Electronic Journal «Technical Acoustics»
http://www.ejta.org

2006, 1

Bibhuti Bikramaditya[1*], Ohyun Kwon[2], Sateesh Kumar Talapuri Venkata Sai[1], Benjamin Ryu[1], Joonki Paik[2]

[1] *IT magic Co. Ltd, 4F, Wonu Bldg., 907-16, Daechi-dong, Gangnam-gu, Seoul 135-280, Korea, http://www.itmagic.co.kr*
[2] *Image Processing and Intelligent Systems Laboratory, Department of Image Engineering, Graduate School of Advanced Imaging Science, Multimedia, and Film, Chung-Ang University, 221 Huksuk-Dong, Tongjak-Ku, Seoul 156-756, Korea, http://ipis.cau.ac.kr*

# Reconfigurable VLSI architecture design for real time image stabilization

*Received 14.09.2005, published 09.01.2006*

This paper proposes reconfigurable VLSI (Very Large Scale Integration) architecture design of real time image stabilization to remove the unwanted displacement due to camera motion and the displacement of the target. This is based on image preprocessing, based many steps (namely light compensation, thresholding, scaling and offset, histogram equalization, LUT operator), followed by sub image phase correlation for motion estimation and kalman filtering for motion correction and stabilization. For the implementation into VLSI, FPGA (Field Programmable Gate Arrays) board consisting 3×2 array of vertex 2 as processing element (PEs) is interfaced with PCI port of the PC and the camcorder is connected to USB port of the FPGA board for capturing, visualization and testing of the image if any.

Image preprocessing techniques are applied on input image before actual processing to suppress the unwanted distortion or to enhance some image features, which is important for further processing. Global motion is estimated from the local motion vector (LMV) and the average of two maximum peak amplitudes from the block of LMV decides its global motion vector, thereby accumulating motion vector for panning. The kalman filtering based motion correction system stabilizes image caused by unwanted movement or camera vibration. This proposed system design is expected to achieve a target frame rate of 30 fps, integration time 3.33 ms per frame producing 7.5 mbps image data when pixel is converted to 8 bit digital value for 320×240 QVGA input image.

The proposed design algorithms has been implemented and verified with Frame grabber board (Matrox Meteor-II Mil-lite software) that is interfaced with camcorder and PC. Similarly, the same design has also been tested using ARM Emulator and DM320 Board (ARM9TDMI core, DSP core).

## INTRODUCTION

The image jittering due to unwanted movements and vibration caused by rotational and transnational motion of the camera is common problem nowadays, particularly in the real time image processing systems and in many vision tasks such as tele-operation, robot navigation, ego motion recovery, scene modeling, video compression and detection of

corresponding author, e-mail: bbikramaditya@itmagic.co.kr

independent moving objects. The unwanted displacement in pixel of the image plane due to hand trembles and heartbeat of the person-carrying camera also affects the quality of image. Many FPGA based custom computing machines (FCCM) have been developed for this purpose by utilizing linear array of FPGAs or single FPGA integrated with SRAM. But the practicality of these approaches was limited by the FPGA's to perform arithmetic functions and store operands. In this direction, the novel attempt to combine FPGA, memory and interconnection chips in the architectures of Virtual computer (VM), CHAMP and Splash 2, require around 14 to 52 FPGAs with fixed support chip resources which is economically costly affair. The FPGA based re-configurable system design to remove unwanted movement in a camera is a noble effort in this direction to give cost effective, reliable, flexible and efficient solution and improve the visual quality through stabilization process

Real time digital image stabilization can be divided into three parts, namely image preprocessing, global motion estimation system and image smoothing/correction system. The image preprocessing is basically low level image processing and characterized by large amount of input data and simple calculations and help in stabilizing input before it comes for real processing or stabilization. Image preprocessing comprises several operations namely light compensation, offset operation, average operation, threshold operation, Look Up Table (LUT) generation and histogram equalization. The motion estimation system is responsible for the estimation of inter-frame global motion vectors, which are forwarded to the motion correction system. The motion correction system accomplishes the stabilization of the image sequence according to the global motion model or objective. Various digital image-stabilizing systems have been developed. Most often, the motion estimation based on block matching algorithm [1, 2] and phase correlation algorithm are being proposed for the same [3, 4, 5]. In block matching algorithm, particularly point matching, edge pattern matching, gray-coded bit-plane matching and block motion vectors filtering have been developed. But this algorithm is not useful for both translational as well as rotational motion of the camera and for that purpose phase correlation method for motion estimation is adding advantages with better quality and results.

In this paper, we propose a FPGA based reconfigurable model system design of digital image stabilization. The design consists of image preprocessing system followed by sub-image novel phase correlation based global motion estimation and kalman filtering based image correction and stabilization. Global motion is estimated from the local motion of four sub-images each of which is detected using sharpening filter before phase correlation (PC) based motion estimation and average of two maximum peak amplitude from the block of LMV decides its global motion vector (GMV). The motion correction system is basically based on kalman filtering.

Figure 1 presents image stabilization system model in which, real time image captured by camcorder is interfaced with 3×2 array of Vertex 2 series based system, known as modified MORRPH board. Algorithm mentioned here is implemented on the MORRPH boards, which is interfaced with PC by its PCI interface to get the stabilized image.

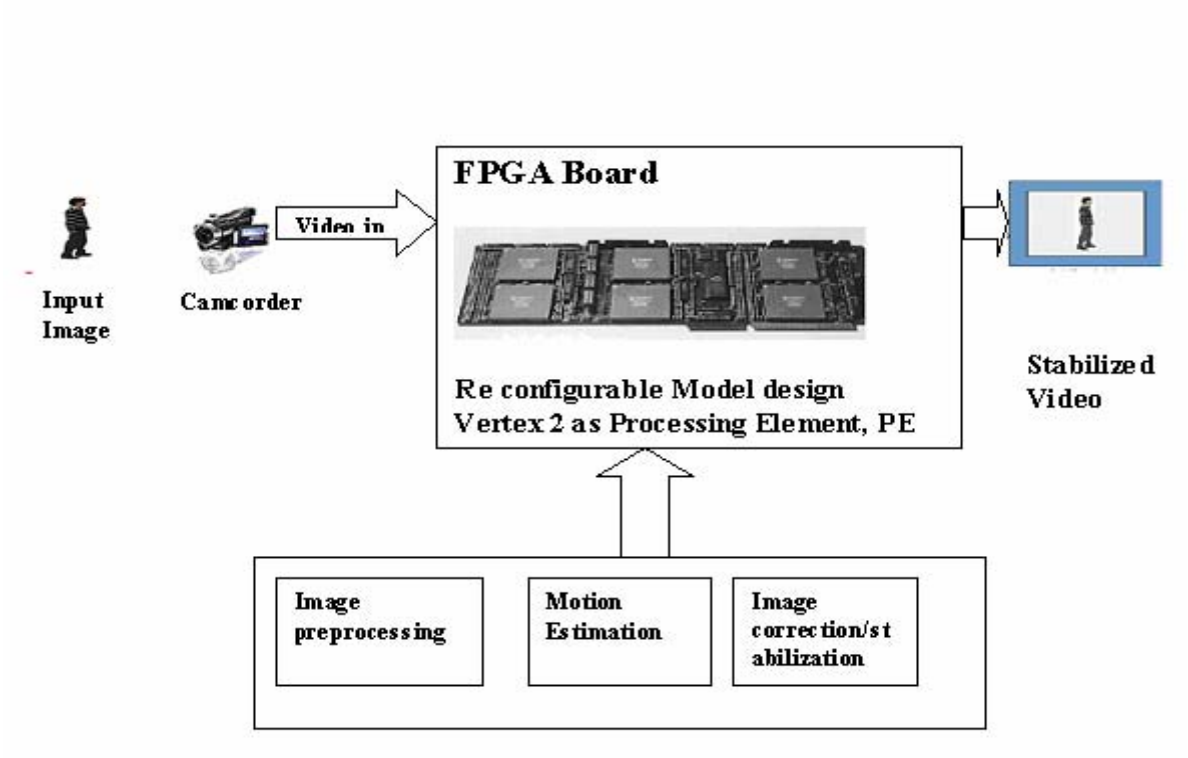


Figure 1. Image stabilization system

The proposed system design is expected to achieve a target frame rate of 30 frames per second, integration time 3.33 ms per frame producing 7.5 mbps image data when each pixel is converted to 8 bit digital value for input image size of 320×240.

This paper is organized as follows: In section 1 we summarize the image preprocessing system. In section 2 we summarize the sub-image phase correlation based global motion estimation algorithm and in section 3 we summarize image correction and stabilization using kalman filtering. Section 4 describes FPGA implementation of kalman filter and in section 5 hardware design for the complete system has been proposed. In section 6 the experimental results based on PCM without VLSI approach are given for reference and validity of the PCM algorithm. Sections 7 and 8 give concluding remarks and references respectively.

## 1. IMAGE PRE-PROCESSING

Image pre-processing is a common name for operations with images at the lowest level of abstraction. The main aim of pre-processing is an improvement of the image data that suppresses unwanted distortions or enhances some image features important for further processing. We can say that image processing design is a combination of low level image processing operation that are intended to provide one solution for a specific image processing task. Normally several low image processing task is to be performed on the input image as per application but no standard is yet fixed for all designs. The following steps of low-level image processing task are needed for the implementation of proposed design:

- Light compensation
- Scaling and offset operator
- Threshold operator
- Window average operator
- LUT operator
- Histogram equalization

To explain in brief, light (shade) compensation and the offset and scaling operator is an example of transformation of image. The light compensation or shade compensation is used to correct the non-uniform lighting condition. Scaling and offset are used to brighten or darken the image. The scaling operation multiplies each operation by a constant value and similarly the offset adds the fixed value to each intensity value. Threshold operation is being done to create binary image. 26 individual template-matching operators, each of which calculates the likeness of fit for each individual letter, process binary image. This is basically used for optical character recognition. Using LUT operation 24 million full color values are mapped to a pallet of 2000 colors. Finally 2000 element histogram is calculated from only pixel values of the object in the processed image. 3×3 window average operator performs image smoothing. To identify areas of the defect within a single object in the 24 bit full color image input containing only object and saturated white back ground, the areas of the defects are characterized by color values that do not appear in the object and always represent less than 50% of the object area, however specific color values that represent the object and any possible defects are not known. The histogram counts the frequency of particular pixel intensity value in an image.

## 2. MOTION ESTIMATION SYSTEM BY PHASE CORRELATION METHODS (PCM)

The idea behind this method is quite simple and is based on the Fourier shift property, which states that a shift in the coordinate frames of two functions is transformed till the Fourier domain as linear phase differences. This can be described as follows.

Let $f_k(x,y)$ and $f_{k+1}(x,y)$ be two functions that are absolute integral over $\Re^2$

$$f_{k+1}(x,y) = f_k(x-d_1, y-d_2). \tag{1}$$

According to the Fourier shift property:

$$\hat{f}_{k+1}(u,v) = \hat{f}_k(u,v)\exp\{-j2\pi(ud_1+vd_2)\}, \tag{2}$$

where $u$, $v$ denote the previous and current image parameter.

Hence the normalized cross power spectrum is given by

$$\frac{\hat{f}_{k+1}(u,v)\,\hat{f}_k^*(u,v)}{\left|\hat{f}_{k+1}(u,v)\,\hat{f}_k^*(u,v)\right|} = \exp\{-j2\pi(ud_1+vd_2)\}, \tag{3}$$

where $*$ indicates the complex conjugate.

The normalized cross power spectrum may also be viewed as the cross power spectrum of whitened signals. There are two possible ways of solving (3) for $(d_1,d_2)$. One way is to

directly work in the Fourier domain. For this purpose, consider a three-dimensional (3-D) Euclidean space whose canonical reference frame is given by the two frequency axes and the phase difference between the two images. The second possible approach, which is more practical and also more robust to noise, is to perform IFFT (Inverse Fast Fourier Transform) of the normalized cross power spectrum. It is then a simple matter to determine $(d_1, d_2)$, since from (3) the result is $\delta(x-d_1, y-d_2)$ which a Dirac delta function is centered at $(d_1, d_2)$, which corresponds to the displacement between the two images.

In each image frame of the sequence, four sub-images are designated as shown in Figure 2. These sub-images are used to determine local motion vectors using phase correlation. To enable FFT computation for phase correlation, sub-images are predetermined to be of square shape with horizontal and vertical pixel dimensions being a power of two. Typically a sub-image size of 64×64 is preferred to keep the computation load of motion estimation low and at the same time include sufficient spatial image content for correct estimation.

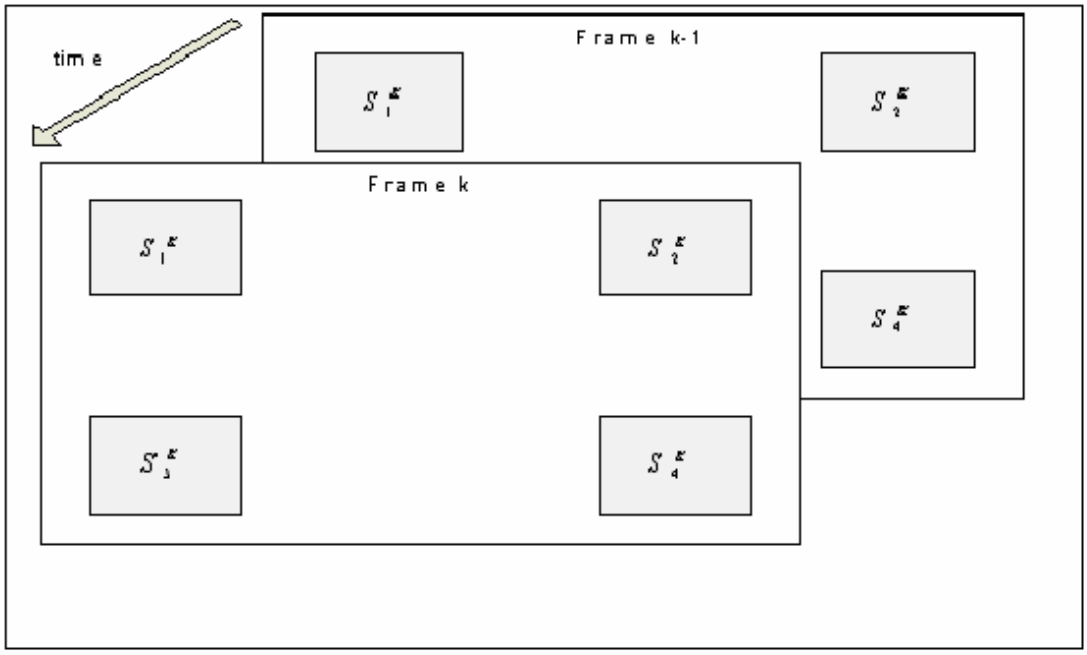


Figure 2. Location for sub images used for local motion estimation

Steps involved in PCM:

Correlation surface determination:

- Divide images $I_{k-1}$ and $I_k$ (Previous image and present image respectively) into pixel 64×64, called sub-image,
- Perform 2D FFT on each of the block,
- Take Inverse 2D FFT and identify the most dominant peak,
- Use the co-ordinates of the peak as the candidate vectors for block matching of 16×16 blocks,
- Normalized cross spectrum of 2D IFFT is the phase correlation surface.

Global motion vector determination:

- Analyzing largest peak amplitude location and their location value, known as local motion vector,
- Take average of two largest local motion vector (LMV) of four-neighborhood pixel,
- Average of determined local motion vector is known as Global Motion Vector, GMV.

For all four sub-images of an image frame, local motion vectors are estimated from the respective sub-images of the previous frame. For each sub-image the largest peak amplitude location of the corresponding phase correlation surface is assigned as the local motion vector and recorded together with the corresponding peak amplitude value. For each image frame, in total four local motion vectors corresponding to the sub-images are estimated and the peak amplitude of each vector is noted. The global motion vector of the image frame can then be decided by taking average of two most peak amplitude values of local motion vectors.

## 3. IMAGE CORRECTION BY KALMAN FILTERING

Global Motion correction by frame position smoothing (FPS) is based on obtaining correction vectors from the difference of original and low pass filtered absolute frame positions. For this purpose, global inter frame motion vectors are accumulated to yield an absolute frame position $X_{raw}(n)$. This signal is low pass filtered to remove the high frequency jitter and retain low frequency part, i.e. smoothened camera displacements representing deliberate camera movements of the stabilized sequences. It has been demonstrated that kalman filter can be used to effectively obtain stabilized frame displacements in real time. Referring to the kalman filter displacements as $X_{kal}(n)$, the correction vector for the frame *n* can be obtained as

$$V_{cor}(n) = X_{kal}(n) - X_{raw}(n), \tag{4}$$

where $X_{raw}(n)$ is the original displacement.

Let x(k) represents a pixel gray scale on frame k. The ideal noise-free pixel values be represented by s(k), assumed to be first order Auto Regressive Model (AR Model), used to represent the temporal behavior of the pixel in video signals. Under this measurement, the process noise model and measurement noise model equation are given by equation (5) and (6) respectively:

$$\mathrm{s(k+1)} = \mathrm{as(k)} + \mathrm{w(k)}, \tag{5}$$

where w(k) is the process noise, assumed to be white independent zero mean Gaussian random process with variance of $\sigma_w^2$, a is the constant depending upon signal statistics.

$$\mathrm{x(k)} = \mathrm{s(k)} + v\mathrm{(k)}, \tag{6}$$

where $v$(k) is the independent additive zero mean Gaussian white noise with variance of $\sigma_v^2$.

It is explicitly assumed that the noise and signals are stationary random processes that are fully determined by their second order statistics. This is practically not true but for local statistics it can be safely used with some approximation.

For the image stabilization system representing horizontal and vertical absolute frame position state estimates by (x1, x2) and (dx1,dx2) representing the corresponding velocity; the state system for image stabilizer is constructed in the form of constant velocity camera model:

$$\begin{pmatrix} x1(n) \\ x2(n) \\ dx1(n) \\ dx2(n) \end{pmatrix} = \begin{pmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x1(n-1) \\ x2(n-1) \\ dx1(n-1) \\ dx2(n-1) \end{pmatrix} + \begin{pmatrix} wx1 \\ wx2 \\ wdx1 \\ wdx2 \end{pmatrix}. \quad (7)$$

The estimated absolute frame position obtained by accumulation of estimated global inter frame motion vectors are used in the observation system. The observation system is defined in terms of

$$\begin{pmatrix} Y1(n) \\ Y2(n) \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix} \begin{pmatrix} x1(n) \\ x2(n) \\ dx1(n) \\ dx2(n) \end{pmatrix} + \begin{pmatrix} vx1 \\ vx2 \end{pmatrix}. \quad (8)$$

The kalman filter estimates the process state from equation (7) known as prediction stage and then obtains feedback in the form of noisy measurements state, known as update stage as given in equation (8). The process and observation model can simply be plugged into a generic form of kalman filter, which carries out the resulting algebra to obtain a state estimates for any instance. The kalman filter output is obtained recursively through prediction/update stages enabling real time operation of the filter.

The recursive kalman filter is developed as per following definitions:

- The filter out put is represented by y(k) which is the estimate of the signal at k,
- The variance of the estimation error is theoretically defined by $\sigma^2(k) = E[\gamma(k) - s(k)]^2$ which is initially unknown,
- The kalman filter gain is given by K (k).

The overall algorithm of the kalman filter is given by

Let y(–1) = 0, $\sigma_w^2 = \sigma_v^2$, and $\sigma^2 = \sigma_v^2$, and start the loop by setting k = 0.

LOOP: for k do the following operations:

$$K = \frac{\sigma^2 + \sigma_w^2}{\sigma^2 + \sigma_w^2 + \sigma_v^2} ;$$

K (k) = K , for k=0.

If K = 0 then

y(k) = y(k–1) + K [x(k) – y(k–1)]

else

y(k) = x(k)χ + (1–K) y(k–1);

If $y = \frac{|x(k) - y(k-1)|}{\sigma_v}$ **? Γ ,**

$\sigma_w^2 = \sigma_v^2$ ;

$\sigma^2 = \sigma_v^2$;

else

$\sigma_w^2$ = K= $\sigma_v^2$ ;

$\sigma^2$ ℜ (1 – K) $\sigma^2$ + $\sigma_w^2$ ;

end-if

Increment k, k ℜ k+1, and go back to LOOP

END:

**Γ** is the threshold values for the motion detection and $\gamma$ is the confidence level. For the purpose of the statistical test of hypothesis for different values of threshold level, confidence levels is measured and are tabulated. It is shown that the highest confidence level (99.9%) results good better performance (**Γ=3.29)**. $\sigma_w^2$ and $\sigma_v^2$ are the Gaussian Variance white noise and random process noises respectively and are unknown. The algorithm is made adaptive by properly estimating parameters, $\sigma_w^2$ and $\sigma_v^2$, using local information. The estimation of these parameters can be based on the criterion functions, like Minimum Mean Square Error, MMSE.

Figure 3. Kalman filter mathematical formulation

## 4. FPGA IMPLEMENTATION OF KALMAN FILTER

The system design for the adaptive (kalman) filter is based on the assumptions that the image sequence and noise are stationary and their statistics are known. Statistics of image sequence and noise can be estimated if these signals are really stationary. Adaptive filters are based on dynamically adjusting the parameters of supposedly optimum filter which are the estimates of unknown parameters. An adaptive kalman filter is an online estimation of motion as well as signal and noise statistics available data.

The FPGA implementation of the adaptive kalman filters is performed with standard memory components and Xilinx FPGA. The memory components are used for frame buffers and parameter buffers and FPGA is used for pixel and parameter calculation. The system architecture, shown in the Figure 4, illustrates three input buffers holding y(k–1), $\sigma_w^2$, $\sigma^2$ and a Xilinx implementation to calculate updated values for these buffers in addition to generating the proper output y(k). The system can easily operate at 66 MHz clock enabling 1024×1024, 60 frames per second operation. System clock rates of 80 MHz and 100 MHz can also be achieved for more system bandwidth requirements.

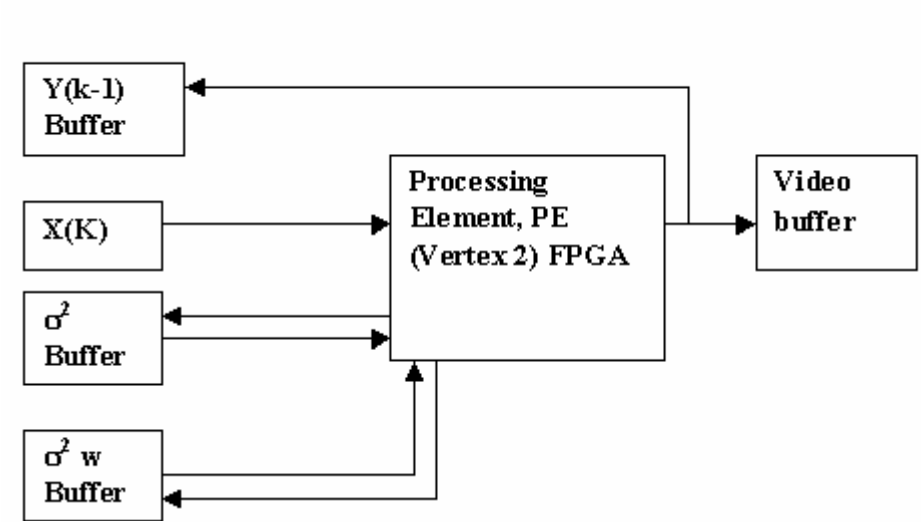


Figure 4. System level kalman filtering FPGA implementation

## 5. PROPOSED MULTI FPGA BASED RC MODEL DESIGN HARDWARE

Many attempts have been made for the implementation of the image processing designs into general Multi FPGA based CCM but lack of modularity and inflexibility, simplicity and cost ineffective solution of the design, paved solid ground for another flexible and efficient architecture called as Re-configurable model special design called as MORRPH, (MOdular and Re-programmable Real Processing). This model relieves much computational burden and increase in the throughput of the commercial system. The said incompletely specified architecture of MORRPH could be operation dependent. The interconnections that exist on the PCB should support types of architectures that are typically used to solve problems for

which the board is intended. For examples, both pipelined or mesh architectures are commonly used to solve the image processing problems. The interconnection of any incompletely specified architecture intended for image processing should support the translation of circuits with these architectures. The MORRPH architecture suffices the problem.

The said proposed MORRPH system based has to be modified and implemented by an PCI interface card with 3×2 array of Processing Elements (PE). Each PE consists of open socket which can be populated with a Xilinx Vertex 2 series FPGA and open support socket which can be populated with the support chips such as external RAM, Math Processors etc.

In implementing the prototype algorithms for the commercial system, a partition may be created between those algorithms that would be implemented on the MORRPH board, and those would be left as implemented on host PC. Figure 5 shows the complete algorithm steps for the proposed real time image stabilization system architecture design.

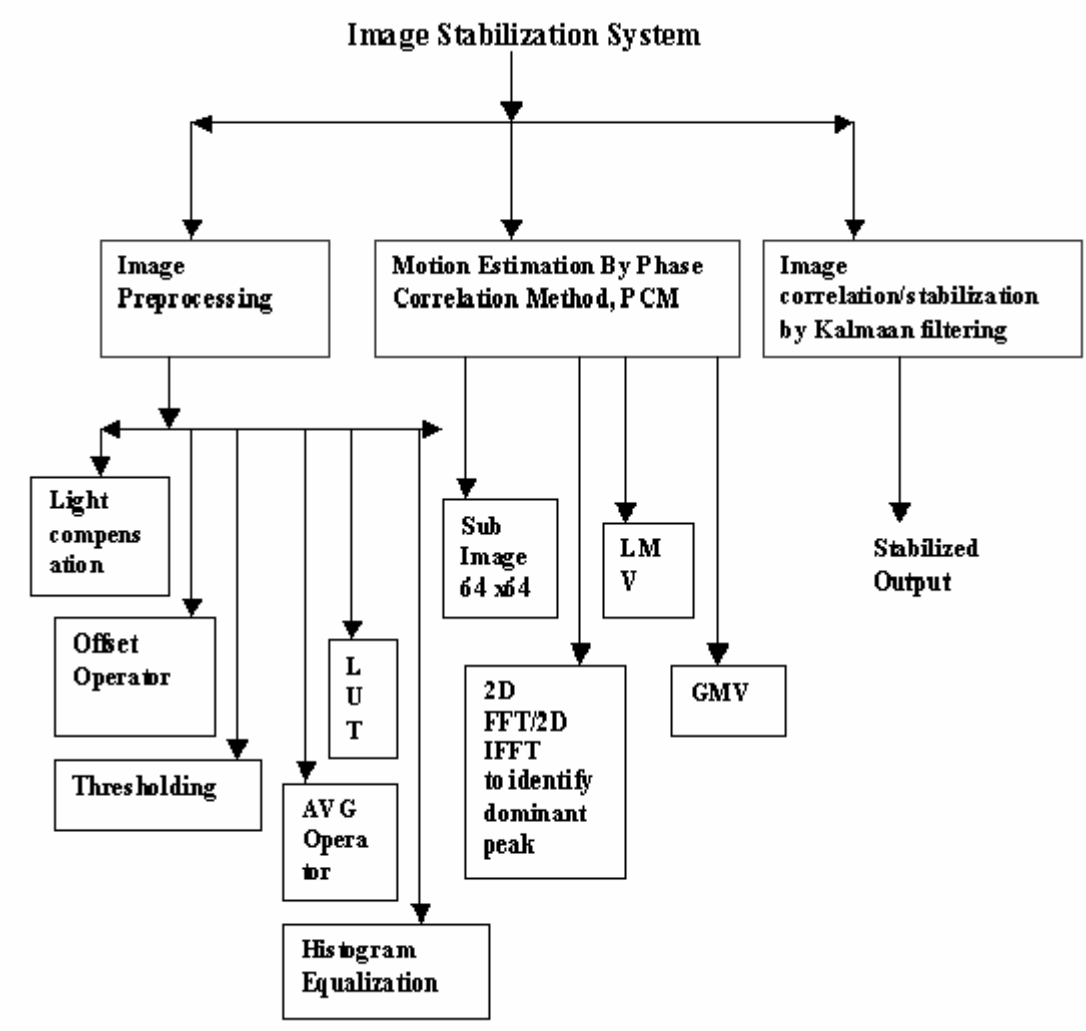


Figure 5. Algorithm steps

The MORRPH architecture can be divided into three subsystems:

- Processing Element (PE) array,
- PCI interface,
- I/O buses.

*Processing Element array (PE array):*

The SRAM based FPGA chips allow the processing unit to be reconfigured an unlimited amount of times after power is established to the FPGA chips. The MORRPH concept specifies two-dimensional rectangular mesh of Processing Elements (PE) called processing array or MORRPH PE array, scaleable in both the dimensions to M×N array.

The structure of PE is given in the Figure 6. Four interconnection buses are used for communication with processing elements to north, south, east and west. The interconnection busses are all the same sizes, providing regular architecture. A final bus that supports the socket bus supports the FPGA sockets to the support chip sockets. A single FPGA can be inserted into the FPGA socket of each PE .The SRAM based FPGAs contain gate level logic resources that can be reconfigured many times. A support chip socket provides enough space to insert several support chips. These storage and processing chips implement functionality that is not efficiently realized by the hardware resources of the FPGA chips.

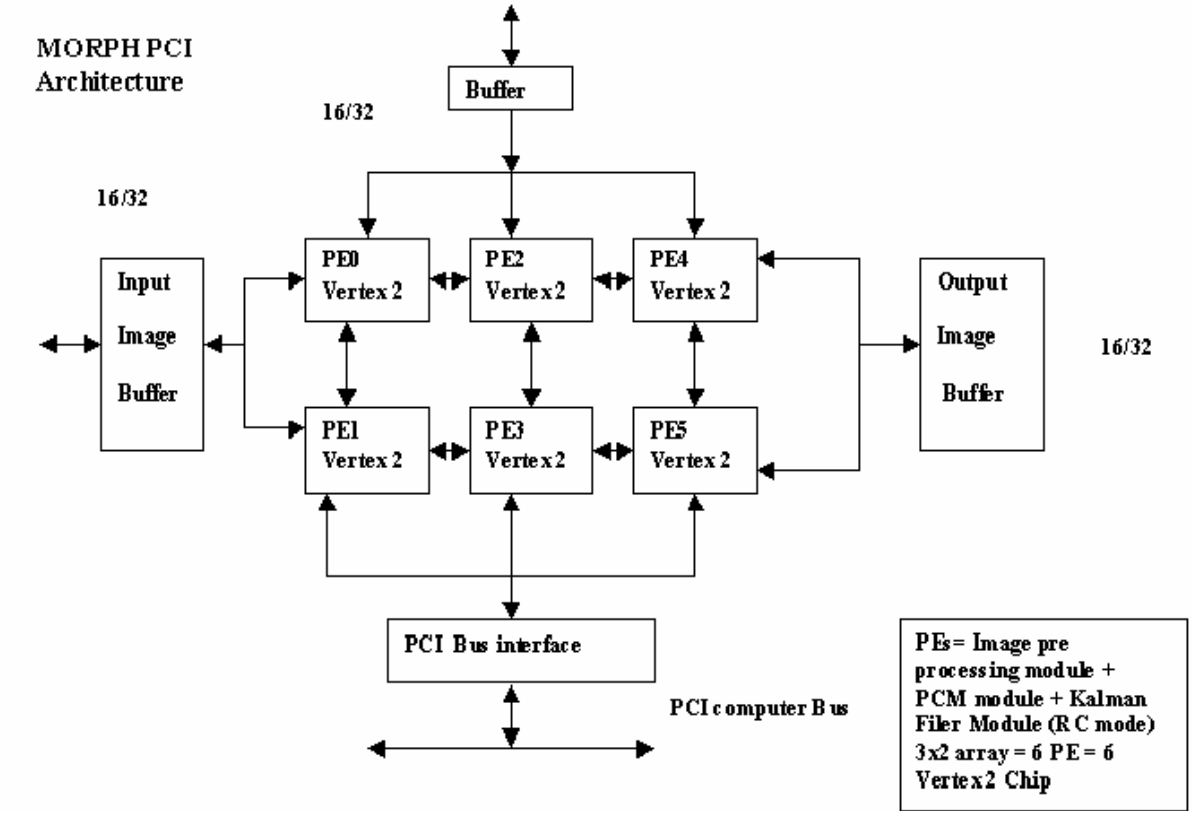


Figure 6. MORRPH-PCI Architecture

Each element of the 3×2-processing array consists of an FPGA and an open support socket. The FPGA is connected to the support socket by 80-bit bus, and each FPGA is connected to its North, South, East and West neighbors by a 32-bit data bus. If a processing element is located in the array such that it doesn't have neighbor in a particular direction, then those pins become part of one of the I/O buses or the PCI interface. Thus the external connections as seen by each processing element are consistent, regardless of the elements location in the array.

*PCI interface:*

The PCI interface has three functions: providing power to the board, performing configuration of the FPGA and providing a sources for low bandwidth communications. The processing element are arranged in a 3×2 array, the PCI interface needs to control only the three FPGAs along the top of the array. Each of the three FPGAs along the top of the array is accessed and configured by the FPGA that is directly below it in the array. This is known as daisy-chained configuration.

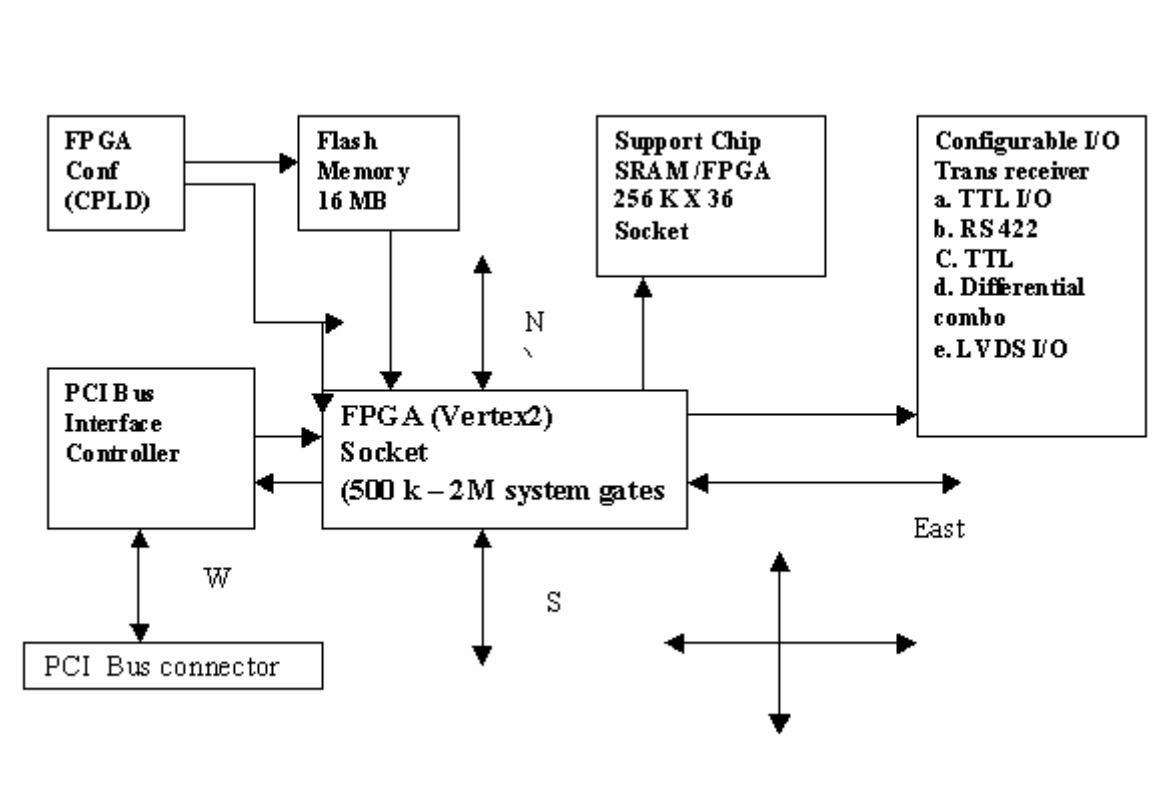


Figure 7. PE array socket and its PCI interface

*I/O buses:*

The real time image data received to and transmitted from MORRPH PCI board using I/O buses at the edge of the processing array. The three I/O buses at the top, left and right of the PE architecture Figure 7 are used to transmit data. The width of the board I/O buses is the same as North, South, East and West interconnection buses of the individual PE. The image data is input from these I/O buses and processed by the PEs in the processing array and then the result data is output to the one of the three I/O buses all in real time.

*Communication protocols:*

The specially designed protocol named as SUIT (Synchronous Unidirectional Image Transfer) for chip to chip and chip to input image data transfer, work as set of reusable modules. This can be implemented on new software development system called TRAVERSE. This protocol is flexible enough to transmit a variety of image or result data formats. This establishes communication between separate modules that are connected in the same FPGA modules or in a separate FPGA chip on the same board. Therefore these module have access to common all global clock signal to synchronize the transfer of data. This eliminates timings and circuitry overheads to generate the handshaking signals associated with synchronous data transfers.

*ZEE protocol:*

The SUIT bus is not suitable for inter board communication when a common clock is not able to synchronize the transfers. To enable communication between separate board level devices with transmission cable in an image processing system, SUIT Bus is modified to include a clock signal. The new 16 bit modified SUIT bus format is called ZEE Format. In order to run the MORRPH board under 32 bit multitasking environment, code for device driver is written that allows to generate interrupts and to access the MORRPH ports VHDL language is used. The TRAVERSE, special software is suitable for this purpose that creates real time image processing solutions for MORRPH Board.

## 6. EXPERIMENTAL RESULTS

In order to test the performance of the proposed system before starting FPGA based system design and implementation, we tested PCM algorithm by connecting camcorder and computer by frame grabber from Meteor II standard, first on a single image frame from SONY Camcorder DCR-TRV900 for an 640×480 image in outdoor and indoor.

To show the performance of the proposed algorithm, Figure 8 shows the original image and stable image in outdoor and Figure 9 shows the original image and stable image in indoors.


Figure 8. Experimental results in outdoor. If you click play button, it will show how image stabilizes (right side) when image is destabilized (left side image) by hand shaking or any other sources

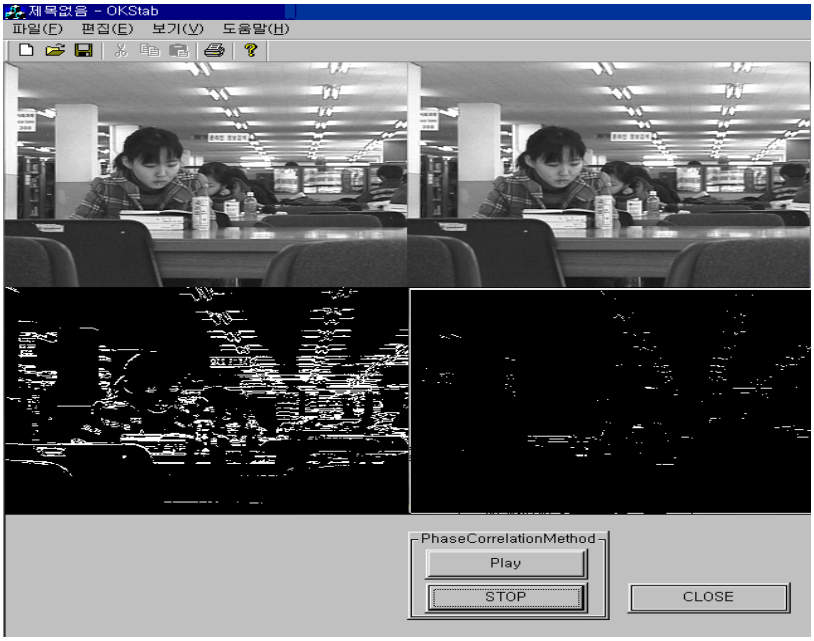


Figure 9. Experimental results in outdoor; button operation is same as in figure 8.

The performance is evaluated using the root mean square error. The proposed method yields better results than other existing methods .The same design has also been tested by us in the ARM Emulator and ported into DM320 Board (ARM9TDMI core, DSP core).

## 7. CONCLUSIONS

In this paper, we proposed a FPGA based re-configurable model design of real time image stabilization to remove unwanted displacement due to camera motion and displacement of the target, which is based on image preprocessing, followed by sub image phase correlation for motion estimation and kalman filtering for motion correction and stabilization. We found that MORRPH based hardware using 3×2 array of Virtex 2 (2M system gates) as Processing Element (PEs) will be best-suited design for the real time image stabilization. The said FPGA Board will be interfaced with PCI port of the PC and camcorder is connected to the USB port of the FPGA board for visualization and testing of the image jittering if any. The above-mentioned experimental result (using PCM based motion estimation and compensation that have been implemented and verified with Frame grabber board that is interfaced with camcorder and PC) is not at all related with this board but it has certainly helped us to decide about the exact hardware of the proposed system for which the work has already been started.

## REFERENCES

[1] J. K. Paik, Y. C. Park, D. W. Kim. An adaptive motion decision system for digital image stabilizer based on edge pattern matching. IEEE Trans. Consumer Electronics, vol. 38, pp. 607–615, 1992.

[2] K. Uomori, A. Morimura, H. Ishii, Y. Kitamura. Automatic image stabilization system by full-digital signal processing. IEEE Trans. Consumer Electronics, vol. 36, pp. 510–519, 1990.

[3] S. Erturk. Digital image stabilization with Sub-image Phase correlation based global Motion Estimation. IEEE Trans. Consumer Electronics, vol. 49, pp. 1320–1325, 2003.

[4] S. Erturk, T. J. Dennis. Image sequence stabilization based on DFT filtering. IEEE Proc. on Image Vision and Signal Processing, vol. 127, pp. 95–102, 2000.

[5] H. Foroosh, J. B. Zerubia, M. Berthod. Extension of phase correlation to subpixel registration. IEEE Trans. Image Processing, vol. 11, pp. 188–200, 2002.

[6] L. Marcenaro, G. Vernazza, C. S. Regazzoni. Image stabilization algorithms for video-surveillance applications. Proc. Int. Conf. on Image Processing, vol. 1, pp. 349–352, 2001.

[7] Thomas H. Drayer, Josheph G. Tront, Tichard W. Conners, Philip A. Araman. A Development system for Real Time Image Vision Hardware Using Field Programmable Gate Arrays. In a proceedings of the 32-nd Hawaii International Conference on System Sciences. 1999.

[8] Robbert D. Turney, Ali M. Reza, Justin G. R. Delva. FPGA Implementation of the Adaptive Temporal Kalman Filter for real time video filtering. http://www.xilinx.com/products/logicore/dsp/temporal_kalman_fltr.pdf

[9] Pedro Cobos Arribus, Felix Monasterio and Huelin Macia. FPGA Board for real time vision development systems. http://ieeexplore.ieee.org/Xplore/login.jsp?url=/iel5/7865/21660/01004106.pdf?arnumber=1004106

[10] Thomas Hudson Drayer. A design methodology for creating Programmable Logic based Real time Image Processing Hardware. PhD Dissertation. Virginia Polytechnic institute and State University.