VLSI IMPLEMENTATION OF ADAPTIVE KALMAN FILTER FOR VIDEO DENOISING

Binsa Mathew[1], M Mathurakani[2]

[1]Asst.professor, Electronics and Communication Engineering, Holy Kings College of Engineering and Technology, Pampakuda, Muvattupuzha, Kerala, India
[2]Formerly Scientist in DRDO, Professor, Electronics and Communication, Toc H Institute of Science and Technology, Kerala, India

Abstract

Video and image noise filtering is needed in many applications like medical imaging. For real time denoising of video signal ,a 3D spatio-temporal filter is required. Such a filter is complex and cannot be used for real time implementation. A new processing algorithm using adaptive filter is discussed in this paper. Here, the video signals are segmented into independent image sequences in spatial domain. The pixels are processed in temporal domain which reduces the complexity of the design. For this algorithm, the video signal considered should be free of impulsive noises. An adaptive temporal Kalman filter is proposed which can be used for real time implementation for temporal processing of video signals. Adaptation is this algorithm is used for motion estimation of consecutive image sequences. The algorithm proposed is relatively simple, effective and efficient. It is implemented on Xilinx SPARTAN 3

Keywords: *Filter, AR model, Adaptive filtering*

---***---

1. INTRODUCTION

Image sequences usually get degraded due to the noise. So filtering technique should be used to denoise it, to get the original image. In the case of video, we need to carry out filtering, both spatially and temporally. All these are usually done with the assumption that image sequences and noise are stationary and noise can be estimated if signals are stationary. For this we need to process each frame independently by using an adaptive spatial filter and filter each pixel in time domain by using an adaptive temporal algorithm. As we are using real time processing and the fact that there is no linear phase restriction on temporal filters, we need to use recursive filters. Here, an adaptive Kalman filter is used to adjust its parameters based on the variations in the noise statistics and the detection of motions in the image sequences. Adaptation in time not only requires estimation of the noise statistics but also a means to control the level of filtering when there is a motion associated to the corresponding pixel in the image sequence.

The usual 3D filtering method is difficult to implement. A much simpler technique for its implementation is to take a pixel of an image sequence at a time and denoise it. Then, repeat the same process for all the pixels of the image sequence of the video. This technique reduces the complexity as the method of noise reduction reduces from 3D to a simple 1D.

The Kalman filter is a multiple-input, multiple-output digital filter that can optimally estimate, in real time, the states of a system based on its noisy outputs. The Kalman filter filters the noisy measurements to estimate the desired signals. The estimates are statistically optimal in the sense that they minimize the mean-square estimation error.

2.LITERATURE REVIEW

[1] describes the Kalman filter algorithm. As given in [2], Kalman filter is an optimal data processing algorithm. It combines all the available measurements of the data and prior knowledge of the system and devices to produce an estimate of the desired variables.

Comparison between Kalman filter and least square estimator is given in [3]. From this, we can see that Kalman filter is much easier to implement than LS algorithm. In Kalman filter algorithm, the estimate of the state is recomputed for every measurement with all the previous measurement data so as to minimize the estimation error.

Natural signals are usually represented by autoregressive (AR) models. [4] describes an adaptive system for AR signals noise enhancement, which comprise of adaptive Kalman filter and a subsystem for AR parameter estimation. Here, an estimation of Kalman filter parameters are used and is fed back between output and estimation parameters.

Usually 3D algorithms are used for removing noise from video. Standard optimum Kalman filter demands complete knowledge of the system parameters, input forcing functions and noise statistics. [5] presents a 3D spatio-temporal Kalman filter for video as an extension of 2D Kalman filter approach for images. The filtering is based on the assumption that, in a noise free video with little motion, each pair of consecutive frames has relatively high correlation in temporal direction but any observed noise will be uncorrelated.

[6] gives an idea of using fixed point arithmetic for Kalman filter parameter calculation.[7] describes methods to tackle the difficulties for implementing the algorithm in FPGA. Even though FPGA reduces the computation time compared to software counterpart, there are many difficulties in implementation. The major issue among them is limited memory resources in FPGA.

3.MULTIDIMENSIONAL KALMAN FILTER

For adaptive filtering, the Wiener filter and Kalman filter can be used to obtain an optimal finite impulse response (FIR) filter with minimum mean squared error. Kalman filter can be used to predict data with varying statistics Kalman filter being applied to a wide range of tracking and navigation problems. Defining the filter in terms of state space methods also simplifies the implementation of the filter in the discrete domain.

3.1.State Space Derivation

The Kalman filter estimates the process state at some time and then obtains feedback in the form of (noisy) measurements. The Kalman filter equations can be time update equations and measurement update equations. The time update equations are responsible for projecting forward the current state and error covariance estimates to obtain the a priori estimates for the next time step. The measurement update equations are responsible for the feedback—i.e. for incorporating a new measurement into the a priori estimate to obtain an improved a posteriori estimate.

3.1.1.Time Update Equations

$$X_{i+1} = \phi X_i + W_i \qquad (1)$$

Where; X_i is the state vector of the process at time i
ϕ is the state transition matrix of the process from the state at i to the state at i+1,and is assumed stationary over time
W_i is the associated white noise process with known covariance

$$P_{i+1}' = \phi P_i \phi^T + Q \qquad (2)$$

Where Q $= E[W_i W_i^T]$ is the process noise covariance

The time update equations above project the state and covariance estimates forward from time step i to step i+1

3.1.2. Measurement Update Equations.

$$K_i = P_i' H^T (HP_i' H^T + R)^{-1} \qquad (3)$$

$$\hat{X}_i = \hat{X}_i + K_i(Z_i - H\hat{X}_i) \qquad (4)$$

$$Pi = (I - K_i H)P_i' \qquad (5)$$

During measurement update, the Kalman Gain is first computed using eqn (3).The estimate of the state is obtained from eqn (4).The error covariance Pi is estimated by eqn (5)
After completing time and measurement updation once, the process is repeated with previous estimates to predict the new estimation values.This feature of kalman filter makes it recursive in nature and made it more preferred in practical implementations.

4.THE DESIGNED SYSTEM.

The adaptive filtering technique to be used should have a good performance and can lend itself for real time implementation. Kalman filter and Weiner filter have steady state solution if the noise and signal are stationary. But in this case ,we have time varying statistics for which kalman filter is the best choice. Another constrain is that the statistics are unknown. An adaptive filtering method is preferred as it adjusts the parameters dynamically. Adaptive kalman filter produces an optimum estimate of the unknown parameter .So it can be used for estimating the motion of signal and noise.

Let X(i) represents a pixel of the ith frame of grayscale video . Using first order AR modeling, let the ideal noise free pixel of the same frame be represented as S(i). AR modeling is used as it is a more realistic and simple model that is usually used to represent the temporal behavior of pixels in video signals. Under this assumption, the process and measurement equations are:

Using these assumption, process and measurement models are as follows;
1.PROCESS MODEL: The ideal pixel of (i+1)th frame is assumed as the sum of ideal pixel of ith and white independent zero mean Gaussian noise with variance σ^2_w

$$S(i+1) = aS(i)+W(i),$$

Here, a is a constant that depends on the signal statistics.
2. MEASUREMENT MODEL: The noisy pixel X(i) have an independent additive zero mean Gaussian white noise with variance of σ^2_v added to the ideal noise free pixel S(i).

$$X(i)= S(i) +V(i)$$

The filter output Y(i) is the estimate of the signal at ith time . σ^2 (i) represents variance of the estimation error and K(i) represent the kalman gain.

The algorithm for denising are given below

4.1 KALMAN FILTERING ALGORITHM

Initialize the values of output and variance.The initial value of variance can be σ_v^2

START: Update the values of Kalman gain, output and variance using the equations

$$K(i) = \frac{a^2\sigma^2(k(i\text{-}1)+\sigma_W^2}{a^2\sigma^2(i\text{-}1)+\sigma_W^2+\sigma_V^2} \qquad (6)$$

$$Y(i) = K(i).X(i)+a.[1\text{-}K(i)].y(i\text{-}1) \qquad (7)$$

$$\sigma^2(i) = a^2[1\text{-}K(i)]\,\sigma^2(i\text{-}1)+\sigma_w^2 \qquad (8)$$

Increment the value of i and execute the loop from START for the desired number of frames.

Here the parameters are σ_v^2 ,σ_w^2 and Y(i-1) are unknown.The parameters are estimated based on optimization of a criterion function like minimum mean square error (MMSE).The following instantaneous estimates can be used to achieve fast and simple implementation:

1. Parameter a , defined by E{X(i)X(i -l)}/E{X^2(i)} , can be estimated by using

$$\hat{a} = \frac{X(i)X(i-1)}{\frac{1}{2}[X^2(i)+X^2(i-1)]} \qquad (9)$$

The value of a is made constant to make the system stable.
2. Simple estimates of

$$\sigma_v^2 = E\{[X(i) - \hat{a}\,Y(i\text{-}1)]^2\} \qquad (10)$$

as well as

$$\sigma_w^2 = E\{[\ Y(i) - \hat{a}\,Y(i\text{-}1\,)]^2\ \} \qquad (11)$$

are calculated, in turn,

$$\widehat{\sigma_v^2} = [X(i)\text{-}\hat{a}Y(i\text{-}1)]^2 \qquad (12)$$

and,

$$\widehat{\sigma_W^2} = [Y(i)\text{-}\hat{a}Y(i\text{-}1)]^2$$

$$= K^2\,\widehat{\sigma_V^2} \qquad (13)$$

While proceeding with the algorithm, the Kalman gain gradually reduces to a small value from a large value. The lowest value of Kalman gain is when temporal signal is stationary. This implies that more filtering occurs as time advances. But for motion signal, the estimation values does not converge and results in less noise filtering.

To estimate the motion, the magnitude difference between the consecutive temporal samples is compared. The assumption made is that if this difference is greater than a threshold value, the motion exist. If σ_v^2 represents the variance of the a difference, then based on Gaussian noise distribution, it can be said that a motion is present if $\frac{|X(i)-aY(i-1)|}{\sigma_v} \geq \Gamma$.

If the test is positive, then the gain calculation in the Kalman filter can be reinitiated by assuming $\sigma^2(i) = \sigma_v^2$. The new gain value reduces the lagging effect and improves the noise filtering.

The adaptive algorithm is represented as follows:

4.2. ADAPTIVE ALGORITHM

The output pixel value is initialised to 0 , σ^2(i) and σ^2 to σ_v^2.

ADAPTIVE: increment the value of i from i=0 using the following equations

$$K= \frac{\sigma^2+\sigma_W^2}{\sigma^2+\sigma_W^2+\sigma_V^2} \qquad (14)$$

$$Y(i) = Y(i\text{-}1)+K[X(i)\text{-}Y(i\text{-}1)]; \qquad (15)$$

IF (Difference $=\frac{|X(i)-aY(i-1)|}{\sigma} \geq \Gamma$)

$$\sigma_w^{2=}\ \sigma_v^2; \qquad (16)$$

$$\sigma^2 = \sigma_v^2; \qquad (17)$$

ELSE

$$\sigma_w^2 = K\sigma_v^2; \qquad (18)$$

$$\sigma^2 \leftarrow (1\text{-}K)\,\sigma^2 + \sigma_w^2$$

END

Increment the value of i and start execution from ADAPTIVE to find the consecutive values.

END

Threshold Γ should be chosen so that the motion can be estimated.

Here, the video considered is assumed to be free of impulsive noise. If impulsive noise is present ,it should be removed by spatial filtering techniques with the help of spatial filters. A median filter can serve the purpose. In short, only spatially filtered noisy frames are given as the input to the adaptive kalman filter..

5. DESIGN CONSIDERATIONS

Certain constraints have to be considered for both simulation and implementation.

5.1. Representation Of Data

The data can be represented either as floating point numbers or as fixed point numbers.

Floating point is a numeral interpretation system in which the mathematical value of a string of digits uses some kind of explicit designation of where the radix point is to be placed relative to that string.

During algorithm development, floating-point numbers are often used because they represent infinite precision. When it comes time to realize the algorithm in hardware, floating-point numbers are not always practical. The solution is to convert very precise floating-point numbers to less precise fixed-point numbers.

Here, floating point representation is used for MATLAB simulation. In VERILOG HDL simulation, 10 bit binary digit precision is used which give a decimal precision of 4 digits, resolution of 0.001.

5.2. Component Usage

FPGA have dedicated adders and multipliers. But there are no dedicated dividers in FPGA platform. So in FPGAs, division is not synthesizable.

For dividing a number by power of 2 (e.g., 4,64,1024…) shifting is used. Shift by one unit to right is equivalent to division by 2.For dividing a number by 4, shift 2 times to right and so on

Division is frequently used in image processing algorithms, thus in order to further improve efficiency of FPGAs, it is reasonable to apply shift divide instead of arbitrary divide if small error is tolerated.

For this project only onchip memory is used. To save storage space , the pixels loaded to RAM at one time is restricted and the data is loaded as a stream from external environment .The data are not stored inside FPGA,It is send back to the computer for storage and further processing.

6.SIMULATION RESULTS

6.1. Matlab Simulation Results

The noisy input video from webcam was divided into different frames using MATLAB. Then one pixel was taken in time domain and fed as the input of the adaptive Kalman filter. The adaptive Kalman filter then denoise each pixel in time domain to get the denoised video.

For simulation, we need to generate a process model and measurement model. The process model represents the original pixel in time domain. This is generated by adding a gaussian noise of small variance (here, variance=1) to a pixel value (here, 100).

Measurement model represents the noisy pixel in time domain. It is generated by adding a gaussian noise of higher variance to the process model (here, variance=25). Its shown in fig 1.This is fed as the input to the adaptive Kalman filter

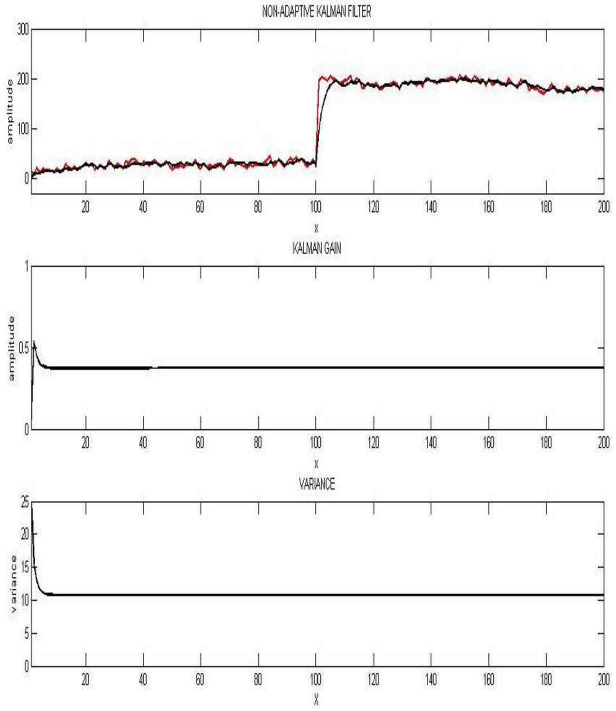

Fig 1:Process and Measurement Models

Consider a change in the value of a pixel in time domain, say from 20 to 200. Its measurement model is the input of the non-adaptive Kalman filter.

Here, we can observe that it takes some time for the output to reach the original pixel value (Fig 2).The value of the Kalman gain and variance first increases and then settles down to a value. It remains around that value for the rest of the process.

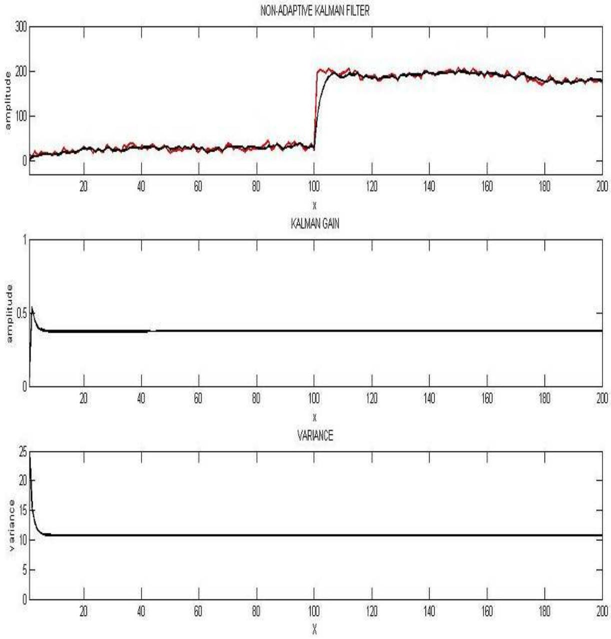

Fig 2: Output,gain and variance in non-adaptive kalman filter

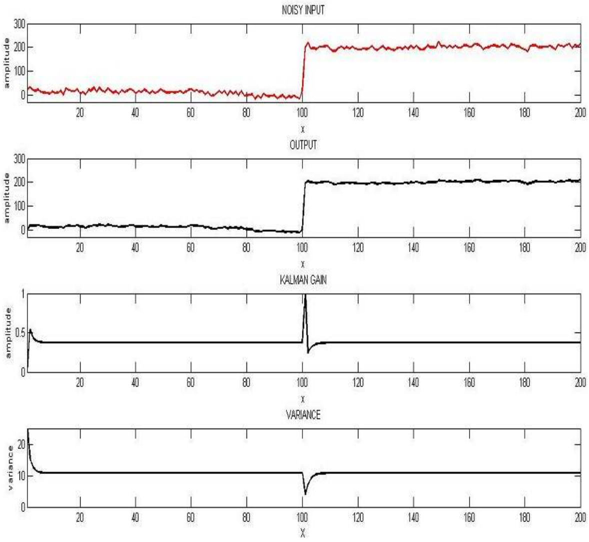

Fig 3: Output, gain and variance in adaptive kalman filter

When the same input is given to an adaptive Kalman filter, (Fig 3), the output is observed to be denoised. This output reaches the desired amplitude at the same time as it detects a threshold greater than or equal to the predefined value. So, there is no delay, as in the case of non-adaptive Kalman filter output is observed here. The Kalman gain of the filter increases to 1 when there the change in magnitude of the adjacent value of temporal pixels is greater than a threshold. When Kalman gain increases to 1, the variance decreases and then settles down to the previous value.Here we can observe that the variance of the estimation error settles to a value around 10 from 25, which indicates that the output have less noise than the input.

For simulation, a video from memory is used. The video is sampled at 16 fps. All the video frames are of size 97*107.The parameters used in Kalman filter calculations are independently stored for every pixel. When passed through an adaptive Kalman filter, the parameters get updated for every temporal pixel in different frames.Fig 4 shows noisy and denoised video frames.

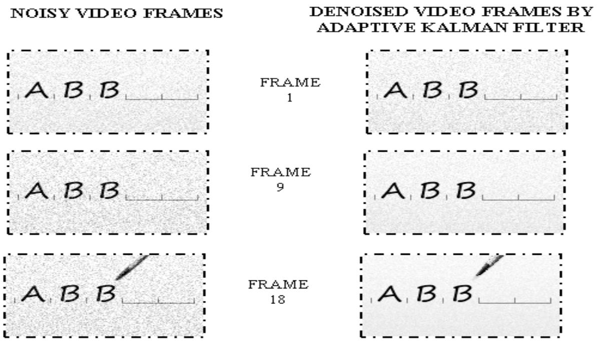

Fig 4. Noisy and denoised video frames

6.2. Verilog Simulation Results

For verilog HDL simulation, value of a temporal pixel from 200 frames of a video was taken from MATLAB. This floating point number is converted to fixed point number with 10 bits binary digit precision for fractional part. The input to the FPGA is 19 bits, where MSB represents the sign bit, next 8 bits represent the integer part and the last 10 bits represent the fractional part.

When the system is in reset state (rst=1), all the parameters a, σ_w, σ_v are assigned its values and the previous values of variance and output are initialized. When rst=0, for every clock edge (rising or falling edge), the program jumps from one state to another, calculating threshold, Kalman gain, variance of estimation error and output pixel value.

The output of FPGA is 19 bits, where MSB represents the sign bit, next 8 bits represent the integer part and the last 10 bits represent the fractional part.

There are mainly 4 blocks, namely, Kalman gain computation, variance of estimation error computation, threshold computation and computation of output pixel value.Fig 5 and 6 shows the HDL simulation results of non-adaptive and adaptive kalman filters .

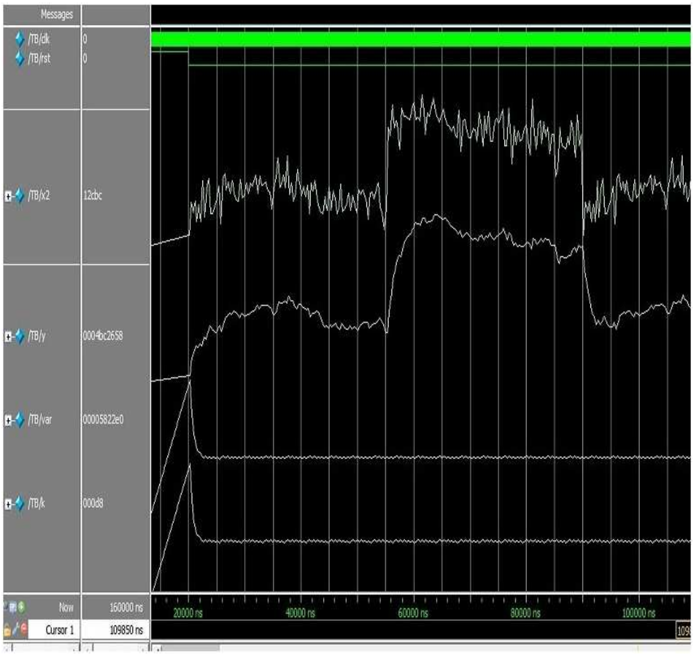

Fig 5: HDL implementation of non-adaptive Kalman Filter

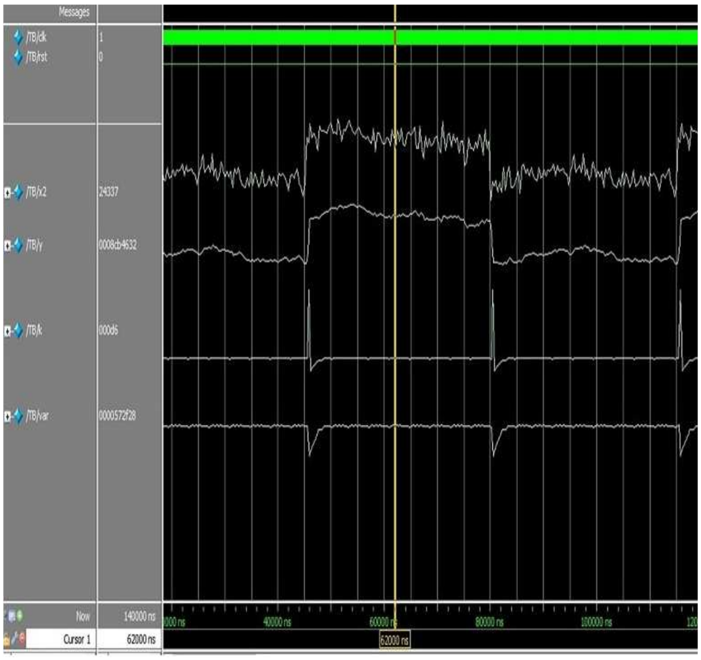

Fig 6: HDL implementation of adaptive Kalman Filter

The result of adaptive kalman filter from fixed point representation in VERILOG HDL and floating point representation in MATLAB is compared using MATLAB tool.The variation between the two was found to be negligible. The plot between the two results is shown in fig 7**.**

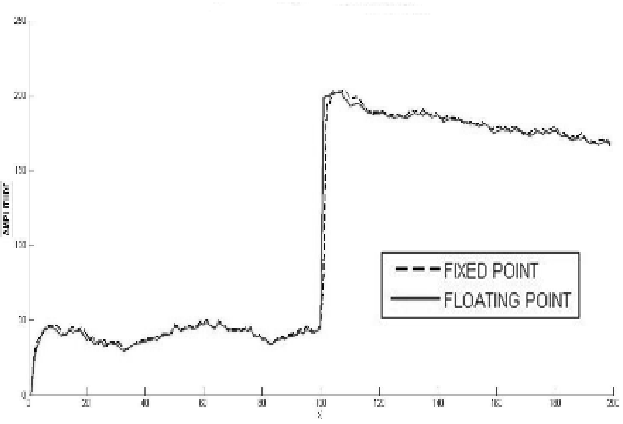

Fig 7: Comparison between fixed and floating point representations

7. CONCLUSIONS

Adaptive filters are important signal processing tools. The basic Kalman filter is a linear, discrete-time, finite-dimensional system, which is endowed with a recursive structure that makes a digital computer well suited for its implementation. Adaptive Kalman filter requires much simpler memory than the general 3D Kalman filter for video denoising. As only 1 pixel is taken at a time for noise removal, the complexity greatly reduces i.e. the model reduces to a 1 dimensional model.

A new architecture for VLSI implementation of simplified adaptive Kalman filter with unknown noise statistics is designed. The implementation is done on XILINX SPARTAN 3 FPGA.

Some of the ideas for future extension are

- The proposed implementation can be extended to colour video signals.
- A better system can be implemented as complex models using advanced Kalman filter techniques for non-linear applications.
- It can also be extended for coloured noise and impulsive noises.

REFERENCES

[1]. Kalman, R., E.; A New Approach to Linear Filtering and Prediction Problems", Transactions of the ASME – Journal of basic Engineering.82. pp.35-45,1960

[2]. Maybeck Peter S, "Stochastic models, estimation and control" volume 1, 1979

[3]. Sorenson H W, "Least square estimation from Gauss to Kalman", 1970

[4]. Gerhard Doblinger ,"An Adaptive Kalman Filter for the Enhancement of Noisy AR signals", Proc. 1998 IEEE Int. Symp. On circuits and Systems, May 31-June3, 1998
[5]. Kim Jaemin and Woods W John ,"Spatio–Temporal Adaptive 3-D Kalman Filter for video " IEEE TRANSACTIONS ON IMAGE PROCESSING , VOL.6, NO.3, 1997
[6]. Ruchi Pasricha, Sanjay Sharma,"An FPGA Based design of fixed point Kalman filter", DSP Journal, vol.9 issue 1, June 2009
[7]. Yiran Li,"FPGA Implementation for image processing algorithms", December 2006
[8]. Simon Dan :"Kalman Filtering" , Embedded Systems Programming, June 2001
[9]. Andres P. Angus, Grewal S. Mohinder :"Kalman Filtering : Theory and Practice Using Matlab", Second Edition
[10]. Bishop Gary, Welch Greg :"An Introduction to the Kalman Filter", Course 8 SIGGRAPH,2001
[11]. Braileam J. C., Kleihorst R.P., Efstratiadis S ,Katsaggelos A. K, and Lagendijk R.L, Noise reduction filters for dynamic image sequences: A review," Proceedings of The IEEE,vol. 83, no. 9, September 1995
[12]. Brown R. G. and Hwang P. Y. C., "Introduction to Random Signals and Applied Kalman Filtering",3rd Ed., John Wiley and Sons, New York 1997
[13]. CHEN GANG and GUO LI,"The FPGA Implementation of Kalman Filter"
[14]. Haykin. S., Adaptive Filter Theory, 3rd Ed., Prentice Hall, New Jersey, 1996
[15]. Kirk. R. E. , Statistics, An Introduction, 3Td Ed., Holt,Rinehart and Winston, Inc., Texas, 1990
[16]. Lacey Tony "Tutorial: The Kalman Filter" Chapter 11
[17]. Promkajin Nicom and Noppanakeepong Suthichai , "An Improvement to the Adaptive Kalman Filter with the Feedback of estimation Error",Student Conference on Research and Development (SCOReD Proceedings, Putrajaya, Malaysia, 2003
[18]. Turney D Robert , Ali M Reza ', and Justin G. R. Delva," FPGA Implementation of adaptive KALMAN filter for real time video filtering", 1999
[19]. XILINX ISE In-Depth Tutorial.
[20]. XILINX Spartan-3 Starter Kit Board User Guide.

BIOGRAPHIES

Binsa Mathew is currently working as Asst.Professor in Holy Kings College of Engineering and Technology. She took her Btech in Electronics and Communication Engineering and Mtech in Vlsi and Embedded systems.

Prof. M. Mathurakani has graduated from Alagappa Chettiar College of Engineering and Technology of Madurai University and completed his masters from PSG college of Technology, Madras University. He has worked as a Scientist in Defence Research and Development Organization (DRDO) in the area of signal processing and embedded system design. He was honoured with the DRDO Scientist of the year award in 2003.Currently he is a professor in Toc H Institute of Science and Technology, Arakunnam. His area of research interest includes signal processing algorithms, embedded system implementations, reusable architecture and communication systems.