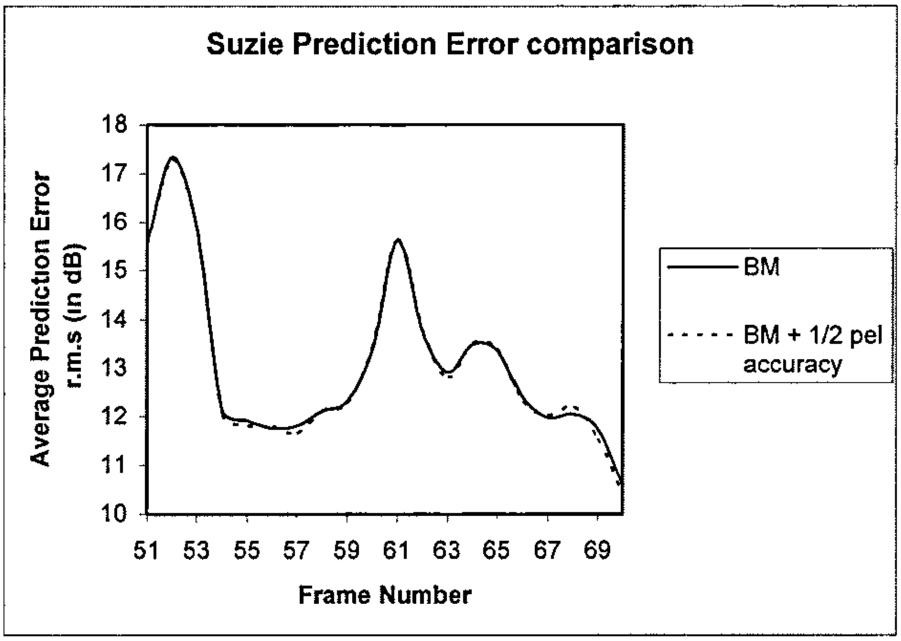


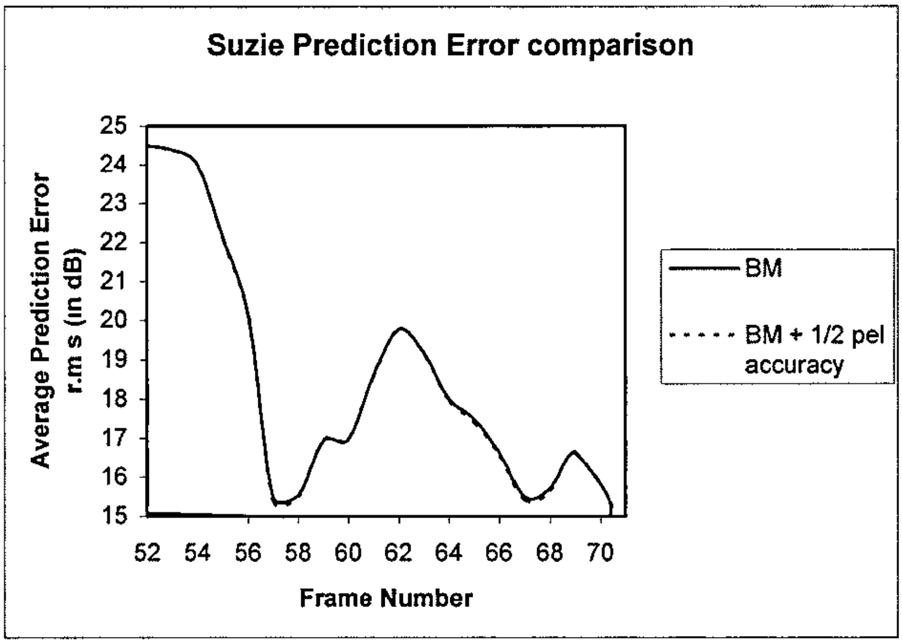
Figure2.4.1.1 Half pixel prediction

Anyhow, it was based on the previous models that all the current existing international standards (i e. H 261, H 263, MPEG-1, MPEG-2, etc) for video compression were built up

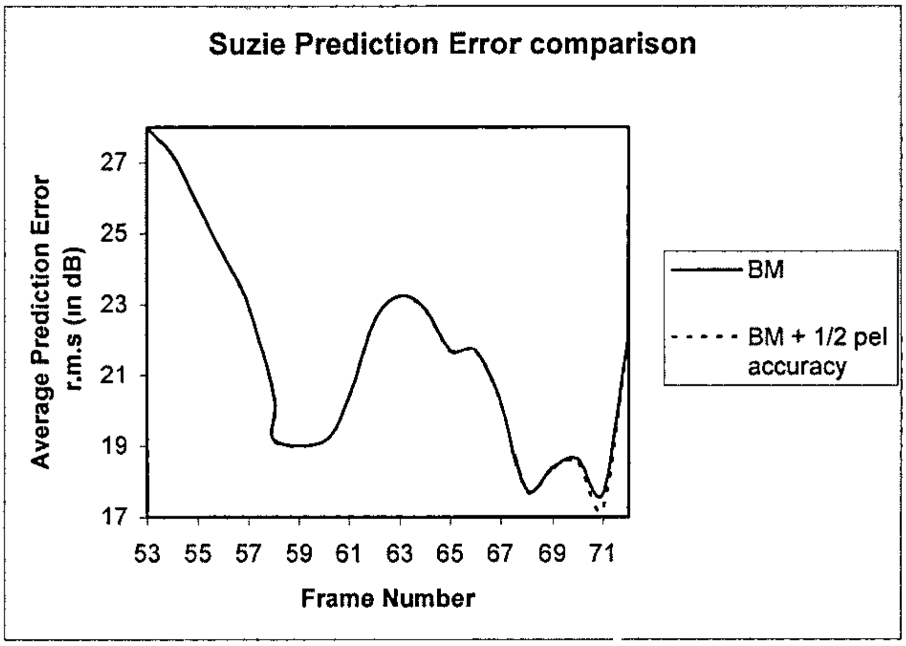
Graphical representation for block matching with half pel accuracy shows on average less than 0 05 dB improvement over traditional block matching without half pel accuracy. To justify the argument two well known sequences of “Suzie” and “Salesman” have been employed and graphs of the Average Mean Square of prediction Error have been plotted for sequences consisting of 20 frames (see Figures 2.4.1.2 and 2 4.1.3). The graphs have also been produced for different frame skips, as it is frequently used in different application . e g. video conferencing and so on In general the graphs shows that using block matching with half pel accuracy contribute very little improvement over block matching without half pel accuracy consideration But it still widely used, for example as an optional feature in H 263



a) No frame skip comparison

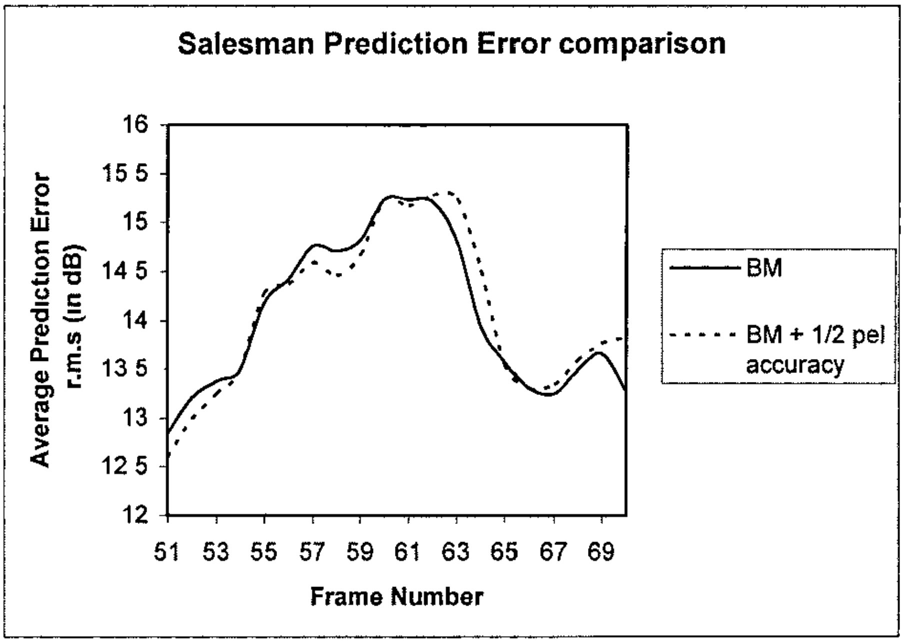


b) One frame skip comparison

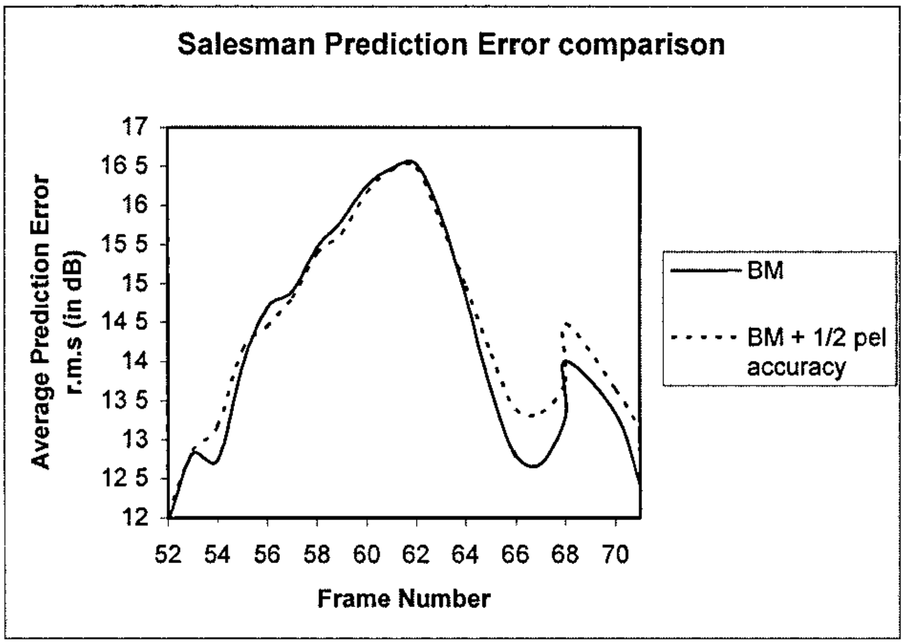


c) Two frame skip comparison

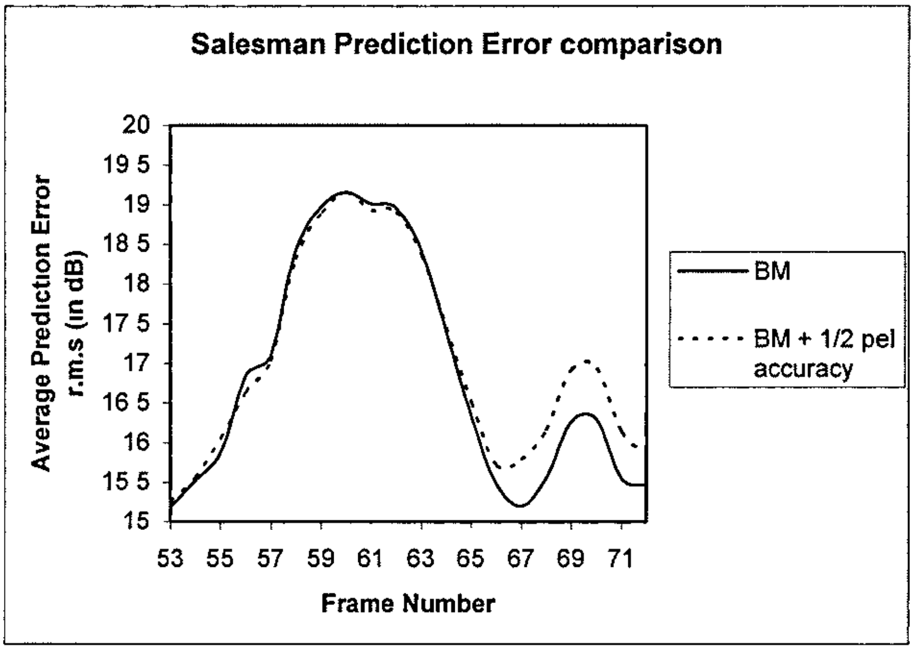
Figure 2.4.1.2 Suzie comparison with previous frame reconstructed.



a) No frame skip comparison



b) One frame skip comparison



c) Two frame skip comparison.

Figure 2.4.1.3 Salesman comparison with previous frame reconstructed

2.5 H.263

2.5.1 Introduction

An outline block diagram of the H.263 codec, the videoconferencing coding standard is given in Figure 2.5.1.1.

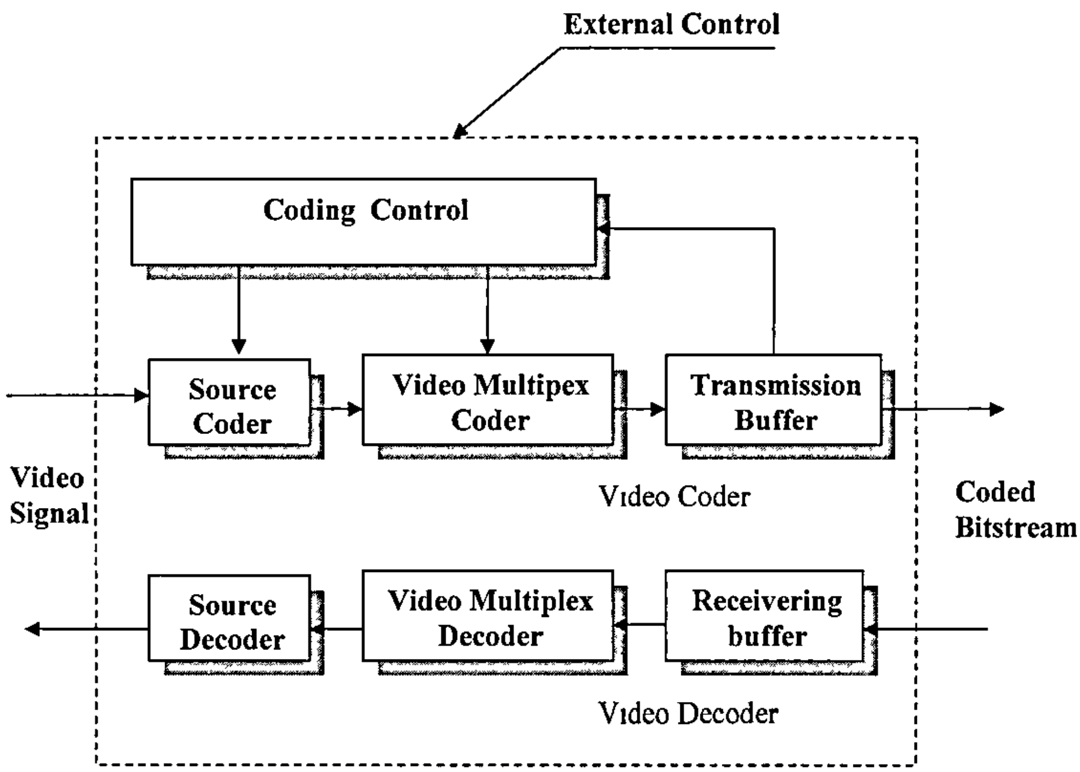
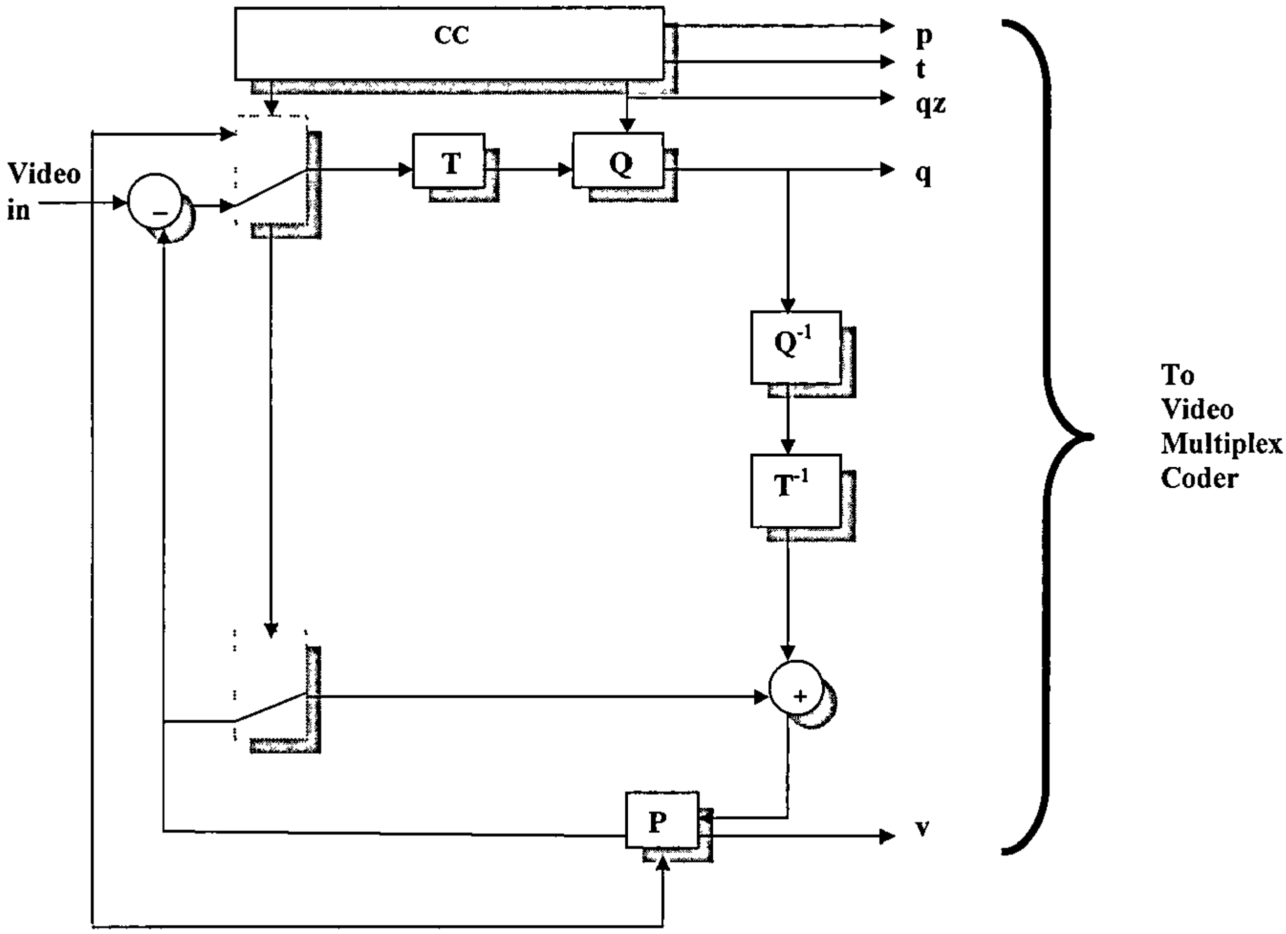


Figure2.5.1.1 H.263 block diagram of the viedeo codec

The H.263 algorithm [53][54], (which evolved from H.261[3]) is broadly based on its predecessor, recommendation H.261. However, there are some changes in the basic implementation and optional processes are available to improve the interframe prediction (Figure 2.5.1.2).



T Transform coding

Q Quantizer

P Picture Memory with motion compensated variable delay

CC Coding control

p Flag for INTRA/INTER

t Flag for transmitted or not

qz Quantizer indicator

q Quantizing index for transform coefficients

v Motion vectors

Figure 2.5.1.2 H 263 Draft recommendation encoder block diagram

The H 261 was originally devised to standardise the transmission of audio visual services and in particular the transmission of videophone and videoconference data The whole idea is that the H 261 has a fixed bit rate of P x 64kbits/s (where P = 1 30) whereas the H 263 which has a capability of achieving lower variable bit rate and is targeted for extensive deployment of any future video services The entire issue of this recommendation centers on bandwidth compression of the video signals The reason is that the video signal has a bandwidth of4 3Mhz (using regular television signal) as compared to 3 4khz of voice bandwidth This is a ratio of 1 : 1265 Therefore in order to transmit the video signal through telephone line, its bandwidth has to be grossly reduced

The H 261 operates on pictures based on a Common Intermediate Format (CIF) which has been derived from 525 and 625 line television standards It uses a hybrid of Discrete Cosine Transform (DCT) and Differential Pulse Coded Modulation (DPCM) and can achieve transmission rates between 16kbps and 2Mbps.

The H 263 also uses a hybrid of DCT and DPCM but has an improved performance when compared with H 261 One of the main reasons for this is that half pixel precision is used for motion compensation whereas full pixel precision is used in H 261. However the H 261 algorithm incorporates a spatial low-pass filter in the encoder feedback loop, which has been omitted from H 263 It has been shown that the pixel interpolation function involved in the half-pixel motion compensation process has the effect of low-pass filtering, without the need for a specific spatial function to remove high frequency noise caused by the quantisation of transform coefficients and also evident at the boundaries of blocks in the motion compensation process. The recommendation can also be applied to a wider range of picture formats and allows variable bit rates to be used therefore increasing the possible uses for the package, for example it further supports QCIF, sub-QCIF, 4CIF, 16CIF (Table 2 5 1 1) resolutions which are more appropriate to the low bit rate environment

Picture Format	number of pixels for luminance (x)	number of pixels for luminance (y)	number of pixels for chrominance (x)	number of pixels for chrominance (y)
sub-QCEF	128	96	64	48
QCIEF	176	144	88	72
CIEF	352	288	176	144
4CIIF	704	576	352	288
16CIIF	1408	1152	704	576

Table 2.5.1.1 ITU-T H 263 picture formats

The compressed ITU-T H.263 video bit stream contains four layers which is the same as ITU-T H 261. From top to bottom the layers are· Picture, Group Of Blocks, Macroblock, and block. Each picture frame is partitioned into 8 x 8 image blocks. A MacroBlock (MB) consists of 4 luminance blocks (Y), 2 chrominance blocks (C_b & C_r) As shown in Figure 2.5 1.3

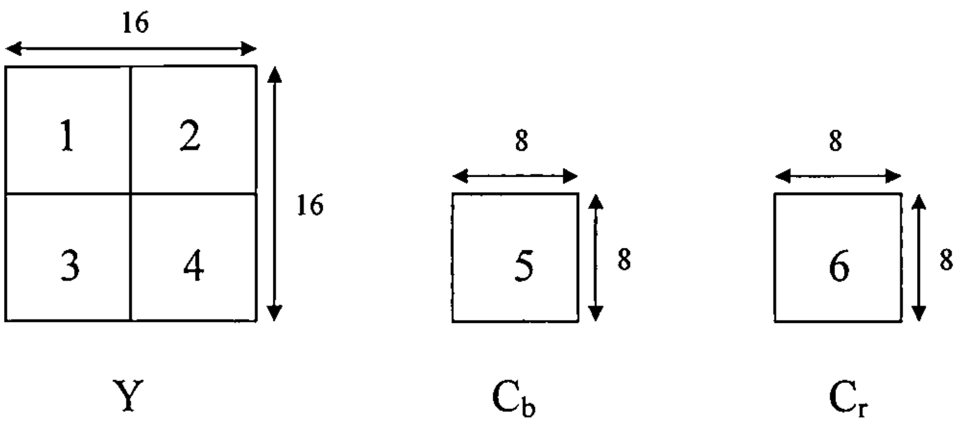


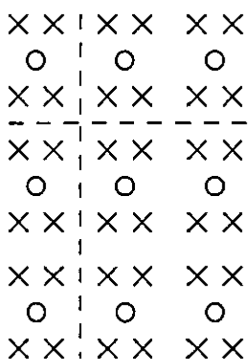
Figure 2.5.1.3 Macroblock structure

However, the Group Of Block (GOB) arrangement for the picture formats are different from ITU-T H 261. A Group Of Block (GOB) comprises of a Kx16 lines, depending on picture format (i e. K = 1 for sub-QCIF, QCIF, and CIF, K = 2 for 4CIF, K = 3 for 16CIF). Each GOB is divided into Macroblocks (Table 2.5.1.2) Similarly each Macroblocks is divided into blocks.

Picture Format	number of group of block (GOB) for picture	number of macroblock (MB) a group of block (GOB)
sub-QCEF	6	8
QCIEF	9	11
CIEF	18	22
4CIIF	18	88
16CIIF	18	352

Table 2.5.1.2 Group of Block and Macroblock arrangement

Macroblocks for colour video sequence comprise 16 x 16 pixels luminance, plus two corresponding 8 x 8 chrominance blocks Vectors can take the form of one per macroblock (Figure 2.5.1.4), or on a block basis, where four vectors per macroblock would exist. The latter forms part of the Annex F "Advanced Prediction Mode" of H 263



× Luminance Sample

O Chrominance Sample

-- Block edge

Figure 2.5.1.4 Positioning of block luminance and chrominance samples

The H 263 algorithm has been demonstrated as a versatile low bit rate video coding procedure, taking account of the growing popularity of home personal computers connecting to PSTN by a modem, having bitrates of 14 4kbits/s or 28 8kbits/s where "software codecs", using the processing power of a contemporary personal computer can do away with the need for an expensive custom receiver

A number of additional optional functions have been included in the H 263 recommendation in order to improve the interframe prediction performance.

- Unrestricted Motion Vectors
- Syntax Based Arithmetic Coding
- Advanced Prediction
- PB-frames

All of these four are optional and can be selected when running the H 263 simulation software.

Graphical Figures 2 5 1 5 - 2 5 1 7 are to show subjectively how well block matching base algorithm behaves for a sequence with relatively a fast motion. Very heavy on prediction error means block matching base algorithms may not perform as they would, if the motion were not relatively so fast These graphs are done using a software program very similar to H 361 (called motion D) on "Car" sequence.

Motion D is a laboratory version software utilizing block matching base algorithms Motion D is meant to be quite versatile It has all the features of the H 261. It can also be employed for non-standard picture sizes such as the Car sequence (720 by 576 pels) used in the thesis

Figure 2.5.1.5
PCM of previous frame for car. (clean frame)

Figure 2.5.1.6
PCM of present frame for car. (clean frame)

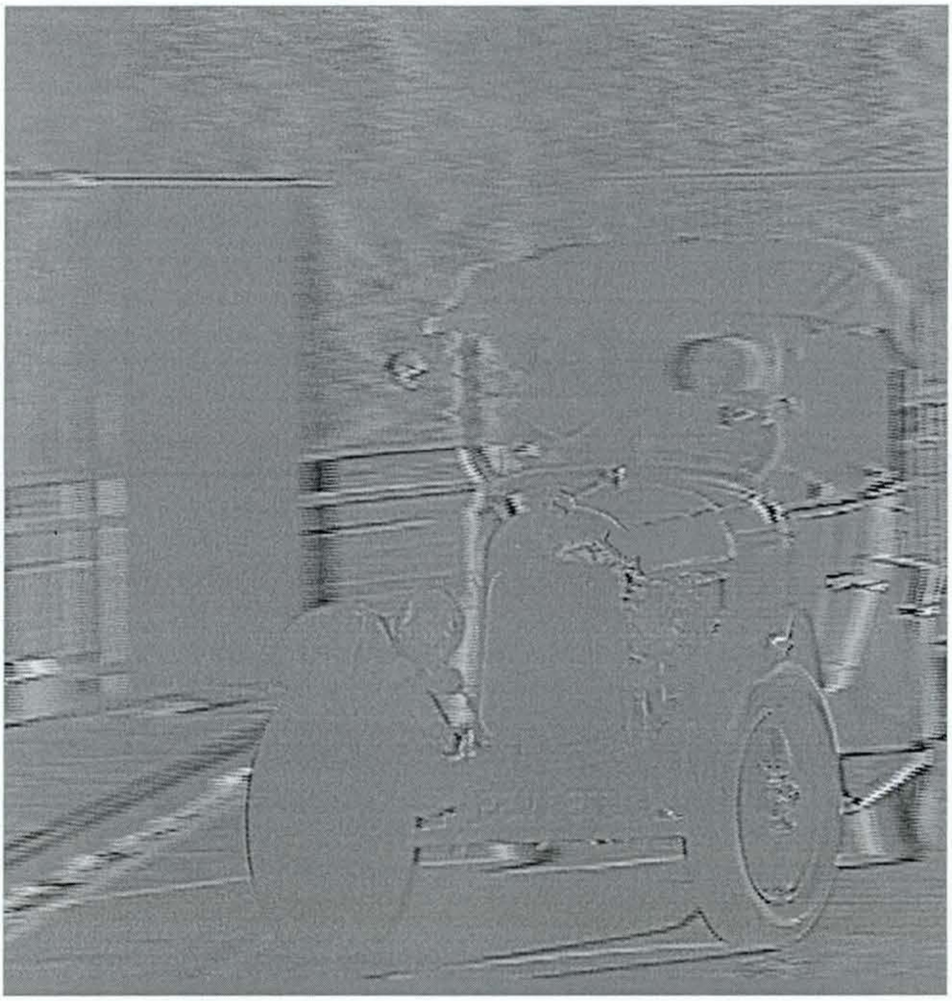


Figure 2.5.1.7 Prediction error for the two successive car frames with MC, with previous frame clean.

2.5.2 Unrestricted motion vectors Mode (Annex D)

In the default prediction mode of H.263, the search for motion vectors can only take place inside the normal picture. In the Unrestricted Motion Vector mode, this requirement is removed and motion vectors are allowed to point outside the picture. The edge pels are used as a prediction for the "not existing" pels. To do this, edge pixel values are extrapolated in the x and y directions as appropriate, producing a virtual search window for the current block to search outside the normal picture boundaries (figure2.5.2).

Extrapolated pixels

4	4	5	6	6	8	8
4	4	5	6	6	8	8
5	5	5	3	2	0	0
6	6	4	4	3	4	4
4	4	5	5	3	1	1
3	3	6	6	4	4	7
2	2	4	5	4	3	5

← Picture area

Figure 2.5.2 Extrapolation for Unrestricted Motion Vectors

With this mode a significant gain is achieved and the image prediction is improved particularly where there is motion involving objects entering or leaving the scene, or there is movement along the edge of the picture, especially for the smaller picture formats. Additionally, this mode includes an extension of the motion vector range so that larger motion vectors can be used. This is especially useful in case of camera movement, where the camera itself is moving in a pan (panning situations). This mode is optional as it does not improve the prediction for static camera and central objects (which would be common in videoconferencing).

2.5.3 Syntax-based Arithmetic Coding Mode SAC (Annex E)

SAC is a variant of Arithmetic Coding [55], used in place of the traditional Variable Length Code for minimum redundancy serial transmission The optimum length of Variable Length Codes is derived from the entropy of the data which tends to be non-integer Syntax-based Arithmetic Coding is an algorithm which encodes the symbols into a fractional number [56]

The implementation of SAC is, however, rather complex and it is impossible to recognize individual symbols in an encoded bit stream. Recovery from errors is difficult and it has a low tolerance to error, since SAC does not resynchronise after a few false symbols, as Variable Length Codes do The SNR and reconstructed frame will be the same, but generally fewer bits will be produced

2.5.4 Advanced Prediction Mode (Annex F)

This option means that Overlapped block motion compensation (OBMC) [57] [58] is used for P-frames Four motion vectors instead of one per macroblock, that is four 8x8 vectors instead of one 16x16 vector are used for some of the macroblocks in the picture, which tends to provide a smoother prediction image and a better spatial quality at the decoder It is necessary that this mode operates in conjunction with the Unrestricted Motion Vector Mode (Annex D), to make a consistent prediction from the availability of extrapolated luminance and chrominance pixels

The four 8 x 8 pixel luminance blocks in some of the macroblock allow a better representation of motion to be made, albeit at the price of a greater data over head It is therefore the responsibility of the implementing organisation to decide the value of this additional motion data

2.5.5 PB Frames Mode (Annex G)

This algorithm allows for the use of forward and bi-directionaly predicted frames. That is two pictures are being coded as one unit called as PB-frame (Figure 2 5 5) The name PB comes from the name of picture types in MPEG where there are P-pictures and B-pictures A PB-frame consist of one P-picture (*P-frame*) which is predicted from the last decoded P-picture and one B-picture (*B-frame*) which is predicted from both the previous decoded P-picture and the P-picture currently being decoded. Motion vectors can be used from the P-frames to generate predictions for the B-frames. This last picture is called a B-picture, because it is bi-directionally predicted from the past and future P-picture For relatively simple sequences, the framerate can be doubled with this mode without increasing the bitrate by much. Additional vectors may also be transmitted as an optional mode, which effectively doubles the temporal resolution of the image with only a small increase in the coded video data rate. However, this tends to produce a less satisfactory prediction in sequences having very fast or complex motion, that is with a lot of motion or low initial frame rates Never the less, the PB-frame does not work as well as the B-frame in MPEG because there is no separate bi-directional vectors in ITU-T H.263 The advantage of ITU-T H.263 over MPEG is that it requires much less overhead which is useful in low bit rate transmission

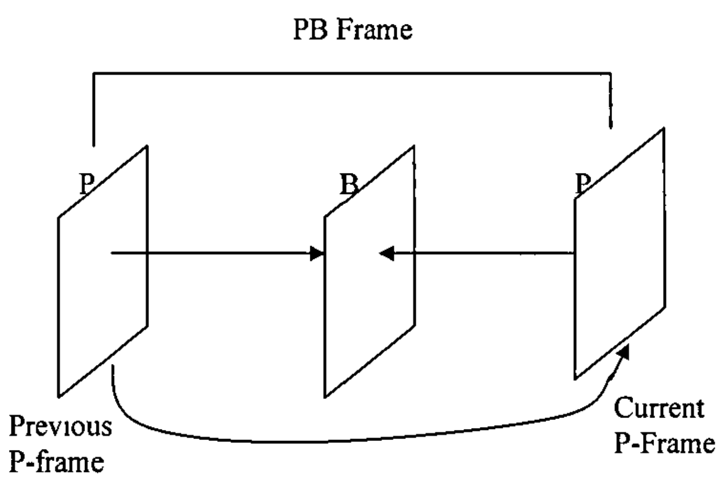


Figure 2.5.5 PB frame Arrangement

For H 263 hierarchy flow diagram and H 263 programming function description refer to appendices A 1 and A 2 respectively.

2.6 Further developments in the standard bodies

The original available videophone standard is the ITU-T H 261 [59]–[63] ITU-T Recommendation H 261 defines a video coding scheme for digital audiovisual services by the ITU-T Study Group XV Two bit-rates which have been established for Integrated Services Digital Networks (ISDN) and are of interest for image transmission are called the B-channel of 64kbits/s and the H0-channel of 384kbits/s The development of ITU-T H 261 went through many stages However, by late 1989, the final CCITT recommendations were made for the range of 64 kbits/s up to 1920 kbits/s Therefore, ITU-T H 261 is also known as a *p* x 64 codec, where *p* is between 1 and 30 Similar to ITU-T Recommendation, the algorithms specified by the Moving Picture coding Experts Group (MPEG) [64] employ a degree of both loss-less and lossy coding techniques. However, whilst the H.261 algorithm is specifically designated as the framework of video codecs working on ISDN channels of *p* x 64kbits/s, the scope of MPEG is more wide-ranging

In the late 1980's an obvious relationship began to emerge between personal computers, digital storage on inexpensive media (such as CD-ROM) and the sale of video entertainment and educational software As the result of that the Motion Picture Expert Group (MPEG) was formed in 1988 to establish a standard for the compression of digital audio and video storage and later on for transmissions. The MPEG-1 [65] [66] is the first phase video compression standard The primary objective of MPEG was to produce a compression algorithm for storage media having a through put of 1 - 1 5 Mbits/s, with other goals of up to 60Mbits/s Whilst the direct application of CD-ROM was an obvious one, the brief of MPEG was to produce a standard that would apply to other storage techniques and applications This scheme is well suited to a wide range of applications such as, Compact Disk Read-Only Memory (CD-ROM), Digital Audio Tape (DAT), Cable Television (CATV), telecommunication networks, and digital video broadcasting MPEG has also been applied to the compression of video for the purposes of Video-on-Demand [67] and for HDTV.

The MPEG-1 video coding algorithm [70] resulted from the requirements of CD-ROM and was greatly influenced by formulation of the ITU-T H 261 algorithm The development and evaluation of the algorithm was performed at bit rates in the region of 1 Mbits/s and video resolutions of 352 pixels x 288 lines, 25 frames per second, for PAL and 352 pixels x 240 lines, with an average of 29 97 frames per second for the NTSC system These rates are not fixed and can be varied according to the requirements of different applications

The essential difference of MPEG-1, compared with H 261, is that, by the nature of the application to CD-ROM, random access is required This allows the end user to arbitrarily choose any point in the video sequence from which to start viewing the moving images To achieve this, MPEG-1 has a number of frames which are encoded on their own and without any reference to other frames in the sequence, which are referred to as key frames and occur typically once in every twelve frame. As a result, MPEG-1 deliberately forces intraframe coding on some frames, whilst the majority are formed as an interframe prediction with reference to temporally adjacent frames

The presence of regularly occurring intraframe coding is one of the reasons why MPEG-1 is unsuitable for real-time coding in audiovisual communications The time taken to process and transmit an intraframe coded frame is considerably higher than for interframe difference data, causing considerable variations in the quantity of bits per frame If the I-frames were to be taken as primary start frames for an interframe sequence, they would have to be encoded with minimal losses, rendering the availability of data for the subsequent interframe coding relatively low in a given time period

One of the essential differences between MPEG-1 and the H 261 algorithm is the way in which interframe predictions are made. H 261 is primarily an interframe coding algorithm using the previous frame as the main prediction source for the generation of the next frame However, since MPEG-1 applies mainly to pre-recorded video sequence, subsequent frames can also be used to make a better prediction of the current frame

Motion vectors used by MPEG have a greater range than would be required for video conferencing applications, since the nature of a wide range of video comprises more interframe motion than would be anticipated in a typical head-and-shoulders scene

Subsequent work on MPEG standards has considered the application of the algorithm for data rates of up to 40Mbits/s MPEG-2 [69] has been adopted for direct satellite broadcasting in Europe and by the US Advanced Television Committee (FCC) for HDTV It is effectively the same as MPEG-1, except that interlace scanning can be retained and interframe delays are less, resulting in a picture of improved quality

The MPEG-1 standard was published in 1993 as ISO/IEC 11172 (Coding of moving pictures and associated for digital storage up to about 1 5 Mbits/s) [68] Part 1 of this standard describes the system, which includes information about the synchronization and multiplexing of video and audio streams. Parts 2, 3 and 4 describe video, audio and conformance testing respectively

The MPEG-2 [70] is the second phase of video compression standard which is aimed at coding above 2 Mbits/s Preparation of the MPEG-2 standard started in 1991 and provides a solution for applications that are not successfully covered by MPEG-1 The next phase of video compression standard, MPEG-3 was dropped in July 1992 A text identical to that of MPEG-2 was published as ITU-T Recommendation H 262. Recently, the MPEG-2 standard has been approved by the Advanced Television System Committee (ATSC) as a Digital High Definition Television (HDTV) [71] [72] Standard in the United States.

Formulation of a new MPEG-4 [73] Standard was begun at the MPEG meeting in Brussels in September, 1993. A draft specification is drawn in 1997 The primary target of this standard is very low bit rate applications The MPEG-4 standard supports a wide range of applications such as videophone over analogue telephone lines, sign language captioning, mobile audiovisual communications and interactive multimedia communications.

H 263 is also better than MPEG-I/MPEG-2 for low resolutions and low bitrates H 263 is less flexible than MPEG, but therefore requires much less overhead Another difference is again the negotiable options in H.263. MPEG has B-frames, but H.263 has PB-frames which are almost as good for moderate amounts of movement, but require much less overhead H 263 has overlapped block motion compensation, motion vectors outside the picture and syntax-based arithmetic coding These options are not in MPEG at all Note that it is only possible to use H 263 at certain resolutions SQCIF , QCIF, CIF, 4CIIF and 16CIF, if you follow the standard H 263 software can be changed to run at every resolution divisible by the macroblock size 16, but the bitstreams generated will not be legal H 263 bitstreams in this case

Chapter 3

Pel-recursive techniques

3.1 Background

Motion compensation techniques predict the frame-to-frame (or field-to-field) motion of an object point and then access the intensity value 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 [74]–[81] [37]–[40] has shown that motion estimation techniques do improve the prediction of the intensity values in the images.

There have been basically two approaches to motion estimation - block-matching and pel recursive techniques [39] [78] [79]. 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 amore exact estimation of the intensity value and has the ability to handle scale changes (zooming, dilating, 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 or the displacement vector reduces the error in the intensity prediction. The majority of the previous approaches have used backward prediction, applying backward prediction leads to 1) reduced bit rates, 2) lower computational requirements. or 3) faster prediction or estimation techniques

The pioneering work in detecting motion in interframe coders was done by estimating the speed (magnitude, but not the direction) by dividing the sum of the frame differences in a moving area by the sum of the element differences in that moving area [75] It was assumed that a speed of half a pel per frame was relatively slow, while a speed of four pels per frame was seldom exceeded. The results were obtained using a fixed camera and a moving object, it was also claimed that the technique could be applied to a panning camera and a moving object. Later the technique was extended to estimate velocity, i e. determine the direction of motion [37] Further pioneering work in the area of motion compensated techniques were done by Cafforio and Rocca [38] [76]. Their work was more theoretical.

The proposed techniques [75] [37] required an estimate of the motion velocity to be sent. Netravali and Robbins [39] [40] [77] 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. They used a 35 level symmetric quantizer, as a result of which the coder performance was only slightly affected by the quantizer. Previous field intensities were used for interpolation. They found that a rather simple interpolator is sufficient. Their algorithm was able to reduce the data transmission rate by up to 50%.

The next algorithm developed was called gain compensation [82]. It should be noted that gain compensation has some inherent motion tracking ability. Separate displacement compensation and gain compensation reduce the bit rate; together they reduced it even more, especially for the cases in which separately they produce minimal reduction Some further theoretical work was done on the implications and constraints of the assumptions which were being made in the motion compensated algorithms

Snyder et al [83] [84] investigated the assumption that frame differences can be expanded as a Taylor series. Followed by Horn and Schunck [85] [86] who segmented the image into moving and stationary regions.By building on the work of Horn and Schunck, Nagel [87] developed a motion estimation technique which can be seen [88] to do a good job of predicting the motion in a scene containing translational motion. No attempt has been made to apply these techniques[83] – [88] to information bandwidth compression. This is mainly because the resulting system of equations is very computationally expensive

Thompson and Barnard [89] reported on ways of estimating and interpreting motion They discussed spatio-temporal gradient techniques; feature point matching (pattern matching) was determined to be too computationally expensive.

Robbins and Netravali [90] 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 is 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 although the reduction factor was the same, the motion compensated algorithm produced better quality reconstruction than the conditional replenishment algorithm

Prabhu and Netravali [91] [92] developed a motion compensated algorithm to compress and transmit component color sequences. The first investigation involved predicting each component separately. Three predictor schemes were evaluated·- 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 component. The luminance information was used to switch the predictor.

Ishiguro and Iinuma [78] 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 implies that the transmitter and receiver both determine the motion prediction from common information (previously transmitted data). 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. This type of pattern matching technique was actually implemented in a production system [93] by NEC. It is interesting to note that it uses pattern matching technique since other researchers had stated that a pattern matching technique would be too computationally intensive [39] [79] and since the spatio-temporal gradient method had received more favorable consideration in the literature [94] – [97]

All the algorithms discussed so far have in effect modelled the motion in the sequence as purely translational Huang and Tsai [95] pointed out that if rotation of object is to be considered, and then pattern matching technique requires a three dimensional data space with an increase in processing bandwidth, indicating a spatio-temporal gradient approach would be more feasible.

Paquin and Dubois [79] investigated spatio-temporal gradient algorithm which employed motion compensated prediction. Although they obtained an algorithm similar to that of Netravali and Robbins [39] [40], they started from a slightly different perspective and with slightly different asumptions The displacements were estimated on a field basis. Their maximum allowable displacement was 10 pels per field while Limb and Murphy [75] assumed 4 pels per frame would seldom be exceeded. They were primarily interested in determining trade-offs between accuracy and computational complexity for interpolator and the estimator.

3.2 Motion compensated image sequence compression

In video conferencing applications, correlation between consecutive frames is significantly high due to the limited amount of motion. This correlation can be exploited more efficiently by taking into consideration the displacements of moving objects in the coding process. Thus in any motion compensated coding scheme, the coding performance depends heavily on the accuracy of the motion estimation

There are instances when the DPCM technique cannot successfully code a segment of an image sequence because motion is a major cause of interframe differences. Motion Compensation (MC) can be used to improve the efficiency of the predictive coding algorithm

If translation of a moving object is available, a more efficient prediction can be estimated using elements in the previous frame(s) that are appropriately spatially displaced. This type of prediction is called *Motion Compensated Prediction.* Furthermore, motion can be a complex combination of translation and rotation Transitional motion is relatively easily estimated and has been used for motion compensated coding, depending on the amount of translation motion in the scene and the ability of an algorithm to estimate translation with the accuracy that is necessary for a good prediction.

The main problem is developing a good algorithm used for motion estimation Various algorithms which have been successfully used in coding application include Block Matching, Pel Recursive, and Gain motion compensated estimation.

Block matching is widely used in coding applications but has its own limitations and weaknesses due to looking at displacement over a block as a whole, which is perhaps a trade-off, i e a less accurate estimation producing less coding which in turn gives higher compression It is not a good idea to trade off the accuracy of estimation for the motion for some over head or perhaps come up with a different algorithm which could take care of the mentioned problem.

The pel recursive method for displacement of motion compensation can overcome the above problem In this method, we look at every pel by pel estimating the displacement vector for every single pel resulting in motion estimation for every single pel rather than every block of pels, therefore higher accuracy is achieved but with the cost of more overhead.

Among the many different algorithms, the one by Netravali [37] [39] [40] [77] [98] – [100] is looked at in more detail.

3.3 Initial estimation of displacement vector

For simplicity the algorithm used for motion estimation in interframe coding follows the assumptions below (these are true for most algorithms for motion estimation in interframe coding)

I - Translation movement of an object is in a plane which is parallel to the camera plane.

II - Illumination is spatially and temporally uniform

III - Occlusion of one object by another, and also uncovered background are neglected.

Under these assumptions, the monochrome intensities b(z, t) and b(z, t - τ) of two consecutive frame are related by

$$b(z,t) = b(z+D, t-\tau) \qquad \text{(Eqn 3.3.1)}$$

Where τ is the time between two frames, D is the two dimensional translation vector of the object during the time interval [t -τ, t], and z is the two dimensional vector [x, y]` of spatial position. Using Eqn (3.3.1) we can write the frame difference signal FDIF (z, t) as

$$FDIF(z,t) \triangleq b(z,t) - b(z,t-\tau) = b(z,t) - b(z+D,t) \qquad \text{(Eqn 3 3 2)}$$

For small D, using Taylor's expansion about z (assuming D to be small)

$$FDIF(z,t) = -D`\nabla_z b(z,t) + \text{higher order terms in D} \qquad \text{(Eqn 3 3.3)}$$

Where ∇_z is the spatial gradient with respect to z.

Assuming that the translation of the object is constant over some moving area R and neglecting higher order terms in D.

$\hat{D}$, the minimum mean square estimate of D can be obtained by minimizing

$$\sum_R [FDIF(z,t) + D`\nabla_z b(z,t)]^2 \qquad \text{(Eqn 3.3.4)}$$

with respect to D, therefore

$$\hat{D} = -\left[\sum_R \nabla_z b(z,t) * \nabla_z{}' b(z,t) \right]^{-1} * \left[\left[\sum_R \text{FDIF}(z,t) * \nabla_z b(z,t) \right] \right. \qquad \text{(Eqn 3.3.5)}$$

∇_z b(z, t) can be approximated as

$$\nabla_z b(z,t) = \begin{bmatrix} EDIF(z) \\ LDIF(z) \end{bmatrix} \qquad \text{(Eqn 3.3 6)}$$

Where EDIF is a horizontal element difference and LDIF is a vertical line difference given by

$$EDIF(z) = 1/2 \ \left[b(z+\Delta x, t) - b(z - \Delta x, t) \right] \qquad \text{(Eqn 3 3 7)}$$

$$LDIF(z) = 1/2 \ \left[b(z+\Delta y, t) - b(z - \Delta y, t) \right] \qquad \text{(Eqn 3.3.8)}$$

Using Eqn (3 3 8)

$$\hat{D} = -\begin{bmatrix} \sum EDIF^2(z) & \sum EDIF(z) * LDIF(Z) \\ \sum EDIF(z) * LIDF(z) & \sum LDIF^2(z) \end{bmatrix}^{-1} * \begin{bmatrix} \sum FDIF(z,t) * EDIF(z) \\ \sum FDIF(z,t) * EDIF(z) \end{bmatrix} \qquad \text{(Eqn 3 3 9)}$$

$\triangleq$ denotes by definition

prime ` denotes their transpose

consider D, Z, ∇ to be column vectors of size (2x1)

An sumption is made to convert the above matrix into diagonal one, that is:-

$$\sum_{R} EDIF(z) * LDIF(z) \approx 0 \qquad \text{(Eqn 3 3 10)}$$

Then

$$\hat{D} = -\begin{bmatrix} \dfrac{\sum FDIF(z,t) * EDIF(z)}{\sum EDIF^2(z)} \\ \dfrac{\sum FDIF(z,t) * LDIF(z)}{\sum LDIF^2(z)} \end{bmatrix} \qquad \text{(Eqn 3 3 11)}$$

in order to proceed with simulation. Moving area segmentation was defined by considering the moving pels That is if the frame difference for the considering pel is less than a threshold value, the pel is considered or classed as moving pel which is chosen in relation with camera noise.

Using Eqn (3 3.11), the initial estimate of local displacement was provided by simulation, giving good results where we were not on the edge of the moving area in the scene.

Careful consideration should be given, in order to estimate initial displacement vectors accurately enough, as these are highly dependant on the implementation of the moving area pels. Therefore the moving area pel should not be classed as a moving pel if the left, right, and upper neighboring pels are not moving pels (and vice versa)

3.4 INTERPOLATION

Having estimated an initial value for the displacement vector, the intensity of the pel displaced by $\hat{D}$ is estimated by means of an interpolating technique and the following formula is used

$$I = I_D + \hat{D}_x(I_A - I_D) + \hat{D}_Y(I_B - I_D) + \hat{D}_X * \hat{D}_Y(I_D + I_C + I_A + I_B) \qquad \text{(Eqn 3 4.1)}$$

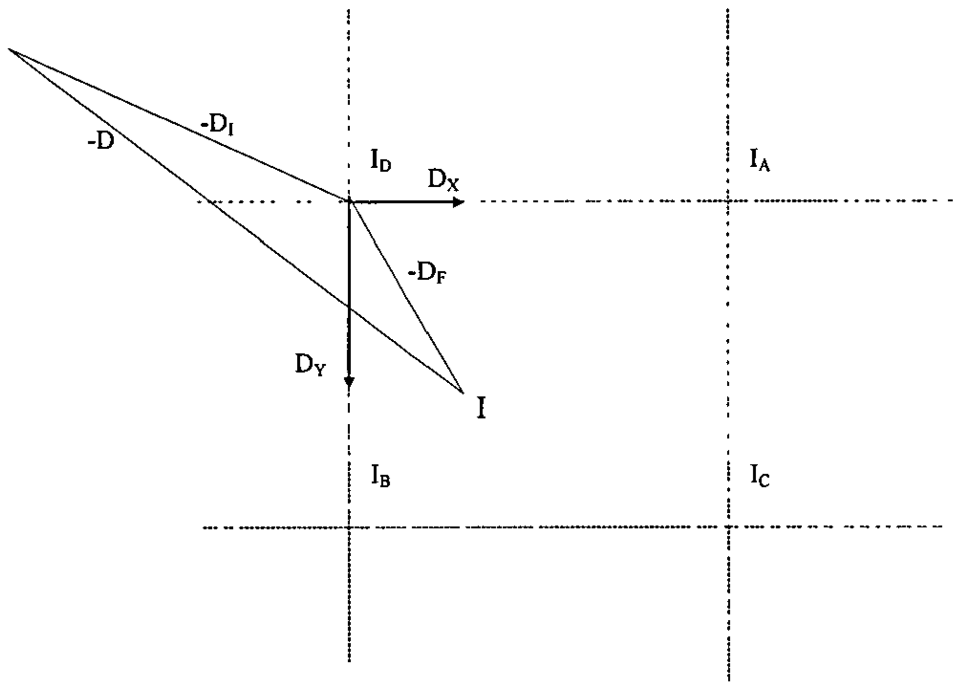


Figure 3.4.1 : Two dimensional linear interpolation.
Displacement D is decomposed into integral part D_I and non-integral part D_F

3.5 PEL RECURSIVE MOTION VECTOR ESTIMATOR

Now we define the Displaced Frame Difference (DFD) as follow -

$$DFD\left(z,\hat{D}\right) = b(z,t) - b\left(z-\hat{D},t-\tau\right) \qquad \text{(Eqn 3 5.1)}$$

In practice , the *DFD*, *DFD*(z, $\hat{D}$), hardly ever becomes exactly zero for any value of $\hat{D}$, because - I) there is observation noise, II)there is occlusion (covered / uncovered background problem), III) errors are introduced by the interpolation step in the of non-integer displacement vectors, and IV) scene illumination may vary from frame to frame. Therefore, it is generally aim to minimize the absolute value or the square of the *DFD*.

Pel recursive displacement estimators tries to minimize recursively $\left[DFD\left(z,\hat{D}\right)\right]^2$ at each moving area pel using a steepest descent algorithm thus

$$\hat{D}_k = \hat{D}_{k-1} - \frac{\varepsilon}{2} * \nabla_{\hat{D}_{k-1}}\left[DFD\left(z,\hat{D}_{k-1}\right)\right]^2 \qquad \text{(Eqn 3 5.2)}$$

Where $\nabla_{\hat{D}}$ is the two dimensional gradient operator with respect to $\hat{D}$.Using Eqn. (3.5.1) therefore

$$\hat{D}_k = \hat{D}_{k-1} - \varepsilon * DFD\left(z_n,\hat{D}_{k-1}\right) * \nabla_z\, b\left(z_n-\hat{D}_{k-1},t-\tau\right) \qquad \text{(Eqn 3.5.3)}$$

Thus, the new value for $\hat{D}$ is the old value plus an update term.

where ε, the convergency parameter is some positive scalar, known as the step size. The step size ε is critical for the convergence of the iterations, because if step size is too small, we move by a very small amount each time, and the iterations will take too long to converge On the other hand, if it is too large the algorithm may become unstable and oscillate about the minimum In the above method, the step size is usually chosen heuristically

The above algorithm can be extended by computing the displaced frame differences at many picture elements in order to estimate *D*

The steepest descent algorithm is used to minimize a weighted sum of the squared displaced frame differences at some previously transmitted neighboring pel, thus

$$\hat{D}_k = \hat{D}_{k-1} - \frac{\varepsilon}{2} * \nabla_{\hat{D}_{k-1}} \left[\sum_{j=0}^{p} W_j \left(DFD\left(z_{n-j}, \hat{D}_{k-1}\right)\right)^2 \right] \qquad \text{(Eqn 3.5.4)}$$

Where $W_i \geq 0$ and $\sum_{j=0}^{p} W_j = 1$

Using Eqn (3 3 1) therefore

$$\hat{D}_k = \hat{D}_{k-1} - \varepsilon * \left[\sum_{j=0}^{p} W_j \; DFD\left(z_{n-j}, \hat{D}_{k-1}\right) * \nabla_z \; b\left(z_{n-j} - \hat{D}_{k-1}, t - \tau\right) \right] \qquad \text{(Eqn 3.5.5)}$$

Where $\nabla_z(\circ)$ can be approximated by finite differences as before

Now recursively $\hat{D}$ is updated using Eqn (3 4 1) and Eqn (3 5 5). For each step, the update term seeks to improve the estimate of $\hat{D}$. The ultimate goal is minimization of the magnitude of the prediction error *DFD*. If a pel at location Z_a is predicted with $\hat{D}_{k-1}$ to have intensity $b\left(z - \hat{D}_{k-1}, t-1\right)$, resulting in a prediction error of $DFD(z, \hat{D}_{k-1})$ the prediction should attempt to create a new estimate, $\hat{D}_k$ such that .-

$$\left| DFD(z, \hat{D}_k) \right| \leq \left| DFD(z, \hat{D}_{k-1}) \right|$$

3.6 Implementation and Experimental results

In order to implement and simulate the previously mentioned technique, calculation of line and element differences in addition to the displacement frame difference are the most crucial and should be given the most concern. In the experiment, different ways of implementation for each parameter should be examined e.g.:- interpolated, none interpolated, averaged using different causal supports (Figure 3.6.1), etc. Epsilon, ε, the convergency parameter is recommended to be 1 / 1024 [77]. Further more, not every pel need to be motion compensated, therefore some kind of masking should be employed e,g . where frame difference, $|FDIF| \leq$ threshold; no prediction is needed. This is classified as non-moving area.

X X X X X X X

X X X X X X X

X X X X O

Figure 3.6.1 : An example of a second order causal support.

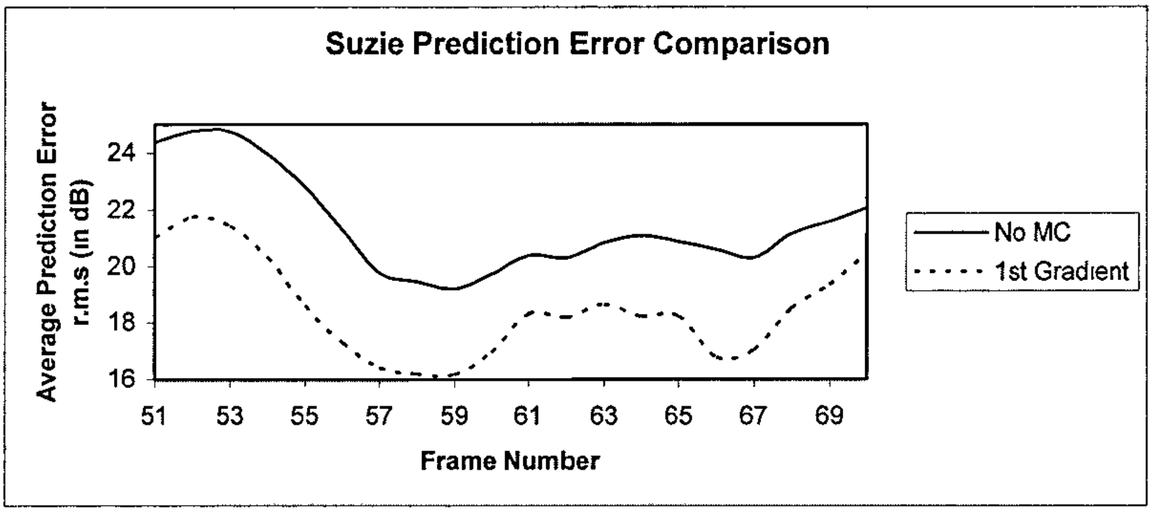
For the experiment a good result is produced having $\varepsilon = 0.9$ and using a 3 by 2 causal support (Figure 3.6.2) for line and element differences. The maximum permitted update term was chosen to be limited to 4. Absolute Frame Difference, $|FDIF| \leq 9$ for a non-moving area. This is done for two different sequences, Suzie and Salesman.

X X X X

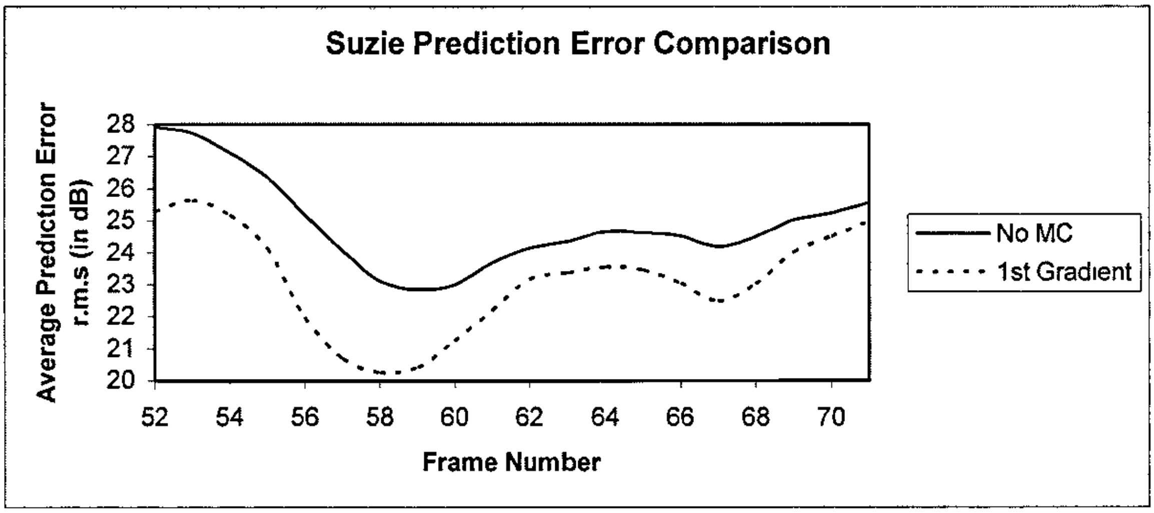
X X O

Figure 3.6.2 : causal support.

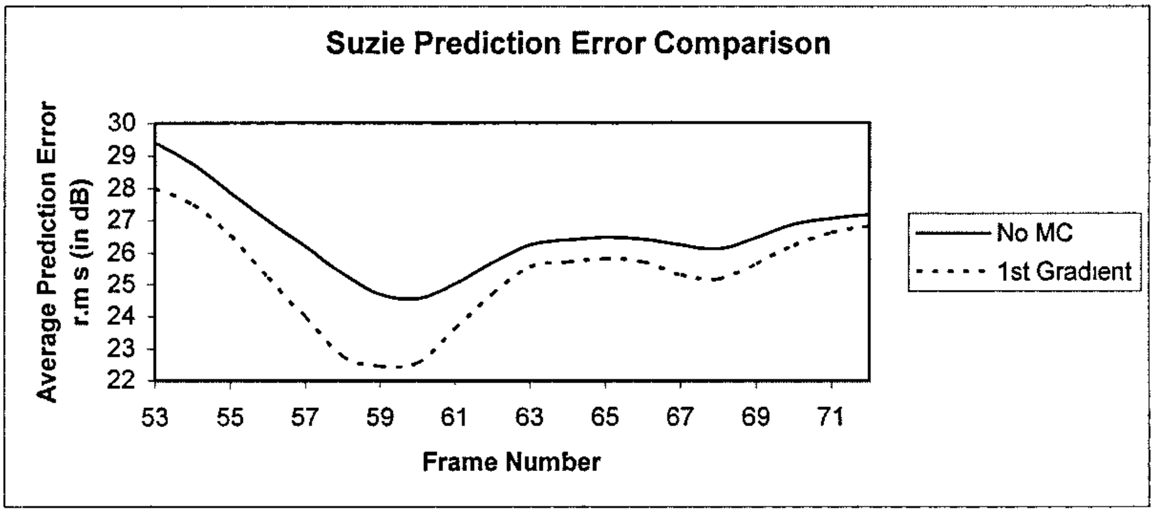
Figures 3.6 3 to 3.6.6 show and indicate the validity of the theory behind the pel-recursive motion estimation. In each of the figures, the graphs represent the energy of the error for the situations in which there is no motion compensation and where the motion is compensated using pel-recursive motion estimation It also looked at different frame skips, that could be frequently used in video conferencing and so on From the graphical and pictorial results (Figure 3.6.3 – 3.6.6) it can be seen that pel-recursive motion compensation does very good job and shows, high dB reduction in transmittable error from the DPCM loop



a) No frame skip comparison.



b) One frame skip comparison.



c) Two frame skip comparison.

Figure 3.6.3 Suzie comparison after three iteration with previous frame clean.

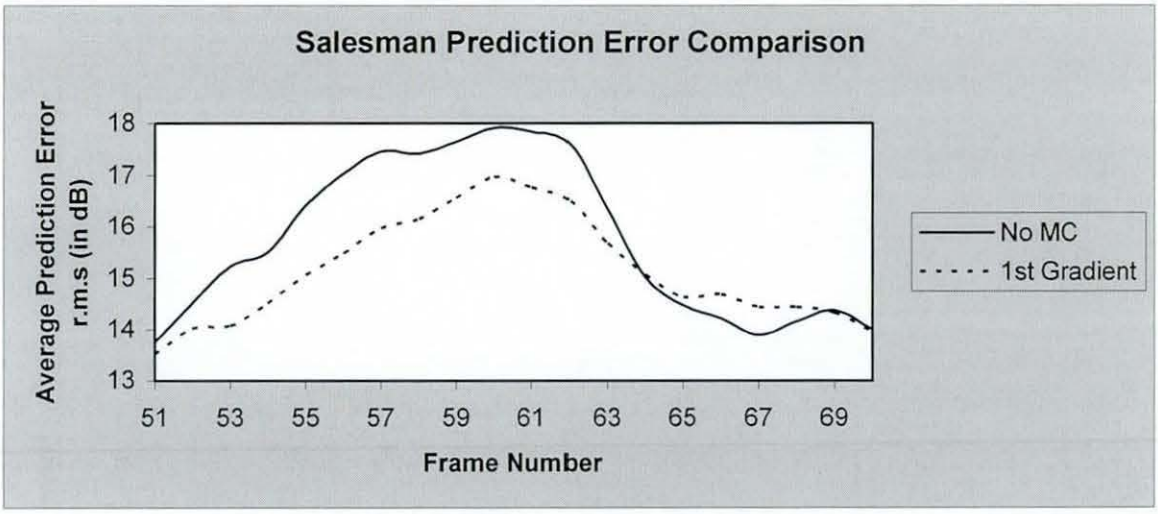
a) PCM frame of Suzie (previous frame)

b) PCM frame of Suzie (present frame)

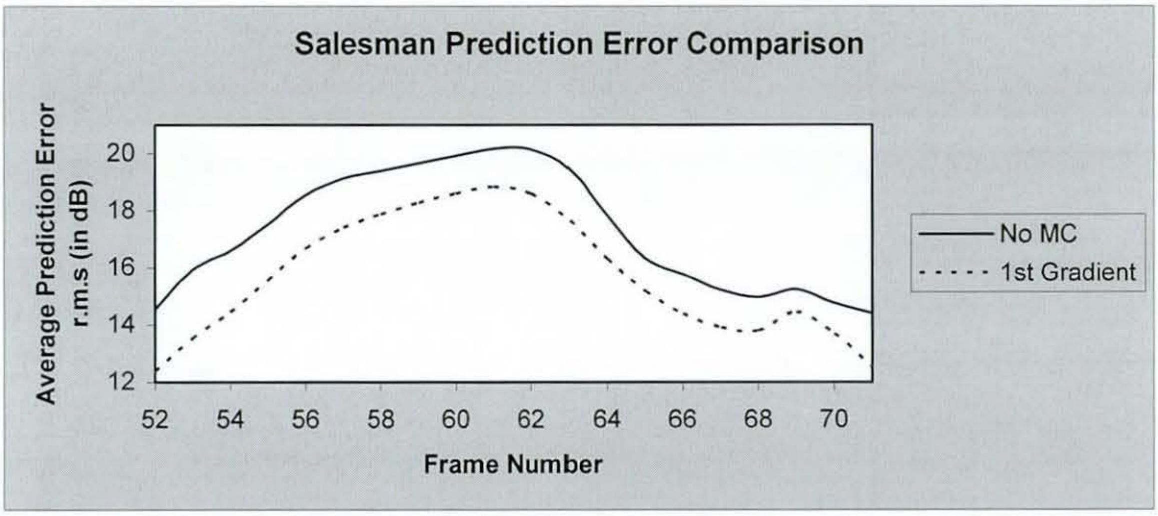
Prediction error with no MC for Suzies in a & b

d) Prediction error for Suzies in a & b (gradient)

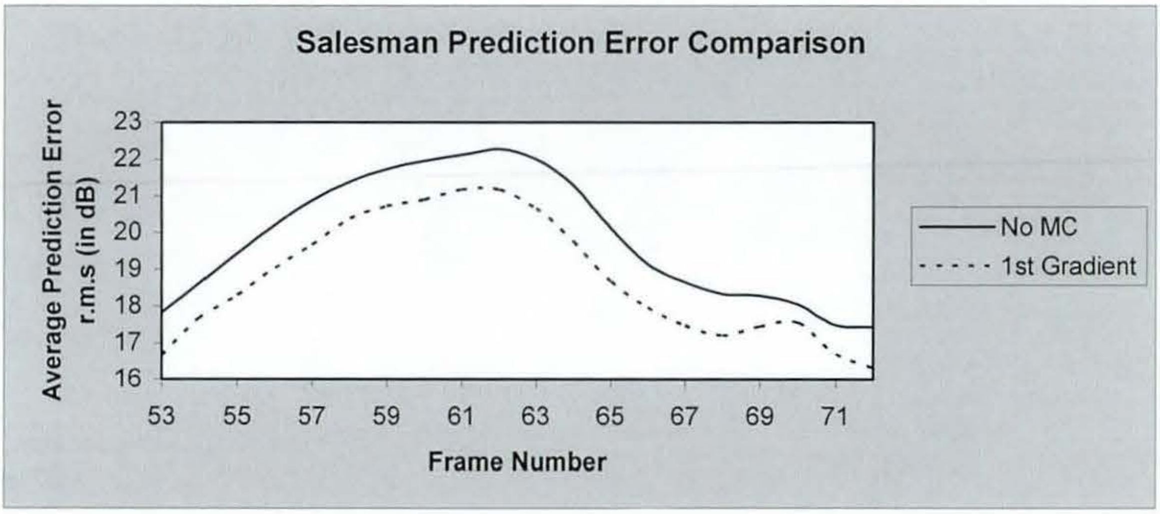
Figure 3.6.4 Prediction error comparison for two successive frame of Suzie after three iteration with previous frame clean.



a) No frame skip comparison.



b) One frame skip comparison.



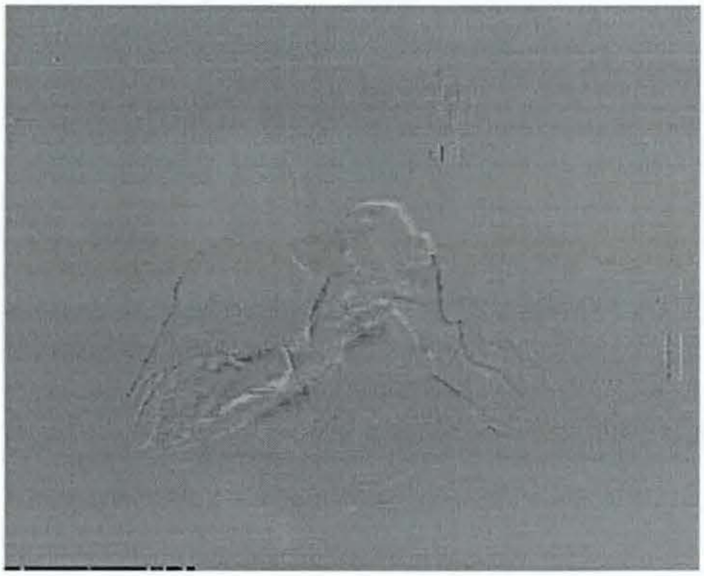
c) Two frame skip comparison.

Figure 3.6.5 Salesman comparison after three iteration with previous frame clean.

a) PCM frame of Salesman (previous frame)

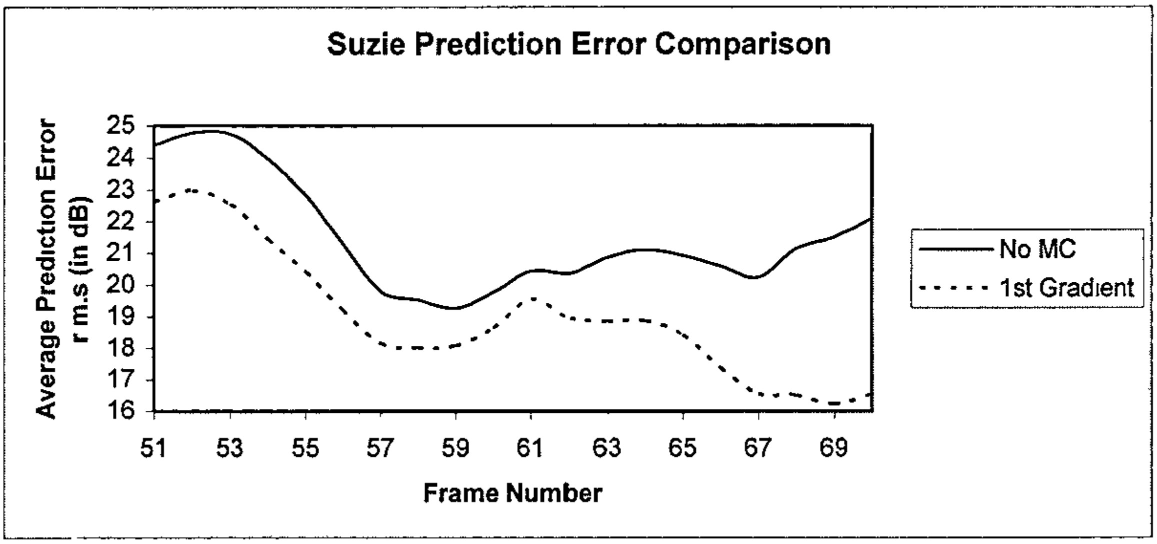
b) PCM frame of Salesman (present frame)

Prediction error with no MC for Salesman in a & b

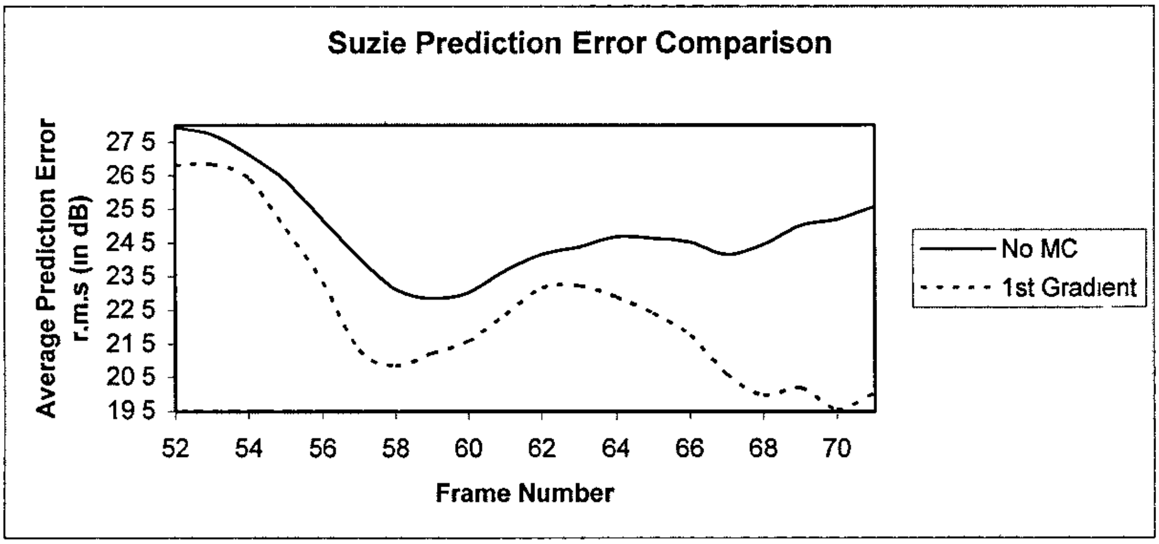


d) Prediction error for Salesman in a & b (gradient)

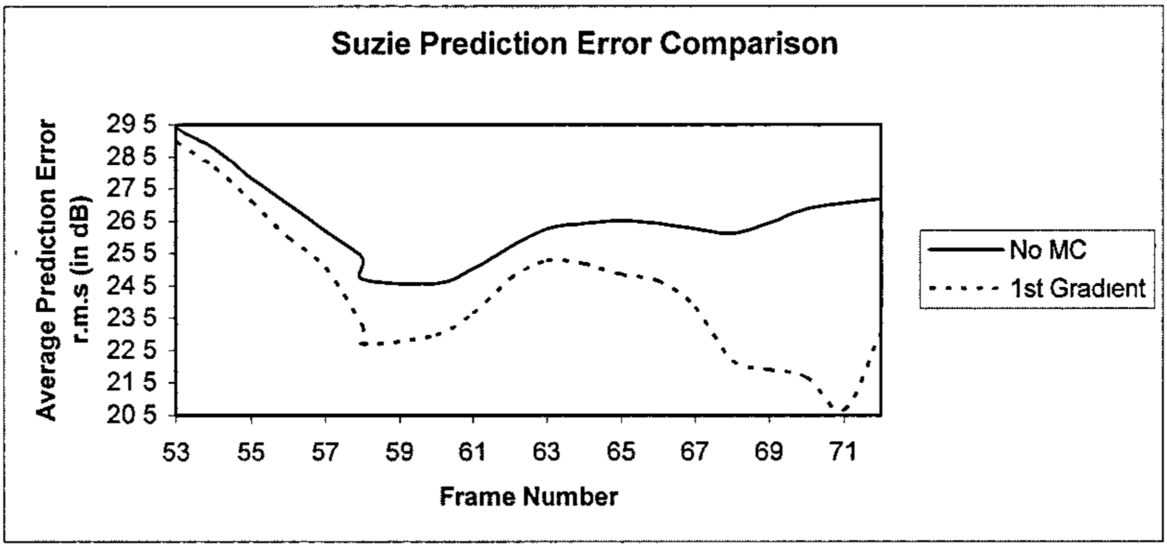
Figure 3.6.6 Prediction error comparison for two successive frame of Salesman after three iteration with previous frame clean.



a) no frame skip comparison.

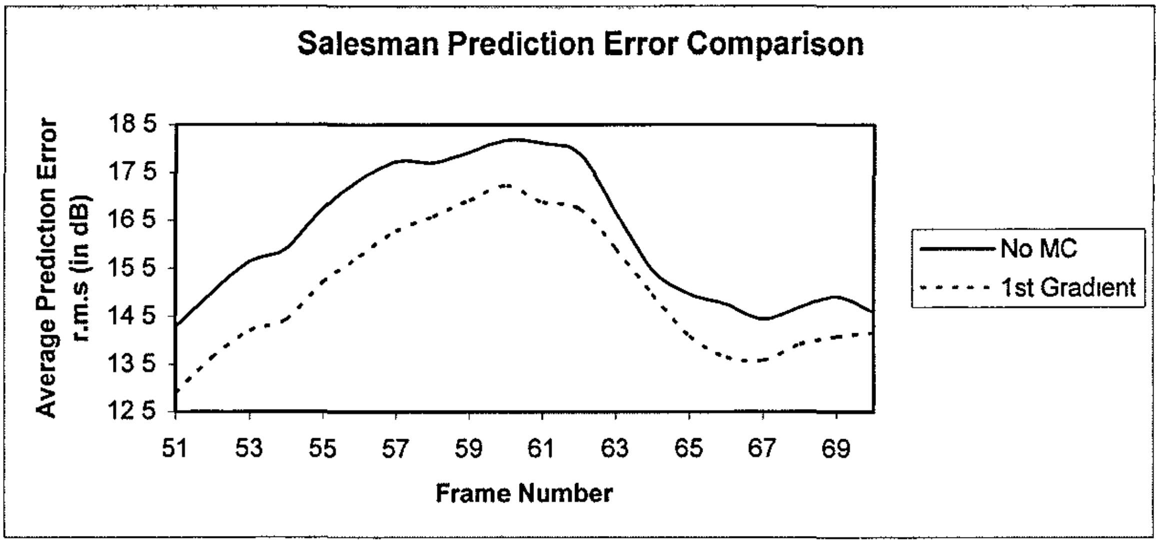


b) One frame skip comparison.

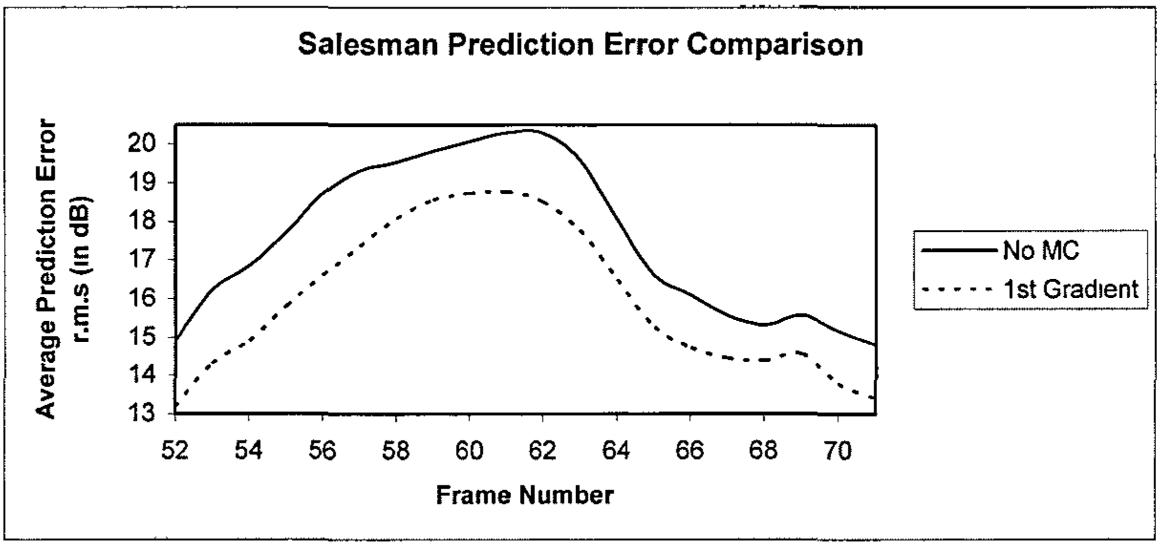


c) Two frame skip comparison.

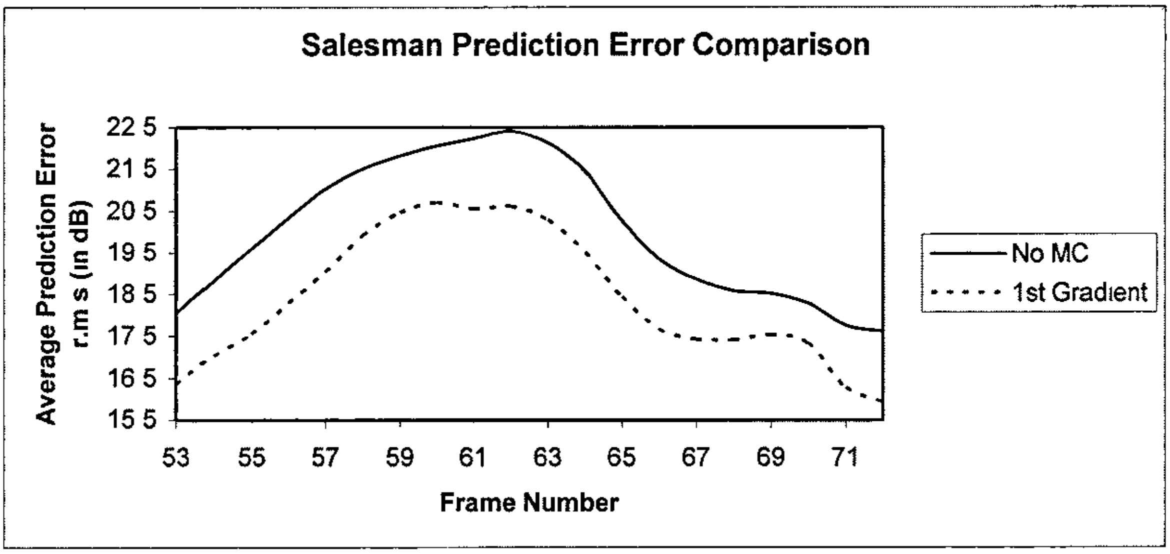
Figure 3.6.7 Suzie comparison after three iteration with previous frame reconstructed, with half pel accuracy on block matching, and system resetting to zero for each pel.



a) No frame skip comparison.



b) One frame skip comparison



c) Two frame skip comparison

Figure 3.6.8 Salesman comparison after three iteration with previous frame reconstructed, with half pel accuracy on block matching, and system resetting to zero for each pel.

3.7 Improved pel-recursive motion compensation

We now consider further improvements for pel-recursive Motion compensation [101] – [106]. Consider the basic algorithm (Eqn 3 5.3 or 3.5.5) for the intensity function at an object edge The condition requiring the largest vector corrections or updates factor are when $|DFD|$ is large and $|\nabla b|$ is small. Conversely, if $|DFD|$ is small and $|\nabla b|$ is large, as could exist at an object edge, the vector correction must be small. For the affirmation algorithms to work, E must be chosen to allow for the case where the correction or update must be small. This gives rise to

$$\varepsilon = 1/2 * \frac{1}{\left(\left|\nabla_z b(z_{n-j} - \hat{D}_{k-1}, t-\tau)\right|\right)^2} \qquad \text{(Eqn 3.7 1)}$$

Or

$$\varepsilon = 1/2 * \frac{1}{\sigma^2 + \left(\left|\nabla_z b(z_{n-j} - \hat{D}_{k-1}, t-\tau)\right|\right)} \qquad \text{(Eqn 3.7.2)}$$

and

$$\left\{\left|\nabla_z b(z_{n-j} - \hat{D}_{k-1}, t-\tau\right|\right\}^2 = \left\{\left|\nabla_x b(z_{n-j} - \hat{D}_{k-1}, t-\tau\right|\right\}^2 + \left\{\left|\nabla y b(z_{n-j} - \hat{D}_{k-1}, t-\tau\right|\right\}^2 \qquad \text{(Eqn 3.7.3)}$$

Where σ is recommended to be of the order of 10 [104], which takes account for $|\nabla b|$ becoming small or zero.

3.8 Helpful implementation details and constrains

Implementation and simulation of algorithms needs many sensible constraints and restrictions in order to show that the algorithms even work. Some of them are as follows -

a) - If $|DFD| \leq$ threshold, the correction term or update of Eqn 3.5.3 or 3.5.5 is zero

b) :- If $|DFD|$ > threshold $|\nabla b|$ is not zero, the update term is calculated. When $|update - term|$ < 1/16, the update term is recommended to be assigned to the value of $\pm$ 1/16.

c) :- If $|DFD|$ > threshold and if $|\nabla b|$ is zero, then the update term again is zero.

d) :- If $|update - term|$ exceeds 2, the update term is recommended to be assigned to the value of ± 2.

It can be seen that as the $|\nabla b|$ or $|gradient|$ becomes large, the update term decreases, and vice versa.

Further, some of the restrictions implemented and applied for simulation are as follow -

a) - Use $\hat{D}_k$ displacement obtained for the previous pel. Predict the current pel by obtaining a pel value from the previous frame at the offset $\hat{D}_k$ from the current pel location z.

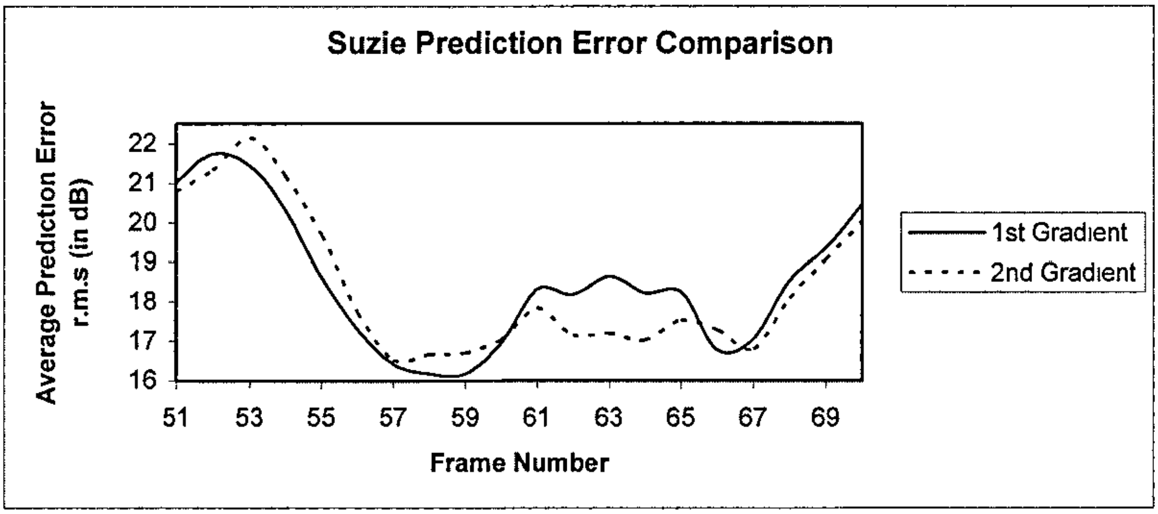
b) - If $|DFD| \leq$ threshold, transmit zero If $|DFD|$ > threshold and $|FDIF| \leq$ threshold, transmit a reset to set $\hat{D}_k = 0$ If $|DFD|$ and $|FDIF|$ > threshold, transmit DFD.

c) :- If $|DFD| \leq$ threshold, use $\hat{D}_k$ as obtained from the previous pel, i.e , $\hat{D}_k = \hat{D}_{k-1}$. And if $|DFD|$ > threshold and $|FDIF|$ < threshold, set $\hat{D}_k = 0$.

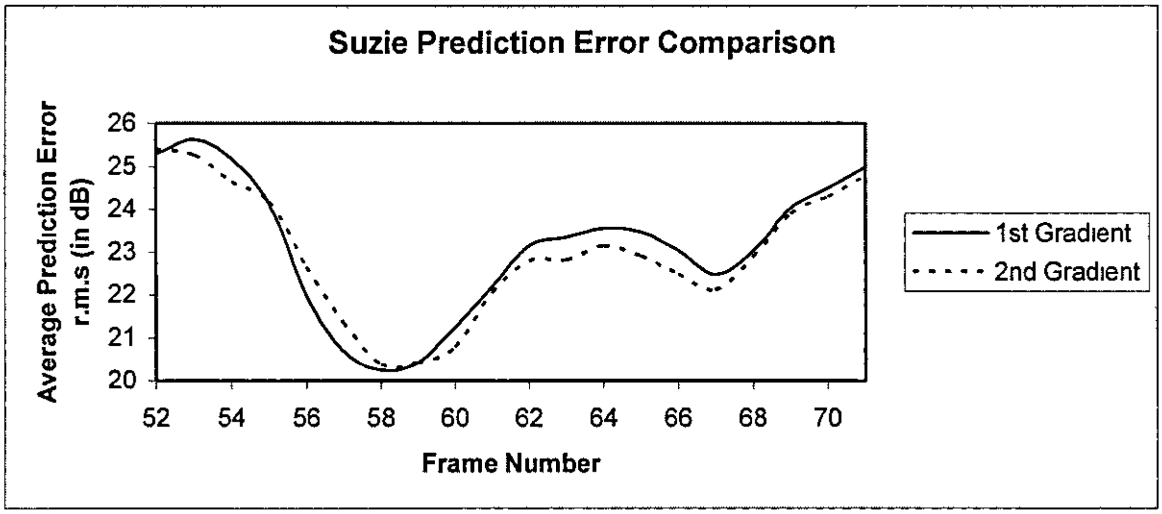
3.9 Implementation and Experimental results

Once again in the experimental work, different ways of implementing each parameter should be examined e g.·- interpolated, non-interpolated, averaged using different causal supports (Figure 3 6 1) For consistency in comparison the threshold value chosen for the experimental work was 9 for frame difference and 20 for displacement frame difference and 3 by two causal support as before The maximum update limit was chosen to be 3. These restrictions give rise to the graphical and pictorial result in (figures 3 9.1 – 3.9 6)

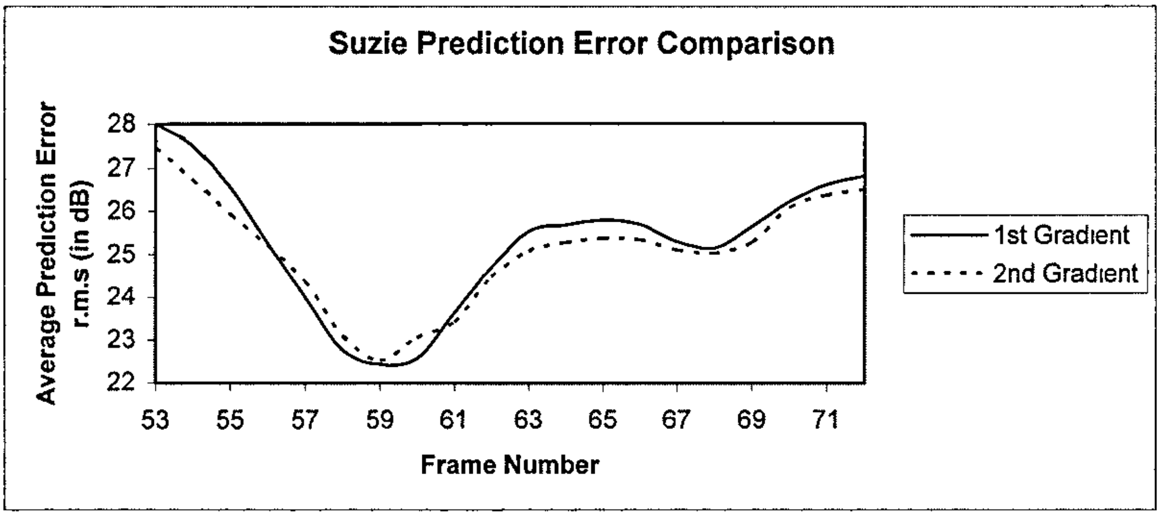
Each figure depicts the graphical representation of the basic state of the art gradient algorithm for ε, the convergence factor, to be non-adaptive and adaptive as first and second gradient Looking at the result from the same sequences of Suzie and Salesman, it can be easily noticed that having ε, the convergence factor, as a variable shows quite substantial improvement over the basic algorithm and reduces the energy of the error



a) No frame skip comparison



b) One frame skip comparison.



c) Two frame skip comparison.

Figure 3.9.1 Suzie comparison after three iteration with previous frame clean.

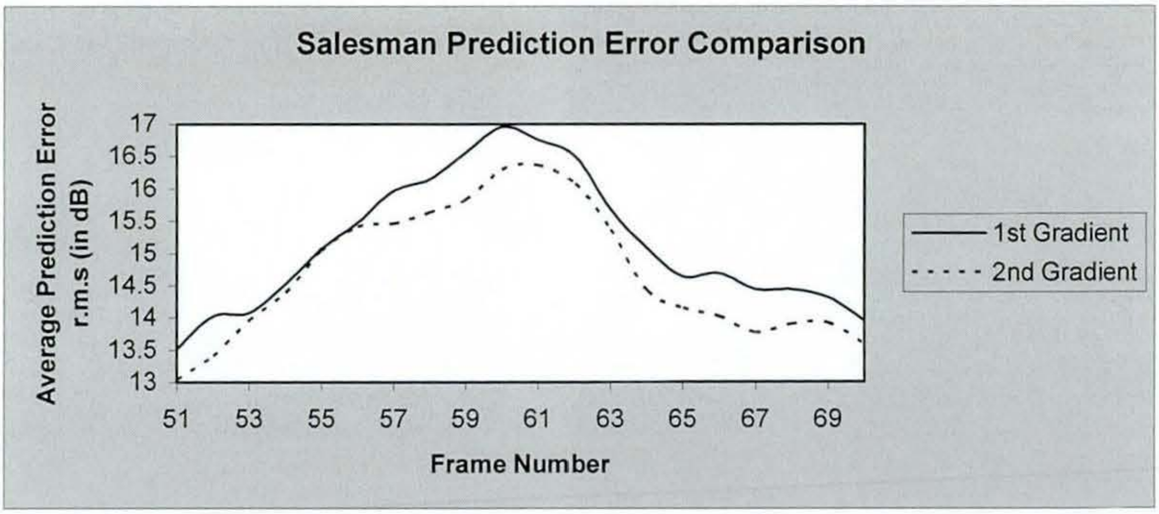
a) PCM frame of Suzie (previous frame)

b) PCM frame of Suzie (present frame)

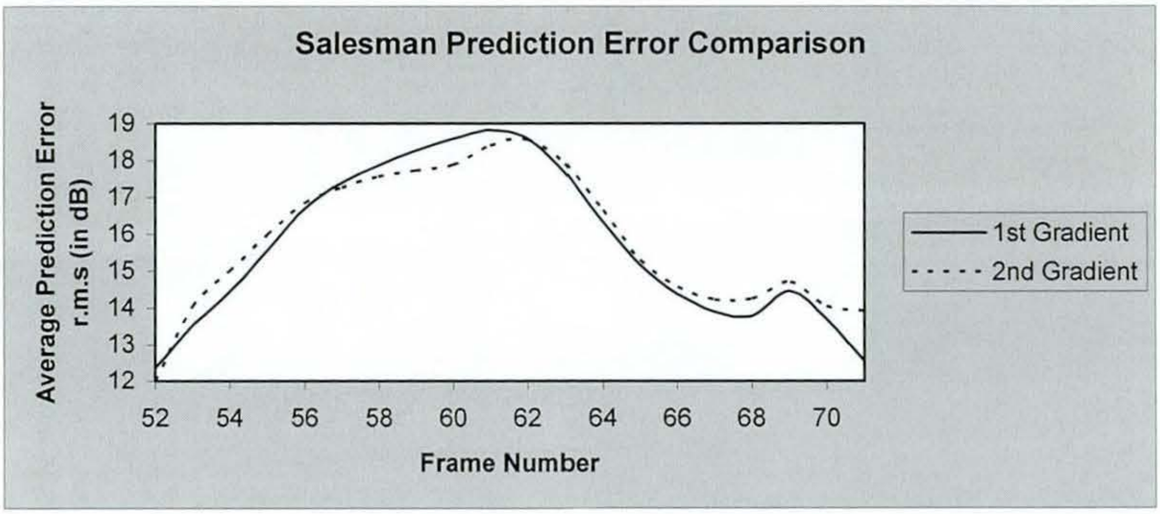
Prediction error with no MC for Suzie in a & b

d) Prediction error for Suzie in a & b (gradient)

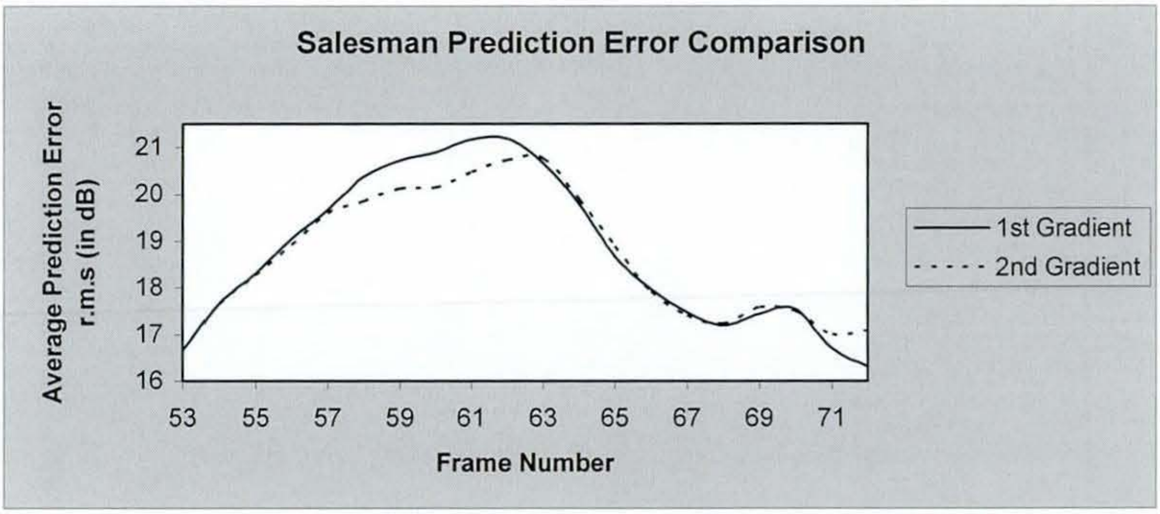
Figure 3.9.2 Prediction error comparison for two successive frame of Suzie after three iteration with previous frame clean.



a) No frame skip comparison.



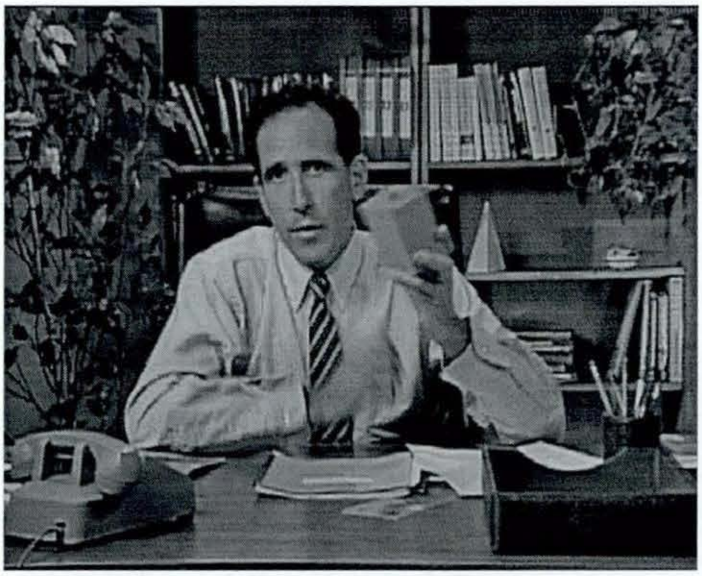
b) One frame skip comparison.



c) Two frame skip comparison.

Figure 3.9.3 Salesman comparison after three iteration with previous frame clean.

a) PCM frame of Salesman (previous frame)

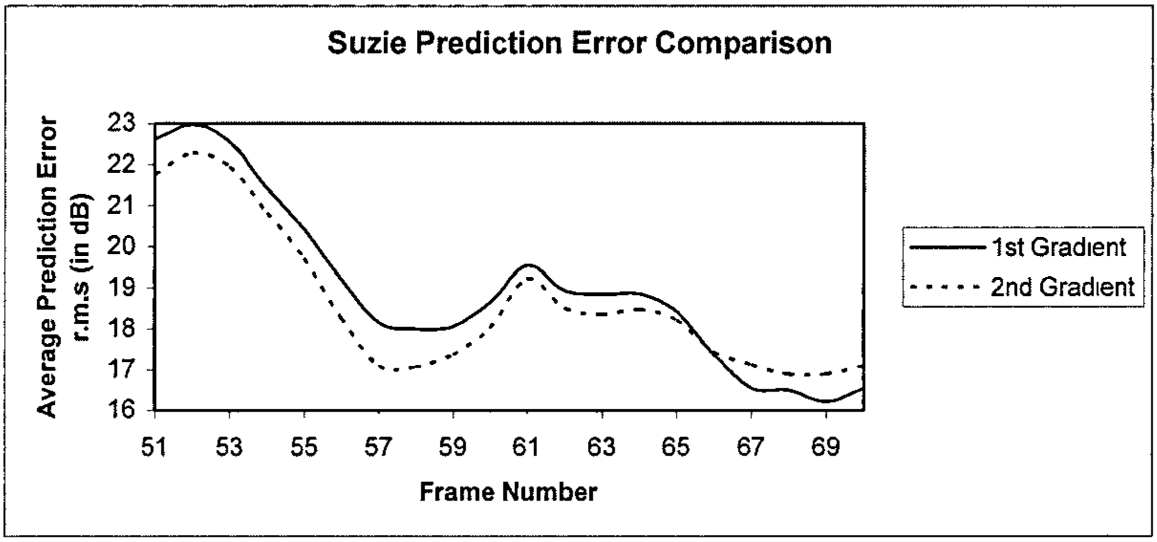


b) PCM frame of Salesman (present frame)

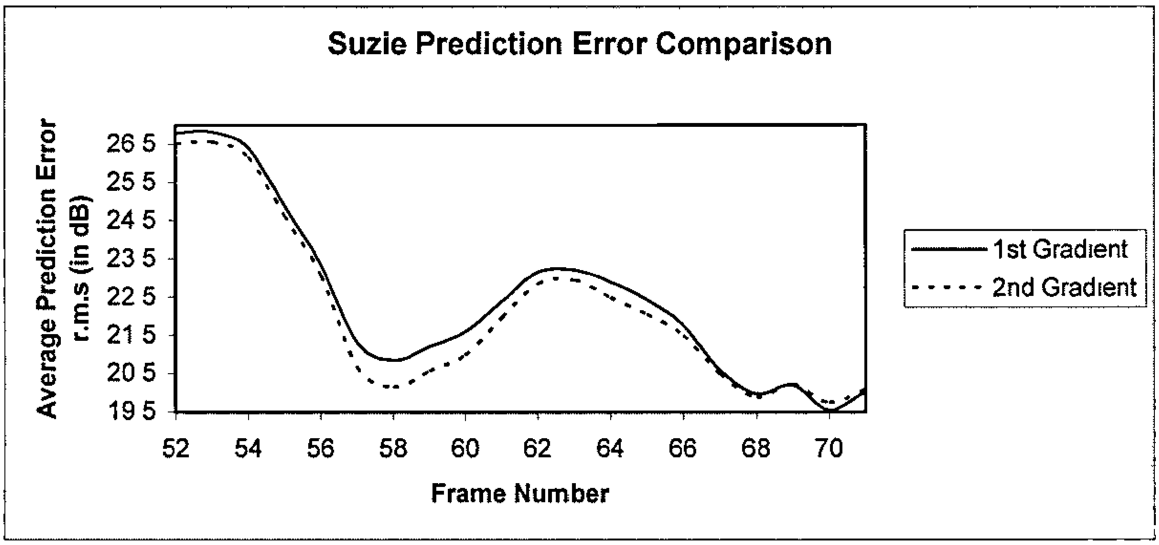
Prediction error with no MC for Salesman in a & b

d) Prediction error for Salesman in a & b (gradient)

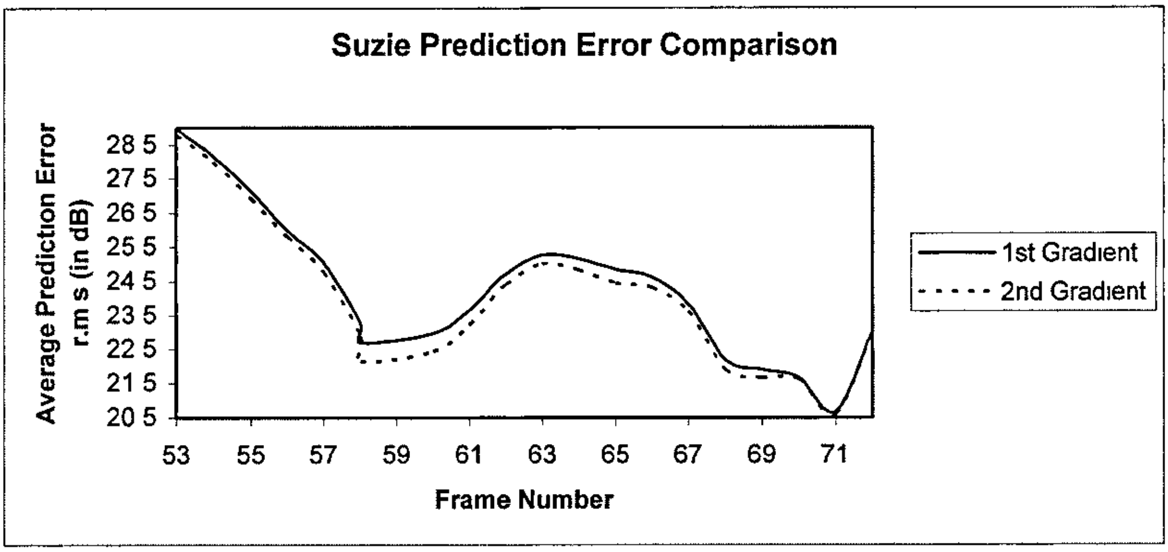
Figure 3.9.4 Prediction error comparison for two successive frame of Salesman after three iteration with previous frame clean.



a) No frame skip comparison

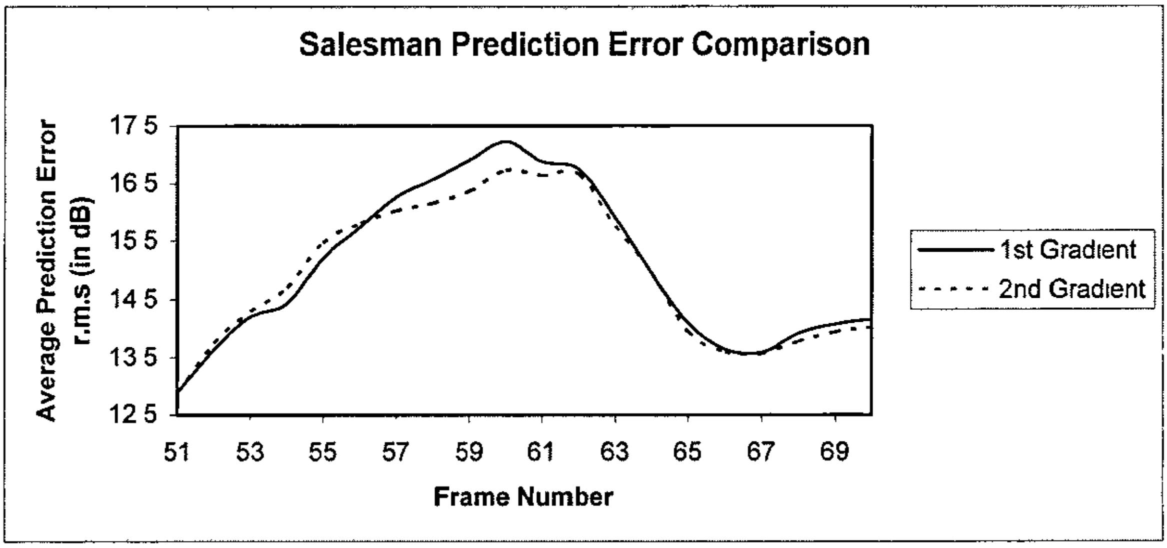


b) One frame skip comparison.

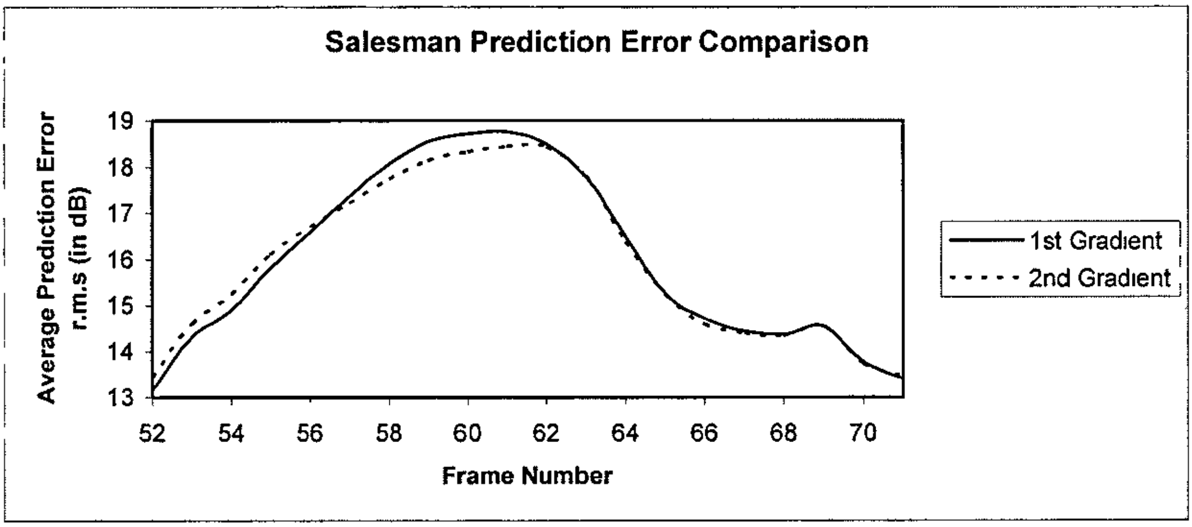


c) Two frame skip comparison.

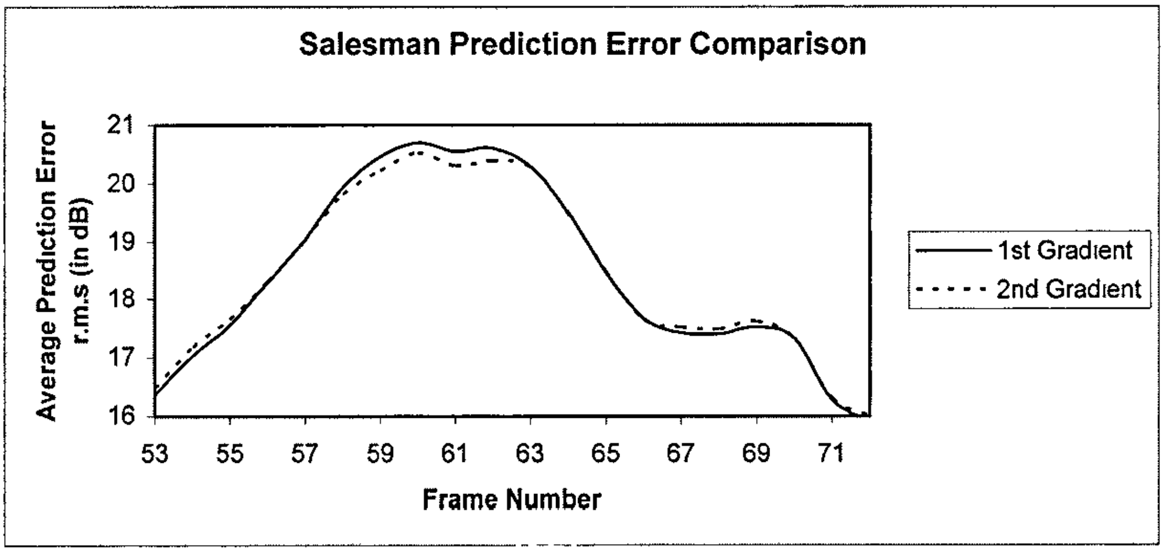
Figure 3.9.5 Suzie comparison after three iteration with previous frame reconstructed, with half pel accuracy on block matching, and system resetting to zero for each pel.



a) No frame skip comparison



b) One frame skip comparison.



c) Two frame skip comparison

Figure 3.9.6 Salesman comparison after three iteration with previous frame reconstructed, with half pel accuracy on block matching, and system resetting to zero for each pel.

Chapter 4

A New pel-recursive technique
For
MOTION COMPENSATED IMAGE SEQUENCE COMPRESSION

4.1 Background

One of the main developments in image coding in recent years is the application of mathematical models describing the motion of objects

For applications in dynamic scene analysis in a sequence of moving images, i e. television pictures, a moving object generates frame-to-frame luminance changes These luminance changes can be used in order to estimate the parameters of a mathematical model that describes the displacement and movement of the object. For instance, consider a simple moving edge as in Figure 4 1 1.

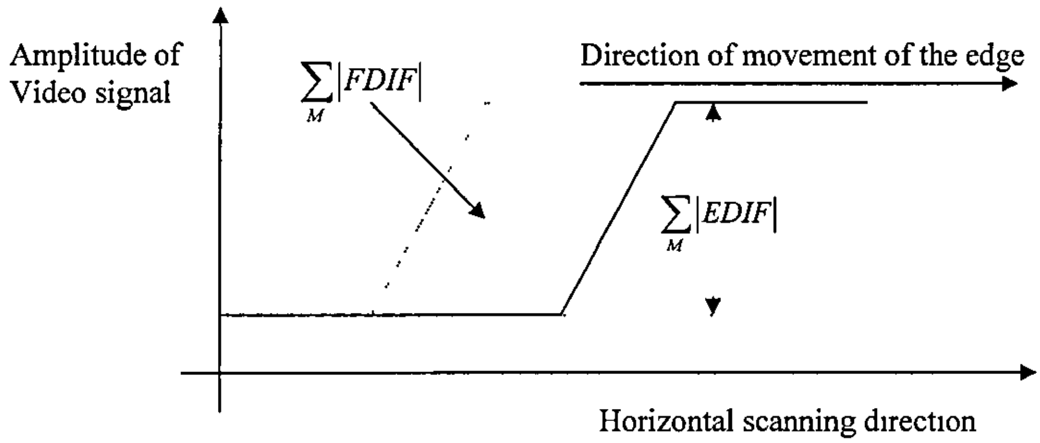


Figure 4.1.1 Illustration of displacement estimation. The dashed line indicates the position of the edge in the previous frame

$$\hat{D}_x = \hat{d}x = \sum_M |FDIF| \Big/ \sum_M |EDIF| \qquad \text{(Eqn 4.1.1)}$$

Where M is the moving area which is generally defined by frame differences greater than a given threshold

For the y direction, similar principle applies, therefore

$$\hat{D}_y = \hat{d}y = \sum_M |FDIF| \Big/ \sum_M |LDIF| \qquad \text{(Eqn 4.1.2)}$$

Motion models can also be used for improving the efficiency of predictive and interpolative television coding techniques. Because of the real-time computing requirements or VLSI architecture implementation [107], only relatively simple and easily realizable models which consider the translational component of motion have been worth while investigating

The x component of the displacement estimate $\hat{D}_{x_i}$, for a few different mathematical models can be summarized below [108] :-

First model

$$\hat{D}_{x_i} = \hat{D}_{x_{i-1}} + \varepsilon * \frac{\partial}{\partial x} R_{s_k s_{k-1}}(z, \hat{D}_{i-1}), \qquad \varepsilon = 1/1024 \text{ (recommended)}$$

(Eqn 4.1.3) [39], [109]

Where

$$R_{s_k s_{k-1}}(z, D) = \mathrm{E}\left[s_K(z) \bullet s_{K-1}(x - dx, y - dy)\right] \qquad \text{(Eqn 4.1.4)}$$

Its simplified update term (the difference between the present and previous estimation) is

$$\mathrm{U}_i = -\frac{1}{2}\varepsilon * \nabla_{\hat{D}_i} \sum_{j \in M} W_j \left[DFD(z, \hat{D}_{i-1})\right]^2$$

(Eqn 4.1.5)

Where $W_i \geq 0$ and $\sum_{j \in M} W = 1$

A quicker update can be achieved by increasing the constant convergence factor, ε However, this also implies a decrease of the achievable estimation accuracy which is limited by ε

Second model

$$\hat{D}_{x_i} = \hat{D}_{x_{i-1}} - \frac{\dfrac{\partial}{\partial x} R_{s_k s_{k-1}}(z, \hat{D}_{i-1})}{\dfrac{\partial^2}{\partial x^2} R_{s_k s_{k-1}}(z, \hat{D}_{i-1})} \qquad \text{(Eqn 4.1.6) [110]}$$

Its simplified update term is

$$U_i = -\frac{\sum_M \frac{\partial}{\partial dx}\left[DFD(z,\hat{D}_{i-1})\right]^2}{\sum_M \frac{\partial^2}{\partial dx^2}\left[DFD(z,\hat{D}_{i-1})\right]} \qquad \text{(Eqn 4 1 7)}$$

Third model

$$\hat{D}_{xi} = \hat{D}_{xi-1} + \frac{\frac{\partial}{\partial x} R_{s_k s_{k-1}}(z,\hat{D}_{i-1})}{\left|\frac{\partial^2}{\partial x^2} R_{s_{k-1} s_{k-1}}(z,0)\right| + \eta^2}, \quad \eta = 10 \qquad \text{(Eqn 4 1 8) [98]}$$

The correction term, η is introduced to avoid problems which would occur in areas of nearly constant luminance where $\partial s_{k-1}/\partial x$ is small and prevents the overshoots

Its simplified update term is

$$U_i = -\frac{\frac{1}{2} * \sum_M \frac{\partial}{\partial dx}\left[DFD(z,\hat{D}_i)\right]^2}{\sum_M \left[\frac{\partial}{\partial x} s_{k-1}(x - \hat{dx}_i, y - \hat{dy}_i)\right]^2 + \eta^2} \qquad \text{(Eqn 4 1 9)}$$

Fourth model

$$\hat{D}_{xi} = \hat{D}_{xi-1} - \frac{\frac{\partial}{\partial x} R_{s_k s_{k-1}}(z,\hat{D}_{i-1})}{\frac{1}{2}\left[\frac{\partial^2}{\partial x^2} R_{s_k s_{k-1}}(z,\hat{D}_i) + \frac{\partial^2}{\partial x^2} R_{s_k s_k}(z,0)\right]} \qquad \text{(Eqn 4.1.10) [111]}$$

Its simplified update term is

$$U_t = -\frac{\frac{1}{2} * \sum_M \frac{\partial}{\partial dx}\left[DFD(z,\hat{D}_t)\right]^2}{\frac{1}{2} * \sum_M \left[\frac{\partial}{\partial x} s_{k-1}(x - dx_t, y - dy_t) + \frac{\partial}{\partial x} s_k(z)\right] \frac{\partial}{\partial x} s_k(z)} \quad \text{(Eqn 4.1.11)}$$

Similarly, the x component of the displacement estimate $\hat{D}_{yt}$, can be deduced and follows the same format.

Comparing the above displacement estimation algorithms shows that these algorithms only differ in the denominator of the update term. The previous algorithm gives rise to faster convergence respectively, while the first algorithm gives slower convergence (virtually damped).

Block matching algorithms work on finding the best matching block by comparison where as pel recursive algorithms work on finding pel by pel estimation. As a good figure of judgment one would expect superiority by pel recursive algorithms over block matching algorithms But experimental results proved otherwise.

In spite of the expectation one might have had for existing pel recursive algorithms somehow, block matching algorithm have shown better performance in digital image compression As a result of this, block matching algorithms dominate compression applications e.g.: JPEG, MPEGs, H.263, and so on. This opportunity has given rise to research to improve the performance of block based algorithms further in many different applications. Less research has been directed in the area of pel recursive algorithms causing the pel recursive base algorithms to be left even further behind.

Of the four existing pel recursive algorithms, the first two were simulated as a bench mark for comparison between pel recursive algorithms and block matching algorithms, (in chapter 3).

As an educated guess one might suggest that the problem may not be with the pel recursive scheme in general. A better pel recursive algorithm may prove that the pel recursive algorithms perhaps should do better than block matching algorithms.

In the light of the above argument a new algorithm is proposed and simulated to show its validity (see the following sections).

4.2 A new algorithm, Motion compensated image sequence compression (algorithm)

For the sake of the analysis, it is assumed that the translational movement of an object is in a plane parallel to the camera and illumination is uniform. It is also assumed that the effect of uncovered background is negligible Under these assumptions, let *S (x, y, t)* denote the monochrome intensities at point (x, y) of a moving object in the image plane where it's translational movement is at a constant velocity of v_x and v_y. It can be shown that after Δt second (one frame period), the object moves to a new location where it can be shown,

$$S(x, y. t + \Delta t) = S[(x + v_x \Delta t)], [(y + v_y \Delta t): t] \qquad \text{(Eqn 4 2 1)}$$

After expanding the field in a power series in Δt and neglecting the higher order terms, the frame difference can be shown as,

$$S(x, y\ t + \Delta t) - S(x, y: t) = \frac{\partial}{\partial x} S(x, y: t)\, dx + \frac{\partial}{\partial y} S(x, y: t)\, dy \qquad \text{(Eqn 4 2 2)}$$

where d_x and d_y correspond to the horizontal and vertical components of the motion vector D Assuming $\frac{\partial}{\partial x} S(x, y. t)$ and $\frac{\partial}{\partial y} S(x, y: t)$ are known for each x, y, t, and defining EDIF, LDIF, and FDIF as the magnitude of the element, line, and frame difference at point n, from (4 2.2), we can write,

$$\text{FDIF} = \Phi_n^{\mathrm{T}} \mathrm{D} \qquad \text{(Eqn 4 2.3)}$$

$$\text{Where } \Phi_n = \begin{bmatrix} \frac{\partial}{\partial x} S(xn, yn\ t) \\ \frac{\partial}{\partial y} S(xn, yn : t) \end{bmatrix} = \begin{bmatrix} EDIF \\ \\ LDIF \end{bmatrix} \qquad \text{(Eqn 4.2.4)}$$

From Eqn(4 2 4) the frame difference (FDIF) measurement is,

$$\zeta_n = \Phi_n^{\mathrm{T}} \overline{D} + \text{noise} \qquad \text{(Eqn 4 2.5)}$$

where $\overline{\mathrm{D}} = [\bar{\mathrm{d}}(\mathrm{x}), \bar{\mathrm{d}}(\mathrm{y})]^{\mathrm{T}}$ is the motion vector estimate.

Now let

$$\mathrm{y} = (\zeta_n - \Phi_n^{\mathrm{T}} \alpha)^2 + \text{noise}$$

For the least-squares of α to be minimized, gives

$$\frac{\partial}{\partial \alpha} \mathrm{y} = -2\, \Phi_n^{\mathrm{T}} (\zeta_n - \Phi_n^{\mathrm{T}} \alpha) = 0$$

That is

$$\zeta_n = \Phi_n^{\mathrm{T}} \alpha$$

Or multiplying each side or equation by Φ_n

$$\Phi_n \zeta_n = \Phi_n \Phi_n^{\mathrm{T}} \alpha$$

For a cluster of M moving pels, the least-squares estimate of D, can be shown as,

$$\sum_{n=1}^{m} \Phi_n \zeta_n = (\sum_{n=1}^{m} \Phi_n \Phi_n^{\mathrm{T}}) \overline{D} \qquad \text{(Eqn 4 2 6)}$$

For, $\eta = \frac{1}{M}\sum_{n=1}^{M} \Phi_n \zeta_n$ and $R = \frac{1}{M}\sum_{n=1}^{M} \Phi_n \Phi_n^T$ (Eqn 4 2.7)

the estimated motion vector from Eqn(4.2.6) is obtained as,

$$\overline{D} = R^{-1}\eta \qquad \text{(Eqn 4 2 8)}$$

For recursive estimation of η and R, we can write

$$\eta_i = \eta_{i-1} + \Phi_n \zeta_n \qquad \text{(Eqn 4 2 9)}$$

$$R_i = R_{i-1} + \Phi_n \Phi_n^T \qquad \text{(Eqn 4 2 10)}$$

Based on the so-called matrix inversion lemma, which is :-

$$(A + X B X^{-T})^{-1} = A^{-1} - A X (B^{-1} + X^{-T} A^{-1} X)^{-1} X^{-T} A^{-1}$$

Substitute as follows :-

$A = R_{i-1}^{-1}$

$B = I$ Unit Matrix

$X = \Phi_n$

That is

$$(R_{i-1} + \Phi_n I \Phi_n^{-T})^{-1} = R_{i-1}^{-1} - R_{i-1} \Phi_n (I^{-1} + \Phi_n^{-T} R_{i-1}^{-1} \Phi_n)^{-1} \Phi_n^{-T} R_{i-1}^{-1}$$

In the above equation, the term in the left hand side bracket can be replaced, using Eqn(4 2 10), therefore

$$R_i^{-1} = R_{i-1}^{-1} - R_{i-1} \Phi_n (I^{-1} + \Phi_n^{-T} R_{i-1}^{-1} \Phi_n)^{-1} \Phi_n^{-T} R_{i-1}^{-1}$$

The term $(I^{-1} + \Phi_n^{-T} R_{i-1}^{-1} \Phi_n)^{-1}$ is scalar, Therefore the inverse of R_i can be obtained as,

$$R_i^{-1} = R_{i-1}^{-1} - \frac{R_{i-1}^{-1} \Phi_n \Phi_n^T R_{i-1}^{-1}}{1+\Phi_n^T R_{i-1}^{-1} \Phi_n} \qquad \text{(Eqn 4 2 11)}$$

Multiplying each side of the Eqn (4 2.11) by η_i and using Eqn (4.2.9)

$$R_i^{-1} \eta_i = R_{i-1}^{-1}(\eta_{i-1} + \Phi_n \zeta_n) - \frac{R_{i-1}^{-1} \Phi_n \Phi_n^T R_{i-1}^{-1}}{1+\Phi_n^T R_{i-1}^{-1} \Phi_n}(\eta_{i-1} + \Phi_n \zeta_n)$$

Simplifying the above, therefore -

$$R_i^{-1} \eta_i = R_{i-1}^{-1} \eta_{i-1} + R_{i-1}^{-1} \Phi_n \zeta_n - \frac{R_{i-1}^{-1} \Phi_n \Phi_n^T R_{i-1}^{-1}}{1+\Phi_n^T R_{i-1}^{-1} \Phi_n} \eta_{i-1} - \frac{R_{i-1}^{-1} \Phi_n \Phi_n^T R_{i-1}^{-1}}{1+\Phi_n^T R_{i-1}^{-1} \Phi_n} \Phi_n \zeta_n$$

Using Eqn(4.2.8)and simplifying further, That is

$$\overline{D}_i = \overline{D}_{i-1} - \frac{R_{i-1}^{-1} \Phi_n}{1+\Phi_n^T R_{i-1}^{-1} \Phi_n} \Phi_n^T R_{i-1}^{-1} \eta_{i-1} + \left(R_{i-1}^{-1} \Phi_n - \frac{R_{i-1}^{-1} \Phi_n \Phi_n^T R_{i-1}^{-1}}{1+\Phi_n^T R_{i-1}^{-1} \Phi_n} \Phi_n\right) \zeta_n$$

Using Eqn(4.2 8)and simplifying the above further, That is

$$\overline{D}_i = \overline{D}_{i-1} - \frac{R_{i-1}^{-1} \Phi_n}{1+\Phi_n^T R_{i-1}^{-1} \Phi_n} \Phi_n^T \overline{D}_{i-1} - \frac{R_{i-1}^{-1} \Phi_n}{1+\Phi_n^T R_{i-1}^{-1} \Phi_n} \left(1 + \Phi_n^T R_{i-1}^{-1} \Phi_n - \Phi_n^T R_{i-1}^{-1} \Phi_n\right) \zeta_n$$

Finally it gives rise to

$$\overline{D}_i = \overline{D}_{i-1} - \frac{R_{i-1}^{-1}\,\Phi_n}{1+\Phi_n^T\,R_{i-1}^{-1}\,\Phi_n}(\Phi^T\,\overline{D}_{i-1} - \zeta) \qquad \text{(Eqn 4 2 12)}$$

In the above equation, the term within the brackets can be replaced by what is known as the Displaced Frame Difference, DFD. Thus,

$$\overline{D}_i = \overline{D}_{i-1} - \frac{R_{i-1}^{-1}\,\Phi_n}{1+\Phi_n^T\,R_{i-1}^{-1}\,\Phi_n}[DFD(\mathrm{x}, \mathrm{y}, \overline{D}_{i-1})] \qquad \text{(Eqn 4 2.13)}$$

To avoid matrix inversion at each iteration, Eqn (4.2.13) can be simplified by ignoring the x and y cross terms in calculating ϕ_n and R. Thus, from Eqn (4.2.4) and Eqn (4.2.7),

$$\Phi_n(\mathrm{x}) = EDIF \quad \text{and} \quad \Phi_n(\mathrm{y}) = LDIF \qquad \text{(Eqn 4.2.14)}$$

$$\mathrm{R(x)} = \frac{1}{M}\sum_m EDIF_m^2 \quad \text{and} \quad \mathrm{R(y)} = \frac{1}{M}\sum_m LDIF_m^2 \qquad \text{(Eqn 4.2 15)}$$

Applying Eqn(4 2.14) to Eqn(4 2.13), the components of the motion displacement

$$\overline{d}_i(\mathrm{x}) = \overline{d}_{i-1}(\mathrm{x}) - \frac{EDIF}{\frac{1}{M}\sum EDIF^2 + EDIF^2}\{DFD[\mathrm{x}, \mathrm{y}, \overline{d}_{i-1}(\mathrm{x})]\} \qquad \text{(Eqn 4.2.16)}$$

$$\overline{d}_i(\mathrm{y}) = \overline{d}_{i-1}(\mathrm{y}) - \frac{LDIF}{\frac{1}{M}\sum LDIF^2 + LDIF^2}\{DFD[\mathrm{x}, \mathrm{y}, \overline{d}_{i-1}(\mathrm{x})]\} \qquad \text{(Eqn 4 2.17)}$$

According to the developed new algorithm for motion compensated image sequence compression [112].

$$\hat{D}_x = \hat{D}_{x-1} - \frac{EDIF}{\sum_R EDIF^2 + EDIF^2} * \left\lfloor \varepsilon * (EDIF)(\hat{D}_{x-1}) - FDIF \right\rfloor \qquad \text{(Eqn 4.2.18)}$$

or

$$\hat{D}_x = \hat{D}_{x-1} - \frac{EDIF}{\sum_R EDIF^2 + EDIF^2} * \left\lfloor \varepsilon * (EDIF)(\hat{D}_{x-1}) + DFD \right\rfloor \qquad \text{(Eqn 4 2 19)}$$

Which can also be simplified as

$$\hat{D}_x = \hat{D}_{x-1} - \frac{EDIF}{\sum_R EDIF^2 + EDIF^2} * \left[\varepsilon * EDIF * DFD \right] \qquad \text{(Eqn 4 2 20)}$$

And similarly

$$\hat{D}_y = \hat{D}_{y-1} - \frac{LDIF}{\sum_R LDIF^2 + LDIF^2} * \left\lfloor \varepsilon * (LDIF)(\hat{D}_{y-1}) - FDIF \right\rfloor \qquad \text{(Eqn 4 2.21)}$$

or

$$\hat{D}_y = \hat{D}_{y-1} - \frac{LDIF}{\sum_R LDIF^2 + LDIF^2} * \left\lfloor \varepsilon * (LDIF)(\hat{D}_{y-1}) + DFD \right\rfloor \qquad \text{(Eqn 4 2 22)}$$

Which can also be simplified as

$$\hat{D}_y = \hat{D}_{y-1} - \frac{LDIF}{\sum_R LDIF^2 + LDIF^2} * \left[\varepsilon * LDIF * DFD \right] \qquad \text{(Eqn 4.2.23)}$$

Where ε, the convergency parameter to control the rate of convergence, is recommended (experimentally) to be in the region of 0.98 to 1.00.

The above recursion to update $\hat{D}_k$ is carried out only in the moving area of the current frame, i.e., for those pels where

$$\sum_{j=-p}^{+p} \left| b(z_{k+j}, t) - b(z_{k+j}, t-\tau) \right| \geq \text{Threshold} \qquad \text{(Eqn 4.2 24)}$$

Otherwise

$$\hat{D}_x = \hat{D}_{x-1} \text{ and } \hat{D}_y = \hat{D}_{y-1} \qquad \text{(Eqn 4 2.25)}$$

The threshold, Threshold, is pre-selected. It should be noted that the choice of the Threshold is mainly based on camera noise, light variation, and so on. In this thesis for the sequences used, a figure of 8 to14 out of 256 intensity levels was chosen. A poor choice of the Threshold figure, far off from the true value will cause errors

Recursively $\hat{D}_x$ and $\hat{D}_y$ are updated using Eqn (4.2.20) and Eqn (4.2.23) where for each step, the update term attempts to improve the estimate of *D*. The ultimate goal is the minimization of the magnitude of prediction error, *DFD* If a pel at location Z_a is predicted with $\hat{D}_{x-1}$ and $\hat{D}_{y-1}$ having intensities of $b\left(z - \hat{D}_{x-1}, t-1\right)$ and $b\left(z - \hat{D}_{y-1}, t-1\right)$ respectively and results in a prediction error of $DFD(\mathrm{z}, \hat{D}_{k-1})$ the prediction should attempt to create new estimations, for $\hat{D}_x$ and $\hat{D}_y$ such that :-

$$\left| DFD(\mathrm{z}, \hat{D}_k) \right| \leq \left| DFD(\mathrm{z}, \hat{D}_{k-1}) \right| \qquad \text{(Eqn 4 2.26)}$$

i e the prediction error is reduced.

The predictor is based on intensities in the previous frame and current frame

4.2.1 Interpolation

Using affirmation for the new algorithm to get the first estimated value for the displacement vector, the intensity of the pel displaced is estimated by means of an interpolation technique. For consistency with the pel-recursive motion estimation shown in pervious chapter and for simplicity, the same algorithm can be used, that is the following formula -

$$I = I_D + \hat{D}_x(I_A - I_D) + \hat{D}_Y(I_B - I_D) + \hat{D}_X * \hat{D}_Y(I_D + I_C + I_A + I_B) \qquad \text{(Eqn 4 2.1 1)}$$

Finally the displacement vector D is decomposed into two parts, the integral part and the non-integral part D_F

4.3 Implementation and Experimental results

As far as implementation and simulation are concerned, the great importance of the work lies in the calculation of the components defining the formula, i.e. such as line, element, and displacement frame differences. For these, different ways and techniques can be examined; e g : interpolation, non-interpolation, averaged using different causal supports (Figure 4.3.1), etc. Not every single pel needs to be motion compensated, therefore a masking mechanism should be utilized e g ·- where frame difference, $|FDIF| \leq$ threshold, no prediction is needed. This is classed as a non-moving area.

X X X X X X X

X X X X X X X

X X X X O

Figure 4.3.1 : A second order causal support

The computation involved in Eqn (4.2.20) and Eqn (4.2.23) is performed recursively. At each iteration the estimated motion displacement is applied to measure a new DFD. This would first require obtaining the location of the displaced pel on the previous frame, based on the estimated components of motion displacement. Since the motion estimates are expected to be non-integer, the luminance value of the displaced pel is predicted by a two dimensional interpolator which uses the four corners of the surrounding pels in a two dimensional grid. In our experiments, the DFD is measured at two locations with reference to the current pel; the pel above (i.e., previous line), and the previous pel along the same line. The average of the two DFDs (with equal weightings) is then used to update the displacement estimates.

In this thesis for simplicity, the non-interpolated averaged 3 by 2 causal support (Figure 4.3.2) is used for line and element differences and displacement frame difference as normal (a pel value of a frame - the interpolated pel value of previous frame), with the convergency parameter, $\varepsilon = 0.98$. The maximum update limit for consistency proposes was chosen to be $|FDIF| \leq 9$ for non-moving areas This is done for two different sequences, "Suzie" and "Salesman".

X(1) X(2) X(3) X(4)

X(5) X(6) O

Figure 4.3.2 : causal support.

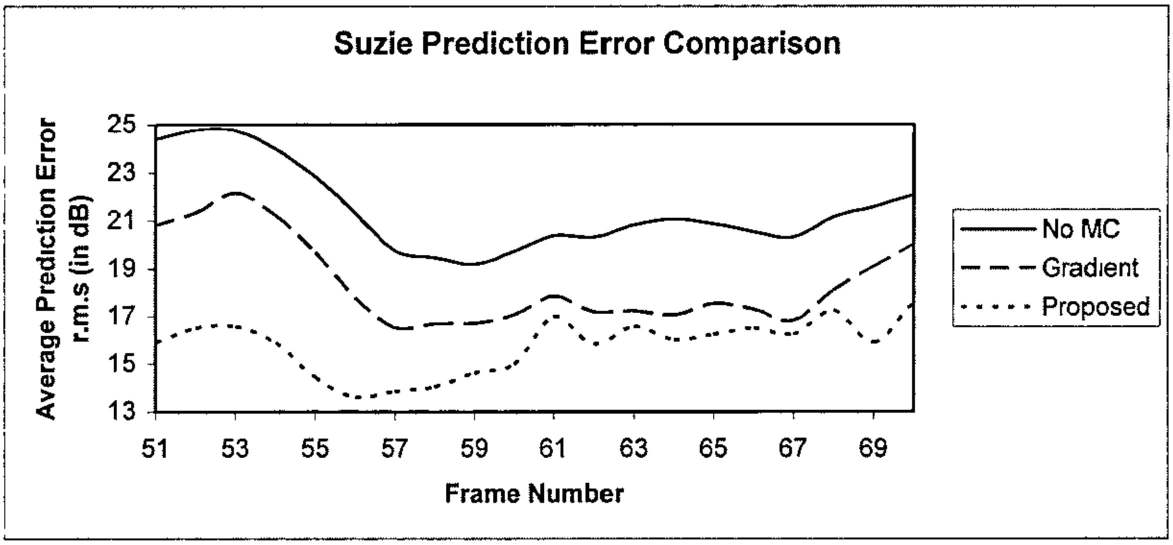
It should be noted that as for causal support concerns, the previous pel value of X(6) and the last two previous line pel values of X(3) and X(4) have the most importance in order to estimate any of element , line, frame or displace frame difference (see Figure 4 3 2)

Further constraint or limitation on the predictor can be used to augment the prediction strategy, the following rule (Eqn 4 3.1) can be used to switch or move adaptively between them on a pel by pel basis.

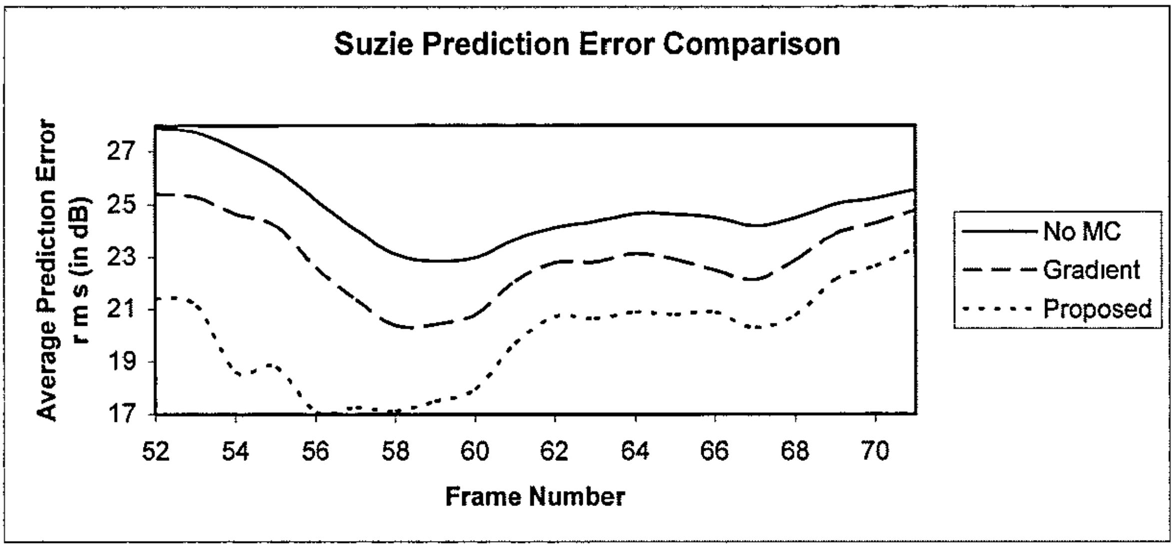
$$\sum_{j=-m}^{+m} w_i \left|FDIF(z_{k+j})\right| \geq \sum_{j=-m}^{+m} w_i \left|DFD(z_{k+j}, \hat{D}_k)\right| \qquad \text{(Eqn 4.3.1)}$$

Figures 4 3 3 to 4.3.11 show and indicate the validity of the new pel-recursive motion estimation algorithm. The graphical and pictorial results are compared for the existing and the new algorithms. In the pictorial results (Figures 4.3 4 and 4.3.6) it can clearly, but subjectively be seen that an improvement occurs in reducing the prediction error for two different successive frames of "Suzie" and "Salesman".

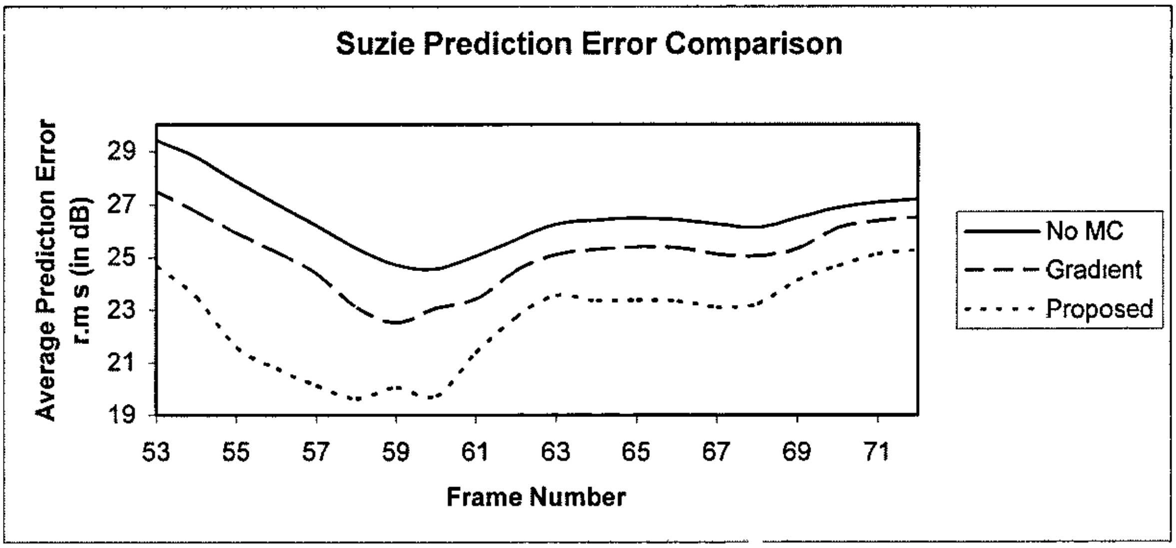
In the Figures 4 3 3 to 4.3.9, the clean frames (the PCM value of pels in the frame) were used In Figures 4.3.10 and 4 3.11, reconstructed frames (as in most codecs, the clean frame is not available in the decoder, therefore the predicted quantized reconstructed frame is used throughout) were used.



a) No frame skip comparison



b) One frame skip comparison.



c) Two frame skip comparison.

Figure 4.3.3 Suzie comparison after three iterations with the previous frame clean.

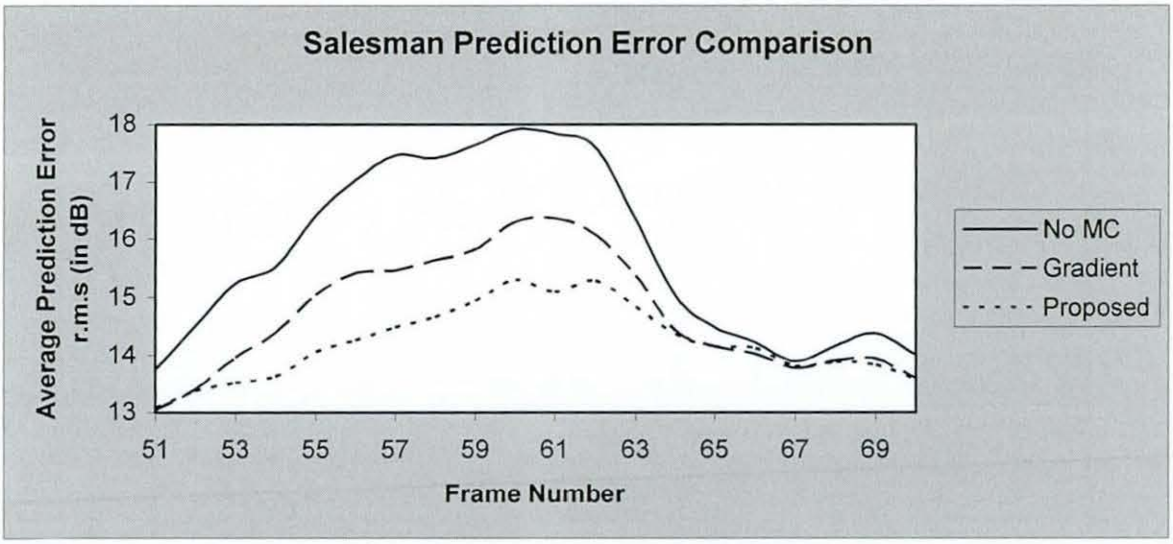
a) PCM frame of Suzie (previous frame)

b) PCM frame of Suzie (present frame)

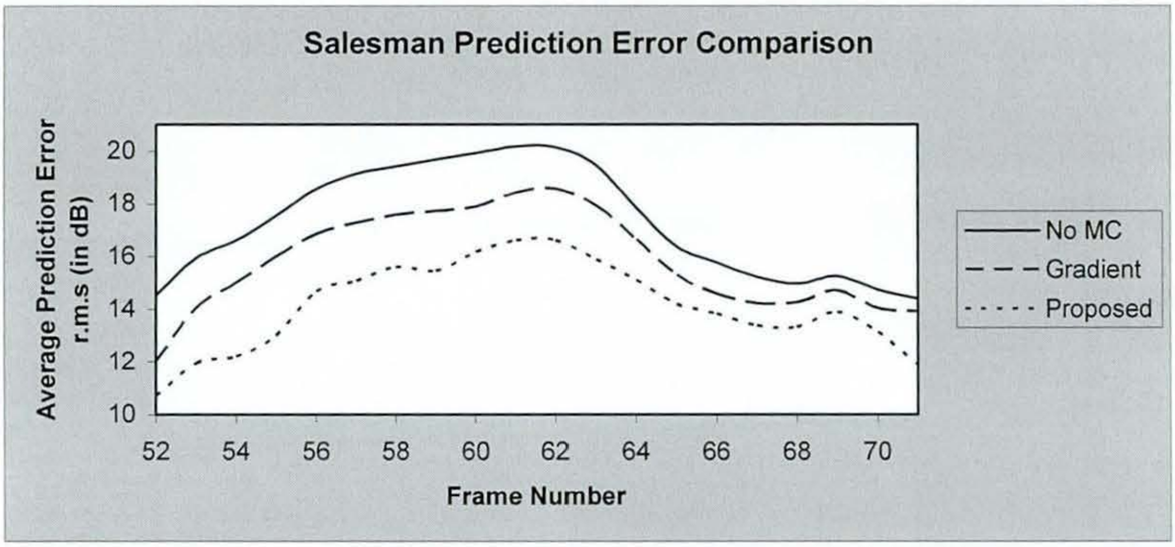
c) Prediction error for Suzies in a & b (for the gradient)

d) Prediction error for Suzies in a & b (for the proposed)

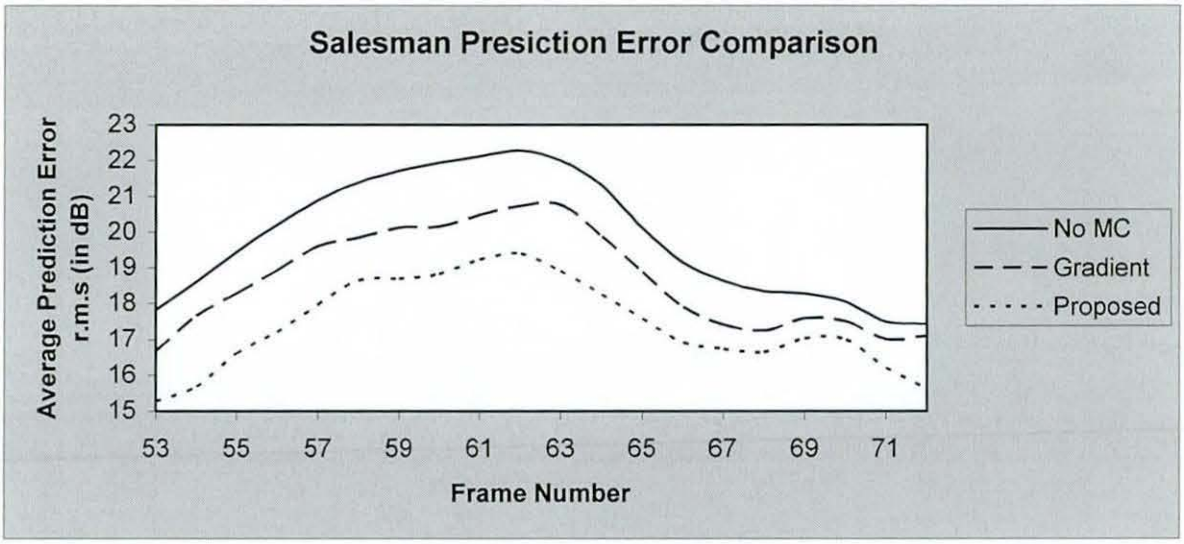
Figure 4.3.4 Prediction error comparison for two successive frames of Suzie after three iterations with previous frame clean.



a) No frame skip comparison.

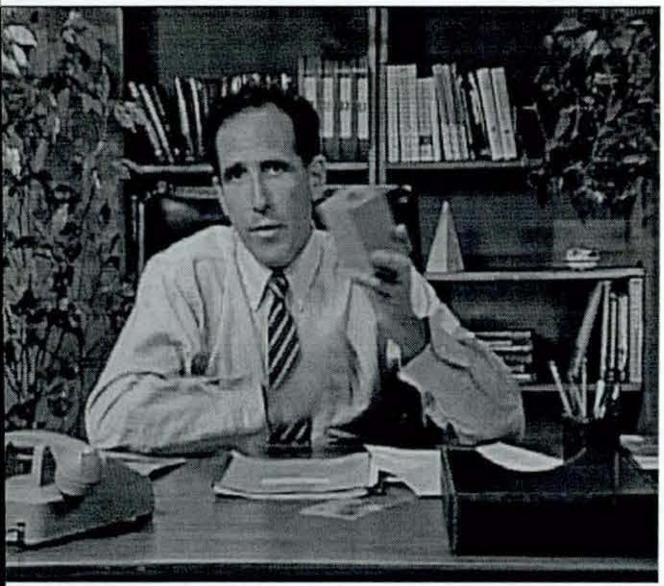


b) One frame skip comparison.

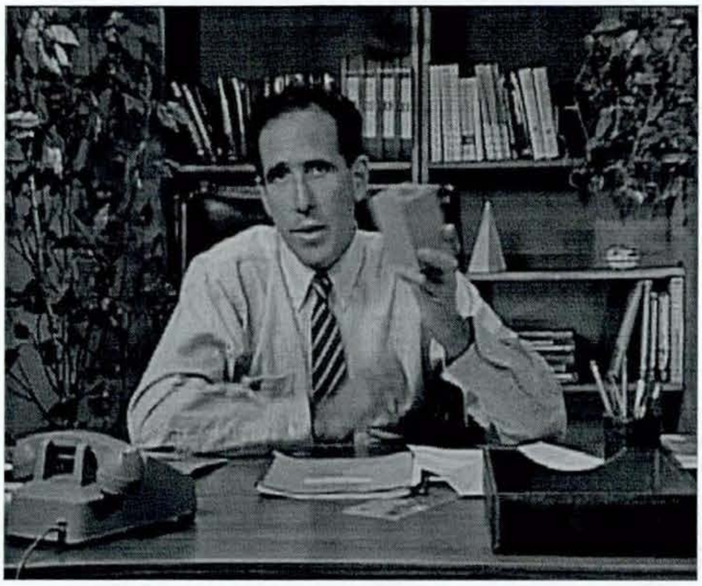


c) Two frame skip comparison.

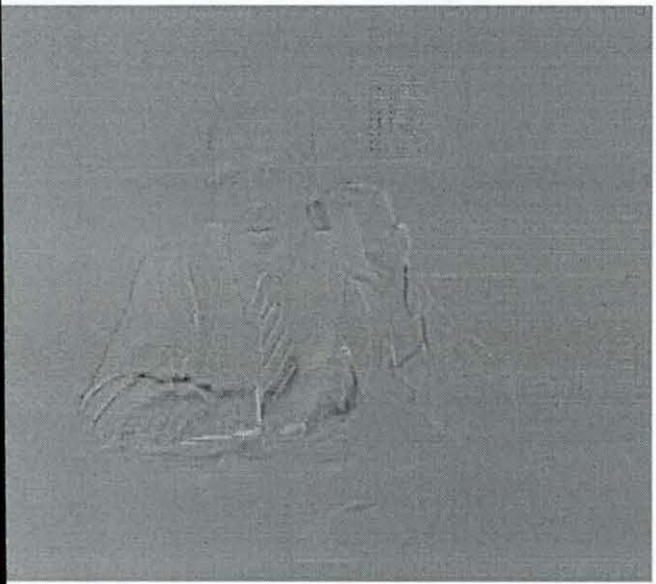
Figure 4.3.5 Salesman comparison after three iterations with the previous frame clean.



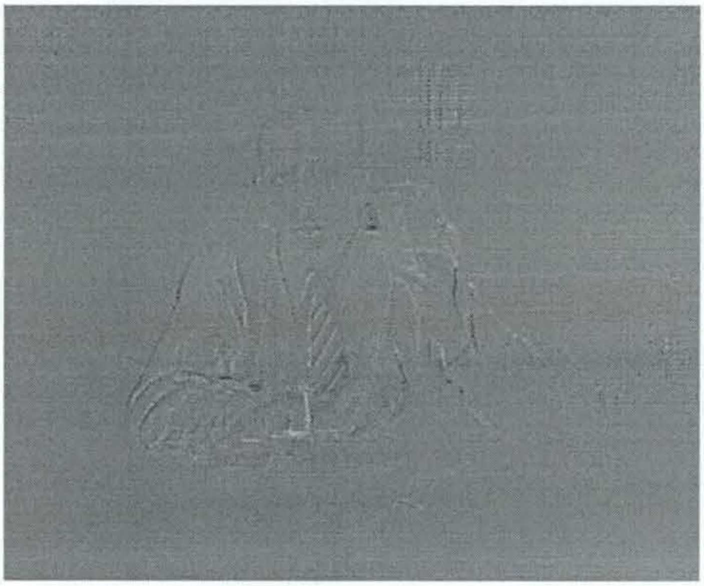
a) PCM frame of Salesman (previous frame)



b) PCM frame of Salesman (present frame)



c) Prediction error for Salesman in a & b
(for the gradient)



d) Prediction error for Salesman in a & b
(for the proposed)

Figure 4.3.6 Prediction error comparison for two successive frame of Salesman after three iterations with the previous frame clean.

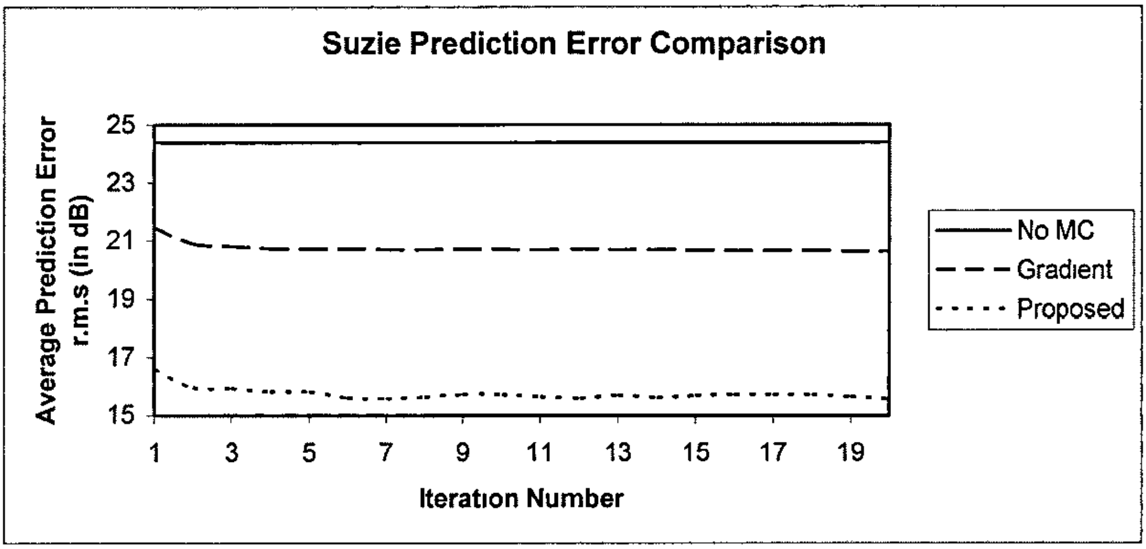


Figure 4.3.7 Suzie comparison for 20 iterations with previous frame clean.

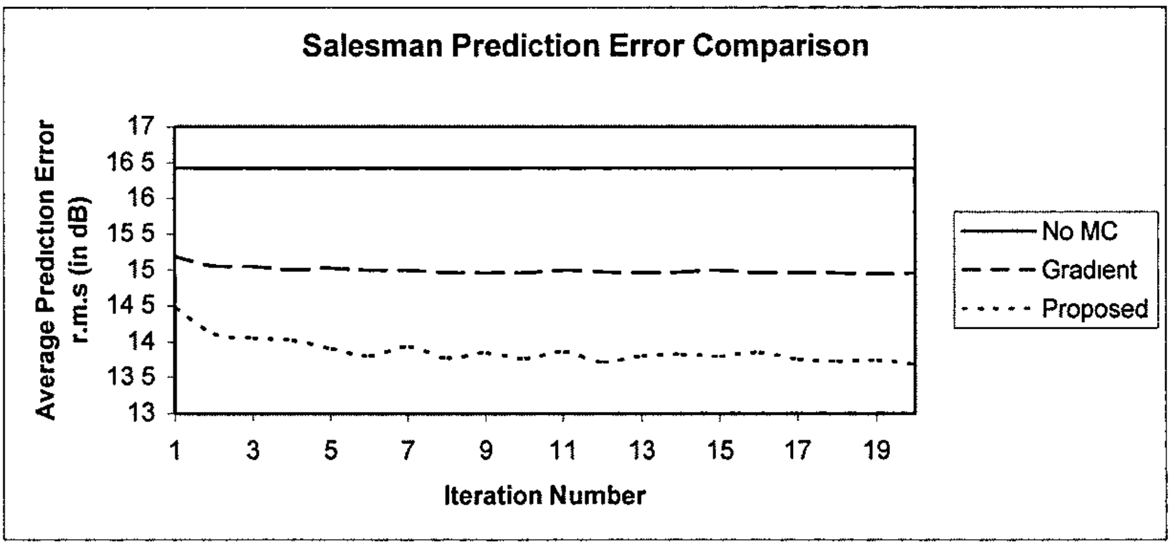
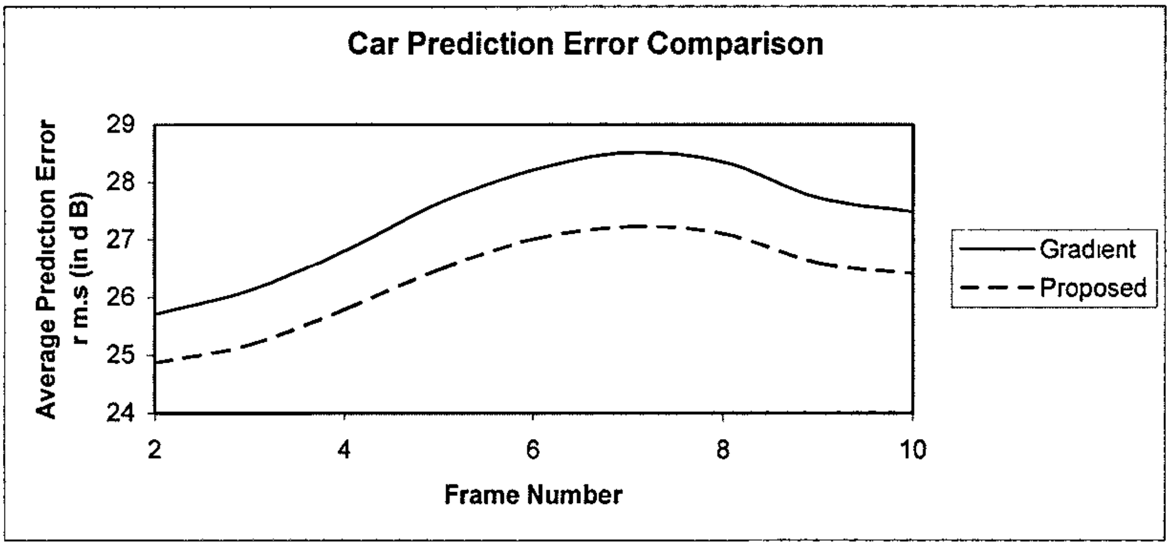
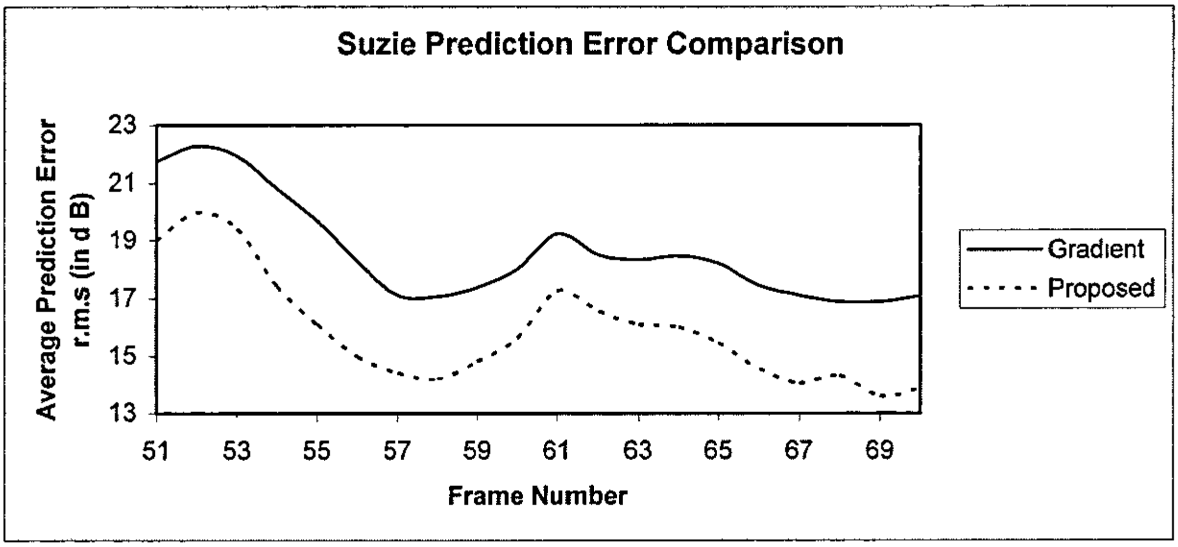


Figure 4.3.8 Salesman comparison for 20 iterations with the previous frame clean

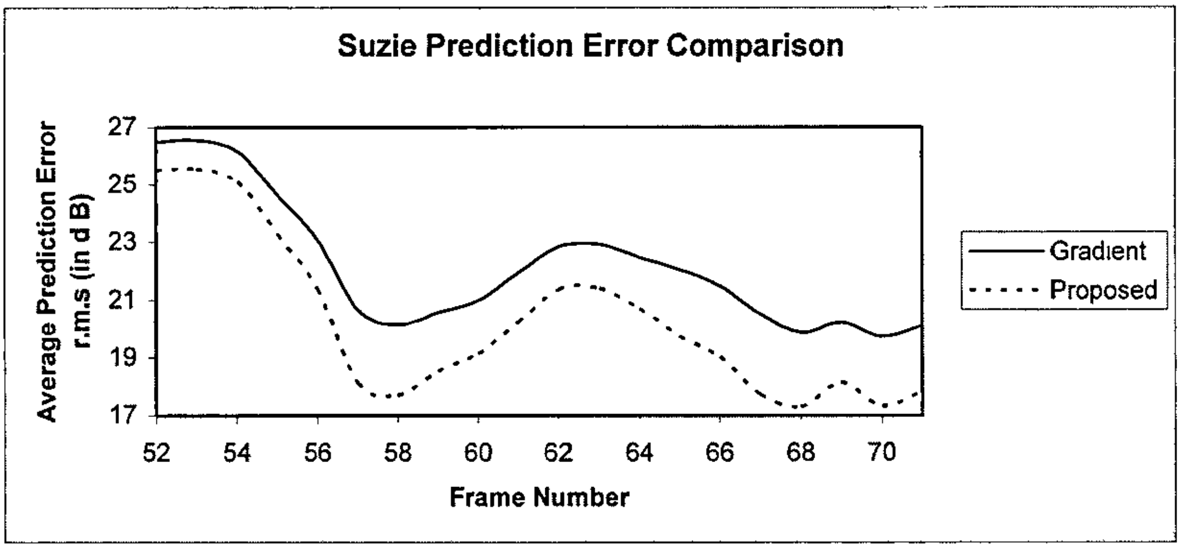


a) No frame skip comparison

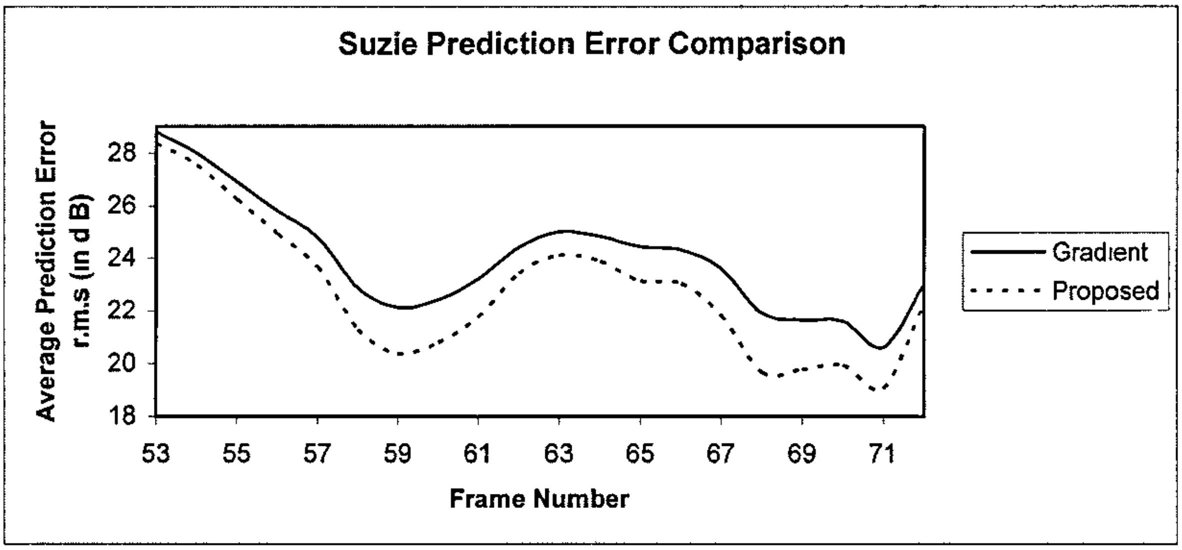
Figure 4.3.9 Car comparison after three iterations with the previous frame clean.



a) No frame skip comparison.

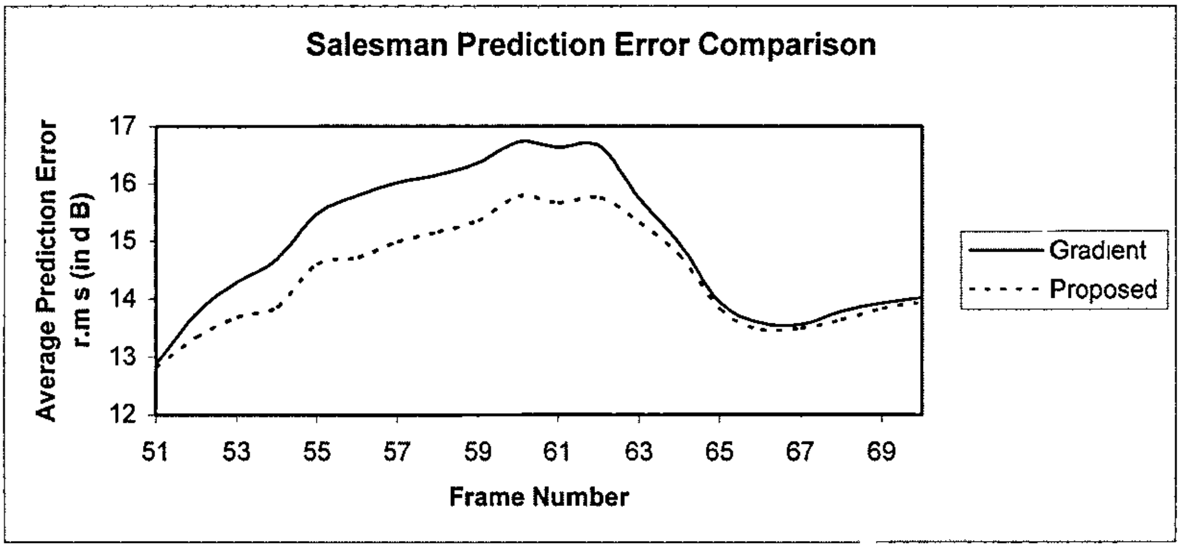


b) One frame skip comparison.

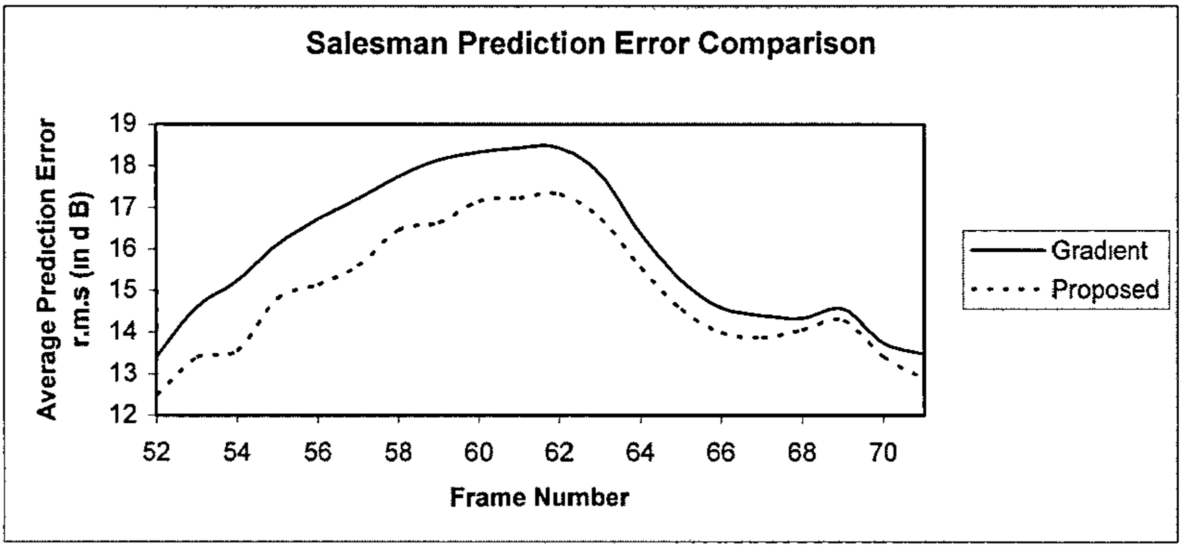


c) Two frame skip comparison.

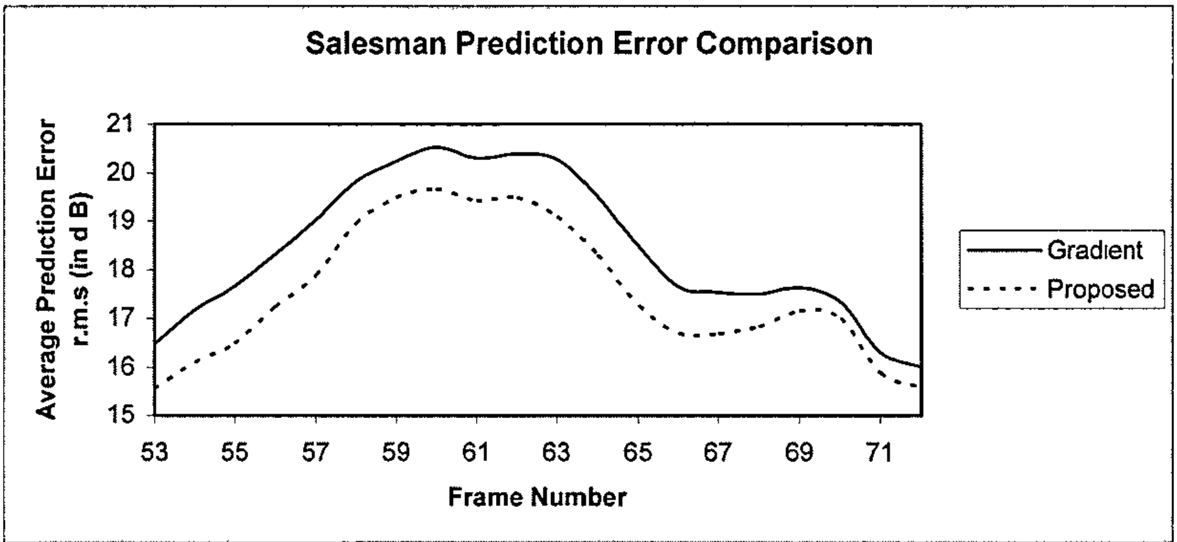
Figure 4.3.10 Suzie comparison after three iterations with the previous frame reconstructed.



a) No frame skip comparison



b) One frame skip comparison.



c) Two frame skip comparison.

Figure 4.3.11 Salesman comparison after three iterations with the previous frame reconstructed.

4.4 Results and summary

As was expected the outcome of the use of the proposed algorithm over the existing modified steepest gradient algorithm is that the new pel recursive algorithm has proven to produce a better result than the existing ones The Figures 4 3 3 to 4 3 11 indicate the statement regardless to whether clean or reconstructed frames are employed.

In the Figures 4 3 3, 4.3 5, 4.3 10, and 4.3.11; the comparisons were done for different frame skips. This is mainly to show that the proposed algorithm does always have better performance over the existing ones. In real time practical applications one may have to use sequences with different frame skips, especially in situations where we are dealing with sequences consisting of bigger or larger frames.

As can be seen from the Figures 4.3 3, 4.3 5, and 4.3.7- 4.3.11; there has been a great improvement of 1 5 dB, over the existing pel recursive algorithm This is achieved by the proposed pel recursive algorithm. This is quite a substantial improvement when compared with the case when no motion compensation is employed. The graphs in Figures 4.3.4 and 4.3.6 (c and d sections) depict that the proposed pel recursive algorithm should result in a good prediction error in comparison with the existing pel recursive algorithm.

Strictly speaking, the proposed pel recursive gradient has quite fast convergency, therefore fewer iterations will be needed. In spit of the fast convergency which is acceptable by most applications, it should not be overlooked that in some sequences little more convergence can be obtain by increasing the number of iterations (see Figures 4 3.7 and 4.3 8)

In some cases we might deal with sequences with very fast motion, where even block matching motion compensation often results in a poor compression; the proposed pel recursive algorithm can show good improvement in reducing the prediction error

Never the less, the result from the proposed algorithm can not compete with the old block matching scheme as it is; this would require a better pel recursive algorithm to be developed in the future.

Finally looking at Figures 4.3.4(c) and 4.3.4(d) and Figures 4.3 6(c) and 4 3 6(d), in these images, relatively darker or lighter patches represent the degree of inaccuracies in estimating the components of the motion displacement. Comparing the two images 4.3.4(c) and 4.3.4(d) and also Figures 4.3.6(c) and 4.3.6(d) confirm the superior performance of the proposed scheme over the modified steepest-descent algorithm, particularly in regions where the motion activities are relatively high.

Chapter 5

Application for hierarchical system

Pel recursive motion compensation has not yet been able to replace block matching motion compensation in hierarchical systems (e g. H.263, MPEGs, and so on); In the light of this, this chapter, looks at an application developed in view of a paper by Bierling [113].

5.1 Overview

Block matching is a widely used displacement estimation method, and can easily be implemented in hardware. Using block matching, a displacement vector is obtained by matching a rectangular measuring window, consisting of a certain number of neighboring picture elements, with a corresponding measuring window within a search area, placed in the next successive or the next preceding image. The match is achieved by searching the spatial position of the extremum of a matching criteria (e g : MAD, the mean absolute

displaced frame difference) The resulting displacement vector is then taken to be the motion vector for all the picture elements inside the measuring window.
The basic assumption of the displacement estimation techniques used is that neighboring picture elements have the same motion parameters. It is not possible to obtain a displacement estimate for every isolated picture element of a block

The known block matching techniques provide fairly good results for motion compensation prediction in general, as their computation and the complexity are low and the prediction error is remarkably small when using the achieved motion compensation However, the match obtained by block matching is an optimum only in the sense of a minimum MAD, the mean absolute displaced frame difference; but frequently it does not correspond to the true motion of the objects.

The reliability of the displacement estimate depends on the size of the chosen measuring windows, in conjunction with the present amount of motion. The estimate tends to be unreliable, if small measuring windows are used and the displacement is large. The smaller the measuring window, the higher is the probability that there are blocks (and hence will be selected by the matching criteria) in the corresponding search area, containing a more similar or identical pattern of picture elements, although there is no correspondence in the sense of motion. Therefore, large measuring windows are required in order to cope with large displacement. Thus, the known block matching techniques fail frequently as a result of using a fixed measuring window size [113]

In order to take into account the above problem, a hierarchical block matching for displacement estimation was suggested by Bierling [113]. The hierarchical structure uses distinct sizes of measuring windows at different levels of the hierarchy The estimator starts with large measuring windows at the highest level. From one level to the next level of the hierarchy, the size of the measuring window is decreased The displacement estimate is obtained recursively, i.e. at each level of the hierarchy, the resulting estimate serves as an initial guess for the next lower level. The first hierarchy levels serve to provide a reliable estimate of the major part of a large displacement, whereas the last levels serve to estimate the remaining part of the displacement accurately. Figure 5.1 shows the principle of hierarchical displacement estimation for the example of three levels. A displacement vector between two successive frame of images is achieved as the sum of three estimates, using three different measuring window sizes [113]. The second hierarchy level starts motion compensation using the results of the first level, i.e. the search points of the search procedure are displaced by the estimate of the first level, and carries on the same way through the rest of the levels recursively.

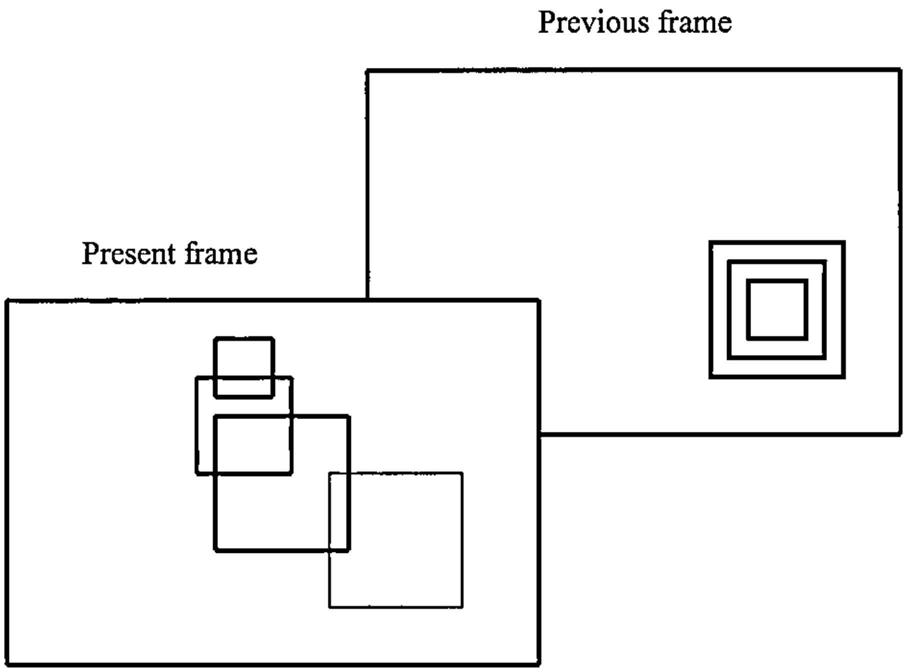


Figure 5.1 Principle of hierarchical displacement estimation for three hierarchy levels

5.2 Application

In order to reduce the computational effort resulting from large windows, sub-sampling inside the measuring window can be performed [113]. This raises the idea of applying the same method of sub-sampling to pel-recursive, in particular for the proposed pel-recursive method. If the task is proven satisfactory, this can perhaps be used to have an affect of final tuning on the displacement which is estimated by block matching.

Looking at the example from another angle the performance of any hierarchical codec can be improved by introducing pel-recursive motion compensation. In view of this let allow and investigate if pel-recursive motion compensation can be active side by side in the presence of block matching motion compensation. This may possibly have some benefit for codecs standards like H 263, MPEGs, or any other.

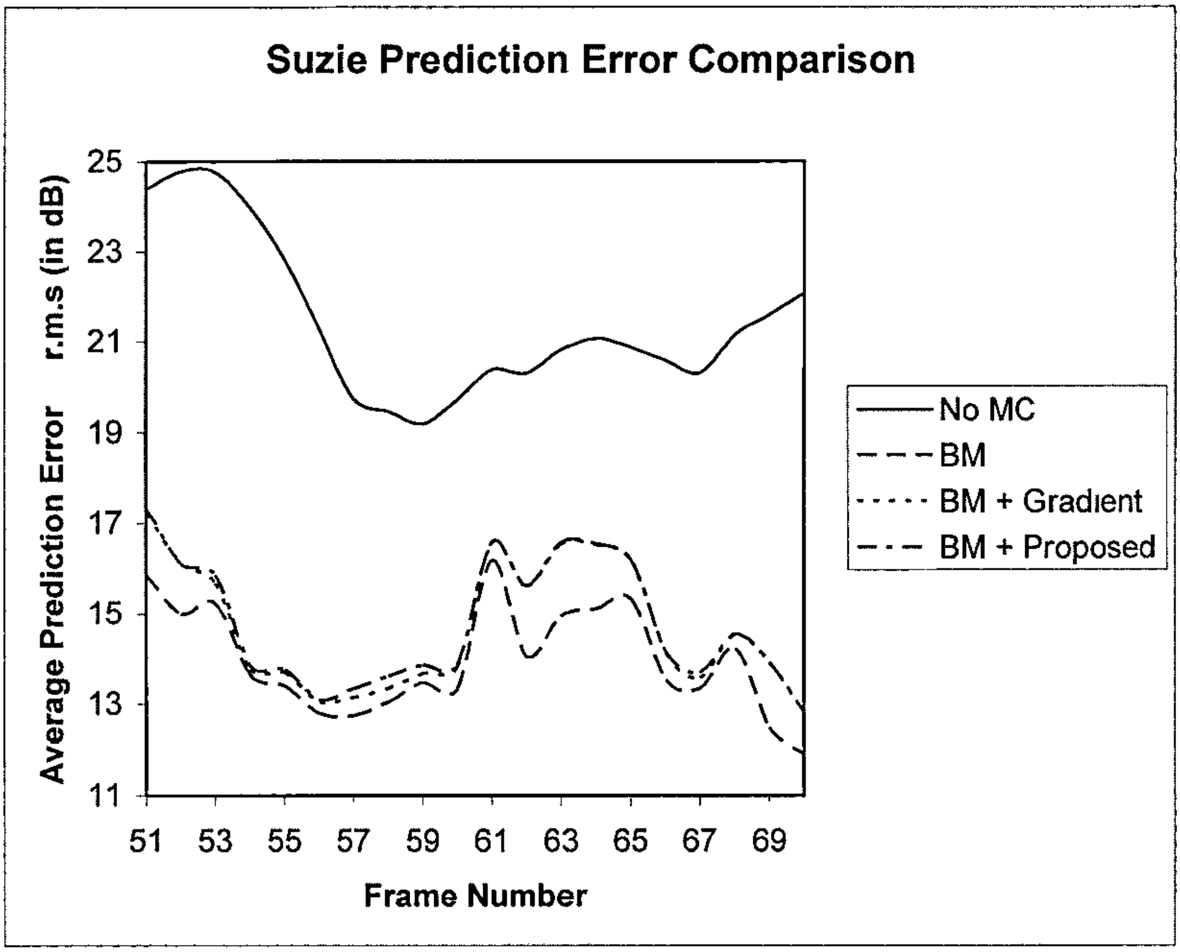
Basically the way the method is structured is as follow:-

1) applying block matching motion estimation on two successive image frames, and producing displacement vectors (i e.- for each block of 16 x 16 pels)
2) Passing the images through a low pass filter in order to have them down sampled, that is to shrink the images (i e:- by 16 x 16). Two dimensional Q.M F (quadrature Mirror Filter) can be used as a crude substitution for the low pass filter. A further rough substitution can be achieved by taking the intensity of the first DCT coefficient (DC coefficient) for each block (i.e - block of 16 x 16 pels); which is really the average intensity of pels in each block.
3) Allocating each block matching displacement vector as the motion vector for every pel of the down sampled images.
4) Apply the pel-recursive motion estimation algorithm on the down sampled (or shrinked) images by taking the motion vectors as the initial iterative estimation of pel-recursive estimation
5) The resulting motion vectors are to be the final tuning on block matching motion estimation displacement vectors

As for the experimental results concerned in this thesis; the above procedure is applied to images with block sizes 8 x 8 Figures 5.2.1 to 5.2.6 shows the graphical result for two sequences of "Suzie" and "Salesman". As it can be seen the outcome is not very promising.

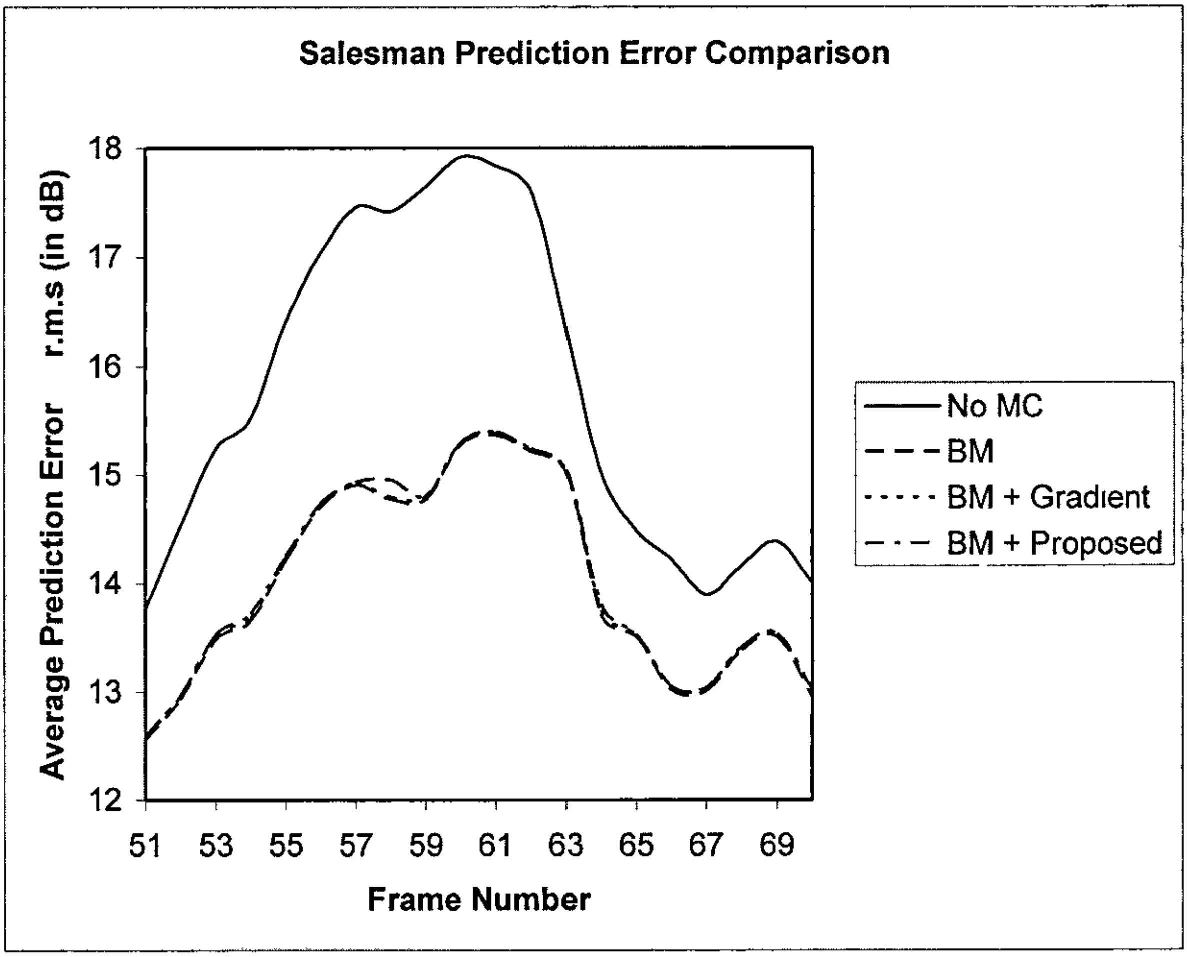
Finally the above procedure could also be carried out for any other block sizes i e a block of 4 x 4....Etc.

One of the drawbacks of the above method is that due to statistical randomness of sub sampled images, which causes an estimation of the error for each pixel, there is some possibility of uncontrolled overshoot as can be seen from the graphs in the figure 5 2.4. This is mainly due to the situation that initial motion vector is independently estimated for every pels.



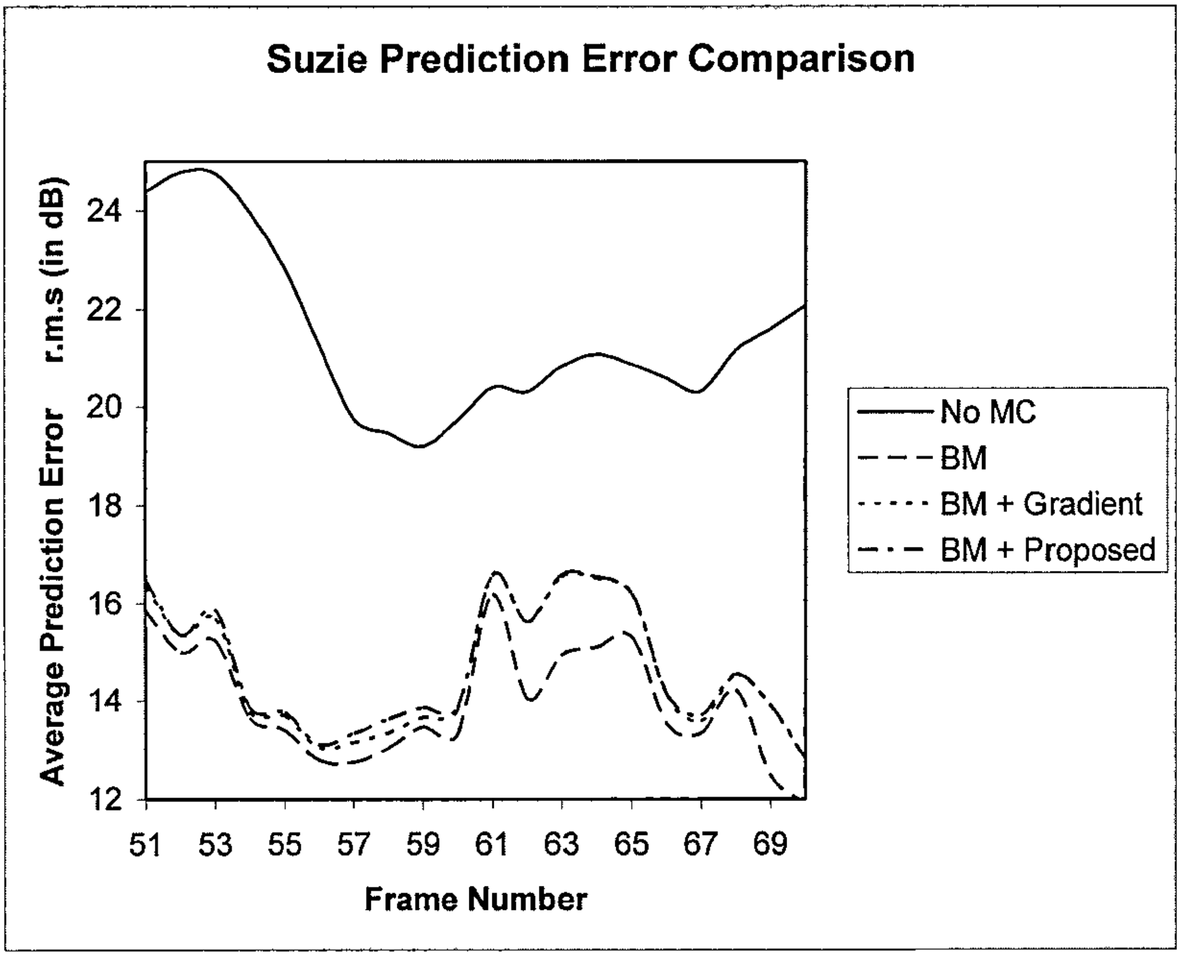
a) No frame skip comparison

Figure 5.2.1 Suzie block recursive comparison after three iterations with previous frame clean, without half pel accuracy on block matching.



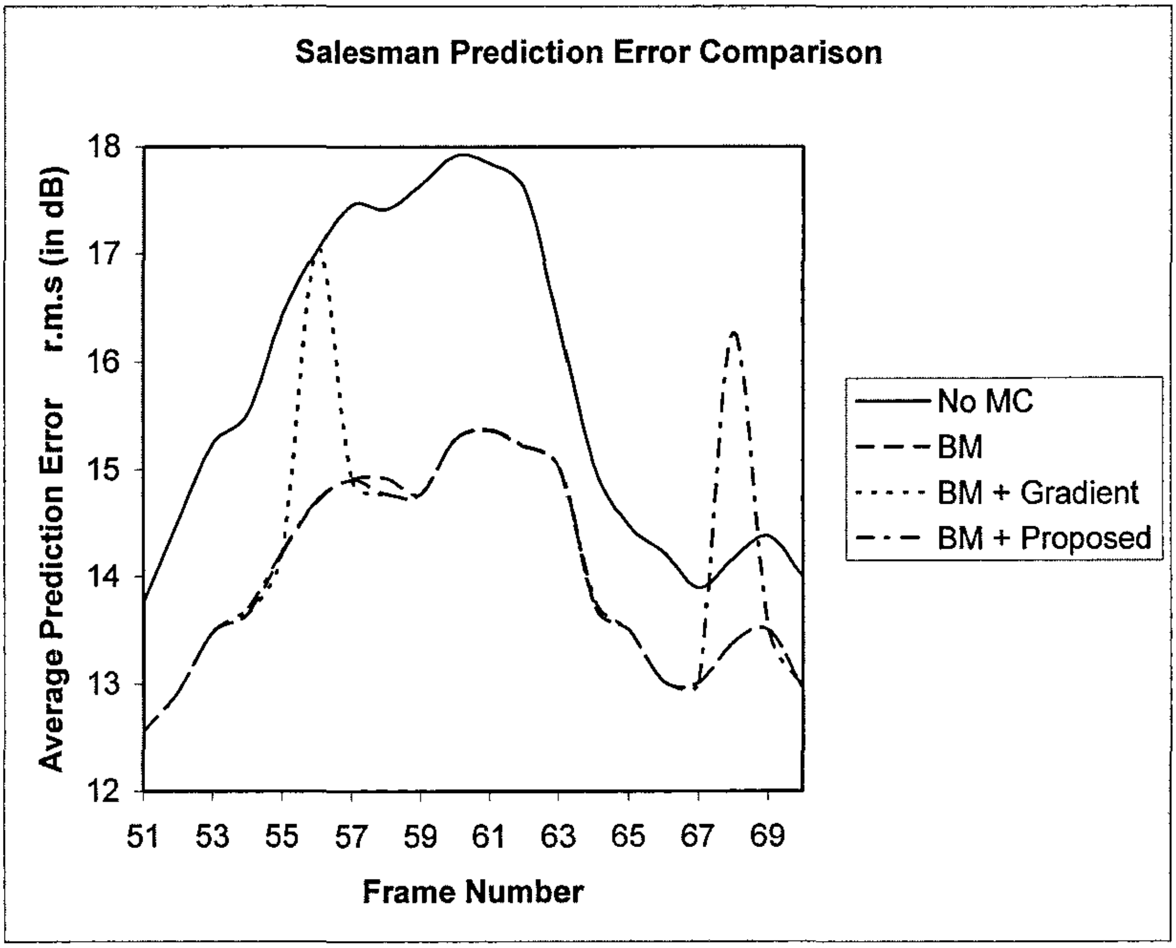
a) No frame skip comparison.

Figure 5.2.2 Salesman block recursive comparison after three iterations with previous frame clean, without half pel accuracy on block matching.



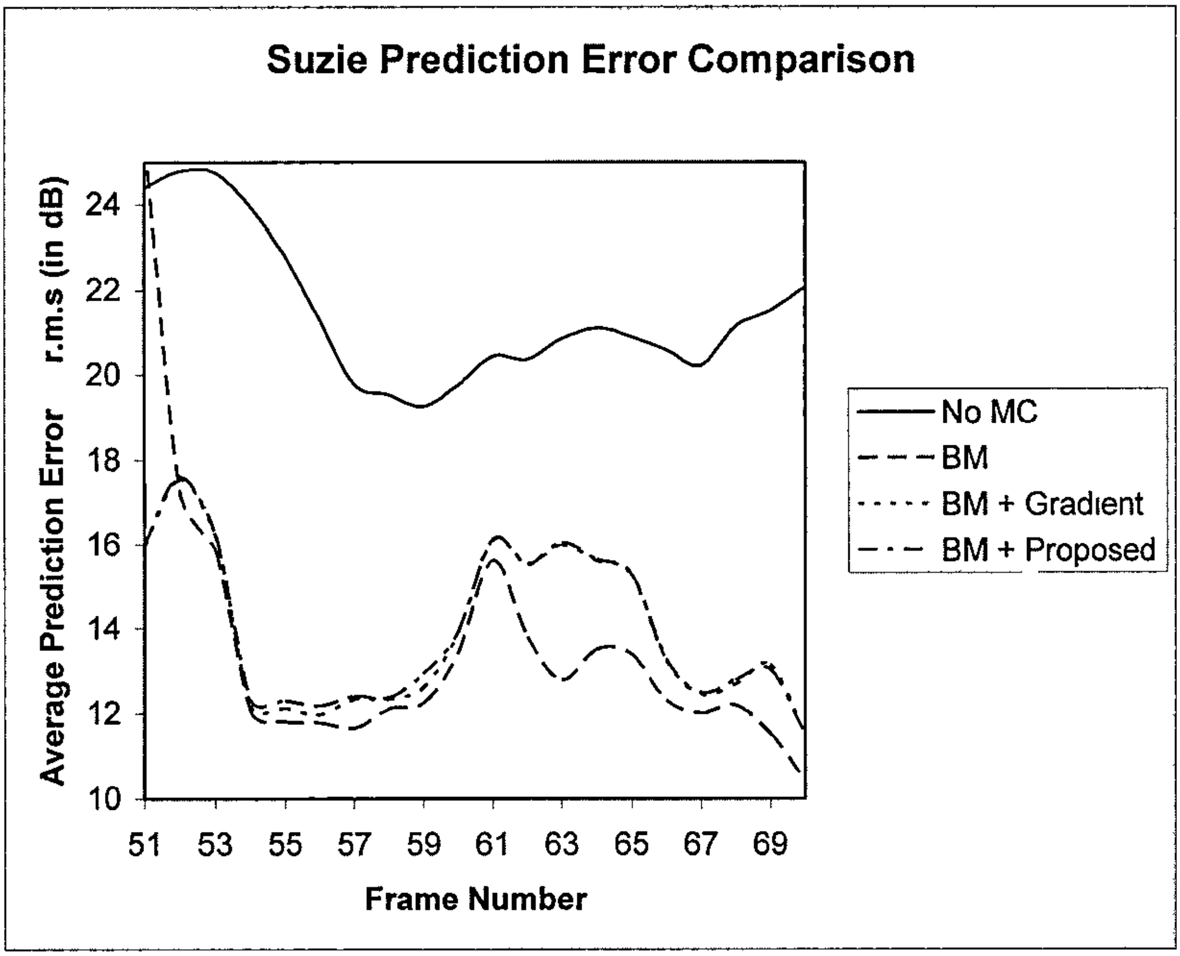
a) No frame skip comparison.

Figure 5.2.3 Suzie comparison after three iterations with previous frame reconstructed, with half pel accuracy on block matching, and initial motion vectors set to zero.



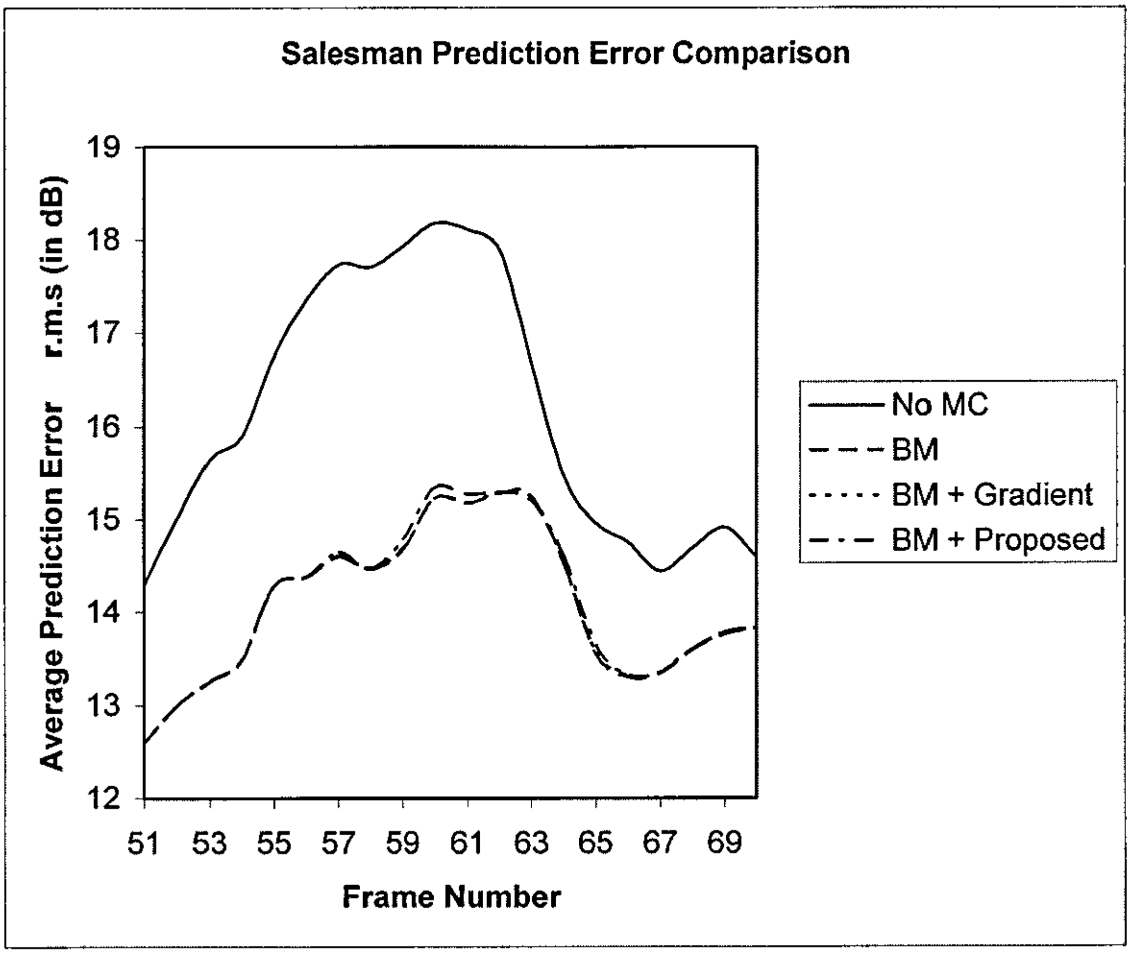
a) No frame skip comparison.

Figure 5.2.4 Salesman comparison after three iterations with previous frame reconstructed, with half pel accuracy on block matching, and initial motion vectors set to zero.



a) No frame skip comparison.

Figure 5.2.5 Suzie block recursive comparison after three iteration with previous frame reconstructed, with half pel accuracy on block matching.



a) No frame skip comparison.

Figure 5.2.6 Salesman block recursive comparison after three iterations with previous frame reconstructed, with half pel accuracy on block matching.

Chapter 6

Combined block matching and pel-recursive techniques

In the application of pel-recursive motion compensation, even the propsed pel-recursive as well as the modified pel-recursive steepest descent gradient did not show a promising performance when used as block recursive algorithm (refer to chapter five) In view of the situation that has arisen, it is a good idea to investigate the possibility of combining the two estimator techniques, pel-recursive and Block matching, in such a manner that block matching can assist the pel-recursive approch to form a Hybrid system. Here we have to investigate further the possibility of developing a hybrid system from block matching and pel-recursive systems

6.1 Local versus Global Minima

Steepest descent is probably the simplest numerical optimization method. It updates the present estimate of the location of the minimum in the direction of the negative gradient, called the steepest descent direction Recall that the gradient vector points in the direction of the maximum. That is, in one dimension (function of a single variable), its sign will be positive on an "uphill" slope Thus, the direction of steepest descent is in the opposite direction.

The descent gradient approach however suffers from a serious drawback·- the solution depends on the initial point. If we start in a "valley", it will be stuck at the bottom of that valley, even if it is a "local" minimum (Figure 6.1.1). Because the gradient vector is zero or nearly zero, at or around a local minimum, the updates become too small for the method to move out of a local minimum. One solution to this problem is to initialize the algorithm at several different starting points, and then pick the solution that gives the smallest value of the criterion function However, this method usually requires significantly more processing time

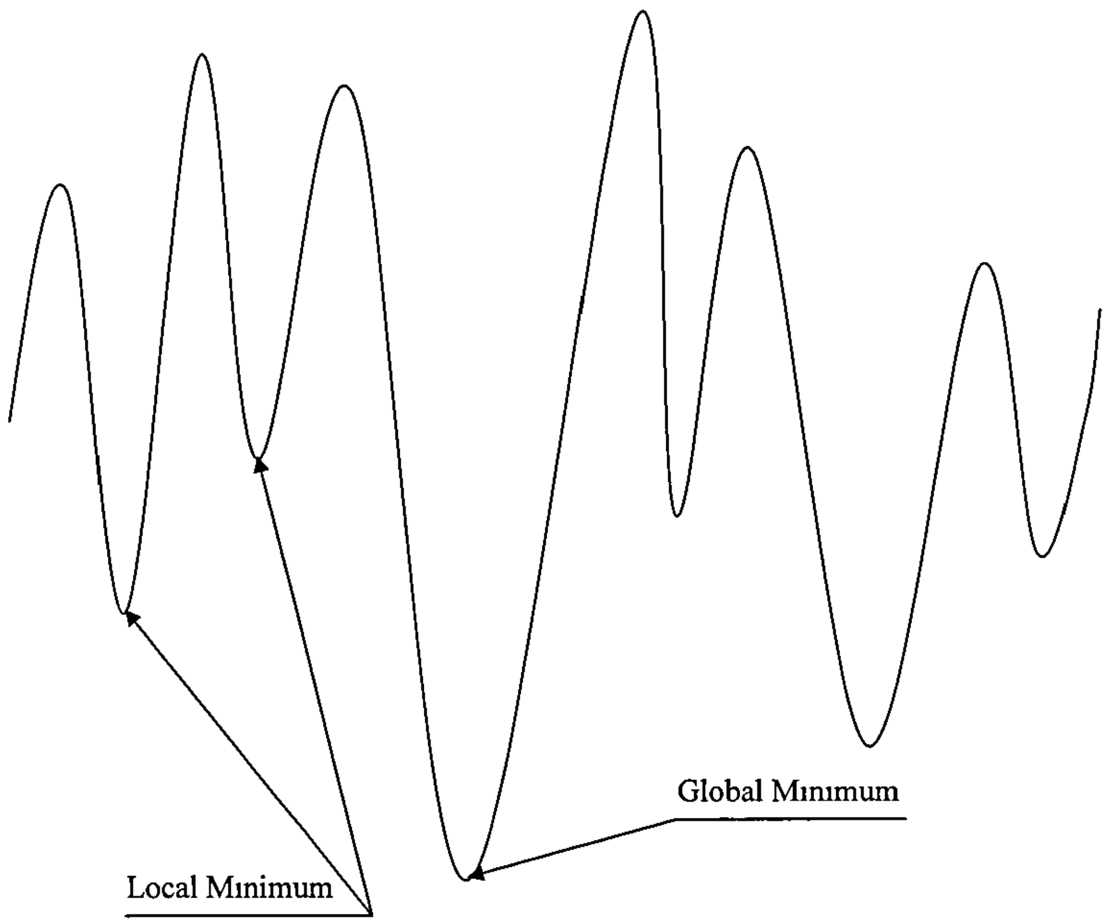


Figure 6.1.1 Demonstrative Graphical sketch of local and global minima.

6.2 Hybrid system

As has been seen from the papers [39] [110] [114] [111] and through the simulation (chapter 3) none of the algorithms by Netravali and Robbins [39], Newton-Raphson [110], Caffero and Rocca [114], or Bergmann [111] give full convergence for every pel to produce a perfect estimation for motion. In addition, some pels converge to unsatisfactory figures and sometimes become unstable leading to the conclusion that the algorithms suffer from some form of instability.

The aim of compression is based on the idea that it is possible to find displacement or motion vectors for each pel so as to have a minimum error image signal Going through an iterative process (i.e. steepest descend algorithm), it is not necessarily true that one can find an area of a global minimum, therefore we face a situation where one lands on a local minimum and perhaps ultimately gets to the actual local minimum or goes into oscillation and becomes unstable.

In spite of all the above, the algorithm by Netravali and Robbins [39] has shown convergence with less overshoot in relation to the other three algorithms [115], with the cost of a high number of iterational computations for estimation of the displacement vectors. If we define the stability constrain criteria as

$$| D - \hat{D}_{i-1} | \langle | D - \hat{D}_i | \qquad (6.1.1)$$

the algorithm by Netravali and Robbins shows better stability as it requires that the update vector be always directed towards and not opposite to the actual displacement

It has been seen that the initial estimation of displacement vectors has a great effect on determining final motion estimation by the iterative process of the steepest descent algorithm or the proposed algorithm. As can be seen from Figure 6.1.1, if the initial estimation of displacement vectors are not well chosen, when the steepest descent algorithm is applied, after a few iterations, one can have a situation where a local minimum is estimated instead of the global minimum estimation.

In view of the affirmation argument, in order to estimate the displacement vectors with more accuracy, virtually for every single pel and particularly the pels where their motion vector happened to be situated on local minimum instead of global minimum, will not be estimated correctly. So the prediction error associated with these pels will not be accurately estimated. Therefore, to overcome this inaccurate estimation it is possible to suggest that an easy and simple solution would be to chose the initial displacement vector by a different mechanism. Having chosen the right initial displacement vector, then the motion vector resulting from first stage can be feed back into the iterative processing of the pel-recursive system This led to the idea of the hybrid system. Relating the above technique to the problem in this thesis, block matching motion estimation is combined with pel-recursive motion estimation to form a hybrid system As for the experimental results, the block matching algorithm is applied to a sequence of a moving images, producing motion vectors for every block of the image and therefore a higher signal to noise ratio.

One of the drawbacks of motion estimation using block matching is that displacement is estimated as one estimation for each block, for example; a block of 16 by 16 pels. This should not necessarily apply to every pixel of the block as some pels may not be moving pels e g.·- blocks containing edges. This also may cause a blocking effect which is one of the drawbacks of the method used.

One needs to transmit a displacement estimation for each block as well as the number of blocks with no motion estimation. This causes more overhead to be transmitted resulting in transmission of a higher number of bits per second.

The Netravali algorithm and the modified algorithm were employed to investigate the advantage and disadvantage of the motion estimation by the pel recursive method. It has been seen that the Netravali algorithm itself suffers from some major defects e g :- lack of divergence and stability which manifests itself through certain pels.

The pel-recursive algorithms try to minimize the prediction error by locking into either a local or the global minimum. As the algorithm iteratively tries to force the error to a minimum value, which is determined by the original motion displacement estimation, one should note that the lack of convergence or stability caused by being in the vicinity of a wrong minimum may give rise to a local minimum instead of global minimum. This may be the main problem associated with pel-recursive algorithms in general.

Combining Block Matching motion estimation and pel recursive motion estimation in a complex manner has shown some improvement of the signal to noise ratio of Block Matching with no extra cost on the overhead, producing new publishable results which still can be improved further. This actually means that, the block matching does the main displacement estimation and the pel-recursive does the fine tuning on each pel.

One of the advantages of this method is that it does not require any extra overhead in transmission because it does not need to transmit any extra information for the motion estimated than is needed for the block matching technique.

In this thesis, for example, by employing H.263 and using block matching without 1/2 pel accuracy; the energy was measured to be 20.52dB for an image in a sequence And also employing H 263 and using block matching with 1/2 pel accuracy, the energy was reduced by a factor of 0.04dB to a figure of 20.48dB. This is also to justify the obvious which is, using block matching with 1/2 pel accuracy is more advance than without 1/2 pel accuracy. This is the one of the main advantages of H 263 over H 261 (H 261 does not have 1/2 pel accuracy feature)

Using block matching (by employing H 263 with 1/2 pel accuracy) and pel recursive motion estimation combined as a hybrid system reduces the energy of the error substantially. Employing the new pel recursive motion estimation would further reduce the energy of the error.

6.3 Implementation and Experimental results

In order to show the outcome resulting from the hybrid system, motion vectors were estimated using the standard traditional method of block matching H.263 with half pel accuracy was employed to generate these displacement vectors. Having estimated the motion vectors for every individual block (for example, block of 16 x 16 pel), they are assigned to be the initial estimation for the pel recursive motion estimation for final tuning of the estimations.

As for implementation of the simulation, great accuracy was needed when calculating components representing the pel recursive formula, such as line, element, displacement frame differences, and so on. For this, different ways and techniques can be utilized and examined; e.g . interpolated, not interpolated, averaged using different causal supports, and etc. It should be noted that not every single pel is to be motion compensated, therefore a masking mechanism needs to be used, e.g.: where frame difference, $|FDIF| \leq$ threshold; no prediction is needed (non-moving area).

In this thesis in order to be uniform throughout the algorithms implementation and simulation for calculation of line and element differences, non-interpolated averaged 3 by 2 causal support (Figure 6.3.1) is used. For displacement frame difference the interpolated pel value of previous frame is subtracted from the average pel intensities of X(3) and X(6) of present frame. As far as the proposed algorithm is concerned, different convergency parameters have been used; that is where the absolute value of an element or line difference is less than 11, the convergency parameter $\varepsilon = 0.8$, other wise $\varepsilon = 0.7$. Here one can have a good educated view that, this is a reasonable indication of improvement by attempting to have the convergency parameter adaptive.

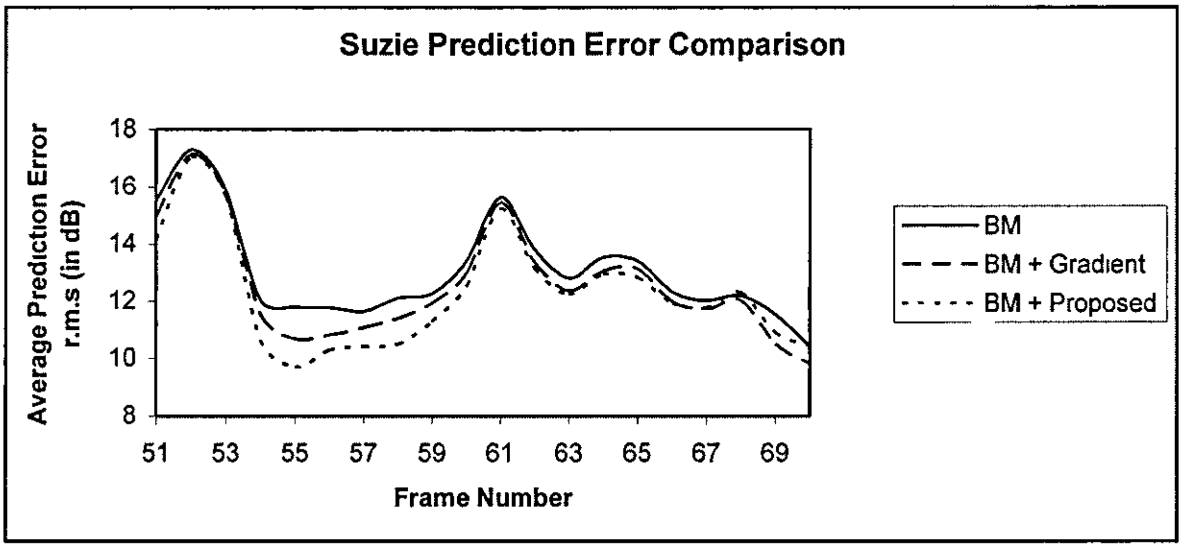
The maximum update limit for consistency purposes were chosen to be $|FDIF| \leq 9$ for non-moving areas. The results show that further constraint or limitation is needed to accomplish a better estimation of motion vectors. As an example, where motion vectors squared are less than or equal to 1, not to update the motion vectors. It should also be mentioned that more constraint or limitation or toggling of the predictor could result in better tuning of the motion vectors. This is done for three different sequences, Suzie, Salesman, and Car.

X(1) X(2) X(3) X(4)
X(5) X (6) O

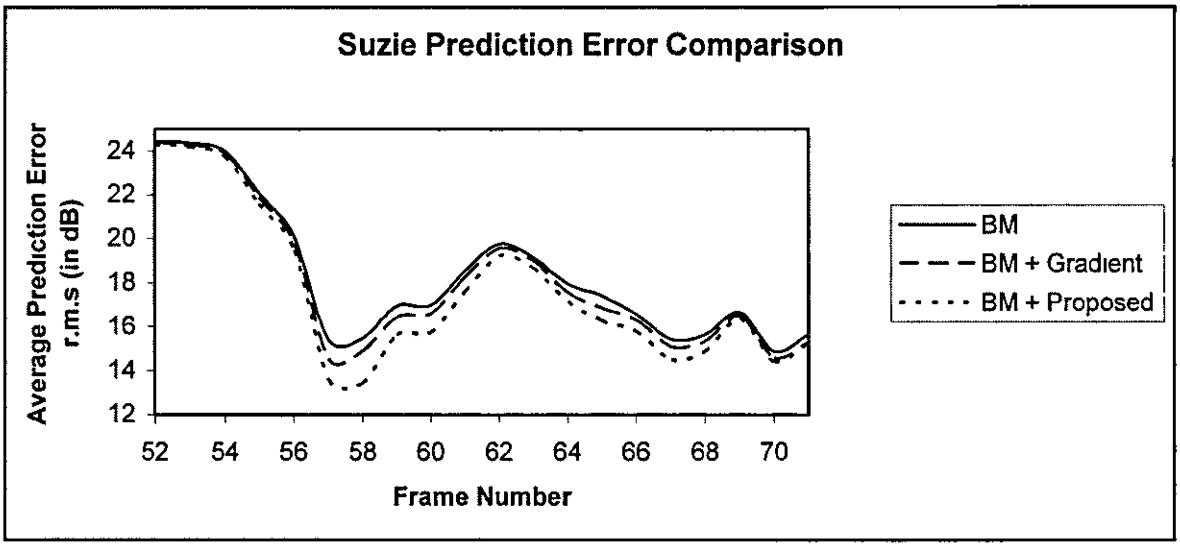
Figure 6.3.1 : A 4 by 2 causal support.

Figures 6.3.2 to 6.3.7 show and indicate the validity of the new pel-recursive motion estimation algorithm [112] in comparison with the existing pel-recursive motion estimation algorithm. The graphical results provide comparison for existing and the new algorithms.

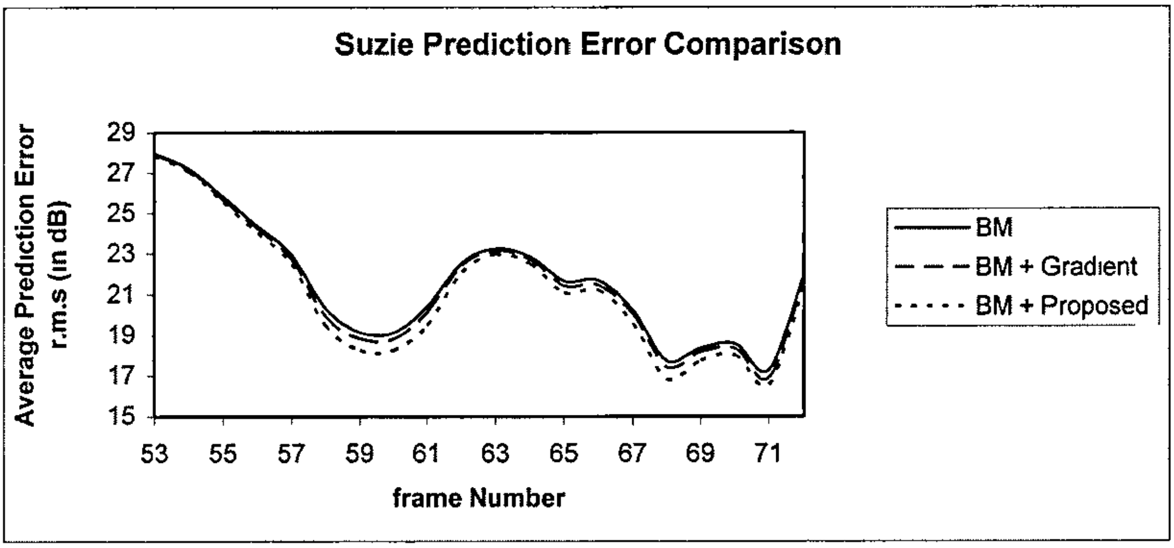
Consider the motion estimation using a pel recursive motion estimation. Experimentally it has been shown that it produces an average improvement of over 0.5dB in signal to noise ratio.



a) No frame skip comparison

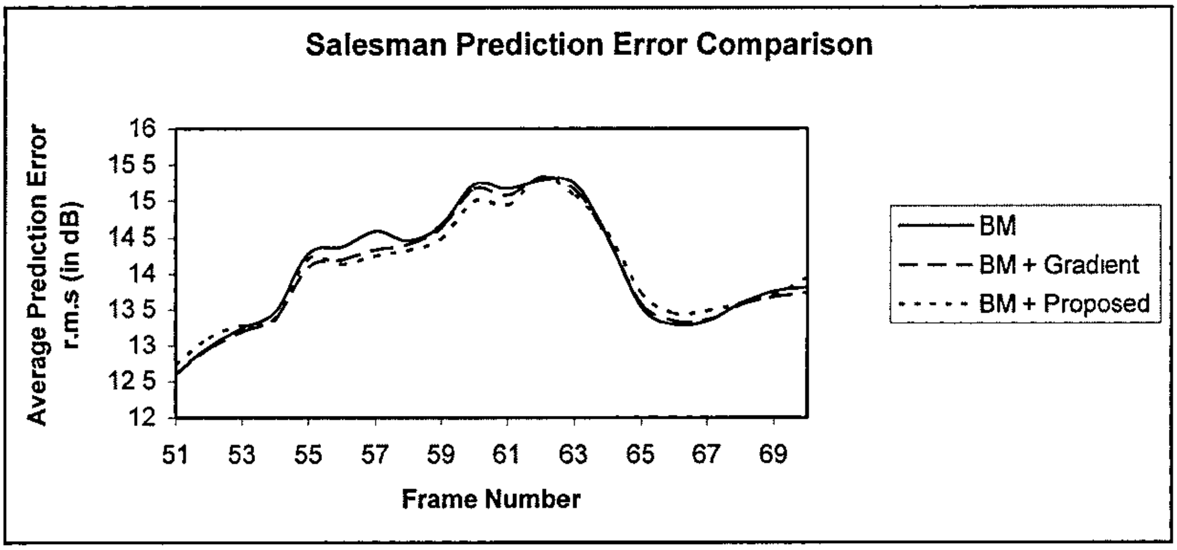


b) One frame skip comparison.

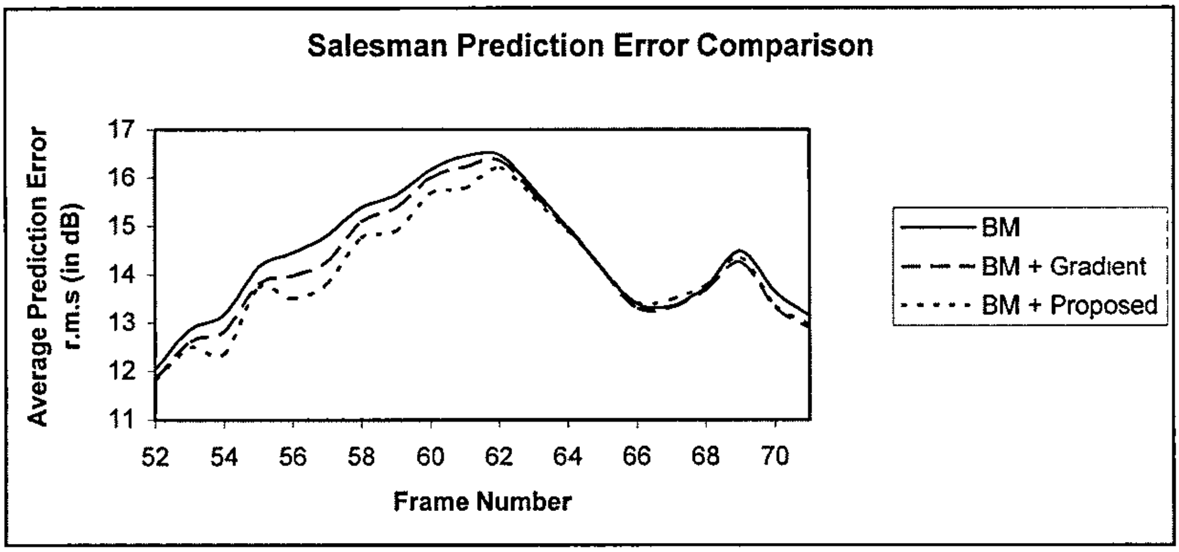


c) Two frame skip comparison.

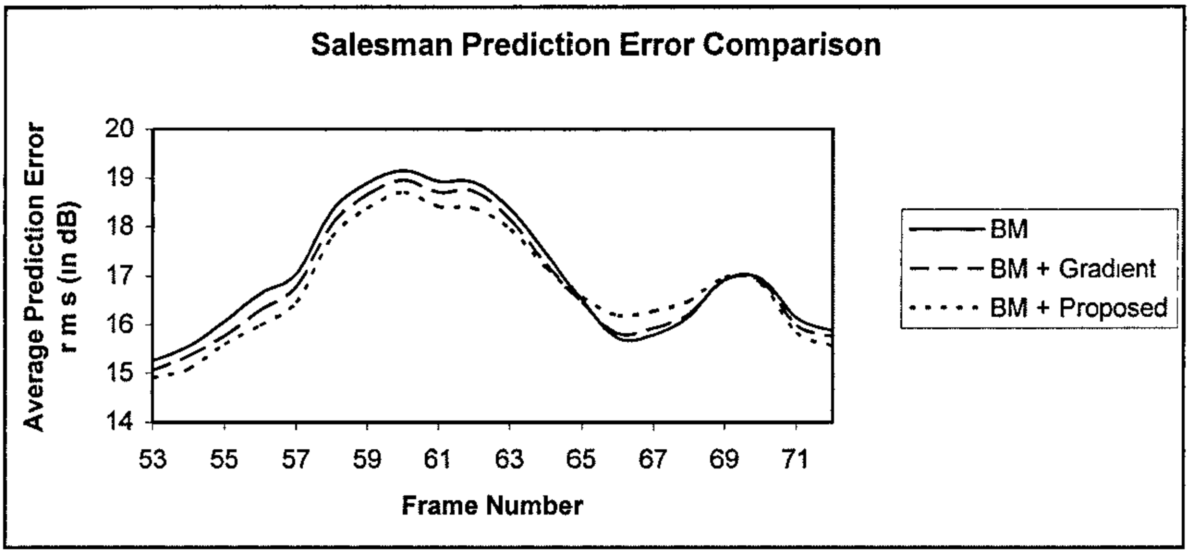
Figure 6.3.2 Suzie comparison after three iterations with the previous frame reconstructed with half pel accuracy on block matching.



a) No frame skip comparison

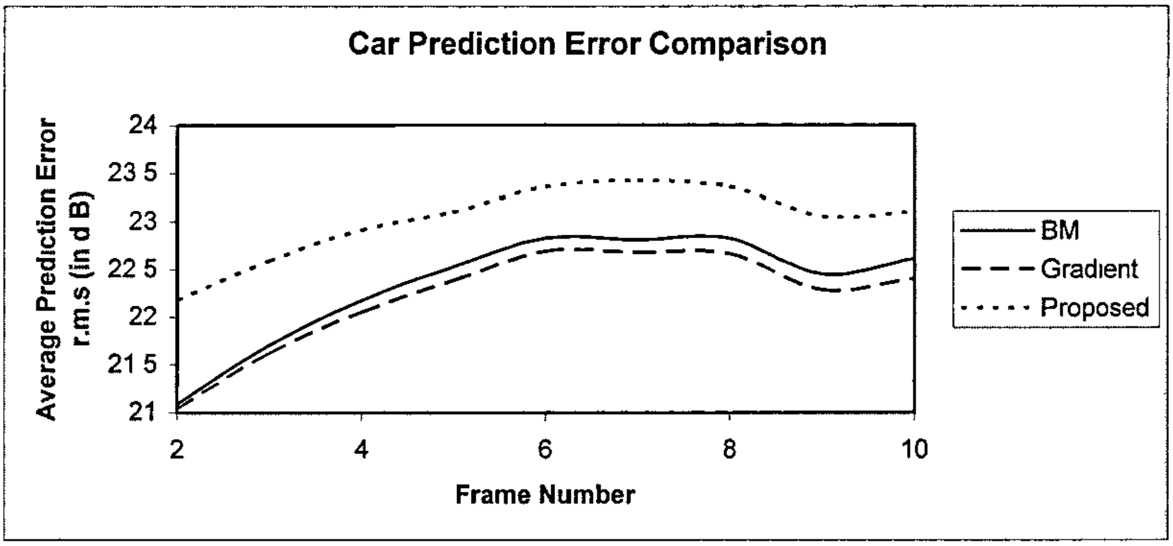


b) One frame skip comparison.



c) Two frame skip comparison

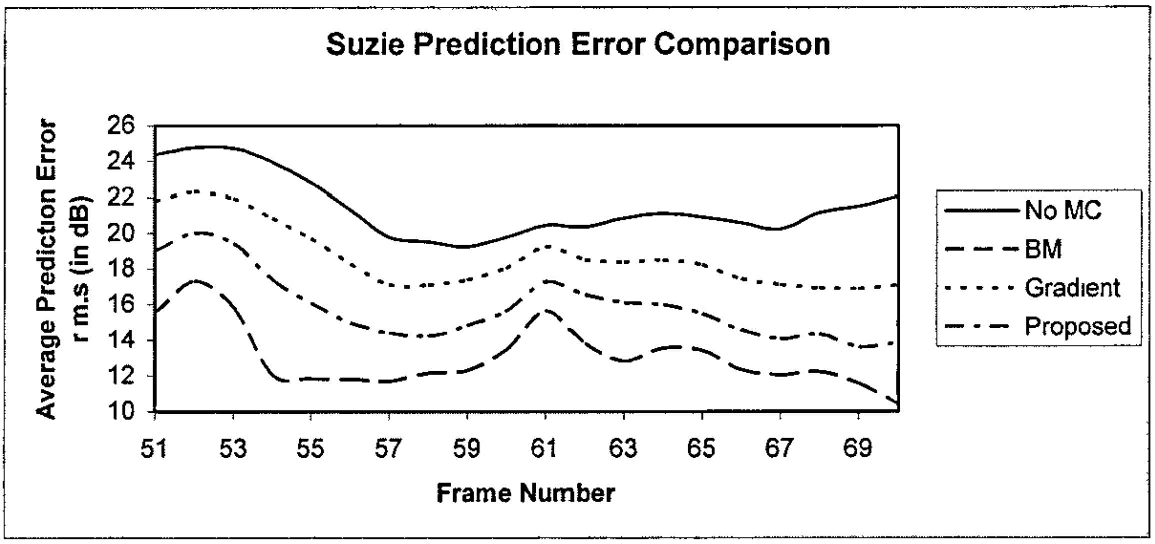
Figure 6.3.3 Salesman comparison after three iterations with the previous frame reconstructed, with half pel accuracy on block matching



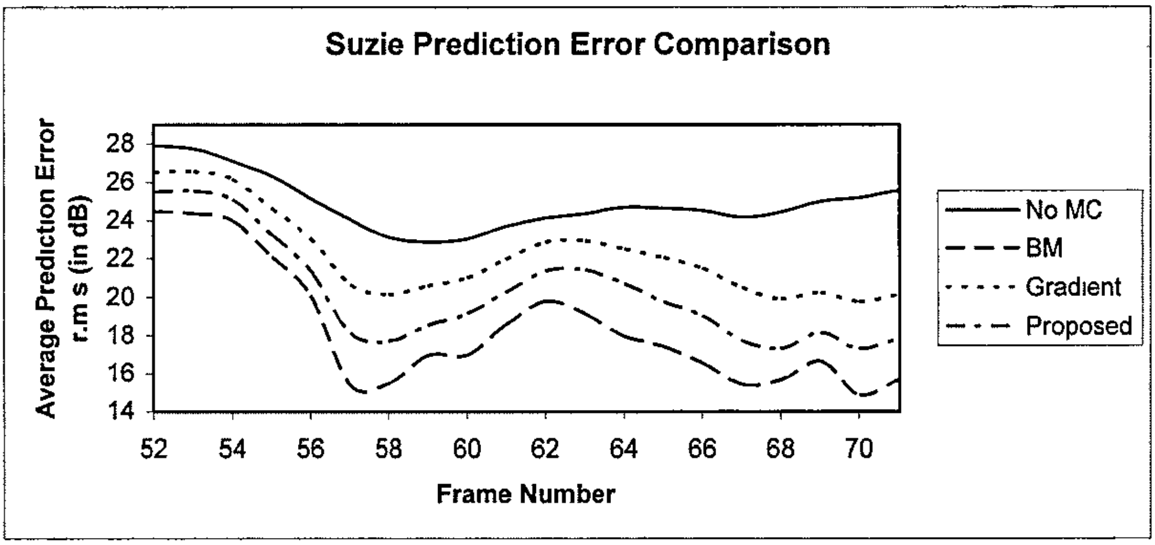
a) No frame skip comparison.

Figure 6.3.4 Car comparison after three iterations with the previous frame clean, without half pel accuracy on block matching.

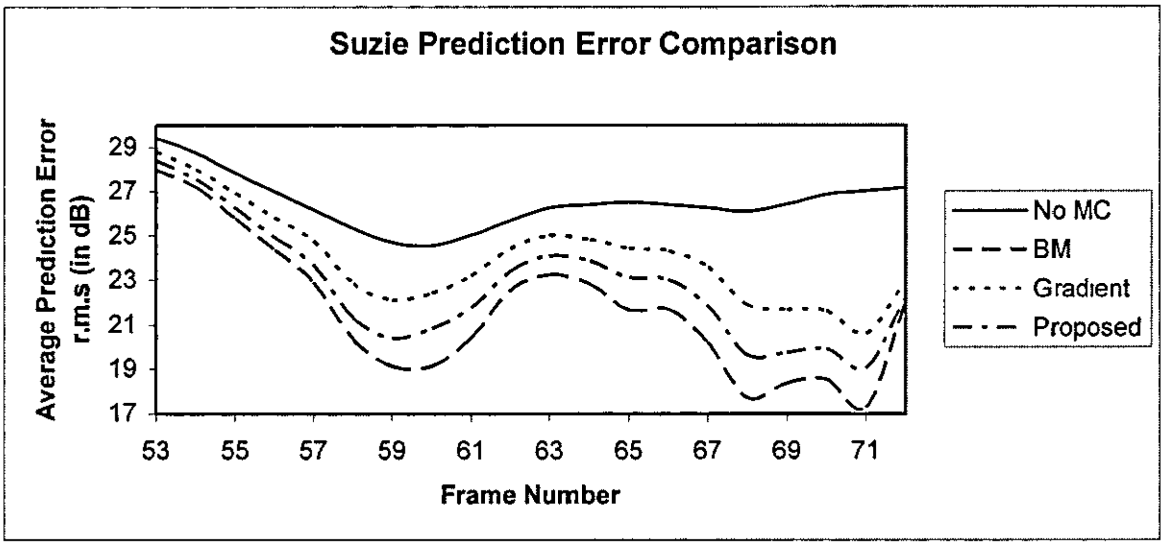
Setting the initial motion vectors to zero which almost in effect is turning off the block matching motion estimator will produce similar results to the one generated by the new proposed pel-recursive motion estimation. This is another justification of the obvious, that the proposed method of pel-recursive motion estimation in general has not being able to compete with block matching motion estimation as it can be seen from the Figures 6 3.5 - 6 3 7



a) No frame skip comparison

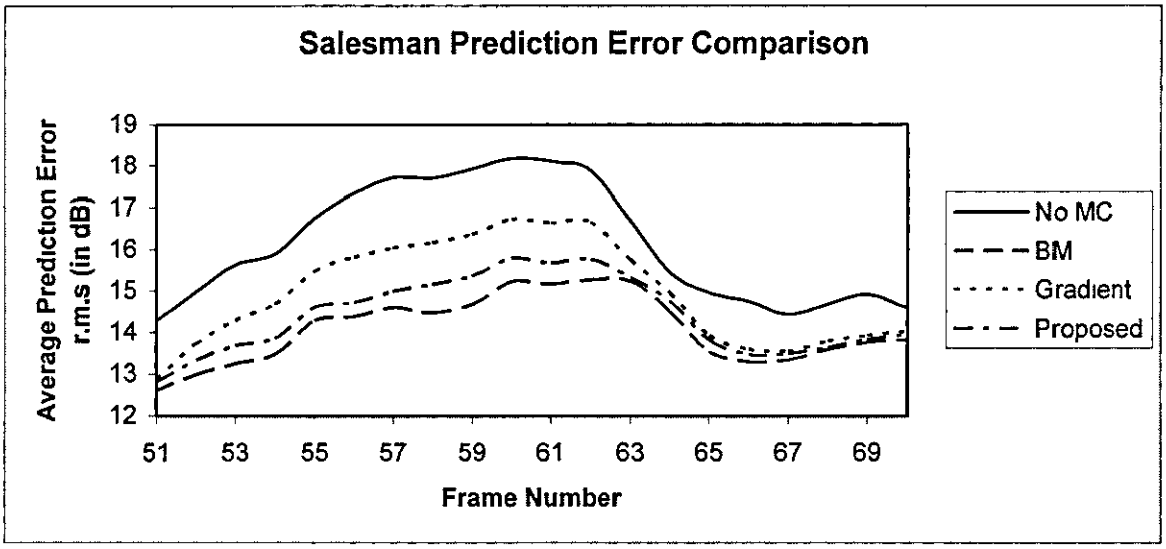


b) One frame skip comparison

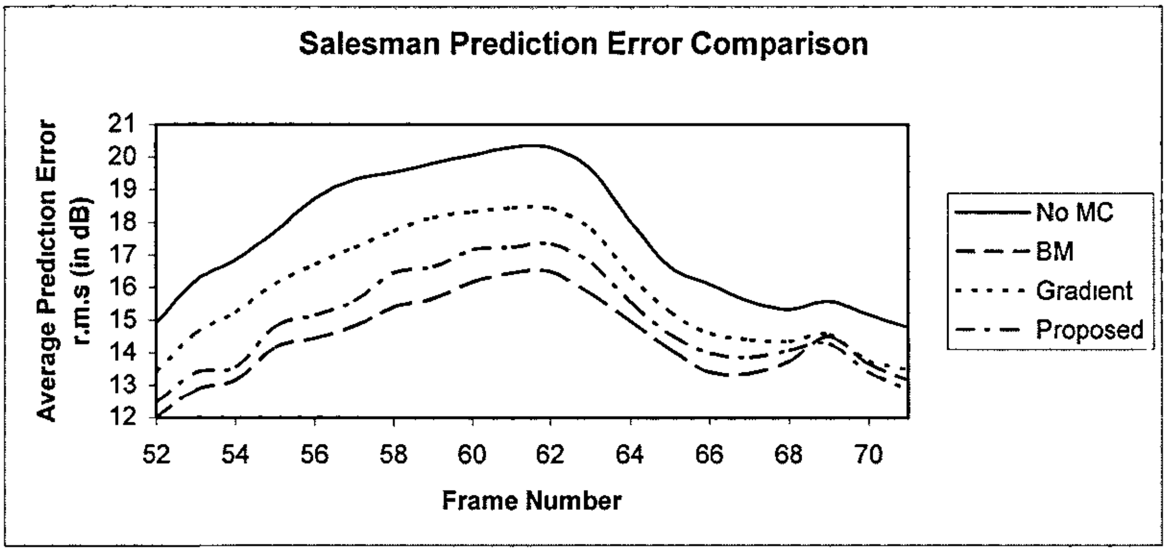


c) Two frame skip comparison.

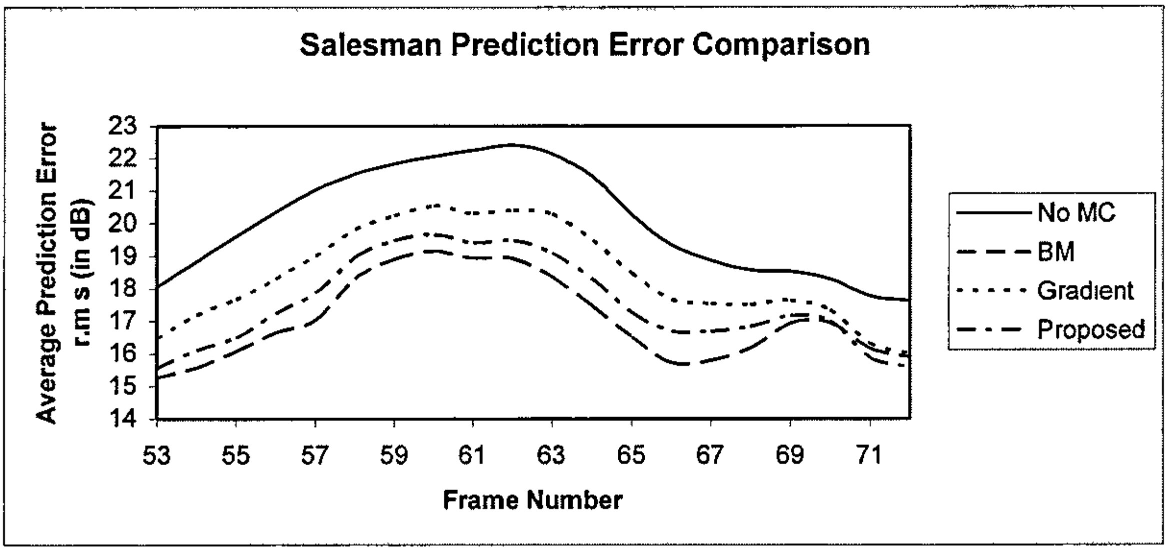
Figure 6.3.5 Suzie comparison after three iterations with the previous frame reconstructed, with half pel accuracy on block matching, and initial motion vectors set to zero



a) No frame skip comparison.

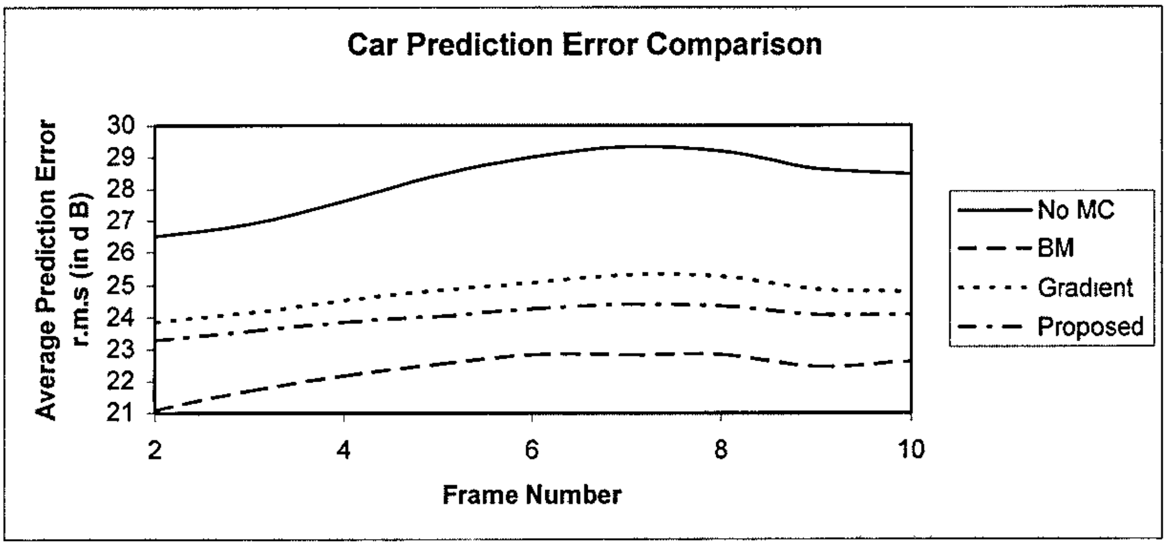


b) One frame skip comparison.



c) Two frame skip comparison.

Figure 6.3.6 Salesman comparison after three iterations with the previous frame reconstructed, with half pel accuracy on block matching, and initial motion vectors set to zero.



a) No frame skip comparison.

Figure 6.3.7 Car comparison after three iterations with the previous frame clean, without half pel accuracy on block matching, and initial motion vectors set to zero.

6.4 Conclusions

According to the results of Figures 6.3.2 and 6.3.3 whenever the hybrid system is used, a substantial improvement in compression is being achieved A greater improvement is shown through using the hybrid system with the proposed algorithm

.

Figure 6.3.4 indicate that for a fast moving image like the "Car" sequences (i e Figures 2 5 1 5 and 2.5.1 6) the hybrid system does show some improvement over the traditional block matching method.

If the initial value motion vectors are set to zero as obtained from the block matching part of the hybrid system (disregarding the effect of block matching from the system), then the situation of pel-recursive versus block matching will arise. Figures 6 3 5 and 6.3 6 show the result when there is no initial value estimator present.

Chapter 7

Conclusions and Further works

This chapter presents a general view of the results described in this thesis and summaries the contribution of new knowledge for the implementation of steepest gradient pel-recursive motion estimation. A further discussion is also developed based on an example in chapter 5, utilizing the new pel recursive algorithm to a certain degree. It goes further to discus the effect of involving block matching motion estimation in pel-recursive motion estimation to form a hybrid system

7.1 Conclusion

As an educated guess or rule of thumb one may suggest that the pel-recursive steepest decent gradient should perform better than block matching in estimating motion for sequences of moving images. The pel-recursive algorithms look at images pel by pel, where as block matching algorithms consider a block of an image as a whole However, experimental results have shown otherwise to the extent that the pel-recursive approach could not get be used in international standards for low bit transmission e.g.:- H 261, H 263, MPEGs and so on. Figures 2.4.1.2 and 2.4.1.3 and Figures 3.6.3 –3 6 8 and Figures 3 6.3 –3 6.8 and Figures 3.9.1 – 3.9 6 show a good indication of this.

Figures 2.4 1.2 and 2.4 1.3 in general depicted that in block matching motion estimation energy of the errors are very low, that is with very high signal to noise ratio, when compared with the situation in which there is no motion estimation present. These results have been generated using H 263 for different situations, estimating motion with and without half pel accuracy. In almost all codec standards, the motion estimator is designed on the basis of a block matching motion algorithm

In chapter 3 the state of the art of existing pel-recursive algorithms have been implemented and simulated for situations where the previous frame has been clean or a reconstructed image. Figures 3.6.3 – 3.6 8 depicted that pel-recursive motion estimation shows a good low value of average energy for the error image signal, but still energy of the error is higher than for block matching. Figures 3 9 1 – 3.9.6 indicate that even where there is some improvement by making ε, the convergency factor, adaptive, the error image signal is not low enough as far as block matching is concerned, even though the error signals of the images are lower than the case where ε is not considered to be adaptive.

Taking a step further, a new algorithm for pel-recursive motion estimation has been proposed, implemented and simulated as detailed in chapter 4. The graphical and pictorial results show and indicate that the average error image signal resulting from the new algorithm is much lower than that for the existing pel-recursive algorithm. Figures 4.2 1 –

4 2 7 compare the existing state of the art with the new algorithm for motion estimation These justify that the new algorithm is working and produces a better result than existing ones. However, the average error resulting from new algorithm on its own is not lower than that for the case of block matching.

Chapter 5 shows a crude example of employing the new pel-recursive algorithm. To see whether or not pel-recursive in general can contribute further improvement into international standards such as H.263 or any other hierarchical codecs, side by side of block matching motion compensation, the new gradient is blocked recursively are applied to sequences of images and detailed in chapter 5, resulting the graphs in Figures 5.2.1 –5.2.7. As can be seen the graphical results are not very promising in comparison to the situation if one would just use the block matching technique.

Considering the above case, it shows that it would not be feasible to have both the pel-recursive motion compensation and block matching motion compensation present as separate elements of estimator in codecs. Then, the thesis moves into new work by combining the two some what different algorithms of block matching and pel-recursive into a hybrid system. The resulting hybrid system does show the average error image signal levels to be lower, when compared with the old block matching algorithm by at least a 0 5 dB (for comparison, introducing half pel accuracy into block matching results in an 0.05 dB improvement over block matching on its own, for, the average error image signal) Figures 6.3.2 and 6.3.3 depict the variation of average error signals for two different sequences, justifying the improvements of the hybrid system over block matching or pel-recursive on their own.

It should be noted that the above conclusions in this thesis are based generally on experiments which were performed for about 20 frames of two different well known sequences of moving images ("Suzie" and "Salesman")

7.2 Suggestions for further research work and recommendations

Since block matching motion compensation has become standardized, research and work on pel recursive motion compensation has not been given any significance and has virtually stopped. This may be due to the better performance of block matching over pel-recursive for motion compensated image compression.

Now hybrid motion compensation can be employed for image compression, resulting in a more advanced performance than each of the two aforementioned motion compensation techniques, "block matching and proposed pel-recursive". This could open a new door for research and development in image compression areas investigating the usage of pel-recursive algorithms or as hybrid systems.

There are many papers relating to the development and further development into block matching techniques since the allocation of the standards. As a starting point a good example of the application suggestion is given in chapter five. In general the developments which were already applied to block matching could be applied to the hybrid system With reference to the these discussions, it is certain that there are many methods and developments, whether small or large, applicable for block matching motion compensation where pel- recursive and block matching motion compensation could work together A good example of this is the application suggestion given in chapter five. The block recursive motion compensation idea was developed with reference to a paper for block matching motion compensation [116].

In pel-recursive work, in general, the predictors used are based on intensities in the previous frame but not previous frames or calculated displaced previous frame. Of course, as for further work, other predictors can be employed in order to augment the prediction strategy. This in turn should enhance the system performance.

In chapter 3; it has been shown that pel-recursive motion estimation can improve its performance by making ε, the convergency factor, adaptive. In the original pel-recursive motion algorithm, ε was chosen to be a fixed variable and this has been improved to be fully adaptive as "modified pel-recursive motion algorithm". In chapter 4; ε, the rate convergency controller, is recommended to be a constant variable But as can be seen from chapter 4, a better result is produced by setting different values for different conditions. In turn this suggests that making ε adaptive should improve the performance of the proposed motion estimation algorithm. In view of this, it would be sensible to conduct further work toward improvement of the proposed algorithm

Finally, further work can be carried out in view of the example in chapter 5. This can basically be done to improve the performance of a hierarchical codec This is done firstly by applying block matching motion estimation on image frames, and obtaining displacement vectors (i e. - for each block of 16 x 16 pels). The images are then passed through a low pass filter in order to have them down sampled, that is to shrink the images (i.e.:- by 16 x 16 pels). Various methods can be employed for this, for example, Two dimensional Q.M.F (quadrature Mirror Filter) can be used as a crude substitution for the low pass filter. A further rough substitution can be achieved by taking the intensity of first DCT coefficient (DC coefficient) for each block (i e ·- block of 16 x 16 pels); which is really the average intensity of pels in each block. Taking the value of each block matching displacement vector as the motion vectors for every pel of the down sampled images. Apply the pel-recursive motion estimation algorithm on the down sampled (or shrunk) images by taking the motion vectors as the initial iterative estimation of the pel-recursive estimation (the current pels for estimating predictive frames is also to be used). This can be looked at as fine tuning on the block matching motion estimation displacement vectors. As for any transmission concern, there will be no further extra over head to be transmitted.

Appendices

Appendices

A.1 Hierarchy flow diagram of H.263

The Hierarchy c files flow diagram for H 263 are as shown in figure A.1.1

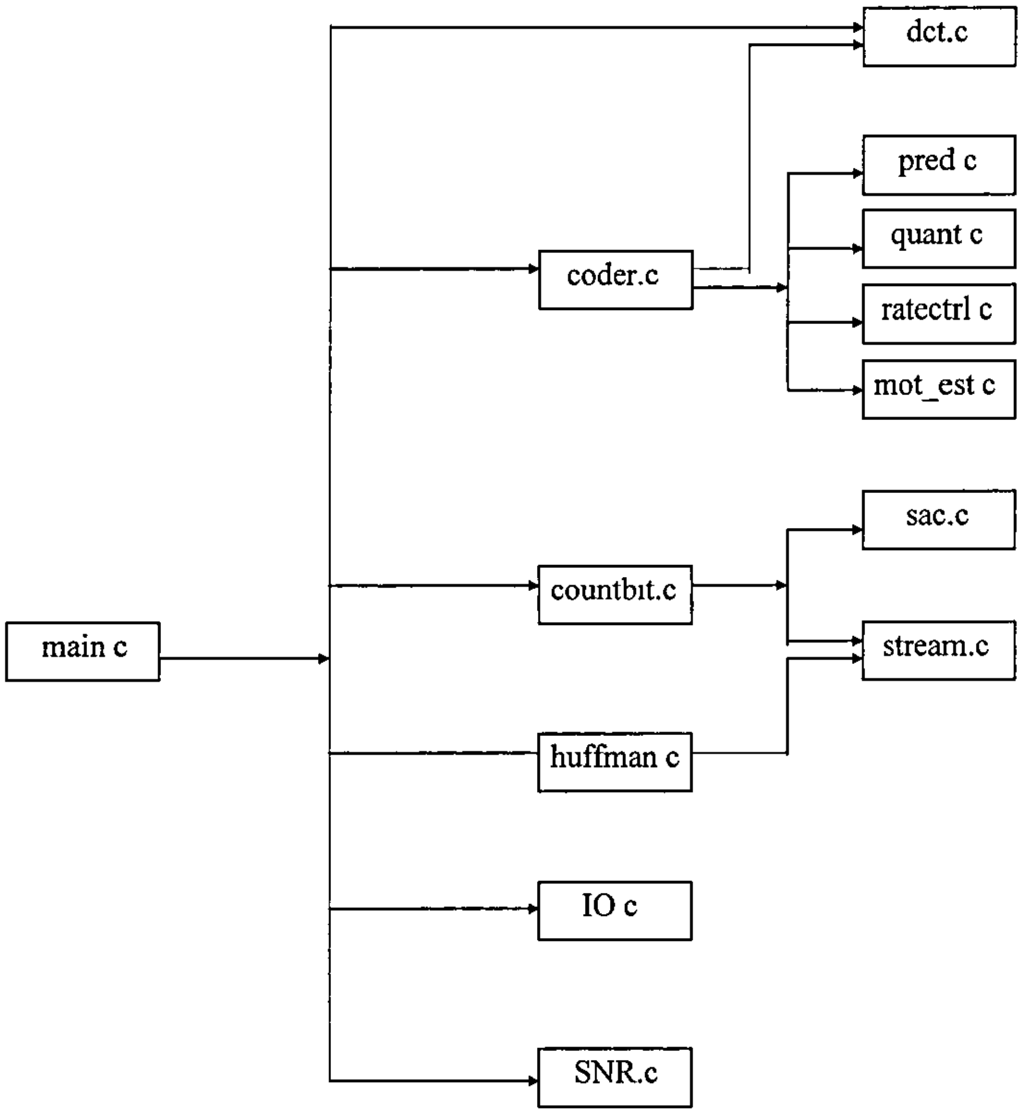


Figure A.1.1 Hierararchy flow diagram of H.263.

main c — The first routıne call It acts on the ınput command lıne (tmn) and set parameter accrdingly.

coder c — Performs all the encoding processes. Activated by main.c.

dct c — Performs the function of Discrete Cosine Transform. Initialized by main c and activated by coder c.

pred c — Relates prediction of PB frames and Advanced Prediction mode. Activated by coder.c

quant c	Sets all the quantisation index during encoding. Controled by coder c
ratectrl c	Organizes the control coding with control parameter as the quantisation index The control parameter is generic within ratectrl.c and doses not have any influence on quant.c.
mot_est c	Performs the motion estimation in the encoding process. Activated by coder c.
countbit c	its functionality is to count the bits during encoding process. activated by main c and coder.c.
sac.c	Performs the Syntax Based Arithmetic coding when this obtion is selected Activated by countbit.c.
stream c	Handels all of the bit level stream commands.
huffman c	Performs the function of huffman encoding routines. Activated by main c.
IO c	Contains the memory management for the component files. Activated by main.c.
snr c	Processes signal to noise ratio for every frame. Activated by main.c.

A.2 Programming function discription

The programming functions as they appear in H.263 software, are described as follow :

main.c

int Next TwoPB (--)	decides whether or not to code the next two images as PB.
void Help ()	help.
void AdvancedHelp ()	help.
void PrintResult (--)	prints results of bits in logn file.
void PrintSNR (--)	print snr of luminance and chrominance in log file.

coder.c

void CodeOneOrTwo (--)	
PictImage *CodeOneIntra (--)	codes one intra image.
int *MB_Encode (--)	performs dct and quantisation of macroblocks
int MB_Dncode (--)	reconstruction of quantised dct coded macroblocks.
void FillLumBlock (--)	fills the luminance of one block of PictImage.
void FillChromBlock (--)	fills the chrominance of one block of PictImage.
void ZeroMBlock (--)	fills one MB with zeros.
void ReconImage (--)	put together reconstructed image.
void MotionEstimatePicture (--)	find integer and half pel motion estimation.
void MakeEdgeImage (--)	copy edge pels for use with unrestricted motion vector mode.
void Clip (--)	clips reconstructed data 0-255.

dct.c

int Dct (--)	perform dct on an 8×8 block and zigzag scanning of coefficients
int idct (--)	descans zigzag scanning coefficients and perform inverse dct on 64 coefficients.
void init_idctref (--)	initiate the inverse dct reference
void idctref (--)	inverse dct reference.

pred.c

MB_Structure *Predict_P (--)	predict P macroblock in advance or normal mode
MB_Structure *Predict_B (--)	predict B macroblock in PB frame prediction
MB_Structure *MB_Recon_B (--)	reconstruct the B macroblock in PB frame prediction
void FindForwLumPredPB (--)	find the forward Luma prediction in PB frame.

void FindBiDirLumPredPB (--)	find the bi-dir Luma pred in PB frame
void FindBiDirChrPredPB (--)	find the bi-dir Chroma pred in PB frame
void FindBiDirLimits (--)	find the bi-dir limits
void FindBiDirChromaLimits (--)	find the bi-dir chroma limits.
void BiDirPredBlock (--)	find the bi-dir prediction block.
void DoPredChrom_P (--)	perform the chrominance pred for P frame.
void FindHalfPel (--)	find the optimum half pel prediction.
void FindPred (--)	find the prediction block.
void FindPred OBMC (--)	find the OBMC prediction block
MB_Structure *MB_Recon_P (--)	reconstruct MB after quantisation for P images.
void ReconLumBlock_P (--)	reconstruct one block of luminance data.
void ReconChromBlock_P (--)	reconstruct chrominance of one block in P frame.
void FindChromBlock_P (--)	find chrominance of one block in P frame.
int ChooseMode (--)	choose coding mode.
int ModifyMode (--)	modify coding mode.

quant.c

void Quant (--)	quantiser for SIM3.
void Dequant (--)	dequantiser for SIM3.

ratectrl.c

void InitializePictureRate (--)	compute the target bitrate and target frame rate for the current picture being coded
int UpdateQuantizer (--)	generate a new quantiser step size base on bits spent until current macroblock and bits spent from the previous picture.
int UpdatePictureRate (--)	updates buffer content and determine frame skip.

mot_est.c

void MotionEstimation (--)	estimate all motion vector for one MB.
unsigned char *LoadArea (--)	fill array with a square of image data.
int SAD_Macroblock (--)	fast way to find the SAD of one vector.
int SAD_Block (--)	fast way to find the SAD of one vector.
int SAD_MB_integer (--)	fast way to find the SAD of one vector.
void FindMB (--)	pick out one field of one MB.

countbit.c

void CountBitsMB (--)	count bits use for MB informatiom.
void Count_sac_BitsMB (--)	count bits use for MB informatiom using sac models modified from CountBitsMB.
int CountBitsSlice (--)	count bits use for slice (GOB) informatiom
void CountBitsCoeff (--)	count bits use for coefficients
void Count_Sac_BitsCoeff (--)	count bits use for sac models.
int CodeTcoef (--)	encode an AC coefficient using the relevant sac model
int FindCBP (--)	find the CBP for a macroblock.
int CountBitsPicture (--)	count the number of bits needed for picture header.

sac.c

int AR_Encode (--) arithmetic	encode a symbol using syntax based coding
int encoder_flush (--)	completes arithmetic coding before stream, or before any fixed length code are transmitted.
void bit_in_psc_Layer (--)	inserts a bit into output bitstream and avoid picture start code emulation by stuffing a one bit
int indexfn (--)	index into frequency cumutative frequency tables or escape code.

stream.c

void mwopen (--)	opens a bit stream for writing.
void mwclose (--)	close the write bitstream and flushes the remaining byte with "1", consistent with -1 returned on EOF.
int Zeroflush (--)	flushes out the rest of the byte with zeros and return number of bits written to bitstream (kol)
void mputv (--)	put a n bits to the stream from byte b
long mwtell (--)	return the position in bits of the write stream.
void mwseek (--)	seek to a specific bit position on the write stream.

huffman.c

void InitHuff (--)	initialized VLC tables.
void FreeHuff (--)	free the VLC tables.
void EHUFF *MakeEhuff (--)	construct an encoder huffman with a designated table size. This table size n, is used for the lookup of huffman values and must represent the largest positive huffman value.
void LoadETable (--)	loads an array into an encoder table. The array is grouped into triplts and the first negative value signals the end of the table.
void printTable (--)	print out 256 elements in a nice byte ordered fashion.
int Encode (--)	encode a symbol according to a designated encoder huffman table out to the stream. It return the number of bits written to the stream and a zero on error.

IO.c

unsigned char *ReadImage (--)	reads one qcif image from disk
PictImage *FillImage (--)	fills Y,C_b and C_r of a PictImage struct.
void WriteImage (--)	write PictImage struct to disk
PictImage *InitImage (--)	allocates memory for structure of 4 2.0 image
void FreeImage (--)	free memory allocated.for structure of 4 2 0 image.
char *StripName (--)	remove character behind ".", and in front of (including) the last "/".

snr.c

void ComputeSNR (--)	compare two image files using SNR.

References

[1] L. Wilkens, and P. A. Wintz, "Data compression and image processing," IEEE Trans Inform. Theory, vol IT-17, pp. 180-198, Mar. 1971.

[2] Griffiths J M., *ISDN Explained - Worldwide Network and Applications Technology*, John Wiley, 1990.

[3] Sekuter R. and Blake R., *Perception*, 3^{rd} Edition, McGraw-Hill 1994, pp158-178.

[4] Thompson J , "European collaboration on Picture Coding Research for 2Mbits/s transmission", *IEEE Trans*, COM-29, no. 12, December 1981, pp 2003-4.

[5] ITU-T Recommendation H 261 *Video codec for audiovisual services at p x 64 kbits/s*, July 1990. [Formerly "CCITT Recommendation"]

[6] Carr M.D., "Video codec hardware to realise a new world standard", *British Telecom Technology Journal*, vol. 8, no. 3, July 1990, pp 28-35.

[7] A. Jain, "Image Data Compression: A review", *Proceedings of the IEEE*, vol. 69, no. 3, pp. 349-389, March 1981

[8] A Netravali, B. Haskell, *"Dgıtal Pıcture· Representatıon and Compressıon"*, Plenum Press, 1988

[9] P. Wıntz, "Transform Pıcture Codıng", *Proceedıngs of the IEEE*, vol. 60, no 7, July 1972.

[10] R. J. Clark, "*Transform Pıcture Codıng*", Academıc Press, 1985

[11] D. Huffman, "A Method for the Constructıon of Mınımum Redundancy Codes", *Proceedıngs of the I R.E.*, vol. 40(10), pp. 1098-1101, September 1952.

[12] I. Wıtten, R. Neal, and J. Cleary, "Arıthmetic Codıng for Data Compressıon", *Communıcatıons of the ACM,* vol 30(10), pp. 520-540, June 1987

[13] W. Pratt, "spatıal Transform Codıng of Colour Images", *IEEE Trans. On Communıcatıon Technology*,Vol. COM-19, December 1971.

[14] N. Ahmed, K. R. Rao, and R. Schultz, " A generalısed discrete transform", *IEEE proc* Vol 59, pp 1360-1362, 1971

[15] K. R. Rao and P. Yıp, "*Dıscrete Cosıne Transform Algorıthms, Advantages, Applıcatıon*", Academıc Press, Inc., 1990.

[16] W. A. Chen, C. Harrıson and S. C. Fralick, " A fast computatıonal Algorıthms for the Dıscrete Cosıne Transform", *IEEE Transactıons communıcatıon*, Vol. 25, No. 9, pp 1004-1011, September 1977

[17] B. Tseng, W. Mıller, "On Computing the Dıscrete Cosıne Transform", *IEEE Trans. On Computers*, Vol C-27, October 1978.

[18] Afael C Gonzalez, and Richard C.Woods, *"Dıgıtal Image Processıng"*, Addıson Wesley publıshıng company, 1992.

[19] M. Miyahara, K. Kotanl, "Block Dıstortıon ın Orthogonal transform Codıng Analysis, Mınimization and Dıstortion Measur", *IEEE Trans. On Com*, No.1, January 1985

[20] F Kamangar, K. Rao, "Fast Algorıthms for the 2-D Dıscrete Cosıne Transform", *IEEE trans On Com* Vol. COM-31, No. 9, September 1982.

[21] J. Makhoul, "A Fast Cosine Transform in One and Two Dımensıons", *IEEE Trans. On ASSP*, Vol. ASSP-25, No 1, February 1987.

[22] N. S. Jayant and P. Noll, "*Dıgıtal Codıng of Waveforms Prıncıples and Applıcatıons to Speech and Vıdeo*", Printice Hall, Inc., 1984

[23] A. Habıbi, "Hybrıd Codıng of Pıctorıal Data", *IEEE Trans. On Com.*, Vol. 29, No. 3, pp 313-332, 1990.

[24] J. A Roese, W. K. Pratt, and G. S. Robınson, "Interframe Cosıne Transform Image Codıng", *IEEE Transactıons on Communıcatıons*, Vol. COM-25, pp 1329-1339, November 1977.

[25] A K. Jain, "Partıal Differentıal Equatıons and Fınıte Dıfference Methods ın Image Processing, Part I :- Image Representatıon", *Journal of Optımızatıon Theory and Applıcatıons*, Vol 23, No. 1, pp 65-91, September 1977.

[26] F. deȷageı, "Delta modulatıon, A method ofPCM transmıssıon usıng a 7-unıt code", *Phılıps Res. Rep* , pp. 442-466, Dec. 1952.

[27] C C. Cutler, "Differential quantization of communication signals," *U S Patent 2 605 361*, July 1952.

[28] P Elias, "Predictive coding," *IRE Trans Inform Theory*, vol. IT-1, pp 16-33, Mar. 1955.

[29] A. Habibi, "Comparison of Nth-order DPCM encoder with linear transformations and block quantization techniques" *IEEE Trans Commun Technol*, vol. COM-19, pp. 948-956, Dec. 1971.

[30] R E Graham, "Predictive quantizing of television signals," *IRE Wescon Cony Rec.* Vol. 2, Pt. 4, pp 147-157, 1958.

[31] W. Zschunke, "DPCM picture coding with adaptive prediction," *IEEE Trans Commun* , Vol. COM-25, no. 11, pp. 1295-1302,Nov. 1977.

[32] I. Dukhovich and J B O'Neal, "A three-dimensional spatial nonlinear predictor for television," *IEEE Trans Commun.*, Vol COM-26, no. 5, pp 578-583, May 1978.

[33] A. N. Netravali and C. B. Rubinstein, "Luminance adaptive coding of the chrominance signals," *IEEE Trans Commun* , Vol. COM-27, pp 703-710, Apr 1979.

[34] H. Yasuda and H. Kawanishi, "Predictor adaptive DPCM," *in Proc SPIE Conf.* Applications of Digital Image Processing, Vol. 149, pp. 189-195, Aug. 1978.

[35] F. Rocca, "Television bandwidth compression utilizing frame-to-frame correlation and movement correlation," *in Symp. Return Bandwidth Compression (MIT)*, April 1969. New York: Gordon and Breach, 1972.

[36] B. G. Haskell and J. 0 Limb, "Predictive video encoding using measured subject velocity," *U S Patent 3 632 865*, Jan. 1972.

[37] J. 0. Limb and J. A. Murphy, "Estimating the velocity of moving images from television signals," *Computer Graphics and image Processing*, Vol. 4, 1975, pp. 311-327.

[38] C Cafforio and F. Rocca, "Methods for measuring small displacements of television images," *IEEE Trans Inform Theory*, Vol 11-22, No 5, pp. 573-579, Sept 1976

[39] A N Netravali and J. D. Robbins, "Motion compensated television coding Part I," *Bell Syst Tech J*, pp 631-670, Mar. 1979.

[40] J. D. Robbins and A. N. Netravali, "Interframe television coding using movement compensation," *in Proe Inter. Conf. Communications (Boston, MA)*, pp. 23 4 1-23 4 5, June 1979.

[41] J. A. Stuller, A. N. Netravali, and J. D. Robbins, "Interframe television coding using gain and movement compensation," *Bell Syst Tech J*, Sept 1980

[42] B. G. Haskell, "Frame replenishment coding of television," *Image Transmission Techniques*, W. K. Pratt, Ed. New York, Academic, 1979.

[43] B. G. Haskell, "Entropy measurements for nonadaptive and adaptive, frame-to-frame, linear predctive coding of video telephone signals", *Bell Syst Tech J*, vol. 54, No 6, pp. 1155-1174, Aug 1975.

[44] J. 0. Limb, R. F. W. Pease, and K. A. Walsh, "Combining intraframe and frame-to-frame coding for television", *Bell Syst. Tech J*, vol. 53, No. 6, pp. 1137-1173, Aug. 1974.

[45] A. N. Netravatı, "On quantızers for DPCM codıng of pıcture sıgnals, " *IEEE Trans. Inform. Theory*, Vol. IT-23, No. 3, pp. 360-370, May, 1977.

[46] J. 0. Lımb, and C. B. Rubınsteın, "On the desıgn of quantızers for DPCM coders: A functional relationship between visıbilıty, probabilıty, and maskıng," *IEEE Trans. Commun.*, Vol. COM-26, pp. 573-578, May 1978

[47] G. Held and T. R Marshall, "*Data Compressıon Technıques and Applıcatıons Hardware and Software Consıderatıons*", Second Edıtion, John Wıley & Sons Ltd , 1987.

[48] A. N. Netravatı and B G. Haskell, " Dıgital Pıctures Representation and Compressıon, and Standards", Second Edıtıon, Plenım Press, Inc., 1989.

[49] ITU-T Draft Recommendation H.263, "Vıdeo Coding for Low Bıtrate Communicatıon", October 1995.

[50] H Gharavı and M Mılls, " Blockmatching Motıon Estimatıon Algorıthms – New Results", *IEEE transactıons on Cırcuıts and Systems,* Vol. 37, No. 5, May 1990.

[51] ITU LBC-95, Study Group 15, Working Party 15/1, "Video Codec Test Model, TMN5", January 1995.

[52] ITU-T Draft Recommendation H.263, "Vıdeo Codıng for Low Bitrate Communication", October 1995.

[53] ITU-T Draft Recommendation H.263, Study Group XV Document, Q2/15, Expert's Group on Very Low Bıtrate Video Telephony, Leıdschendam, Aprıl 1995.

[54] M W. Whybray and W Ellis, "H.263 - Video Codıng Recommendatıon for PSTN Vıdeophone and Multimedıa", *Proc IEE Colloquıum on low bıt-rate ımage codıng*, Jun 1995, pp 6/1-6/9.

[55] G Paul Howard and Jeffrey Swtt Vitter, "Arıthmetic Codıng for Data Compressıon" ın *Proccedıngs or the IEEE*, June 1994.

[56] G. Held and T.R.Marshall, "*Data and Image Compressıon Tools and Technıques*", Fourth Editıon, John Wıley & Sons Ltd., 1996.

[57] M.T. Orchard and G J Sullivan, "Overlapped Block Motıon Compensatıon: An Estımatıon - Theoretıc Approach", *IEEE Trans. On Immage Processıng*, vol 3, no 5, Sep. 1994, pp 693-699.

[58] H Gharavı and Mıke Mılls, "Block Matching Motıon Estımatıon Algorithms – New Results" ın *IEEE Transactıons on Cırcuıts and Systems*, vol. 37, no. 5, May 1993.

[59] CCITT Draft Recommendatıon H.261, "Vıdeo Codec for Audıovısual Servıces atpx64 kbıts/s", Geneva, 1989.

[60] ITU-T Recommendation H.261, "Vıdeo Codec for Audıovisual Servıces at px64kbıts/s", March 1993.

[61] M. L. Lıou, "Vısual Telephony as an ISDN Applicatıon", *IEEE Communıcatıons Magazıne*, pp. 30-38, February 1990.

[62] M. L. Liou and H. Fujıwara, "VLSI Implemantatıon of a Low Bit-Rate Video Codec", *Proceedıngs of IEEE Internatıonal Symposıum on Cırcuıts and Systems*, Singapore, Vol. 1, pp. 180-183, June 1991.

[63] A C P Loui and M L. Liou, "High-Resolution Still-Image Transmission Basedon CCITT H.261 Codec", *IEEE Transactions on Circuits and Systems for Video Technology*, Vol. 3, No 2, pp 164-169, April 1993

[64] D G. Morrison, "Standardization by ISO/MPEG of Digital Video Coding for Storage Applications", *in Audiovisual Telecommunications, ed* N. D. Kenyon and C Nightingale, Chapman and Hall, 1992, pp 177-193.

[65] ISO/IEC IS 11 172-2, "Information Technology - Coding of Moving Pictures and Associated Audio for Digital Storage media at up to about 1 5 Mbits/s - Part 2Video", MPEG-1, 1993.

[66] A. C. Hung, "PVRG - MPEG CODEC 1.1", Portable Video Research Group(PVRG), Stanford University, November, 1993.

[67] W. Hodge, S. Mabon and J. T. Powers (Jr.), "Video on Demand· Architecture, Systems and Applications", *Journal of the Society of Motion Picture and Television Engineers*, vol. 102, no. 9, 1993, pp 791-803.

[68] ISO/IEC "Coding of moving pictures and audio for digital storage media at up to about 1 5 Mbits/s", *IS 11172-2.*

[69] ISO/IEC "Generic coding of moving pictures and associated audio", *IS 13818-2*

[70] ISO/IEC IS 13818-2, "Information Technology - Generic Coding of Moving Pictures and Associated Audio Information - Part 2: Video", MPEG-2, 1994

[71] ATSC, "Digital Television Standard for HDTV Transmission", Document A/53,1995.

[72] ATSC, "Guide to the use of the Dıgıtal Televısion Standard for HDTV Transmission", Document A/54, 1995.

[73] ISO/IEC JTC1/SC29/WG11 N0999, "MPEG-4 Testing and Evaluatıon Procedures Document", Tokyo, July 1995.

[74] E Dubos, B. Prasada, and M. S. Sabrı, "Image Sequence Coding", *ın Image Sequence Analysıs*, (T. S. Huang, Ed.), pp 229-287, Sprınger-Verlag, Berlin, 1981.

[75] J. O. Limb and J. A. Murphy, "Measurıng the Speed of moving Objects from Television Signals", *IEEE Trans on Communıcatıons*, Vol COM-23, No 4, pp. 474-478, Aprıl 1975.

[76] C Caffonıo and F. Rocca, "Tracking Moving Objects ın Televisıon Images", *Sıgnal Processıng*, Vol. 1, pp. 133-140, 1979.

[77] A. N Netravalı and J. D. Robbins, "Motion Compensated Codıng Some New Results", *Bell system technıcal journal*, Vol. 59, No. 9, pp. 1735-1745, Nov. 1980

[78] T. Ishiguro and K. Iinuma, "Televısıon Bandwidth Compressıon Transmıssıon by Motion Compensated Codıng", *Communıcatıons*, Vol 20, No 6, pp 24-30, Nov 1982

[79] R. Paquin and E. Dubois, "A Spatıo Temporal Gradıent Method for Estımatıng the Dısplacement Fıeld in Tıme Varyıng Imagery", *Computer Vısıon, Graphıcs, and Image Processıng*, Vol. 21, pp 205-221, 1983.

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

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

[82] J. A. Stuller, A. N. Netravalı, and J. D. Robbins, "Intertrame Televısion Codıng Usıng Gaın and Dısplacement Compensatıon", *BSTJ*, Vol. 59, No. 7, pp. 1227-1240, Sept. 1980.

[83] W. E. Snyder, S. A. Rajala and Hırzınger, "Image Modeling: The Contınuıty Assumption and Tracking", *Proc 1980 Int. Joınt Conf. Pattern Recognıtıon*, pp. 1111-1114, 1980.

[84] R. W. Hornbeck, "*Numerıcal Methods*", Quantam Publısher, New York, 1975.

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

[86] B. G. Schunck and B. K. P. Horn, "Constraınts on Optıcal Flow Computation", *Proe IEEE Computer Socıety Conf Pattern Recognıtıon and Image Proceasıng*, pp. 205-210, Aug. 1081.

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

[88] H -H. Nagel and W. Enkelmann, "Towards The Estimation of Dısplacement Vector Fıelds by 'Orıented Smoothness' Constraints," *Proe Int Conf on Pattern Recognıtıon*, pp. 6-8, Aug. 1984.

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

[90] J D Robbıns and A. N. Netravali, "Spatıal Subsamplıng in Motıon Compensated Televısion Coders", *BSTJ*, Vol. 61, No. 8, pp 1895-1910, Oct. 1982.

[91] K A Prabhu and A. N. Netravalı, "Motion Compensated Component Color Codıng", *IEEE Trans On Communıcatıons*, Vol. COM-30, No. 12, pp. 2519-2527, Dee.1982.

[92] K. A. Prabhu and A. N. Netravalı, "Motıon Compensated Composıte Color Codıng", *IEEE Trans on Communıcatıons*, Vol COM-31, No. 2, pp. 216-223, Feb.1983.

[93] Technıcal Descnption of NETEC-X1(MC) TV CODEC, DEX-5547, Issue 1, NEC Amenca, Inc., NEC Corporation, Tokyo, Japan, March 1983

[94] H H. Nagal, "Image Sequence Analysıs: What Can Learn from Applications?", *ın Image Sequence Analysıs*, (T. S. Huang, Ed.), pp. 19-228, Spnnger-Verlag, Berlin, 1981.

[95] T. S. Huang and R Y Tsai, "Image Sequence Analysıs: Motıon Estımation", *ın Image Sequence Analysıs*, (T. S. Huang, Ed.), pp. 1-18, Spnnger-Verlag, Berlın, 1981.

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

[97] B. G. Haskell and J. 0. Lımb, "Predıctıve Vıdeo Encodıng Usıng Subject Velocıty", *U S Patent 3,632,865*. Jan, 1972

[98] C Cafforio and F. Rocca, "The dıfferential method for motıon estimation", *ın Image Sequence processıng and Dynamıc Scene Analysıs (NATO ASI Serıes, Vol 12)*, T S. Huang, Ed. New York : Spnnger-Verlag, pp. 104-124, 1983

[99] C. Labıt and A. Benevıste, "Motıon estımatıon in televısıon pictures", *ın Image Sequence processıng and Dynamıc Scene Analysıs (NATO ASI Serıes, Vol. 12)*, T. S Huang, Ed. New York : Springer-Verlag, pp. 296-306, 1983.

[100] A. N. Netravalı, and J. O. Lımb, " Pıcture Coding : A revıew", *Proc IEEE,* Vol. 68, pp. 366-406, Mar 1980.

[101] J. Biemond, L. Looijenga, and D E Boekee, "A pel-recursıve Wıener based dısplacement estimatıon algorithm", *Sıgnal Processıng,* Vol. 13, No. 4, pp. 399-412, December 1987.

[102] L. Boroczky, J. N. Drıessen, and J. Biemond, "Adaptive algorithms for Pel-Recursıve Displacement Estımation", *Proceesıngs of the SPIE Conference on Vısual Communıcatıon and Image Processıngs '90,* Lausanne, Swıtzerland, 1-4, pp 1210-1221, Oct. 1990.

[103] J. Bıemond, J. N. Drıessen, A. M. Geurtz and D.E.Boekee, "A pel-recursive Wıener based algorithm for the sımultaneous estımatıon of rotation and translatıon", *Proceedıngs of the SPIE Conference on Vısual communıcatıons and Image Processıng,* Cambrıdge, MA, Vol. 1001, pp. 917-924. Nov. 1988.

[104] D. R. Walker and K. R. Rao, "Improved pel-recursıve motıon compensation", *IEEE Transactıons on communıcatıons,* Vol. COM-32, pp. 1128-1134, No. 10, October 1984.

[105] L. Boroczky, k. Fazekas, and T Szabados, "Theoretıcal and experımental analysıs of a pel-recursive Wiener based Motion Estimatıon algorıthm", *Annales des Telecommunıcatıon,* Vol. 45, No 9-10, pp 471-476, Sept /Oct. 1990.

[106] T. Szabados, L. Boroczky, and k. Fazekas, "analysis of a pel-recursıve Wıener based Motıon Estımation algorithm", Proceedıngs of URSI Int Symp. On Sıgnals, Systems and Electronıcs, Erlangen, Germany, 18-20, pp. 318-320 Sept 1989.

[107] Rin Chul Kım, and Sang Uk Lee, "AVLSI Archıtecture for a Pel Recursive Motion Estimatoın Algorıthm", IEEE Transactıons on Cırcuıts and System, Vol. 36, No. 10, pp 1291-1300, October 1989.

[108] Hans Georg Musmann, Peter Pirsch, and H. J. Grallert, "Advances in Pıcture Coding", Proceedings of the IEEE, Vol 73, No 4, pp 523-548, April 1985

[109] J. D. Robbins and A. N. Netravali, "Recursıve motion compensation : A revıew", ın Image Sequence Processing and Dynamic Scene Analysis, T. S Huang, Ed. Berlın, Germany : Sprınger-Verlag, 1983, pp. 76-103.

[110] H. C Bergmann, "Dısplacement estımation based on the corrolatıon of ımage segments", in IEEE Proc. Int. Conf On Electronic Image Processıng (York; England), pp. 215-219, July 1982.

[111] H. C. Bergmann, "Eın schnell konvergıerendos Dısplacement-Schatzverfahren fur dıe ınterpolatıon von Fernsehbild sequenzen", ph.D. dıssertatıon, tech. Unıv. of Hannover, Hannover, Germany, Feb. 1984.

[112] H. Gharavi, and H. Reza-Alıkhanı, " Pel-Recursıve Motion Estimation for video compression", *IEE, Electronıcs Letters,* Vol. 37, No. 21, pp. 1285 – 1286, October 2001.

[113] M. Bierling,"Dısplacement estımstıon by hıerarchıcal block matching", SPIE, pp 942-951, Vol 1001, Vısual Communıcatıons and Image Processing, 1988.

[114] C.Cafforio, and F Rocca, "The differential method for ımage motıon estımatıon", *NATO ASI serıes*, vol.F2, Image sequence processıng and dynamıc scene analysis edıted by T.S Huang, sprınger- verlag Berlin Heidelberg 1983.

[115] H.G Musmann, P Pırsch, and H J Grallert, "Advances in picture coding", *Proceedıngs of the IEEE*, vol.73, no.4, Aprıl 1985.

[116] M. Bierling, "Displacement estimation by hierarchical block matching", SPIE, pp. 942-951, Vol.1001, Visual Communications and Image Processing, 1988.

Bibliography

Masanao. Aoki, Introduction to Optimization Techniques, the Macmillan Company, New York

Michael P Ekstrom, Digital Image Processing Techniques, Academic prees,INC, 1984

Leon Cooper, and David Steinberg, Introduction to Methods of Optimization, W B Saunders company

N B Jones and J D Mck. Watson, Dgital Signal Processing· principles, devices, and application, IEE control Engineering series 42, Peter Peregrinus Ltd. 1990

Athanasios Papoulis, Probability, Random Variables, and Stochastic Processes, Second Edition, McGraw-Hill Company

N J Fliege, Multirate Digital Signal Processing, Multirate Systems, Filter Banks, Wavelets, John Wiley and Sons, 1994

Ali N Akansu and Mark J T Smith, Subband and Wavelet Transforms, Design and Applications, Kluwer Academic Publishers, 1996.

T. S Huang, Image Sequence Processing and Dynamic Scene Analysis, NATO ASI Series, Springer-Verlag, 1983

Ronald E Crochiere and Lawrence R. Rabiner, Multirate Digital Signal Processing, Printice-Hall Signal Processing Series, 1983

E Oran Brigham, The Fast Fourier Transform, Printice-Hall, Inc 1974

Ferrel G Stremier, Introduction to Communication Systems, Second Edition, Addison-Wesley publishing Company, 1982.

Jae S. Lim, Tow Dimensional Signal and Image Processing, Printice-Hall Signal Processing Series, 1990.

Emmanuel C Ifeachor and Barrie W. Jervis, Digital Signal Processing, A Practical Approach, Addison-Wesley Publishers Ltd, 1993

Dennis Bodson and Richard Schaphorst, Teleconferencing, IEEE Press, 1989