FPGA IMPLEMENTATION OF REAL TIME DIGITAL VIDEO SUPER RESOLUTION FOR INFRARED CAMERAS

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
OF
MIDDLE EAST TECHNICAL UNIVERSITY

BY

MEHMET AKTUKMAK

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR
THE DEGREE OF MASTER OF SCIENCE
IN
ELECTRICAL AND ELECTRONICS ENGINEERING

JANUARY 2013

Approval of the thesis:

FPGA IMPLEMENTATION OF REAL TIME DIGITAL VIDEO SUPERRESOLUTION FOR INFRARED CAMERAS

submitted by **MEHMET AKTUKMAK** in partial fulfillment of the requirements for the degree of **Master of Science in Electrical and Electronics Engineering Department, Middle East Technical University** by,

Prof. Dr. Canan Özgen _____
Dean, Graduate School of **Natural and Applied Sciences**

Prof. Dr. İsmet Erkmen _____
Head of Department, **Electrical and Electronics Engineering**

Prof. Dr. Uğur Halıcı _____
Supervisor, **Electrical and Electronics Eng. Dept., METU**

Examining Committee Members:

Prof. Dr. A. Aydın Alatan _____
Electrical and Electronics Engineering Dept., METU

Prof. Dr. Uğur Halıcı _____
Electrical and Electronics Engineering Dept., METU

Prof. Dr. Gözde Bozdağı Akar _____
Electrical and Electronics Engineering Dept., METU

Asst. Prof. İlkay Ulusoy _____
Electrical and Electronics Engineering Dept., METU

M.Sc. Uğur Güngör _____
Electronic Design Department, ASELSAN, MGEO

Date: **January 31, 2013**

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name, Last Name : Mehmet AKTUKMAK

Signature :

ABSTRACT

FPGA IMPLEMENTATION OF REAL TIME DIGITAL VIDEO SUPERRESOLUTION FOR INFRARED CAMERAS

Aktukmak, Mehmet

M.Sc., Department of Electrical and Electronics Engineering
Supervisor: Prof. Dr. Uğur Halıcı

January 2013, 64 pages

At present, the quality of image taken from infrared cameras is low compared to the other cameras because of manufacturing technology. So, resolution enhancement processes are becoming more important for these cameras. Super resolution is a good approach to solve this resolution problem. In general, the systems that infrared cameras used require video processing to perform in real time. So, a suitable approach should be selected and implemented to work in real time. The computational load and processing time are big issues in this case. FPGAs are proven to be suitable hardware devices for these types of works.

Super resolution involves two parts as global motion estimation and high resolution image reconstruction. In this study, one suitable algorithm, namely as PM, for global motion estimation in the literature is selected to be implemented in real time. On the other hand, for high resolution image reconstruction part, FPGA structures of some well known algorithms in the literature, namely as POCS, MLE, MAP and LMS are proposed and their performance, resource requirements and timing considerations are discussed. Most efficient one is selected and implemented in FPGA.

Keywords: real time super resolution, real time image registration, infrared camera, resolution enhancement, FPGA implementation

ÖZ

KIZILÖTESİ KAMERALAR İÇİN GERÇEK ZAMANLI SAYISAL VİDEO SÜPER ÇÖZÜNÜRLÜK FPGA UYGULAMASI

Aktukmak, Mehmet

Yüksek Lisans, Elektrik ve Elektronik Mühendisliği Bölümü
Tez Yöneticisi: Prof. Dr. Uğur Halıcı

Ocak 2013, 64 sayfa

Günümüzde, üretim teknolojisinden dolayı kızılötesi kameralardan alınan görüntülerin kalitesi diğer kameralardan alınanlara göre düşük olmaktadır. Bu yüzden görüntü iyileştirme işlemleri bu kameralar için daha önemli bir hale gelmektedir. Süper çözünürlük bu problemin çözümü için iyi bir yaklaşımdır. Genel olarak, kızılötesi kamera kullanan sistemler video işlemlerinin gerçek zamanlı yapılmasına gereksinim duymaktadır. Bu yüzden uygun bir yaklaşım seçilmeli ve gerçek zamanlı çalışacak şekilde gerçeklenmelidir. Bu durumda hesaplama yükü ve işlem süresi önemli hale gelmektedir. FPGA'ler bu tip işler için uygun donanımsal elementlerdir.

Süper çözünürlük, global hareket tahmini ve yüksek çözünürlüklü görüntü oluşturma kısımları olarak iki kısımdan oluşur. Hareket tahmin kısmı için literatürde bulunan PM isimli uygun bir algoritma seçilir ve gerçek zamanlı olarak gerçeklenir. Yüksek çözünürlüklü görüntü oluşturma kısmı için ise literatürdeki POCS, MLE, MAP and LMS isimli temel algoritmalardan bazıları için FPGA yapıları önerilir ve performans, kaynak kullanımı ve zamanlama kriterleri tartışılır. Bu algoritmalardan en verimli olanı seçilir ve FPGA'da gerçeklenir.

Anahtar Kelimeler: gerçek zamanlı süper çözünürlük, gerçek zamanlı görüntü eşleştirme, kızılötesi kamera, çözünürlük artırma, FPGA gerçeklemesi

To My Family…

ACKNOWLEDGMENTS

Firstly, I would like to express my sincere thanks to my supervisor Prof. Dr. Uğur HALICI, for her support, friendly attitude and encouragement at each stage of this thesis study.

I would like to thank to ASELSAN, MGEO Electronics Design and Image Processing Departments for the support given throughout this study.

I would like to forward my appreciation to all colleagues for their continuous encouragement.

TABLE OF CONTENS

TABLES

FIGURES

LIST OF ABBREVIATIONS

ARM	: Acorn RISC Machine
BMA	: Block Matching Algorithm
CLS	: Constrained Least Squares
DCT	: Discrete Cosine Transform
DDR RAM	: Double Data Rate Random Access Memory
DSP	: Digital Signal Processor
EM	: Expectation Maximization
FF	: Flip Flop
FIFO	: First In First Out
FPGA	: Field Programmable Gate Arrays
FT	: Fourier Transform
GPU	: Graphics Processing Units
HR	: High Resolution
IBP	: Iterative Back Projection
LMS	: Least Mean Square
LR	: Low Resolution
LSI	: Linear Shift Invariant
LUT	: Look Up Table
MAP	: Maximum A Posteriori
ML	: Maximum Likelihood
MLE	: Maximum Likelihood Estimation
MRF	: Markov Random Fields
MSE	: Mean Square Error
PM	: Projection Method
POCS	: Projection Onto Convex Sets
PSF	: Point Spread Function
PSNR	: Peak Signal to Noise Ratio

RAM	: Random Access Memory
RANSAC	: Random Sample Consensus
RLS	: Recursive Least Squares
SAD	: Sum of Absolute Differences
SD	: Steepest Descent
SNR	: Signal to Noise Ratio
SR	: Super Resolution
SRAM	: Synchronous Random Access Memory
VHDL	: Very High Speed Integrated Circuit Hardware Description Language
VLIW	: Very Long Instruction Word

LIST OF SYMBOLS

$o(x, y)$	: True scene intensity image
$x(n_1, n_2)$	: Desired high resolution image which is discrete version of
y_k	: *k*'th observed LR image
f	: Upsampling factor
L_1	: Horizontal resolution of LR images
L_2	: Vertical resolution of LR images
L_1	: Horizontal resolution of desired HR image
L_2	: Vertical resolution of desired HR image
D	: Downsampling matrix
H	: Blurring matrix
W_k	: Geometric warp matrix for *k*'th LR image
e_k	: Additive noise in the *k*'th LR image
$P'_Y(x)$	: Horizontal projection vector
$P'_D(y)$	: Vertical projection vector
$\overline{P'_Y}$	: Mean of horizontal projection vector
$\overline{P'_D}$	: Mean of vertical projection vector
$P_Y(x)$	: Normalized horizontal projection vector
$P_D(y)$	: Normalized vertical projection vector
S_{max}	: Maximum expected shift value for PM algorithm
$SAD_H(m)$	: Sum of absolute differences for horizontal projection vectors
$SAD_V(m)$	: Sum of absolute differences for vertical projection vectors
$\hat{x}^n$	: Estimated HR image at n'th iteration
λ	: Coefficient for data fidelity term
β	: Coefficient for regularization term

CHAPTER 1

INTRODUCTION

1.1 Motivation

At present, digital imaging has an important role in our life. It is desired to have better quality images in all applications of digital imaging. Better quality can be achieved by increasing spatial resolution. One way to increase resolution is to improve sensor manufacturing techniques [1]. However, it is an expensive solution and there are limits of this process. Super resolution is a signal processing approach which proposes a cheap solution to this problem. This approach produces high resolution images from low resolution images by using image processing techniques. Since existing low resolution imaging devices are still used, it is a feasible and cheap solution [2]. There are several areas which use super resolution approaches such as satellite imaging, astronomical imaging, video resolution enhancement, confocal microscopy and medical imaging.

In this study, video resolution enhancement for infrared cameras will be considered. Infrared cameras are widely used in several areas such as night vision, chemical imaging, flame detection, military imaging, target detection, etc. At present, this type of cameras produces lower quality images than other cameras because of manufacturing technology where the quality degradations are due to optics and detectors [3]. So, the image enhancement processes become more important issue for infrared cameras.. This study will focus on increasing resolution of these cameras on real time.

1.2 Aim and Scope of the Thesis

Generally, imaging systems with video cameras require real time video processing. So, real time implementation of video processing algorithms becomes an important issue. Although there are many super resolution algorithms proposed in the literature (See Section 1.5), most of them are not suitable for real time implementation. In this thesis, real time implementation of suitable super resolution algorithms for infrared cameras will be concerned. As a processing unit, Field Programmable Gate Arrays (FPGA) will be used.

1.3 Contribution

In this study, real time implementation of super resolution by using FPGA is concerned. FPGA structures of suitable super resolution approaches are proposed. Performances, resource usage, timing considerations of these approaches are discussed. For motion estimation stage, an integral

projection based method is implemented on FPGA. This method provides a cheap solution for translation estimation. On the other hand, FPGA structures for Projection onto Convex Sets (POCS), Maximum Likelihood Estimation (MLE), Maximum a Posteriori (MAP) and Least Mean Squares (LMS) algorithms are proposed for high resolution image reconstruction stage. Since LMS algorithm is found to be efficient in terms of performance, resource usage and timing considerations, this algorithm is selected to be implemented on FPGA.

1.4 Organization

The thesis consists of five chapters. Brief information and literature survey for super resolution technique are given in this chapter.

In Chapter 2, theoretical background of super resolution is given in details. Firstly, the image acquisition model is explained which forms an important basis for high resolution image reconstruction algorithms. Then, motion estimation algorithm used for global motion estimation is given. Lastly, high resolution image reconstruction algorithms POCS, MLE, MAP and LMS are explained in details.

In chapter 3, FPGA structure of real time super resolution is considered. Firstly, FPGA structure of motion estimation algorithm is proposed. Then, the common blocks designed for high resolution image reconstruction algorithms are explained. After that, FPGA structures of POCS, MLE, MAP and LMS algorithms are proposed by using these common blocks. Finally, memory management of the designs is discussed.

In chapter 4, high resolution image reconstruction algorithms are compared in terms of resource usage, timing considerations and resulting image quality. In the light of comparison, LMS algorithm is selected to be implemented. After that, performance drawbacks of FPGA implementation of LMS algorithm are discussed. Then, details of FPGA implementation and simulation results are given. Finally, resource usage for final FPGA implementation is mentioned.

In chapter 5, the study is concluded and possible future works are discussed.

1.5 Literature Survey

Super resolution algorithms require several low resolution images to reconstruct a high resolution image. Low resolution images are assumed to be different looks at the same scene with motion differences. The basic idea is to fuse low resolution images which consist of different information about the same scene. The motion differences between images are assumed to be at sub pixel level. Video sequence captured from a camera satisfies above properties. If a stationary scene is captured by a video camera, the frames of video sequence will be low resolution images which have different information about the same scene. The sub pixel motion differences between images are assumed to be due to vibrations of camera platform. For instance, when an infrared camera placed on an aircraft and a stationary far scene is observed, there will be sub pixel shifts between captured images due to vibration of the craft on the air which means the captured low resolution images will become useful data for super resolution reconstruction.

In order to reconstruct a high resolution image, firstly the low resolution images should be registered. This process compensates motion differences between low resolution images. Since

super resolution uses sub pixel information, the motion between low resolution images should be estimated with sub pixel accuracy by using a suitable motion estimation algorithm. After motion estimation, low resolution images are registered by using estimated motion information and registered images are used to reconstruct a high resolution image. Several works performed for registration and high resolution image reconstruction in the literature. The literature review is given below.

Registration is studied for many image processing applications. [4] and [5] are valuable literature surveys for registration algorithms. Nevertheless, there are several works performed considering registration for super resolution in the literature. These methods can be categorized into spatial and frequency domain approaches. A spatial domain image registration algorithm for super resolution in which the Taylor series expansion of spatial transformation between images is used in order to estimate translation and rotation is introduced in [6]. A feature based spatial domain approach for super resolution is discussed in [7]. Feature points are detected by Harris feature detector [8]. Putative correspondences are identified around the features and they are refined by RANSAC methodology. Maximum Likelihood approach is used to estimate the motion information by using refined correspondences.

Most of frequency domain motion estimation algorithms use the shift property of Fourier transform. The Fourier transform of two shifted images differ from each other by only phase shift which can be found by correlating them. This method is called as phase correlation. [9] and [10] describes such a method in order to estimate the motion information. A rotation estimation algorithm based on the magnitude of Fourier transforms of images is developed in [11]. After rotation estimation, translation motion is estimated by using standard phase correlation method. Standard phase correlation method is modified in [12] in order to extend the accuracy to sub pixel level. For downsampled images, it is shown that the signal power in phase correlation is not concentrated in a single peak, but rather several coherent peaks adjacent to each other. The sub pixel shift values are found by using these coherent peaks. The low frequency part of the spectra is used in [13] in order to avoid the effects of aliasing while estimating motion information. Rotation is estimated by using 1D correlation instead of 2D which results reduced computational load. After rotation estimation, slope of phase difference is used for translation estimation. A performance comparison of these registration algorithms for different motion models is given in [14].

Since real time implementation is considered in this thesis, the algorithm used to estimate motion has to be efficient in terms of computing and memory. Integral projection based methods used widely in video stabilization [15] and image compression [16] are attractive methods for this purpose. They only require 1D matching procedure which results in a reduced computational load and memory requirements. Integral projection based method is explained in details in [17]. The robustness of the algorithm to additive noise is also discussed in this paper. This method is extended to sub pixel accuracy in [18] which is necessary property for super resolution. The gradients of integral projections are used for motion estimation in [15]. It makes the algorithm robust for illumination changes. These approaches provide cheap solution for motion estimation which is charming for real time implementation. However, only translation is considered in these methods.

There are several high resolution image reconstruction algorithms in the literature. Similar to registration algorithms, HR image reconstruction algorithms can also be categorized into spatial domain and frequency domain approaches.

Frequency domain approaches use the relationship between Fourier transform of desired HR image and observed LR images. These approaches assume that aliasing exist in each LR images. The shift property of FT, aliasing in LR images and bandlimiteness of desired HR image are used to estimate desired HR image. First frequency domain approach for SR reconstruction is proposed in [19]. A relationship between Fourier transforms of HR image and observed LR images are

derived. HR image is estimated by solving the inverse problem. When enough frames are available, a unique solution is obtained. If this is not the case, a least-squares approximated solution is reached. An extension is performed for noisy and blurred images in [1]. It results weighted least squares formulation. In this approach, blur and noise parameters are assumed as same for all LR images. This method is extended to consider different blur parameters for LR images in [20]. Tikhonov regularization is used in order to overcome ill-posedness of the problem. A recursive total least square solution is proposed in [21]. This approach reduces the effects of registration errors. In order to reduce computational load, Discrete Cosine Transform (DCT) based approach is proposed in [22]. Multichannel adaptive regularization parameters are used to overcome ill-posedness of problem. Frequency domain approaches have simple theoretical background in which the relationship between LR images and desired HR image can be demonstrated clearly [2]. On the other hand, the motion model is restricted to translation and blur model is Linear Shift Invariant (LSI). A prior knowledge is also difficult to apply for regularization.

Most basic approach in spatial domain for HR image reconstruction is non-uniform interpolation approach. This approach uses relative sub pixel motion information for LR images to obtain non-uniformly spaced sampling points. After that, a direct or iterative process is used in order to obtain uniformly spaced sampling points on HR grid [23, 24 and 25]. After interpolation stage, a restoration stage that removes blur and noise is applied. An interpolation-restoration HR image reconstruction approach for infrared cameras is introduced in [26]. Weighted nearest neighbor interpolation is used to obtain uniform sampling points. After that, wiener filtering is applied to reduce the effects of blurring and noise. A wavelet based HR image reconstruction algorithm is proposed in [27]. A computationally efficient wavelet interpolation method is explained for two dimensional data. The computational load for non-uniform interpolation methods is low which makes it suitable for real time applications. But it is applicable when the blur and noise parameters are same for all LR images. Additionally, since restoration step ignores the error in interpolation step, the optimality of the solution is not guaranteed.

Another spatial domain approach is regularized SR reconstruction. Because of insufficient number of LR images and ill-conditioned blur operators, SR reconstruction is an ill-posed problem. Regularization stabilizes the solution of ill-posed problems. A multichannel regularized SR approach is given in [28, 29]. Generalized multichannel deconvolution method including multichannel regularized SR approach is proposed in [30]. SR reconstruction approach by minimizing regularized cost functional for infrared cameras is proposed in [3]. Steepest descent and conjugate gradient optimization procedures are used in order to minimize the cost function. Additionally, PSF estimation for infrared cameras is described by using observation model, optic and detector specifications. Constrained Least Square (CLS) based SR reconstruction is proposed in [31]. L-curve method is used in order to find the optimal value of regularization parameter of CLS solution.

Stochastic approach can also be used for SR reconstruction. It provides to incorporate a priori knowledge about the solution. MAP (Maximum a Posteriori) estimator is used in order to find the solution. Maximum Likelihood (ML) is a special case of MAP. It has no prior term. SR problem is ill-posed because of insufficient number of observed LR images and ill-conditioned blur operators [2]. Since MAP gives stabilized solutions, it is mostly preferred for SR problem. This approach also can provide to find registration parameters and high resolution image simultaneously. A ML approach which estimates the sub pixel shifts, noise variance and HR image simultaneously is proposed in [32]. Expectation maximization (EM) is used in order to solve ML estimation problem. SR reconstruction for video sequences using MAP estimator is described in [33]. Huber-Markov Gibbs prior model is used for prior information. A MAP solution for joint estimation of motion parameters and HR image is proposed in [34]. Gaussian prior model is used for regularization. Bayesian estimation with Gaussian prior model is also used for integrating satellite

images in [35]. Joint estimation of motion vectors and high resolution image is proposed for compressed videos in [36]. Object segmentation, motion field and high resolution image are jointly estimated by MAP approach in [37]. Multiple moving objects are considered in this work. The number of moving objects needs to be known. Recently, MAP formulation is solved by using variational methods in [38]. Optical flow and high resolution image are estimated simultaneously by performing energy minimization via Euler-Lagrange equations. Gauss-Seidel method is used in order to reach the solution iteratively. Main advantages of stochastic approaches are robustness, flexibility in modeling noise characteristics and prior knowledge concerning the solution. Efficient optimization procedures can be used to reach the solution iteratively.

Another iterative approach for solving SR problem is Projection onto Convex Sets (POCS). It uses prior knowledge about the solution to solve the interpolation and restoration problem simultaneously. POCS formulation is first proposed in [39]. Observation noise is considered for POCS formulation in [40]. Space varying blur, nonzero aperture time, nonzero physical dimension of detectors, sensor noise and arbitrary sampling lattices are considered for POCS solution in [41]. The method is extended to consider multiple moving objects by validity map and segmentation map in [42]. Validity map provides robustness when registration errors exist and segmentation map provides object based SR reconstruction. In [43], image acquisition model is improved to allow higher order interpolation methods and a POCS-based method is proposed which has adaptive constraint sets to reduce the ringing artifact in the edges. A set theoretic regularization approach is proposed in [44]. Ellipsoidal constraint sets are used to find the solution. POCS methods are simple and allow incorporating prior information. But the computational load is high and solution is not unique. Also the convergence to the solution is slow.

Iterative back projection is another iterative approach for SR reconstruction. Firstly, it is formulated in [45]. The approach uses the difference between simulated LR images through acquisition model and observed LR images to find solution iteratively. In [46], the approach is extended to consider perspective motion model. After that, more general motion model is adapted to the algorithm in [47]. The advantage of this approach is that it can be understood easily. But the solution is not unique because of ill-posed nature of the problem.

Adaptive filtering theory is a suitable approach for video sequences. It considers the dependence of LR frames in time. Firstly, it is proposed in [48]. Least squares (LS) estimators based on Recursive Least Squares (RLS) and Least Mean Squares (LMS) algorithms are proposed. The steepest descent (SD) and normalized steepest descent optimization techniques are used to reach final HR image. This method is adopted for the case of translational motion and common space-invariant blur which is computationally less complex in [49]. Classic mean-covariance approach is used to interpolate LR images to high resolution grid. After interpolation, deblurring is applied by using MAP approach with bilateral filter [30].

The literature that is related to real time implementation of super resolution is limited comparatively. SR is implemented by using VLIW (Very Long Instruction Words) and ARM processors in [50]. A method similar to non-uniform interpolation is implemented in this work. Existing motion estimation information from another block is used. FPGA implementation of SR is proposed in [51]. Non-uniform interpolation with weighted mean approach is used for initial estimate of HR image. After that, bilateral total variation is used to remove the blur as in [30]. Motion estimation is not considered in their work. Another FPGA implementation which uses IBP algorithm is proposed in [52]. IBP algorithm is modified in order to account for the presence of noise in [53] and FPGA implementation without motion estimation is proposed.

CHAPTER 2

SUPER RESOLUTION BACKGROUND

In this chapter, the theoretical background for super resolution approach is given. The block diagram of the flow is shown in Figure 1. In the figure, image acquisition corresponds to the process that captures real word scene and produces LR images. An acquisition model for infrared camera is assumed for this process. It is described in Section 2.1. Observed LR images are used to reconstruct HR image. This process is referred as SR Image Reconstruction. It consists of two parts. Motion estimation is the first part that calculates motion information between LR images. This information is used to register LR images in HR image reconstruction part. Motion estimation algorithm used in this study is explained in Section 2.3. Finally, high resolution images reconstruction approaches are introduced in Section 2.4. POCS, MLE, MAP and Adaptive Filtering approaches are given in details.

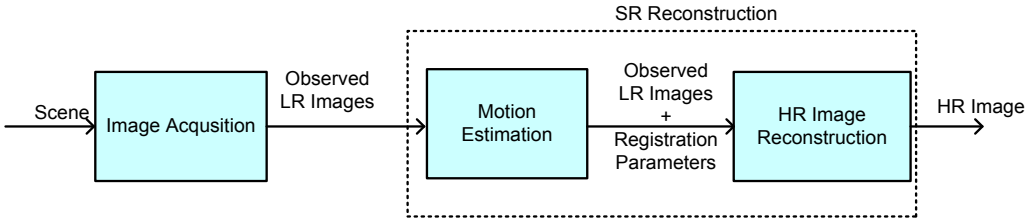


Figure 1 Super Resolution Block Diagram

2.1 Image Acquisition Model

Super resolution is an inverse problem. So, the forward model that relates desired solution to the observations forms important basis for reconstruction algorithms. This model is called as image acquisition model [3]. Block diagram of the model is shown in Figure 2.

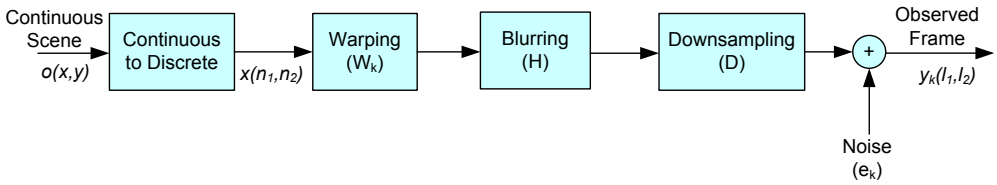


Figure 2 Image Acquisition Model

According to the model, the formulation for *k'*th observed LR image is given as

$$y_k = DHW_k x + e_k \qquad k = 1, \dots, p \tag{1}$$

The sizes of the matrices considered are as follows:

x : $N_1 N_2 x 1$
y_k : $L_1 L_2 x 1$
W_k : $N_1 N_2 x N_1 N_2$
H : $N_1 N_2 x N_1 N_2$
D : $L_1 L_2 x N_1 N_2$
e_k : $L_1 L_2 x 1$

where $N_1 = fL_1$ and $N_2 = fL_2$; f is resolution upsampling factor for both axes.

The true scene intensity image is defined as $o(x, y)$ which is assumed to be bandlimited. $x(n_1, n_2)$ is discrete version of $o(x, y)$ sampled at or above the Nyquist rate. It is the desired high resolution image of size $N_1 x N_2$ that SR algorithms aim to reach. It is assumed p different observed LR images of size $L_1 x L_2$ are produced from single desired HR image according to the model. The differences between LR images occur due to warping, blurring and downsampling operations.

Warping (W_k): It consists the motion occurring while acquisition of LR images. It may contain global or local motion between captured images. For a stationary scene in the far field, the local motion can be ignored and the motion model can be restricted to global translation and rotation for video sequences. In this thesis, only translation implementation is considered and rotation implementation is decided to be future work.

Blurring (H): It may occur due to optics, finiteness of the size of detectors and motion blurring. Low quality optics degrades the imaging quality. The detector structure also forms a point spread effect. These effects are examined for infrared cameras and a theoretical point spread function which consists of the effects of optics and detectors is estimated in [3]. This estimation procedure is used and a LSI blur function is found by using the specifications of infrared camera used in the tests. Motion blurring is ignored.

Dowsampling (D): It is a matrix that decimates warped and blurred high resolution images to form LR images. The decimation factor may be different for each LR images used for reconstructing high resolution images for some applications. But, the factor is generally same for video SR applications.

Noise: According to the model, an additive noise is added to decimated LR images in order to form observed LR images. Zero mean Gaussian noise model is sufficiently accurate for infrared cameras [3].

2.2 Motion Estimation

In order to reconstruct high resolution images, input low resolution images should be registered. Registration requires knowledge of motion information between LR images. The accuracy of motion estimation should be at sub-pixel level since SR algorithms assume that observed LR images are shifted with sub-pixel precision. Several works are performed for SR in order to estimate motion as explained in Section 1.1. Since real time implementation in FPGA is considered in this thesis, a suitable motion estimation algorithm for hardware architecture should

be selected. A method based on integral projections is proposed for global motion estimation in [54]. Integral projections based methods [54, 17] are also cheap solutions for translation estimation and suitable for FPGA implementations. The motion model is restricted to only translation which is sufficient for our application as explained in Section 2.1. So, in this study, motion estimation will be performed by using this method which is named as projection method (PM).

The method is based on the projections of the images' intensity values on horizontal and vertical axes [17]. Horizontal and vertical projection vectors are formed by projecting the pixel values. Instead of using all pixel values in the original frame, the pixels in a frame whose size is smaller than original frame size is used. So, the possible artifacts on the boundaries are avoided. In addition, the projection vectors are upsampled so that the sub-pixel motion estimation can be performed.

The projection vectors are calculated for an MxN image $F(x,y)$ by using following formulas.

$$P_H'(x) = \sum_{y=1}^{M} F(x,y)\,, for\ x = 1, \dots, N \tag{2}$$

$$P_V'(y) = \sum_{x=1}^{N} F(x,y)\,, for\ y = 1, \dots, M \tag{3}$$

where $P_H'(x)$ is horizontal projection vector and $P_V'(y)$ is vertical projection vector. The calculations of the horizontal and vertical projection vectors are demonstrated in Figure 3 and Figure 4.

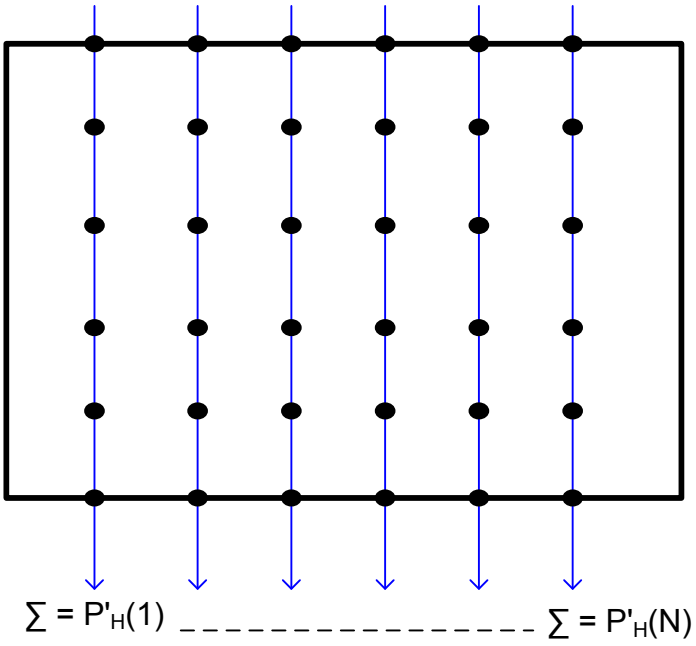


Figure 3 The calculation of horizantal projection vector

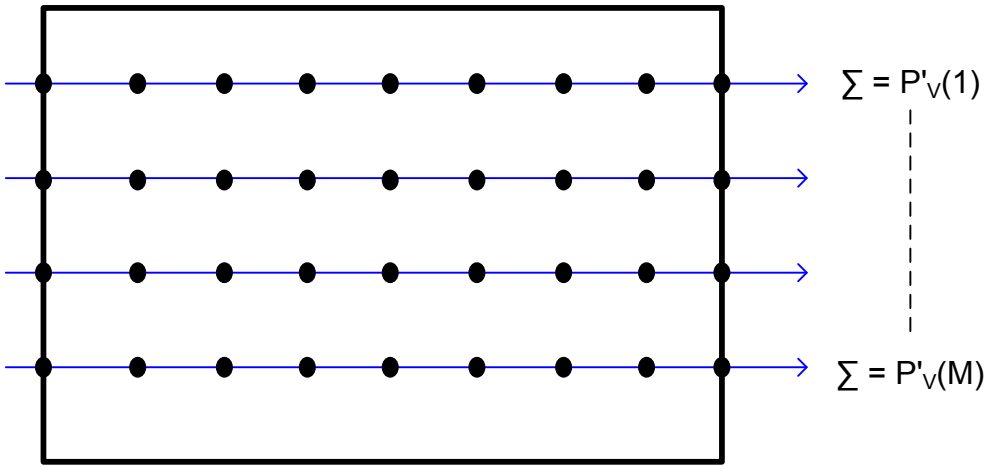


Figure 4 The calculation of vertical projection vector

In order to normalize the projection values, means of the vectors are subtracted from corresponding projection vectors. This process makes algorithm robust when the intensity bias exists between the images. The means and the normalized projection vectors are calculated using the following formulas.

$$\overline{P_H'} = \sum_{x=1}^{N} P_H'(x)\,/N \tag{4}$$

$$\overline{P_V'} = \sum_{y=1}^{M} P_V'(y)\,/M \tag{5}$$

$$P_H(x) = P_H'(x) - \overline{P_H'} \tag{6}$$

$$P_V(y) = P_V'(y) - \overline{P_V'} \tag{7}$$

In order to find shift values on x and y directions between the reference and current images, firstly the projection vectors are calculated using above formulas for both images. Then the optimum shift values are decided considering the sum of absolute differences between the projection vectors while shifting the projection vectors of current images over the projection vectors of reference image. A maximum shift value is set as a system parameter (S_{max}) which is decided with respect to maximum expected shift values.

The calculation of the shift values on x and y directions are performed independently from each other. The shift value on y direction is calculated by using similar way to the shift value on x direction which is calculated using following steps:

1. The first and last S_{max} values of the projection vector of current image are ignored. So the length of the vector will be shorter by amount of 2*S_{max} as shown in Figure 5.

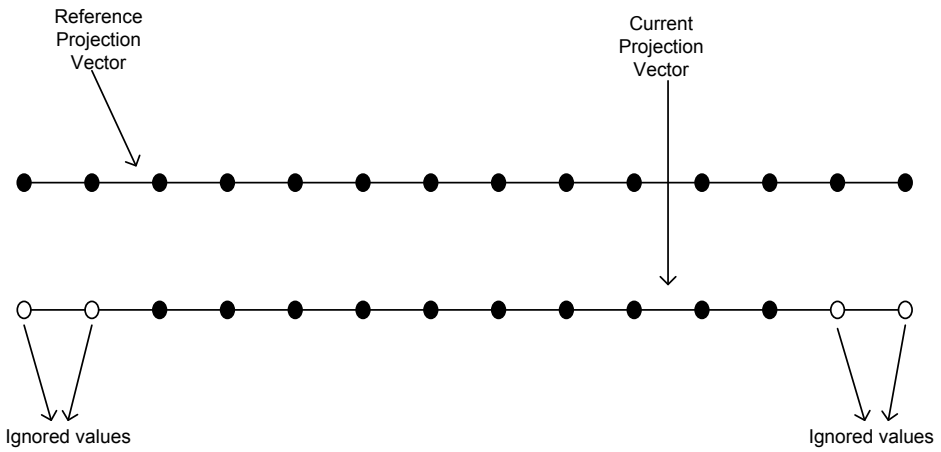


Figure 5 The values ignored (empty circles) on horizontal projection vector of current frame

2. The projection vector of current image is shifted over the projection vector of reference frame and sum of absolute differences are calculated as shown in Figure 6 using following formula.

$$SAD_H(m) = \sum_{k=S_{max}}^{N-S_{max}} \left| P_{H,ref}(k+m) - P_{H,cur}(k) \right|, \quad m = -S_{max}, \dots, S_{max} \tag{8}$$

where $P_{H,ref}$ and $P_{H,cur}$ are the horizontal projection vectors of reference and current images respectively.

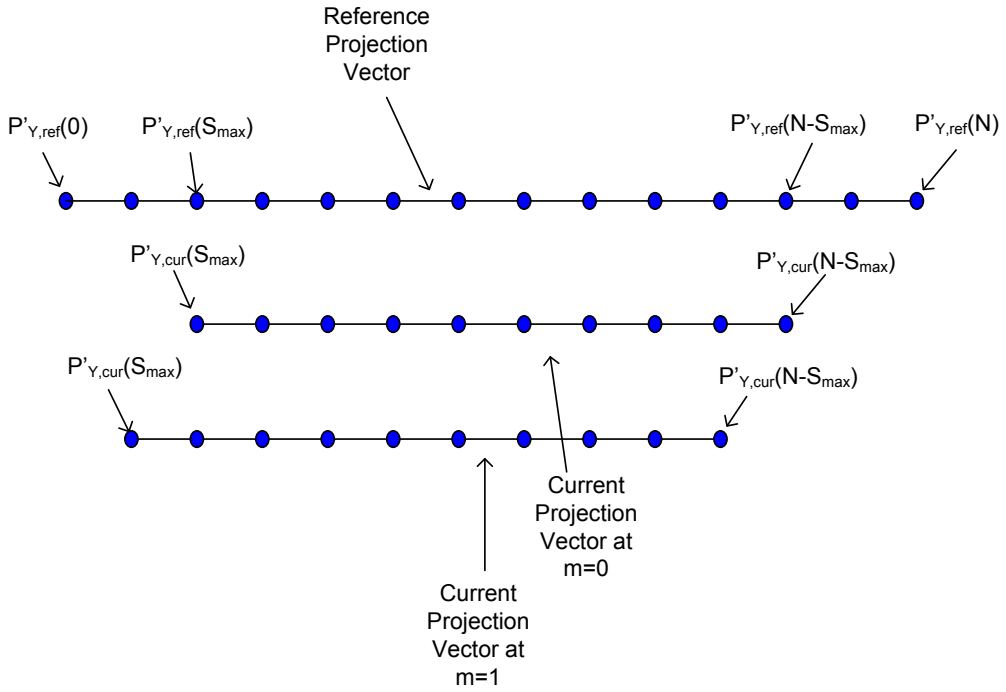


Figure 6 The positions of horizontal projection vectors of reference and current frame with respect to different shift values

3. 2*S_{max}+1 SAD values are calculated using step 2 for $m = -S_{max}, \dots, S_{max}$. The index that corresponds to minimum SAD value is predicted shift value.

$$S_H = argmin\ SAD_H(m) \tag{9}$$

Instead of using all pixel values in the original frame while calculating the projection vectors, a limited box is used in order to make algorithm more robust as shown in Figure 7. Effects of artifacts on boundaries of the images are extracted by this way.

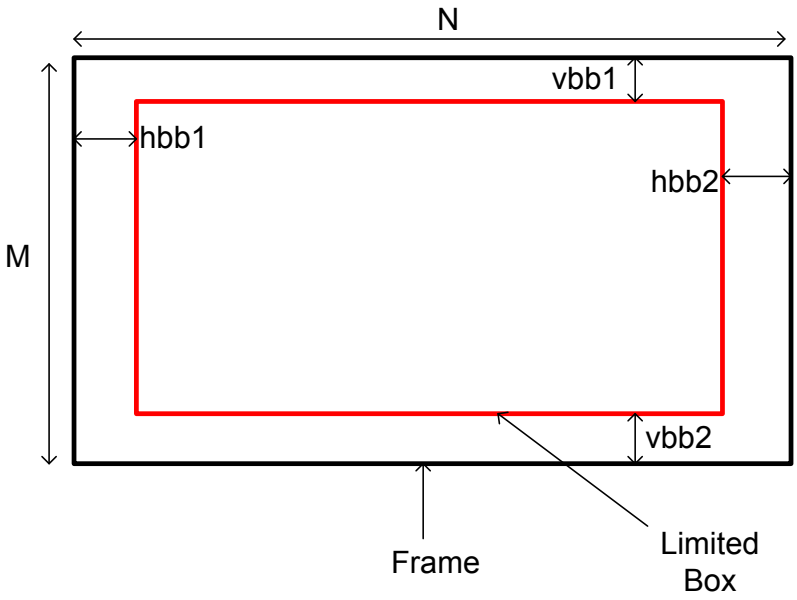


Figure 7 Original frame and limited box

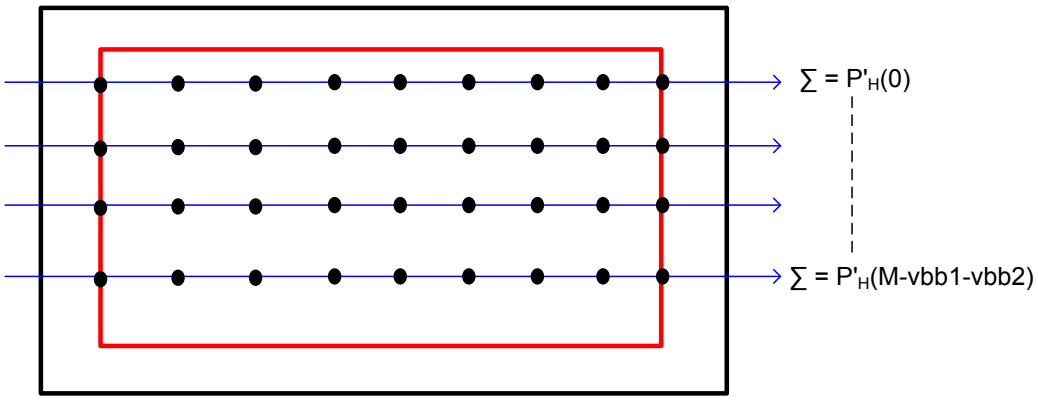


Figure 8 Horizontal projections when limited box is used

$Hbb1$, $hbb2$, $vbb1$ and $vbb2$ are parameters of the limited box. The formulas used in the calculation of projection vectors are changed as:

$$P_H^{'}(x) = \sum_{y=vbb1}^{M-vbb2} F(x,y)\,, for\ x = 1, \dots, N - hbb1 - hbb2 \tag{10}$$

$$P_V^{'}(y) = \sum_{x=hbb1}^{N-hbb2} F(x,y)\,, for\ y = 1, \dots, M - vbb1 - vbb2 \tag{11}$$

$$\overline{P_H'} = \sum_{x=1}^{N-hbb1-hbb2} P_H'(x)\,/(N - hbb1 - hbb2) \quad (12)$$

$$\overline{P_V'} = \sum_{y=vbb1}^{M-vbb1-vbb2} P_V'(y)\,/(M - vbb1 - vbb2) \quad (13)$$

The formulas for normalized projection vectors $P_H(x)$ and $P_V(y)$ are the same as before (Equation (6) and Equation (7)).

High resolution image reconstruction algorithms need sub pixel motion estimation. The projection method is extended to the sub pixel precision with some modifications [18]. The projection vectors of reference and current images are upsampled using bilinear interpolation [54] as shown in Figure 9. In the figure, upsampling factor is chosen as K=4 which provide a 0.25 sub-pixel accuracy. Bicubic interpolation [54] can be used for better results but it will increase computational load.

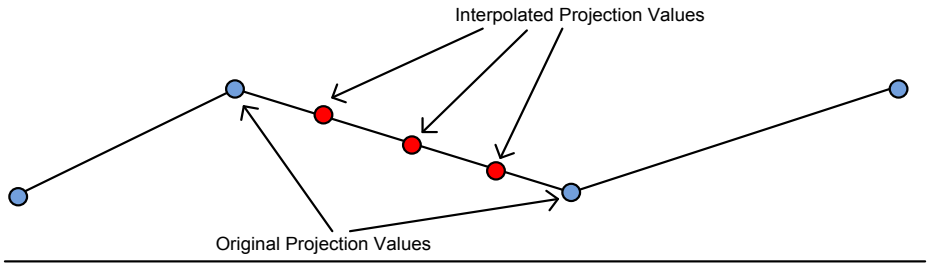


Figure 9 Upsampled projection vector with factor K=4

If it is assumed that the sizes of P_H and P_V before upsampling are N_P and M_P respectively, the size of resulting projection vectors ($N_{P,sp}$ and $M_{P,sp}$) after upsampling by factor K will be

$$M_{P,sp} = (M_P - 1) * K + 1 \quad (14)$$

$$N_{P,sp} = (N_P - 1) * K + 1 \quad (15)$$

The resulting larger projection vectors are used for SAD calculations and the shift values are estimated with sub pixel accuracy by this way.

2.3 High Resolution Image Reconstruction

The process that produces high resolution images from observed LR images and motion information is called as HR image reconstruction. Several algorithms are proposed for this process in the literature. Most well known HR image reconstruction algorithms are POCS, MLE, MAP and adaptive filtering approaches. Generally, more complex algorithms use these algorithms as a basis. Since real time implementation is considered in this thesis, these basis algorithms are considered. The theoretical backgrounds for the algorithms are explained in the following sections.

2.3.1 Reconstruction with Projection onto Convex Sets

Projection onto convex sets (POCS) algorithm solves the problem of super resolution image reconstruction by using closed convex constraint sets. The solution will be at intersection of these predefined constraint sets [39, 41]. The algorithm reaches to solution by successively projecting an initial estimate to the constraint sets.

Projection operators are used to map a point to the closest point within the associated sets. The order of the projections onto the sets is important. The final solution will depend on this order. If total of *N* projection operators are applied to an arbitrary point x^n, the mapped point x^{n+1} is given as

$$x^{n+1} = P_N P_{N-1} \dots P_2 P_1 x^n \tag{16}$$

where P_i is the projection operator associated with related constraint set and *n* is the iteration number. Data consistency constraint sets are the most commonly used convex sets. They provide consistency with observed low resolution images. These constraint sets are defined for all pixels within each observed LR images as

$$C_D^k[l_1, l_2] = \{x(n_1, n_2) : \left|r^{(x)}(l_1, l_2)\right| \leq T_k(l_1, l_2)\} \tag{17}$$

where the residual is

$$r^{(x)}(l_1, l_2) = y_k(l_1, l_2) - \sum_{n_1, n_2} x(n_1, n_2) h_k(n_1, n_2; l_1, l_2) \tag{18}$$

$T_k(l_1, l_2)$ is a bound that reflects statistical confidence. It is generally selected with respect to noise statistics in observed LR images. (n_1, n_2) is high resolution grid coordinates and (l_1, l_2) is low resolution grid coordinates. h_k is PSF that maps associated motion compensated pixels in HR grid to a single pixel in LR grid. The projection operator which projects an arbitrary point $x(n_1, n_2)$ onto $C_D^k[l_1, l_2]$ is defined as

$$P_k[x(n_1, n_2)] \tag{19}$$
$$= x(n_1, n_2) + \begin{cases} \dfrac{(r^x(l_1, l_2) - T_k(l_1, l_2)) h_k(n_1, n_2; l_1, l_2)}{\sum_{o_1, o_2} {h_k}^2(o_1, o_2; l_1, l_2)}, r^x(l_1, l_2) > T_k(l_1, l_2) \\ 0, \quad -T_k(l_1, l_2) \leq r^x(l_1, l_2) \leq T_k(l_1, l_2) \\ \dfrac{(r^x(l_1, l_2) + T_k(l_1, l_2)) h_k(n_1, n_2; l_1, l_2)}{\sum_{o_1, o_2} {h_k}^2(o_1, o_2; l_1, l_2)}, r^x(l_1, l_2) < -T_k(l_1, l_2) \end{cases}$$

In order to improve results, additional constraints such as amplitude constraint can be used [41]. This constraint set is defined as

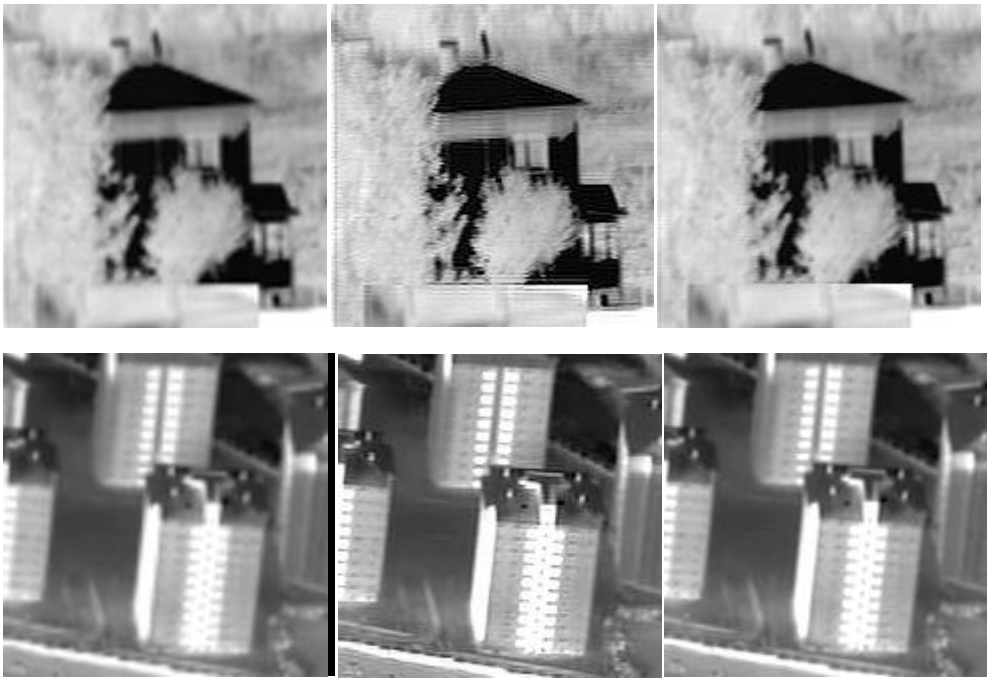
$$C_A[n_1, n_2] = \{x(n_1, n_2) : \alpha \leq x(n_1, n_2) \leq \beta\} \tag{20}$$

Projection operator which projects a point to C_A is defined as clipping operator. It limits the value of $x(n_1, n_2)$ between α and β. For single update, total projection operators in the case of amplitude and data consistency constraints used are given as

$$x^{n+1} = P_A P_N \dots P_2 P_1 x^n \tag{21}$$

where *N* is total number of pixels within all observed LR images. After these projections are performed consecutively, the update is completed. Several updates are required in order to reach the final solution. If the intersection set of constraint sets are non-empty, the solution will converge to a point in this set. Since the intersection sets may contain many points, there may be several solutions. So, the solution may not be unique for this approach [41].

As an example, the images at left column of Figure 10 are resulting images of bicubic interpolation which are upsampled by factor two on *x* and *y* axis. On the other hand, the images at middle column are resulting images of POCS algorithm with same upsampling factor. Four LR images for POCS are used in order to reconstruct HR image. The shift values between these images are generated with 0.5 pixel precision. The images at right column correspond to the original ground truth images.



(a) (b) (c)

Figure 10 Images Reconstructed by a) Bicubic Interpolation b) POCS Method c) HR Image

2.3.2 Reconstruction with Maximum Likelihood Estimation

The low resolution images are assumed to be degraded observed images of a high resolution image. The degradations are due to geometric warping, optical blur, downsampling and observation noise as explained in Section 2.1. The aim is to find the high resolution image from these observed LR images. This reconstruction method produces simulated LR images via image acquisition model from temporal solution and tries to minimize error between simulated and observed LR images using MLE [7].

Image acquisition process is formulated in Equation (1) given in Section 2.1. Linear operators D, H and W_k in the equation is combined into a single matrix A_k as

$$y_k = A_k x + e_k \qquad k = 1, \dots, p \tag{22}$$

where p is the number of observed LR images. The equations for each observed LR images are combined into a single equation as

$$\begin{bmatrix} y_1 \\ \vdots \\ y_p \end{bmatrix} = \begin{bmatrix} A_1 \\ \vdots \\ A_p \end{bmatrix} x + \begin{bmatrix} e_1 \\ \vdots \\ e_p \end{bmatrix} \tag{23}$$

Let $\underline{y} = \begin{bmatrix} y_1 \\ \vdots \\ y_p \end{bmatrix}$, $\underline{A} = \begin{bmatrix} A_1 \\ \vdots \\ A_p \end{bmatrix}$ and $\underline{e} = \begin{bmatrix} e_1 \\ \vdots \\ e_p \end{bmatrix}$ then

$$\underline{y} = \underline{A}x + \underline{e} \tag{24}$$

Given observed LR images y_k and imaging matrix A_k, the solution for high resolution image x is found via maximum likelihood estimation. It is assumed that the observation noise is Gaussian with zero mean and variance ${\sigma_k}^2$ [3]. Then, the probability of an observed LR image given estimate of high resolution image is

$$\Pr(y_k|\hat{x}) = \frac{1}{\sigma_k\sqrt{2\pi}} \exp\left(-\frac{\|\hat{y}_k - y_k\|^2}{2{\sigma_k}^2}\right) \tag{25}$$

where $\hat{x}$ is estimate of HR image and $\hat{y}_k$ is simulated LR image that is given by $\hat{y}_k = A_k\hat{x}$. The log-likelihood function is obtained by considering the logarithm of $\Pr(y_k|\hat{x})$ and given by

$$\mathrm{L}(y_k) = -\|\hat{y}_k - y_k\|^2 = -\|A_k\hat{x} - y_k\|^2 \tag{26}$$

The constant σ_k is dropped since it is unnecessary during calculation of maximum of log-likelihood function. It is assumed that the noises in observed LR images are uncorrelated [55]. So, log-likelihood function for observed LR images is given by

$$\sum_{\forall \mathrm{k}} \mathrm{L}(y_k) = -\sum_{\forall \mathrm{k}} \|A_k\hat{x} - y_k\|^2 = -\left\|\underline{A}\hat{x} - \underline{y}\right\|^2 \tag{27}$$

Maximum likelihood estimate is found by maximizing log-likelihood function as

$$\hat{x} = argmax_x\left(-\left\|\underline{A}x - \underline{y}\right\|^2\right) = argmin_x \left\|\underline{A}x - \underline{y}\right\|^2 \tag{28}$$

This problem is a linear minimization problem. The solution is obtained by using Moore-Penrose pseudo inverse [7] as below

$$\hat{x} = (\underline{A}^T\underline{A})^{-1}\underline{A}^T\,\underline{y} \tag{29}$$

$\underline{A}$ is a very large sparse matrix of size $pL_1L_2xN_1N_2$. It is not possible to directly take the inverse of this matrix in practice. So, iterative solutions are considered. Steepest descent, normalized steepest descent or conjugate gradient methods can be used in order to reach the solution. Steepest descent solution is given below. The cost function that is to be minimized is given as

$$E(x) = \left\|\underline{A}x - \underline{y}\right\|^2 \tag{30}$$

The gradient of the cost function with respect to x is

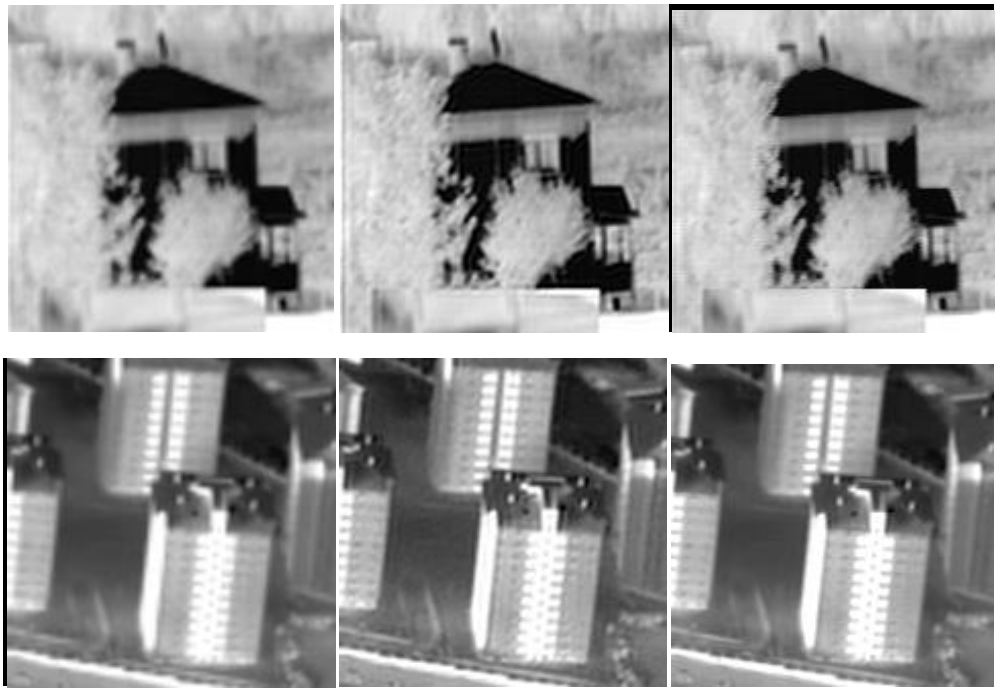
$$\nabla E = 2\underline{A}^T\left(\underline{A}x - \underline{y}\right) = 2\sum_{k=1}^{p} W_k^T H_k^T D^T(DH_kW_kx - y_k) \tag{31}$$

Then, the steepest descent solution of the problem is given as

$$\hat{x}^{n+1} = \hat{x}^n - \lambda\nabla E = \hat{x}^n - \lambda\sum_{k=1}^{p} W_k^T H_k^T D^T(DH_kW_k\hat{x}^n - y_k) \tag{32}$$

where λ is step size. The matrix operations in the equation correspond to image operations. The matrix D^T is implemented by upsampling the image by using an interpolation method which can be nearest, bilinear or bicubic interpolation. H_k^T is implemented by forward projection of intensity values using the weights of transpose of original blur filter. On the other hand, W_k^T is implemented by warping the image with inverse motion information. Figure 11 shows the results of MLE algorithm. The images at left column are resulting images of single image bicubic interpolation when upsampling factor is selected as 2 for both axes. On the other hand, the images at middle column are resulting images of MLE approach with same upsampling factor. Four for MLE LR

images are used in order to reconstruct HR image. The shift values between these images are generated with 0.5 pixel precision.



(a) (b) (c)

Figure 11 Images Reconstructed by a) Bicubic Interpolation b) MLE Method c) HR Image

2.3.3 Reconstruction with Maximum a Posteriori

Since super resolution is an ill-posed problem, the solution is not stable. Small amounts of noise in observed LR images may result large perturbations in the solution. This is an inevitable occurrence in inverse problems. In order to find a stable solution, regularization is considered. Regularization also helps to remove the artifacts from the solution. It also improves the rate of convergence [30].

Regularization term corresponds to a priori information about high resolution image solution. The estimate of high resolution image is computed given the observed LR images and a priori information. Bayes' theorem is used for this purpose. Posterior probability function is given as

$$\Pr(\hat{x}|\underline{y}) = \frac{\Pr(\underline{y}|\hat{x})\Pr(\hat{x})}{\Pr(\underline{y})} \tag{33}$$

where $\Pr(\underline{y}|\hat{x})$ is given in Equation (27) in Section 2.3.2. $\Pr(\hat{x})$ is prior information about the solution. Since the term $\Pr(\underline{y})$ does not depend on x, it has no role in the optimization. Then, the maximum a posteriori solution is given as below

$$\begin{aligned}\hat{x} &= argmax_x\ (\lg \Pr(x) + \lg \Pr\left(\underline{y}\middle|\hat{x}\right)) \\ &= argmax_x\ (\lg \Pr(x) - \left\|\underline{A}x - \underline{y}\right\|^2) \\ &= argmin_{\ x}(\left\|\underline{A}x - \underline{y}\right\|^2 - \lg \Pr(x))\end{aligned} \tag{34}$$

The cost function that is to be minimized is given below

$$\begin{aligned}E(x) &= \left\|\underline{A}x - \underline{y}\right\|^2 - \lg \Pr(x) \\ &= \left\|\underline{A}x - \underline{y}\right\|^2 - \mathrm{Y}(x)\end{aligned} \tag{35}$$

First term corresponds to the data fidelity term and second term corresponds to the regularization term. The most common functions used for regularization terms are potential functions which are quadratic in the pixel values [7] as given below.

$$\Pr(x) = \frac{1}{Z} \exp\left(-x^T Q x\right) \tag{36}$$

where Q is a symmetric and positive-definite matrix. Then, regularization term is written as

$$\mathrm{Y}(x) = \lg \frac{1}{Z} \exp(-x^T Q x) = -\frac{1}{Z}\, x^T Q x \tag{37}$$

and the cost function becomes

$$E(x) = \left\|\underline{A}x - \underline{y}\right\|^2 + \frac{1}{Z}\, x^T Q x \tag{38}$$

The simplest matrix for Q which satisfies the condition is identity matrix [7]. When identity matrix is used, the cost function becomes

$$E(x) = \left\|\underline{A}x - \underline{y}\right\|^2 + \rho \left\|x\right\|^2 \tag{39}$$

Another matrix Q which satisfies the symmetric and positive-definite property is derived from a linear operator T which is an image filter applied to x.

$$E(x) = \left\|\underline{A}x - \underline{y}\right\|^2 + \rho \left\|\mathrm{T}x\right\|^2 \tag{40}$$

where Q is replaced with $\mathrm{T}^{\mathrm{T}}\mathrm{T}$. This type of regularization is known as Tikhonov regularization [56]. T can be a high pass operator such as derivative, Laplacian or identity matrix. The main purpose of the regularization is to limit total energy of the final image or forcing spatial smoothness. The resulting image when regularization is used will be denoised and will not contain sharp edges [57]. Different regularization methods can be used. Total Variation is a regularization method which preserves edges in the reconstructed image [58]. Another regularization method is bilateral filter which is derived from Total Variation [30]. The computational load is low and edges are preserved in this method. Gaussian Markov Random Fields (Gaussian MRFs) [35] and Huber MRFs [33] are also used for regularization.

In this thesis, Tikhonov regularization is used since the implementation of this regularization type requires only image filtering operations which are already necessary for PSF blurring operations. So, an image filter can be designed and used for both operations which provide simplicity during FPGA implementation. When Tikhonov regularization is used, the gradient of cost function with respect to x will be

$$\nabla E = 2\underline{A}^T\left(\underline{A}x - \underline{y}\right) + 2\rho \mathrm{T}^{\mathrm{T}}\mathrm{T}x \tag{41}$$

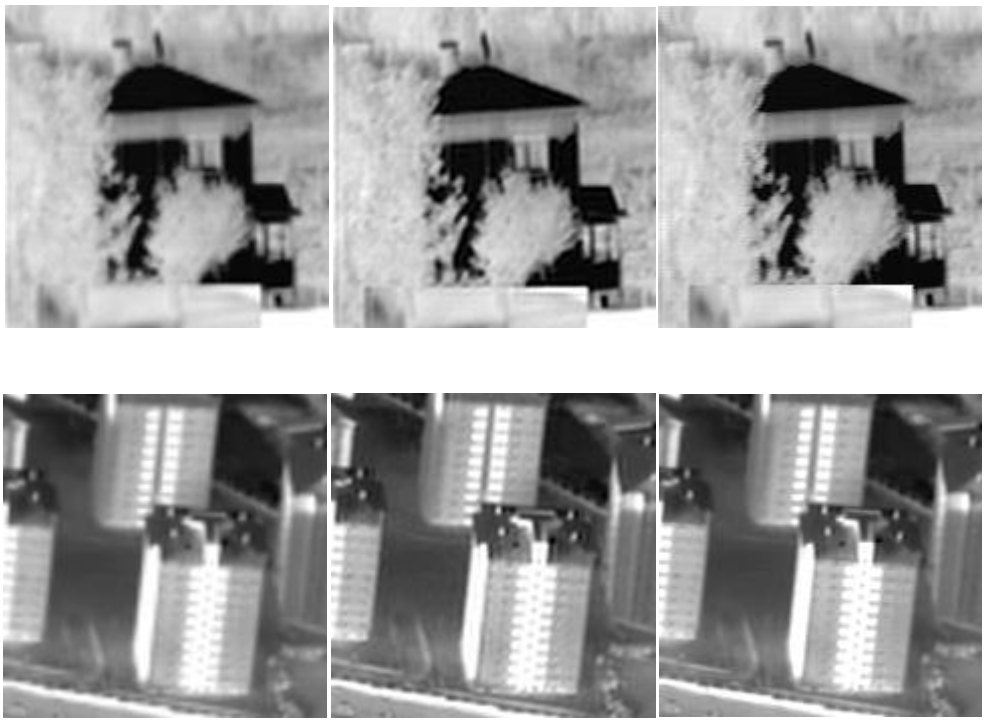
$$= 2\sum_{k=1}^{p} W_k^T H_k^T D^T (D H_k W_k x - y_k) + 2\rho \mathrm{T}^{\mathrm{T}} \mathrm{T} x$$

In this case, the steepest descent solution for MAP will be given as

$$\hat{x}^{n+1} = \hat{x}^n - \lambda \nabla E \tag{42}$$
$$= \hat{x}^n - \lambda \sum_{k=1}^{p} W_k^T H_k^T D^T (D H_k W_k \hat{x}^n - y_k) - \beta \mathrm{T}^{\mathrm{T}} \mathrm{T} x$$

where λ is coefficient of data fidelity term and β is coefficient of regularization term.

The images at left column of Figure 12 correspond to the resulting images of single image bicubic interpolation. The images at middle column show the results of MAP algorithm. Upsampling factor is selected as 2 for both methods. Four LR images for MAP are used in order to reconstruct HR image. The shift values between these images are generated with 0.5 pixel precision.



(a) (b) (c)

Figure 12 Images Reconstructed by a) Bicubic Interpolation b) MAP Method c) HR Image

2.3.4 Reconstruction with Adaptive Filtering Approach

It is shown that adaptive filtering theory can also be used in order to solve super resolution problem in [48]. This theory considers the continuous flow of images in time which makes it attractive for video applications. It can be implemented using both LMS (Least Mean Square) and RLS (Recursive Least Squares) algorithms. The derivations of both algorithms are given below.

A low resolution image sequence $\{y(t)\}_{t\geq 0}$ of size [M x M] is considered. The high resolution image sequence $\{x(t)\}_{t\geq 0}$ of size [NxN] is related to the LR image sequence via the relationship

$$y(t) = DHx(t) + U(t) \tag{43}$$

This equation is similar to Equation (1), with indexes replaced by temporal correspondence and warp matrix is omitted. Downsample matrix D is assumed to be constant for all LR images and H is the linear space and time invariant blur operator as in Equation (1). $U(t)$ is measurement error and it is assumed to be random noise [48].

The ideal image sequence in time is given by

$$x(t-i) = W(t,i)x(t) + S(t,i) \tag{44}$$

where $W(t,i)$ is the geometric warp matrix which represents motion between $x(t)$ and $x(t-i)$. $S(t,i)$ is error which is assumed to be additive random noise [48].

When Equation (43) and Equation (44) are combined, the resulting formula becomes

$$y(t-i) = DHW(t,i)x(t) + e(t,i) \quad for\ 0 \leq i \leq N-1 \tag{45}$$

where N is number of past measured images and $e(t,i)$ is overall error between the model at time t and the measured image at time $t-i$. The error is assumed to be Gaussian with zero mean and variance ${\sigma_k}^2$ as in MLE and MAP cases [48]. In this case, the weighted least square cost function that is to be minimized is defined as

$$E(t) = \sum_{i=0}^{N-1} \mu^{\mathrm{k}} \| y(t-i) - DHW(t,i)\hat{x}(t) \|^2 \tag{46}$$

where $0 < \mu \leq 1$ is forgetting factor which gives exponentially less weight to older error samples and $k = N-1-i$. This cost function is similar to the cost function of MLE approach (Equation (32)) except time indices inserted and past error samples used. The derivative of the cost function with respect to $\hat{x}(t)$ is set to zero in order to find optimal solution.

$$\frac{\partial E(t)}{\partial \hat{x}(t)} = 0 =>$$

$$\sum_{i=0}^{N-1} \mu^{\mathrm{k}} W(t,i)^T H^T D^T y(t-i) = \left[\sum_{i=0}^{N-1} \mu^{\mathrm{k}} W(t,i)^T H^T D^T DHW(t,i) \right] \hat{x}(t) \tag{47}$$

The solution of above equation gives the optimal $\hat{x}(t)$ minimizing $E(t)$. Equation (47) is written as

$$P(t) = R(t)\hat{x}(t) \tag{48}$$

where

$$P(t) = \sum_{i=0}^{N-1} \mu^{\mathrm{k}} W(t,i)^T H^T D^T y(t-i) \tag{49}$$

and

$$R(t) = \left[\sum_{i=0}^{N-1} \mu^{\mathrm{k}} W(t,i)^T H^T D^T DHW(t,i) \right] \tag{50}$$

$W(t,i)$ can be decomposed as

$$W(t,i) = W(t-1,i-1)W(t,1) = \cdots \quad (51)$$
$$= W(t-i+1,1)W(t-i+2,1)W(t-i+3,1)$$
$$\ldots W(t-1,1)W(t,1)$$
$$= \prod_{j=1}^{i} W(t-i+j,1)$$

where $W(t,0) = \mathrm{I}$ (Identity matrix). Then, $P(t)$ and $R(t)$ are written as

$$P(t) = \mu W(t,1)^T P(t-1) + H^T D^T Y(t) \quad (52)$$
$$R(t) = \mu W(t,1)^T R(t-1) W(t,1) + H^T D^T \hat{Z} D H(t) \quad (53)$$

In order to obtain optimal solution, the following equation should be solved.

$$\hat{x}(t) = R(t)^{-1} P(t) \quad (54)$$

Iterative methods such as steepest descent, normalized steepest descent or conjugate gradient can be used to solve above equation [48]. The steepest descent solution of the problem is

$$\hat{x}^{n+1}(t) = \hat{x}^n(t) + \lambda[P(t) - R(t)\hat{x}^n(t)] \quad (55)$$

At each time, the initialization is done using the previous temporal solution as

$$\hat{x}^0(t) = W(t,1)\hat{x}^r(t-1) \quad (56)$$

$W(t,1)$ is the forward warp operator which compensates motion between consecutive frames. Then, r iterations are performed in order to obtain $\hat{x}^r(t)$ which is the temporal solution at time t. The solution described above is named as Pseudo-RLS algorithm [59, 60]. If μ is replaced with zero in the Equation (52) and Equation (53), the optimality criteria will become instantaneous error and only present data will be used in order to reach temporal solution [48]. In this case, $P(t)$ and $R(t)$ are written as

$$P(t) = H^T D^T Y(t) \quad (57)$$
$$R(t) = H^T D^T D H(t) \quad (58)$$

and the solution of the problem using steepest descent is described as

$$\hat{x}^{n+1}(t) = \hat{x}^n(t) + \lambda H^T D^T [y(t) - DH\hat{x}^n(t)] \quad (59)$$

This solution is named as LMS solution [59]. Initial estimate of first high resolution image is chosen as interpolated version of first low resolution image. The next high resolution images are estimated using Equation (59) and Equation (56).

A problem with this approach is the possibility of $R(t)$ being singular. To avoid this problem, regularization term is added to the weighted cost function as in MAP case. It guarantees its non-singularity and increases stability of the solution [48]. If the regularization term is added to cost function, new cost function becomes

$$E(t) = \sum_{i=0}^{N-1} \mu^{\mathrm{k}} \| y(t-i) - DHW(t,i)\hat{x}(t) \|^2 + \frac{1}{Z} x^T Q x \quad (60)$$

Similar to reconstruction by MAP approach, Tikhonov regularization [56], Total Variation [58], Bilateral Total Variation [30], Gaussian MRF [35] and Huber MRF [33] can also be used for regularization. (See Section 2.3.3)

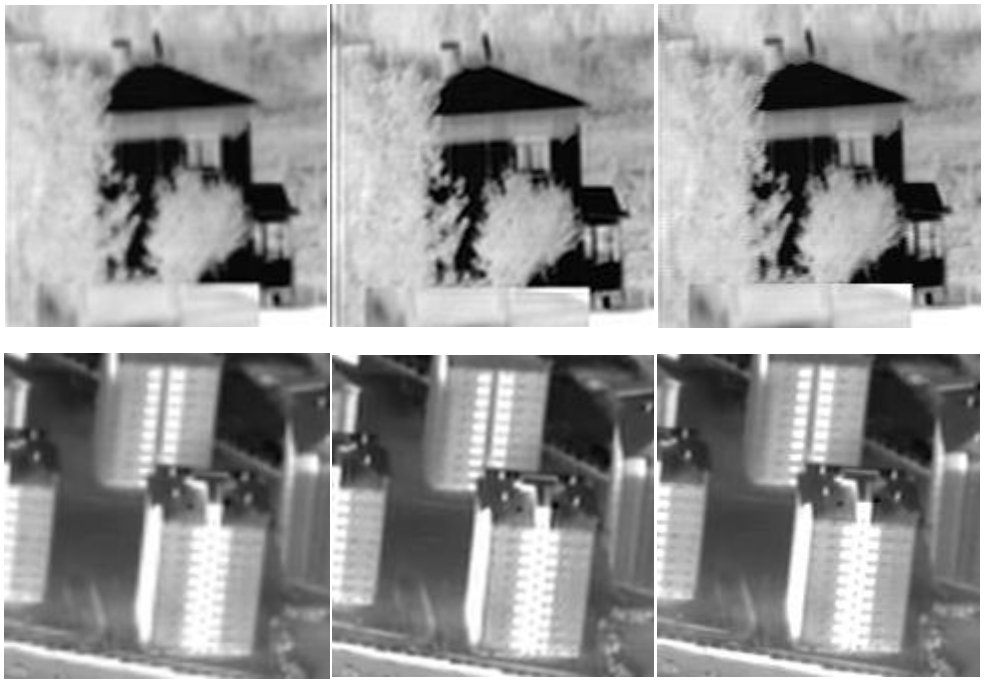
Since real time implementation is considered in this thesis, computational complexity is an important issue. LMS algorithm is shown to be had lower computational load than Pseudo-RLS algorithm in [48]. Also, LMS algorithm requires only present frame in order to reach temporal solution which means only single frame should be stored in memory regions. So, this algorithm is

also efficient in terms of memory requirements. Due to these reasons, LMS algorithm is considered for real time implementation.

When Tikhonov regularization is used, the final steepest descent solution of LMS approach will be given as

$$\hat{x}^{n+1}(t) = \hat{x}^{n}(t) + \lambda H^{T} D^{T} [y(t) - DH\hat{x}^{n}(t)] - \beta \mathrm{T}^{\mathrm{T}} \mathrm{T} \hat{x}^{n}(t) \qquad (61)$$

The images at left column of Figure 13 show the resulting images of single image bicubic interpolation. On the other hand, the images at middle column correspond to the results of LMS algorithm with Tikhonov regularization. Upsampling factor is selected as 2 for both methods. Four LR images for LMS are used in order to reconstruct HR image. The shift values between these images are generated with 0.5 pixel precision.



(a) (b) (c)

Figure 13 Images Reconstructed by a) Bicubic Interpolation b) LMS Method c) HR Image

CHAPTER 3

FPGA DESIGN OF SUPER RESOLUTION

As discussed earlier, super resolution methods consists of two parts which are global motion estimation and high resolution image reconstruction. Considered methods for these stages are PM, POCS, MLE, MAP and LMS which the theoretical backgrounds for these methods are given in details in Chapter 2. In this chapter, only FPGA structures of these methods will be proposed for real time implementation. The comparisons of these methods and FPGA implementation will be given at next chapter.

Firstly, FPGA will be introduced as a processing unit for real time implementation in Section 3.1. Then, the main FPGA block diagram of super resolution approach will be given in Section 3.2. There are three main blocks in the structure. These blocks are *Iteration Timing Generator, Motion Estimation* and *High Resolution Image Reconstruction*. The explanations of these blocks will be given in Section 3.2.1, Section 3.2.2 and Section 3.2.3 respectively.

3.1 Processing Unit

For a real time video application, the processing unit is selected in accordance to the application requirements. Digital Signal Processors (DSP), Graphics Processing Units (GPU) and FPGAs are suitable processing units for these types of applications. The computation characteristics of the application which will be implemented are important issue for deciding between the processing units. Our application super resolution has following computation characteristics [61]:

- Relatively simple data objects (Fixed point data format rather than floating point)
- Relatively modest arithmetic (Multiply and add rather than divide and transcendental functions)
- Pipeline processing structures
- Extensive data parallelism

FPGAs are approved to be suitable units for computations with above characteristics compared to DSPs [62] and GPUs [61].

The main sub-unit of FPGAs is LE (Logic Element) [63]. Every LE includes flip-flops and look-up tables. So, they can be used in order to construct asynchronous and synchronous circuits. Since LEs are individual units and they work independently from each other, they are suitable in order to construct parallel and pipeline structures. Furthermore, FPGAs have built-in memory blocks and dedicated multipliers which increase computational power.

In this thesis, Virtex V XC5VFX130T FPGA is used as processing unit. This unit contains 81,920 LEs, 10,728KB internal memory and 320 dedicated multipliers [63].

3.2 Super resolution FPGA Structure

The block diagram of FPGA structure is shown in Figure 14. There are three main blocks. *Motion Estimation* block performs motion estimation between successive frames by using projection method which is explained in Section 2.2. Horizontal and vertical shift values are computed for each frame of input video sequence. The details of the implementation are given in Section 3.1.2.

Since HR image reconstruction algorithms consists iterative structures, video timings are necessary for iterations during a single frame. *Iteration Timing Generator* block produces proper video timings for iterations. The explanation of the block is given in Section 3.1.1.

HR Image Reconstruction block uses estimated shift values, iteration timings and input video sequence to produce HR video sequence. Four algorithms were considered in this study. These algorithms are POCS, MLE, MAP and LMS which were explained in Section 2.3.1, Section 2.3.2, Section 2.3.3 and Section 2.3.4 respectively. Since MLE approach can be performed by applying little changes to MAP approach, FPGA structure of this algorithm was not proposed individually. The structures for POCS, MAP/MLE and LMS algorithms are presented in Section 3.1.3.2, Section 3.1.3.3 and Section 3.1.3.4 respectively.

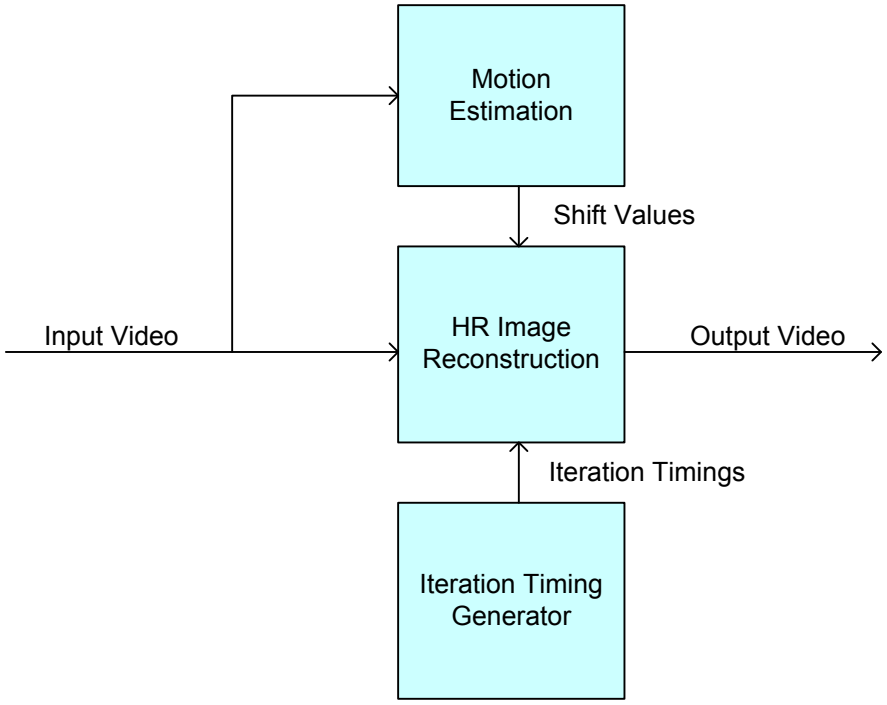


Figure 14 Structure of Super resolution Block

3.2.1 Iteration Timing Generator

POCS, MLE, MAP and LMS algorithms are implemented by iterative manner. During a single frame period, an amount of iterations is required for these algorithms in order reach HR image estimate solution. For these iterations, video timings are required. These timings are named as

iteration timings. *Iteration Timing Generator* block generates necessary video signals according to iteration timings.

In real time implementation, a region is extracted from input video sequence and SR is applied to this region. There are three reasons for this process:

1. In most of real time systems, the resolution of output is fixed. Super resolution process should not change output resolution.
2. Total number of iterations during a single frame depends on the resolution of region that SR is applied. If the resolution increases, the duration of single iteration will increase and total number of iterations in a frame period will decrease.
3. If the resolution of region that SR is applied increases, memory requirement for the structure will increase proportionally.

Extracted region from input video sequence is named as LR interested region. This region is processed and resolution is increased. The region produced after SR process is named as HR interested region. Since output resolution of the block is fixed, HR interested region is placed to the center of output grid. If any remaining regions exist, they are filled with black. The case that resolution of LR interested region is 160x120, resolution upsampling factor is 2 and output resolution is 640x480 is shown in Figure 15.

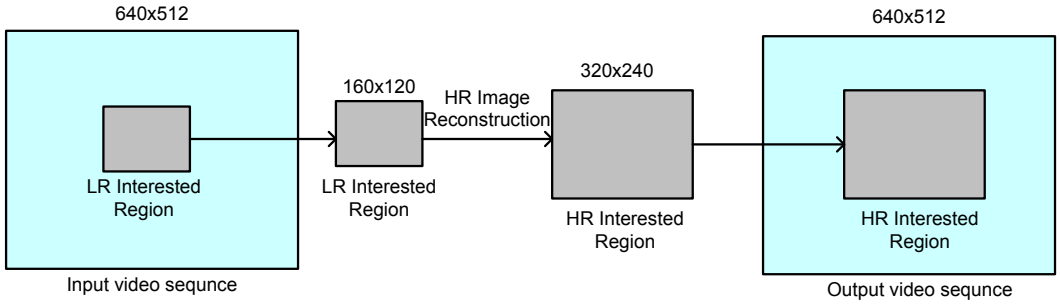


Figure 15 HR and LR Interested Regions

Iteration Timing Generator block generates necessary video signals according to the resolutions of LR interested region and HR interested region. These signals are used by child blocks of *HR image reconstruction* block. *Iteration frame* which is a signal covers a single iteration is shown in Figure 16. This signal is kept high during iteration.

In Figure 16, initial iteration produces initial estimation for HR interested region. Other iterations are performed in order to update estimation. *r* is the total number of iterations performed during a video frame period.

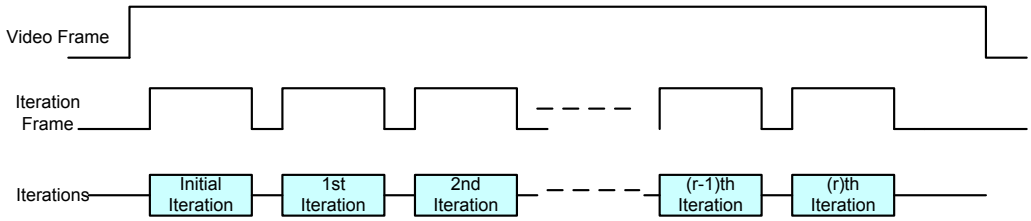


Figure 16 Iteration Frame

3.2.2 Motion Estimation

In order to estimate sub pixel shift values on x and y directions between successive frames of input video sequence, projection method in Section 2.2 is used. The block diagram of FPGA structure is shown in Figure 17. In the figure, *n* corresponds to current frame number and *n-1* to previous one.

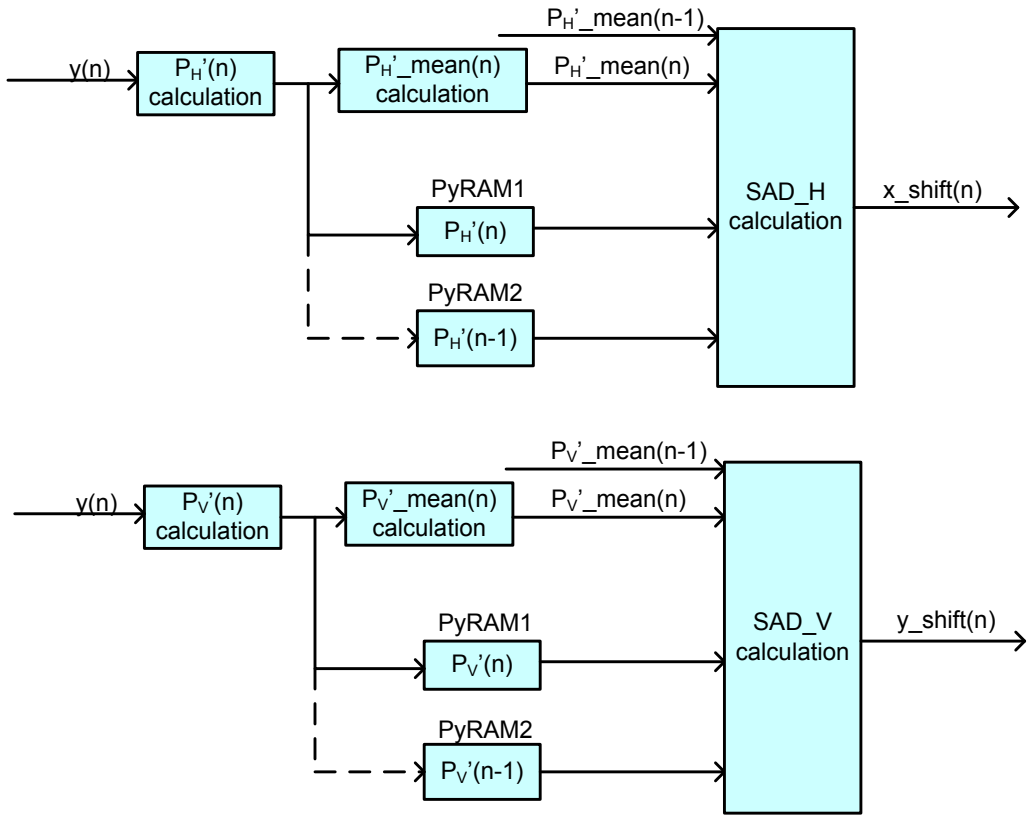


Figure 17 FPGA Structure of PM

The block $P_H'(n)$ *calculation* calculates horizontal projection vector of current image and the block $P_V'(n)$ *calculation* calculates vertical projection vector of current image. The vector formulations are rearranged in accordance to video sequence as

$$P_H'(n,x) = \sum_{y=vbb1}^{M-vbb2} y(n,x,y)\,, for\ x = hbb1, \dots, N - hbb2 \quad (62)$$

$$P_V'(n,y) = \sum_{x=hbb1}^{N-hbb2} y(n,x,y)\,, for\ y = vbb1, \dots, M - vbb2 \quad (63)$$

$$\boldsymbol{P_H'}(n) = [P_H'(n,hbb1)\ P_H'(n,hbb1+1)\ \dots\ P_H'(n,N-hbb2)] \quad (64)$$

$$\boldsymbol{P_V'}(n) = [P_V'(n,vbb1)\ P_V'(n,vbb1+1)\ \dots\ P_V'(n,M-vbb2)] \quad (65)$$

where $y(n,x,y)$ is the intensity value of pixel at location (*x, y)* in current frame of input video sequence. $hbb1$, $hbb2$, $vbb1$ and $vbb2$ are the parameters of the limited box (See Section 2.3.2). $P_H'(n,x)$ is *x*'th element of horizontal projection vector and $P_V'(n,y)$ is *y*'th element of vertical projection vector. $\boldsymbol{P_H'}(n)$ and $\boldsymbol{P_V'}(n)$ are horizontal and vertical projection vectors of current frame respectively.

The blocks *P_H'_mean calculation* and *P_V'_mean calculation* calculates the mean of horizontal and vertical projection vectors of current frame using the following formulas:

$$\overline{P_H'}(n) = \sum_{x=hbb1}^{N-hbb2} P_H'(n,x)\,/(N - hbb1 - hbb2) \quad (66)$$

$$\overline{P_V'}(n) = \sum_{y=vbb1}^{M-vbb2} P_V'(n,y)\,/(M - vbb1 - vbb2) \quad (67)$$

where $\overline{P_H'}(n)$ and $\overline{P_V'}(n)$ are the mean values of projection vectors.

In the structure, the projection vectors and mean values of previous frame and current frame should be stored. The projection vectors are stored on RAMs. Four RAMs are needed for this structure. Two RAMs are used to store previous projection vectors $\boldsymbol{P_H'}(n-1)$ and $\boldsymbol{P_V'}(n-1)$and two RAMs are used to store current projection vectors $\boldsymbol{P_H'}(n)$ and $\boldsymbol{P_V'}(n)$. The sizes of RAMs depend on the sizes of vectors. The mean values $\overline{P_H'}(n)$, $\overline{P_H'}(n-1)$, $\overline{P_V'}(n)$, $\overline{P_V'}(n-1)$ are stored on registers since they are just single integer values.

The blocks *SAD_H calculation* and *SAD_V calculation* calculate the shift values on x and y directions respectively. *SAD_H calculation* reads $\boldsymbol{P_H'}(n-1)$, $\boldsymbol{P_H'}(n)$, $\overline{P_H'}(n)$, $\overline{P_H'}(n-1)$ from RAMs and registers. As the same way, *SAD_V calculation* reads $\boldsymbol{P_V'}(n-1)$, $\boldsymbol{P_V'}(n)$, $\overline{P_V'}(n)$, $\overline{P_V'}(n-1)$. Then, the blocks compute normalized projection vectors by using following formulas.

$$\boldsymbol{P_H}(n) = \boldsymbol{P_H'}(n) - \overline{P_H'}(n) \quad (68)$$

$$\boldsymbol{P_V}(n) = \boldsymbol{P_V'}(n) - \overline{P_V'}(n) \quad (69)$$

$$\boldsymbol{P_H}(n-1) = \boldsymbol{P_H'}(n-1) - \overline{P_H'}(n-1) \quad (70)$$

$$\boldsymbol{P_V}(n-1) = \boldsymbol{P_V'}(n-1) - \overline{P_V'}(n-1) \quad (71)$$

After normalization process, upsampling is applied to the vectors for sub pixel estimation. The normalized projection vectors are upsampled by a factor *f* which is upsampling factor. So the estimated shift values will have *1/f* precision. The upsampling is done using bilinear interpolation.

$$\boldsymbol{P_{H,U}}(n) = \boldsymbol{P_H}(n) \uparrow f \quad (72)$$

$$\boldsymbol{P_{V,U}}(n) = \boldsymbol{P_V}(n) \uparrow f \quad (73)$$

$$\boldsymbol{P_{H,U}}(n-1) = \boldsymbol{P_H}(n-1) \uparrow f \quad (74)$$

$$\boldsymbol{P_{V,U}}(n-1) = \boldsymbol{P_V}(n-1) \uparrow f \quad (75)$$

$\boldsymbol{P_{H,U}}(n)$, $\boldsymbol{P_{H,U}}(n-1)$, $\boldsymbol{P_{V,U}}(n)$ and $\boldsymbol{P_{V,U}}(n-1)$ are upsampled normalized projection vectors. These vectors are used to calculate the sum of absolute differences.

$$SAD_H(n,m) = \sum_{k=S_{max}+hbb1}^{N-S_{max}-hbb2} \left|P_{H,U}(n-1,k+m) - P_{H,U}(n,k)\right| \quad (76)$$

$$SAD_V(n,l) = \sum_{k=S_{max}+vbb1}^{M-S_{max}-vbb2} \left|P_{V,U}(n-1,k+m) - P_{V,U}(n,k)\right| \quad (77)$$

$$m = -S_{max} \dots S_{max}$$

The calculated SAD values are compared to each other. Values of *m, l* that SAD values get their minimum are the estimated shift values.

$$x\_shift(n) = argmin_m\, SAD_H(n,m) \quad (78)$$

$$y\_shift(n) = argmin_l\, SAD_V(n,l) \quad (79)$$

The shift values calculated by *SAD_H calculation* and *SAD_V calculation* blocks are used in *HR image reconstruction* block in order to register input LR images.

3.2.3 High Resolution Image Reconstruction

The theoretical backgrounds for POCS, MLE, MAP and LMS algorithms were given in Section 2.3. In this section, FPGA structure used for implementation of these algorithms will be proposed. Firstly, the common blocks designed for FPGA implementation will be introduced. After that, the structure of POCS, MLE, MAP and LMS algorithms will be introduced by using these common blocks respectively.

3.2.3.1 Common FPGA Blocks

High resolution image reconstruction algorithms require common operations such as interpolation, downsampling, filtering and motion compensation for image registration. So, common blocks were designed in order to be able to be used commonly for the structures of reconstruction algorithms. They were designed by using VHDL coding and built-in structures of FPGA such as RAM and multipliers. VHDL coding leads to usage of Look Up Tables (LUT) and Flip Flops (FF) after synthesis the codes. The blocks are listed below:

- Bilinear Interpolation
- Image Filter
- Downsampling
- Motion Compensation
- Interlace to Progressive Converter
- Progressive to Interlace Converter
- Extraction of Interested Region
- Line Buffer
- LR RAM

- HR RAM
- POCS Back Projection

Bilinear Interpolation: All of high resolution image reconstruction algorithms require upsampling operation with an interpolation method. For this purpose, a bilinear interpolation block was designed. It enhances the resolution of input video twice for both axes by using bilinear interpolation method [54]. Two RAMs were used in the implementation since bilinear interpolation requires accessing two consecutive lines of input image. The sizes of RAMs were chosen as the line size of LR interested region.

Image Filter: Image filtering is used for PSF blurring and regularization operations. An image filter of size 5x5 was designed for this purpose. The coefficients of the kernel are configurable so it could be used for all image filtering operations. Since filtering operation requires accessing all pixel values that kernel covers, four lines should be stored. In order to achieve this, four FIFOs were used. The sizes were chosen as line size of HR interested region because all filtering operations are performed on HR grid for all algorithms.

Downsampling: Downsampling is performed in order to simulate image acquisition model. For this purpose, a block that reduces the resolution of video input to the half on both axes is designed. It masks the discarded pixels and lines in order to reduce the resolution.

Motion Compensation: Motion compensation block is designed for shifting an image on x and y axes with respect to estimated motion information. The block reads the image from RAM by starting a determined address with respect to registration parameters and produces an image with proper video timings. The produced image is motion compensated with estimated shift values.

Interlace to Progressive Converter: Since interpolation, image filtering, motion compensation operations are performed easily when the video format is progressive, a converter is needed to convert interlace video to progressive one when input video format is interlaced. This block converts the input video to progressive and writes it to the RAM.

Progressive to Interlace Converter: HR image reconstruction block should not change video format. The output video format should be same as input's. This block read the video from RAM and produces an interlaced video with proper video timings. If the input video format is interlaced, this block should be used.

Extraction of Interested Region: This block extracts LR interested region from input video sequence. A masking operation is performed to discard the other regions of input video.

Line Buffer: Sometimes two or more video sequences are needed to be synchronized when the applied operation requires parallelism. This operation is performed by storing some lines and adding required delay to the video sequences. Line Buffer block is used for this purpose. It consists of a RAM. The size of RAM is adjusted with respect to line size of HR interested region.

LR RAM: This block consists of a RAM which is capable of storing a LR interested region.

HR RAM: This block consists of a RAM which is capable of storing a HR interested region.

POCS Back Projection: POCS algorithm requires a special back projection method. This block is designed for this operation. The other algorithms do not use this block. The block reads residual and result of previous projection and produces a set of HR pixel values for each LR pixel according to Equation (21). After that, these pixel values are clipped according to the amplitude constraint. Then, the resulting pixel values are written to appropriate region of determined RAM.

FPGA structures of the HR image reconstruction algorithms are formed by connecting above blocks. Figure 18, Figure 19 and Figure 20 shows the block diagrams of POCS, MAP/MLE and LMS algorithms respectively. In the block diagrams, color lines are used in order to show the activation time of the processes performed. The explanations of the color lines are given below:

Green: Active all time
Red: Performed at only first iteration
Black: Performed at every iteration
Light Blue: Performed at last iteration
Grey: Performed at (iteration mod 4 == 1 except first iteration)
Dark Blue: Performed at (iteration mod 4 == 2)
Pink: Performed at (iteration mod 4 == 3)
Brown: Performed at (iteration mod 4 == 3)

3.2.3.2 FPGA Structure of POCS Approach

In the proposed structure shown in Figure 18, four previous frames of input video sequence are used in order to estimate current high resolution image. POCS algorithm requires consecutive projections according to the pixel values of input LR frames. Total number of pixels of input LR frames determines the number of projection operations needed for data consistency constraints as given in Equation (23). Figure 18 shows appropriate FPGA structure for POCS algorithm.

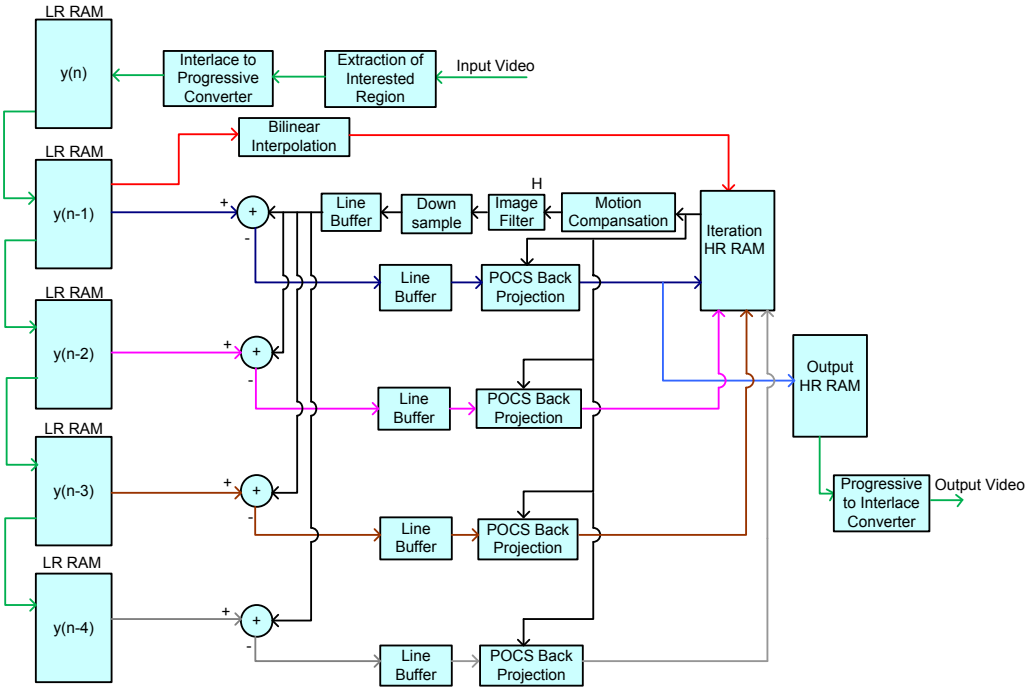


Figure 18 FPGA Structure of POCS

In the structure, LR interested region of input video is extracted and written to a LR RAM with progressive format. Five LR RAMs are used in order to store four previous and one current LR

interested regions. Four previous ones are used in order to reconstruct HR interested region. HR interested region is created at frame (n) according to following steps below.

1. K = 1, Iteration = 1
2. LR interested region at frame (n-K) is read from corresponding LR RAM
3. Bilinear interpolation is applied for initial estimation of HR interested region
4. It is written to iteration HR RAM
5. Iteration = Iteration +1
6. Last updated HR interested region is read from iteration HR RAM
7. Motion compensation is applied using registration parameters of (n-K)'th frame
8. Filtering is applied by using PSF
9. Downsampling is performed
10. The resulting image is subtracted from LR interested region at frame (n-K) which is read from LR RAM
11. Residual pixel values are stored in Line Buffer for timing synchronization purposes
12. POCS Back projection is applied to residual pixels and last updated HR pixels (Equation 21) in order to update HR interested region
13. Updated HR interested region is written to iteration HR RAM
14. If (Iteration == r), Updated HR interested region is also written to output HR RAM
15. If (K==4) K=1 else K=K+1
16. If (Iteration < r), Iteration = Iteration +1 and go to step 6

According to the proposed architecture, four previous LR interested regions are used consecutively in order to update HR interested region. So, total of four iterations are required in order to complete projection operations of data consistency constraints for single update of HR interested region. This is a disadvantage of POCS algorithm which makes it unsuitable for parallel processing. It reduces total amount of updates during a single frame period.

3.2.3.3 FPGA Structure of MAP/MLE Approach

Equation (34) in Section 2.3.3 is rearranged as below since MAP formulation given has not an appropriate notation for video sequence.

$$\hat{x}^{n+1}(n) = \hat{x}^{n}(n) - \lambda \sum_{k=1}^{4} W_k^T H^T D^T (DHW_k \hat{x}^{n}(n) - y_k(n-k)) \quad + \beta \mathrm{T}^{\mathrm{T}} \mathrm{T} \hat{x}^{n}(n) \tag{80}$$

Since four previous LR interested regions are used, four simulated LR interested regions are created through image acquisition model and used for obtaining four residual HR interested regions. These processes can be performed in parallel. Figure 19 shows block diagram of FPGA structure. As it is seen from the figure, four parallel process flows are used to form residual HR interested region.

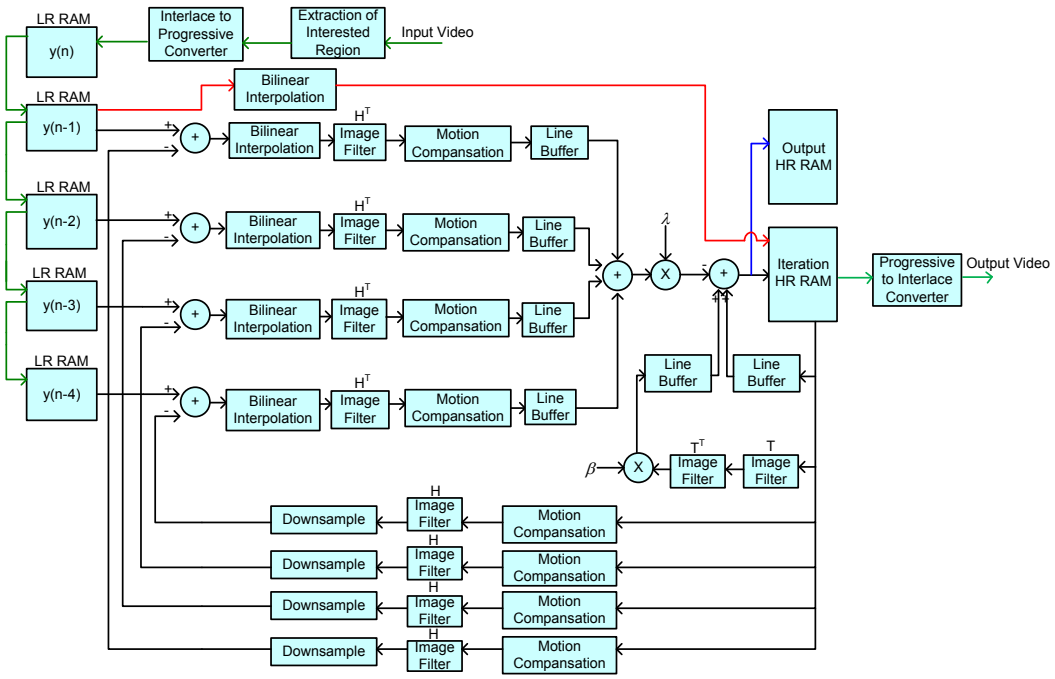


Figure 19 FPGA Structure of MAP/MLE

LR image buffering structure is same as in POCS structure. So, four previous LR interested region is processed in parallel in order to reconstruct HR interested region. HR interested region at frame (n) is created according to following steps below.

1. Iteration = 1
2. LR interested region at frame (n-1) is read from LR RAM
3. Bilinear interpolation is applied for initial estimation of HR interested region
4. It is written to iteration HR RAM
5. Iteration = Iteration +1
6. Last updated HR interested region is read from iteration HR RAM
7. Four HR image is created by motion compensating using corresponding registration parameters
8. HR images are filtered by PSF
9. HR images are downsampled in order to create simulated LR interested regions
10. The resulting image is subtracted from corresponding LR interested regions which are read from LR RAMs
11. Residual LR interested regions are upsampled using bilinear interpolation
12. Resulting images are filtered by transpose of PSF
13. Motion compensation is applied by using registration parameters corresponding to inverse motion
14. Line buffers are used in order to synchronize timings
15. The images are summed together to form residual HR interested region
16. HR interested region which is read in step 6 is filtered by regularization kernel and its transpose to form regularization term
17. Last updated HR interested region, residual HR interested region with λ coefficient and regularization term with β coefficient are used in order to update HR interested region

18. Updated HR interested region is written to iteration HR RAM
19. If (Iteration == r), updated HR interested region is also written to output HR RAM
20. If (Iteration < r), Iteration = Iteration +1 and go to step 6

MLE solution is a special case of MAP solution where regularization term is omitted. By setting β coefficient to zero of the above structure, the effect of regularization term will disappear and obtained solution will be MLE solution.

3.2.3.4 FPGA Structure of LMS Approach

LMS equations shown in Equation (61) and Equation (58) are rearranged for appropriate discrete notation as shown below

$$\hat{x}^{n+1}(n) = \hat{x}^n(n) + \lambda H^T D^T [y(n) - DH\hat{x}^n(n)] - \beta \mathrm{T}^{\mathrm{T}}\mathrm{T}\hat{x}^n(n) \quad (81)$$

$$\hat{x}^0(n) = W(n,1)\hat{x}^k(n-1) \quad (82)$$

FPGA block diagram for above equations is shown in Figure 20. The previous LR interested region $y(n-1)$ is used to estimate HR interested region. So, two LR RAMs are sufficient to store previous and current LR interested regions. Since POCS and MAP/MLE algorithms require four LR RAMs, this algorithm is more efficient in terms of memory.

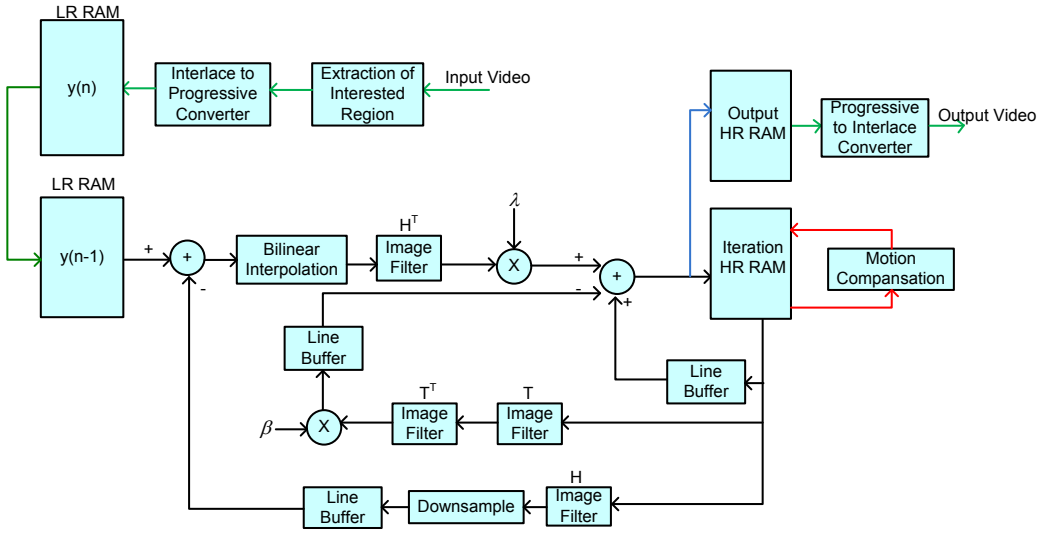


Figure 20 FPGA Structure of LMS

HR interested region at frame (n) is created according to following steps below.

1. Iteration = 1
2. Estimated HR interested region at frame (n-1) is read from Iteration HR RAM
3. It is registered with LR interested region at frame (n) by using estimated registration parameters
4. Resulting initial estimate for HR interested region is written to Iteration HR RAM
5. Iteration = Iteration +1
6. Last updated HR interested region is read from iteration HR RAM
7. It is filtered with PSF
8. It is downsampled in order to create simulated LR interested region
9. Line buffer is used for synchronizing timings

10. The resulting image is subtracted from LR interested region read from LR RAM
11. Residual LR interested region is upsampled using bilinear interpolation
12. Resulting image is filtered by transpose of PSF to form residual HR interested region
13. HR interested region which is read in step 6 is filtered by regularization kernel and its transpose to form regularization term
14. Last updated HR interested region, residual HR interested region with λ coefficient and regularization term with β coefficient are used in order to update HR interested region
15. Updated HR interested region is written to iteration HR RAM
16. If (Iteration == r), updated HR interested region is also written to output HR RAM
17. If (Iteration < r), Iteration = Iteration +1 and go to step 6

3.2.3.5 RAM Structure of Algorithms

During iterations in frame period, LR interested regions are read and used more than once. So, these regions should be stored in a memory region for reusability. In order to achieve this, input frames should be buffered. For POCS and MAP/MLE algorithms, total of four previous LR interested region should be buffered in RAMs while LMS requires only one LR interested region to be buffered. For this purpose, LR interested region of current frame is written to LR RAM. This region is used at next frame period during iterations. Since previous LR interested region is used while estimating current HR interested region, one frame delay is added to the system.

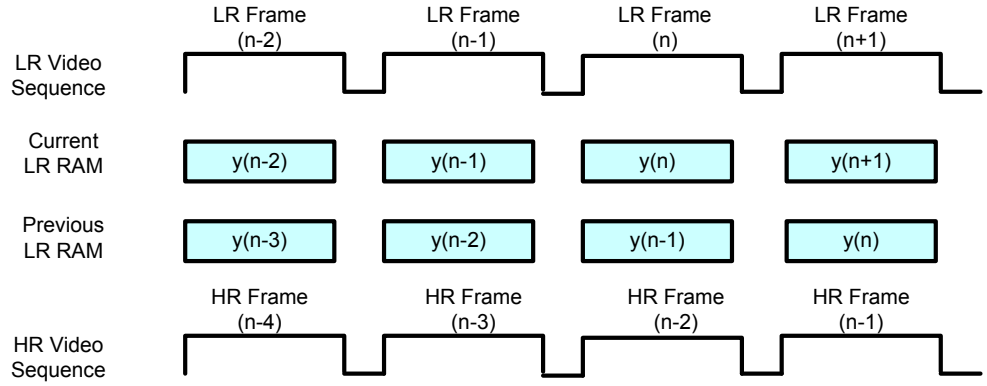


Figure 21 RAM Strcutures

Iterations require reading, processing and writing of HR interested region. For this purpose, two HR RAMs named as Iteration HR RAM and Output HR RAM are used as shown in Figure 18, Figure 19 and Figure 20. Iteration HR RAM in the figures is used to store HR interested region for updating processes. During iterations in a frame period, HR interested region stored in this memory region is updated.

In order to output estimated HR interested region properly, it should not change during current frame period because of iterations. So, resulting HR interested region at the last iteration is also written to Output HR RAM. This RAM is just used for outputting final HR interested region. While iterations are being performed at next frame, estimated HR interested region at previous frame is fed to the output of system by means of Output HR RAM. This adds another frame delay to the system. So, total amount of delay of overall system will be two frames as shown in Figure 21.

CHAPTER 4

COMPARISONS, FPGA IMPLEMENTATION AND RESULTS

FPGA structures of super resolution approaches were given in Chapter 3. Projection Method was considered for motion estimation stage. On the other hand, POCS, MLE, MAP and LMS algorithms were considered for high resolution image reconstruction stage. In this chapter, a comparison between high resolution image reconstruction algorithms is performed in terms of resource usage and output quality .Resource usage analysis covers Look-up Tables (LUTs), Flip-flops (FFs) and RAMs usage in FPGA. As a result of comparison, LMS algorithm is selected to be implemented. After that, FPGA design of PM and LMS algorithms are performed by using VHDL coding. The design is simulated in computer for validation purposes. After simulation, it is synthesized and embedded to a real time system which consists of an infrared camera.

The outline of this chapter is as follows. Firstly, results for motion estimation algorithm are given in Section 4.1. Then, comparison results between high resolution image reconstruction algorithms in terms of PSNR values and resource usage are given in Section 4.2. After that, performance of the implemented algorithm in FGPA is analyzed in Section 4.3. Finally, the details and results of FPGA implementation are given Section 4.4.

4.1 Results for Motion Estimation

Motion estimation algorithm is applied to a video sequence captured by the infrared camera used in this study as an experiment. The scene is stationary. The camera which is fixed on a platform placed on an aircraft has vibrations. So, captured frames have motion differences. Figure 22 shows a single frame of the video sequence used.

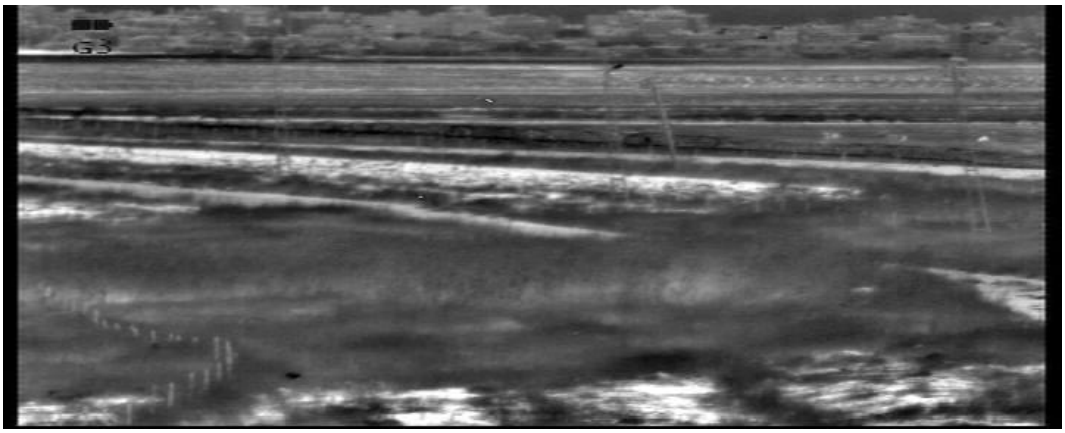


Figure 22 Test Image for Motion Estimation

The frames are registered by estimated shift values. PSNR (Peak Signal to Noise Ratio) values are calculated between consecutive frames as

$$MSE = \frac{\sum_{i=1}^{M}\sum_{j=1}^{N}(im1(i,j) - im2(i,j))^2}{MxN} \tag{83}$$

$$PSNR = 10\log_{10}\frac{255^2}{MSE} \tag{84}$$

Two different upsampling factors K=2 and K=4 are used for projection vectors in the experiment. The resulting PSNR values with respect to frame no are shown in Figure 23.

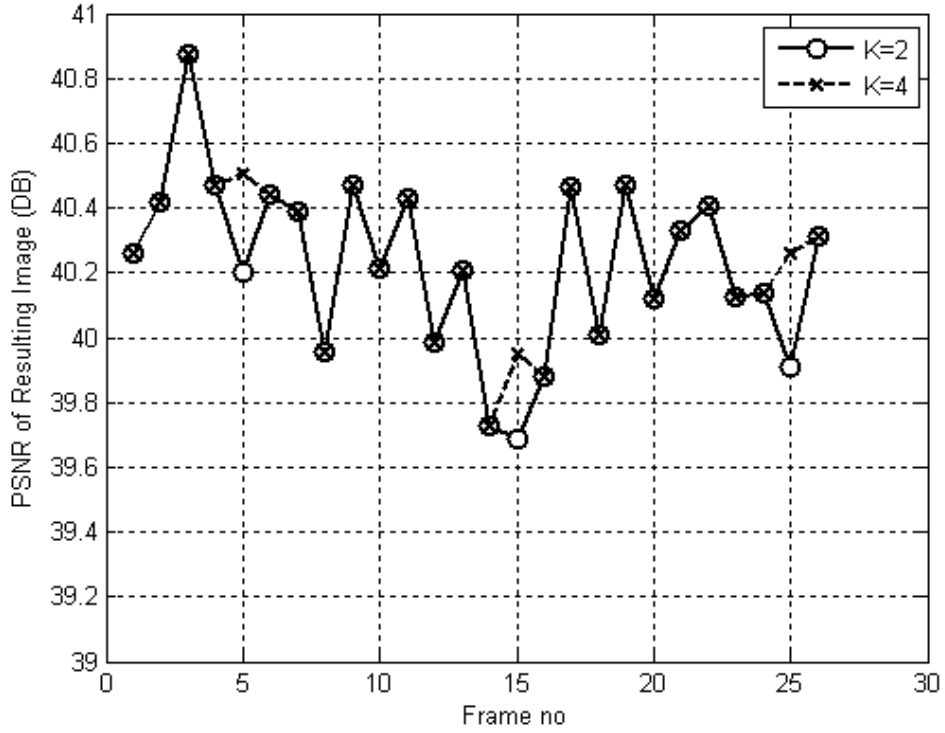


Figure 23 PSNR graph of PM

As shown in the resulting graph, when upsampling factor is increased, sub pixel registration accuracy also increases.

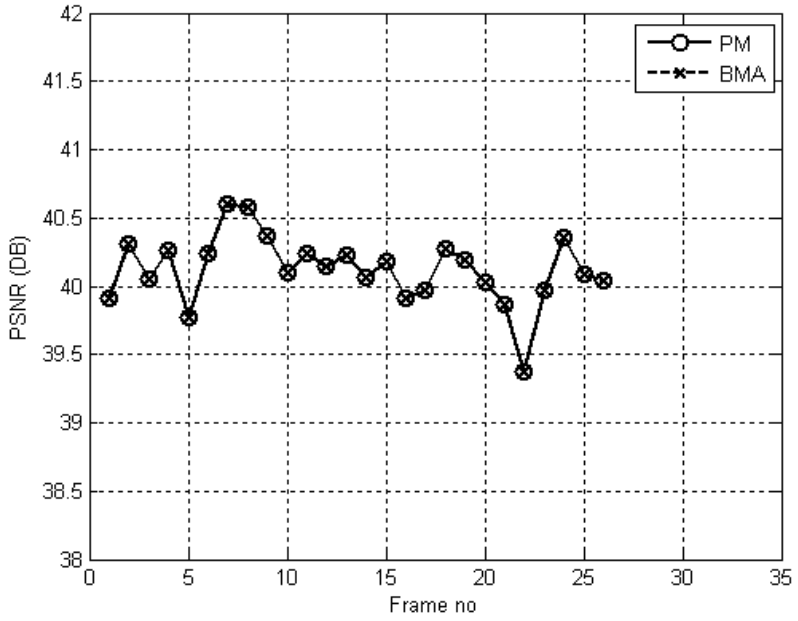
PM is a simplified version of conventional Block Matching Algorithm (BMA). BMA requires storing of previous frame in order to calculate shift values. On the other hand, PM requires storing only projection vectors of previous frame which makes this algorithm efficient in terms of memory requirements. In addition to this, PM requires only 1D correlation instead of 2D correlations which makes it also efficient in terms of computational load. Due to these reasons, PM is attractive for real time implementation.

For performance comparison between PM and BMA algorithms, an experiment is performed. Two video sequences captured from day camera and infrared camera are used in this experiment. The observed scene is stationary and the camera platforms have vibrations that result in translational motion between consecutive frames. Sample frames of the video sequences are shown in Figure 24.

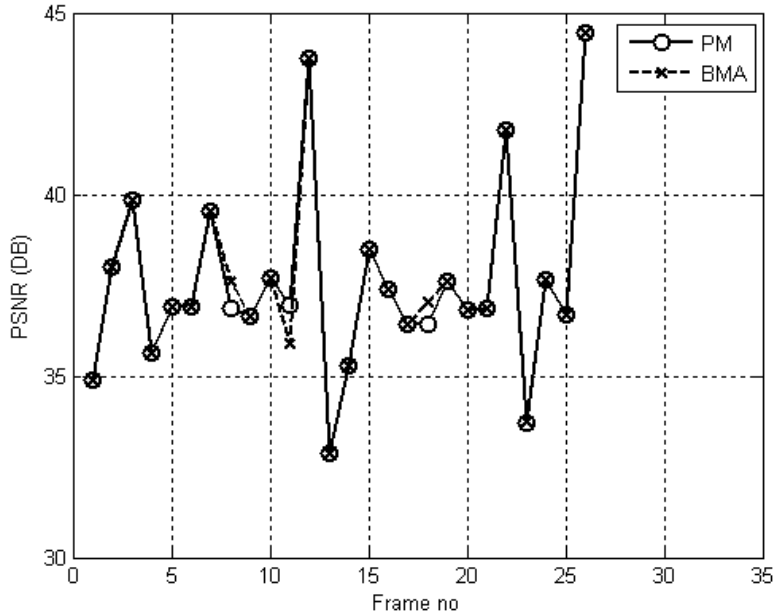
(a) (b)

Figure 24 Sample frames captured from a) Infrared Camera b) Day Camera

The video sequences are registered by using both PM and BMA algorithms seperately. PSNR values between consecutive frames for infrared and day camera video sequences after registration are shown in Figure 25.



(a)



(b)

Figure 25 PSNR values after registration for a) IR camera video sequence b) Day camera video sequence

As it is seen from Figure 25, PSNR values for this experiment are almost same. Especially, for infrared camera, the results are exactly same. Since PM algorithm is a suitable algorithm for real time implementation and gives sufficient performance, this algorithm is selected to be implemented on FPGA in this study.

4.2 Comparison of High Resolution Image Reconstruction Algorithms

POCS, MLE, MAP and LMS algorithms are implemented in Matlab for performance comparison. The performance comparison for these algorithms is performed by producing synthetic low resolution images. Single high resolution image captured from an infrared camera which is shown in Figure 26 is used to produce synthetic low resolution images through image acquisition model. This ground truth image is shifted, blurred, downsampled and contaminated with Gaussian noise in order to produce synthetic images. The shift values between images are generated with 0.5 pixel precision.

Figure 26 Ground truth image used in the experiments

The Gaussian noises are added to input images with various SNR levels while producing synthetic LR images. SNR value of an input image is defined as

$$SNR = 10\log_{10}\frac{var(lr\_im)^2}{var(noise)^2} \tag{85}$$

where *var(lr_im)* corresponds to standard deviation of synthetic low resolution image without noise and *var(noise)*corresponds to the standard deviation of contaminated Gaussian noise. Eight

sets of LR images are produced with different SNR levels vary from 10 DB to 45 DB with 5 DB steps. These video sequences are used to analyze the performance of the algorithms with respect to noise levels.

The outputs of the algorithms are compared to the ground truth high resolution image. Peak Signal to Noise Ratio (PSNR) is used for performance comparison. PSNR is calculated between ground truth high resolution image and an estimated high resolution image by using Equation (83) and Equation (84). Evaluated PSNR values with respect to different noise levels are shown in Figure 27.

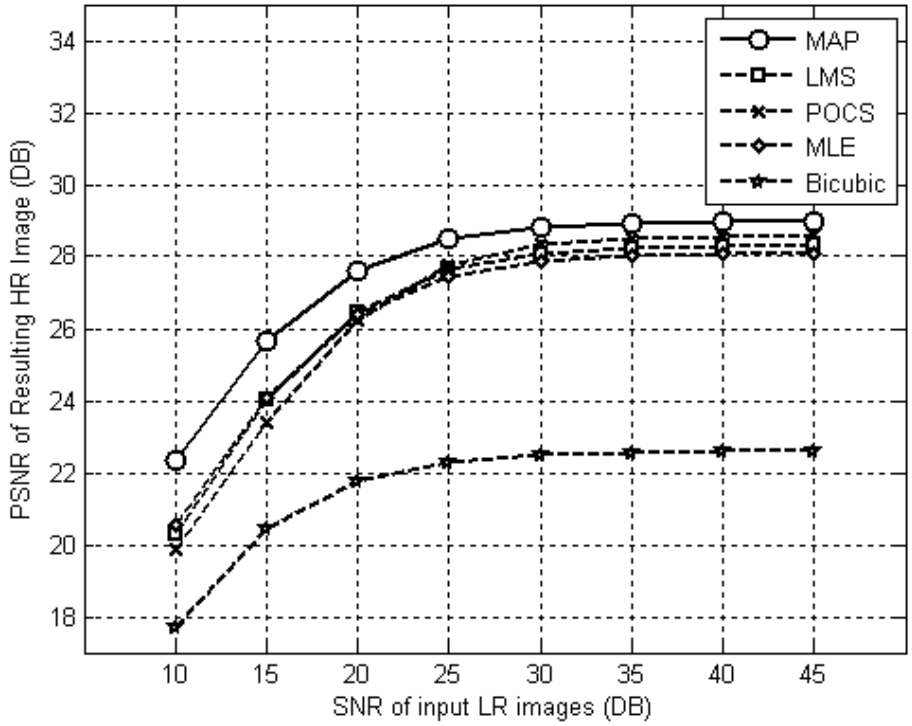


Figure 27 PSNR values of HR Image Reconstruction Algorithms

As seen in the graph, considered HR image reconstruction algorithms lead to close PSNR values in this experiment. Single image bicubic interpolation is used as baseline in order to show performances of HR image reconstruction algorithms. Since the PSNR performances of the HR image reconstruction algorithms close to each other, the resource usage for FPGA implementation becomes more important issue which is the main point in this study. The comparison of resource usage of HR image reconstruction algorithms are given below.

Since the structures designed for POCS, MLE, MAP and LMS algorithms use common blocks as shown in Figure 18, Figure 19 and Figure 20, a rough comparison between resource usages of these algorithms will be given. The resource usages of common blocks are found by synthesizing the blocks individually by Xilinx ISE 12.4 synthesis tool. During the synthesis, the resolution of LR interested region is selected as 160x120, resolution upsampling factor is chosen as 2 and the

word length used is selected as 8 bit (See Section 4.3). The resulting usage of LUTs, FFs and RAMs of common blocks are given in Table 1.

Table 1 Resource usage of common blocks

Blocks/Resources	**LUTs**	**FFs**	**RAMs**
Bilinear Interpolation	227	221	4 KB
Image Filter	481	452	6 KB
Downsampling	11	11	0
Motion Compensation	241	203	2 KB
Extraction of Interested Region	127	99	0
Line Buffer	0	0	2 KB
LR RAM	0	0	20 KB
HR RAM	0	0	80 KB

Since FPGA structures of HR image reconstruction algorithms are formed by combination of above blocks, a rough estimation of resource usage can be performed for each algorithm. Numbers of common blocks used in the FPGA structures of POCS, MLE, MAP and LMS are given in Table 2.

Table 2 Usage counts of common blocks

Blocks/Algorithms	POCS	MLE	MAP	LMS
Bilinear Interpolation	1	5	5	1
Image Filter	1	8	10	4
Downsampling	1	4	4	1
Motion Compensation	1	8	8	1
Extraction of Interested Region	1	1	1	1
Line Buffer	5	9	10	3
LR RAM	5	5	5	2
HR RAM	2	2	2	2

Total amount of resource usages for each algorithm are estimated by summing the products of block resources needed and usage counts. The resulting estimated LUTs, FFs and RAMs usage of the algorithms are given in Table 3.

Table 3 Resource usage of Algorithms

Algorithms/Resources	LUTs	FFs	RAMs
POCS	1671	986	282 KB
MLE	7082	6538	362 KB
MAP	8044	7442	376 KB
LMS	2530	2342	236 KB

According to above results, MAP has highest resource requirements. MLE requires fewer resources since no regularization term is used which exists in MAP case. LMS and POCS algorithms require fewer resources than MLE and MAP algorithms.

As seen from resource comparison, LMS and POCS algorithms are good candidates for real time implementation because of their better resource requirements. There is also an important issue that affects the real time performance. Total amount of updates of high resolution image that can be performed during single frame period is an important factor to converge final solution. As given in Section 3.1.3, MLE, MAP and LMS algorithms updates high resolution image estimation using a single iteration. But, this case is different for POCS algorithm. POCS is a projection based

method. It requires consecutive projections according to pixel values of observed LR images. Since consecutive processes are required, four LR images cannot be processed as parallel (See Section 3.1.3.2). So, a single update for high resolution image estimation requires four iterations. This reduces total amount of updates during a frame period. This case does not exist for LMS algorithm. Total amount of updates for LMS is four times of total amount of updated for POCS during a single frame period. For this reason, LMS algorithm is preferable to POCS algorithm in terms of real time implementation in FPGA. Due to these reasons, LMS is selected as high resolution image reconstruction algorithm for FPGA implementation.

4.3 Performance Evaluation and Results

During implementation of LMS algorithm in FPGA, some modifications are needed to be performed. In this section, performance considerations of FPGA implementation are given compared to original LMS algorithm. There are two reasons that cause performance differences between FPGA implementation and original LMS algorithm. The first one is word length used in the implementation and the other one is interpolation type used at upsampling operations.

Word length corresponds to bit width used for storing variables and pixel values at intermediate steps. Original LMS algorithm is implemented with double precision floating point processes. On the other hand, FPGA implementation is performed by fixed word length. When word length is increased, resource usage of the implementation will increase proportionally. So, in order to reach reasonable resource usage, the fractional values are fitted to some determined word length which will decrease the performance of the algorithm a little bit. In our implementation, the word length is chosen as eight bit. The fractional values are truncated to this word length.

The other factor that affects the performance of the implementation is interpolation type used for upsampling operations. As it is known, bicubic interpolation gives better results than bilinear interpolation for single image resolution enhancement. Nevertheless, bilinear interpolation is a low cost solution. In this section, a comparison is performed in order to analyze performance differences between these methods when they are used for super resolution.

The performance evaluation for above factors is performed by producing synthetic low resolution images. Single high resolution image captured from an infrared camera is used to produce low resolution image sequence through image acquisition model. The ground truth image is shifted, blurred, downsampled and contaminated with Gaussian noise and synthetic video sequence which consists of 25 frames is formed by this way. The shift values between frames are generated with 0.5 pixel precision.

Obtained LR sequences are applied to the original LMS algorithm, FPGA implementation algorithm with bilinear interpolation, FPGA implementation algorithm with bicubic interpolation and single image bicubic interpolation method in Matlab. Resulting graph is shown in Figure 28. Circle indicators show PSNR values of original LMS algorithm with respect to noise levels. Square indicators and asteriks indicators show PSNR values of FPGA implementation algorithm with bilinear and bicubic interpolation respectively. Diamond indicators correspond to PSNR values of single image bicubic interpolation.

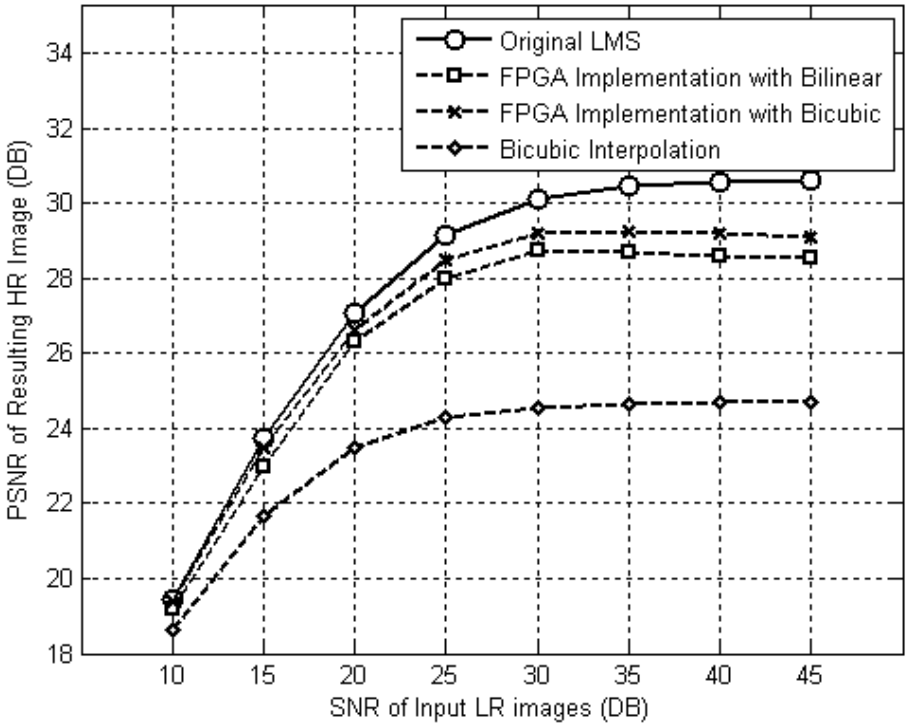


Figure 28 Performance Results with respect to Noise Levels

Original LMS algorithm reaches to highest PSNR value as shown in Figure 28. The performance decreases about 1.5 DB when word length is fixed to 8 bit which corresponds to FPGA implementation algorithm with bicubic. FPGA implementation algorithms with bilinear interpolation and FPGA implementation algorithms bicubic interpolation have slightly different PSNR values. Bilinear interpolation leads to close values to bicubic interpolation about 0.5 DB difference with lower computational cost. Single image bicubic interpolation has lowest performance compared to other algorithms.

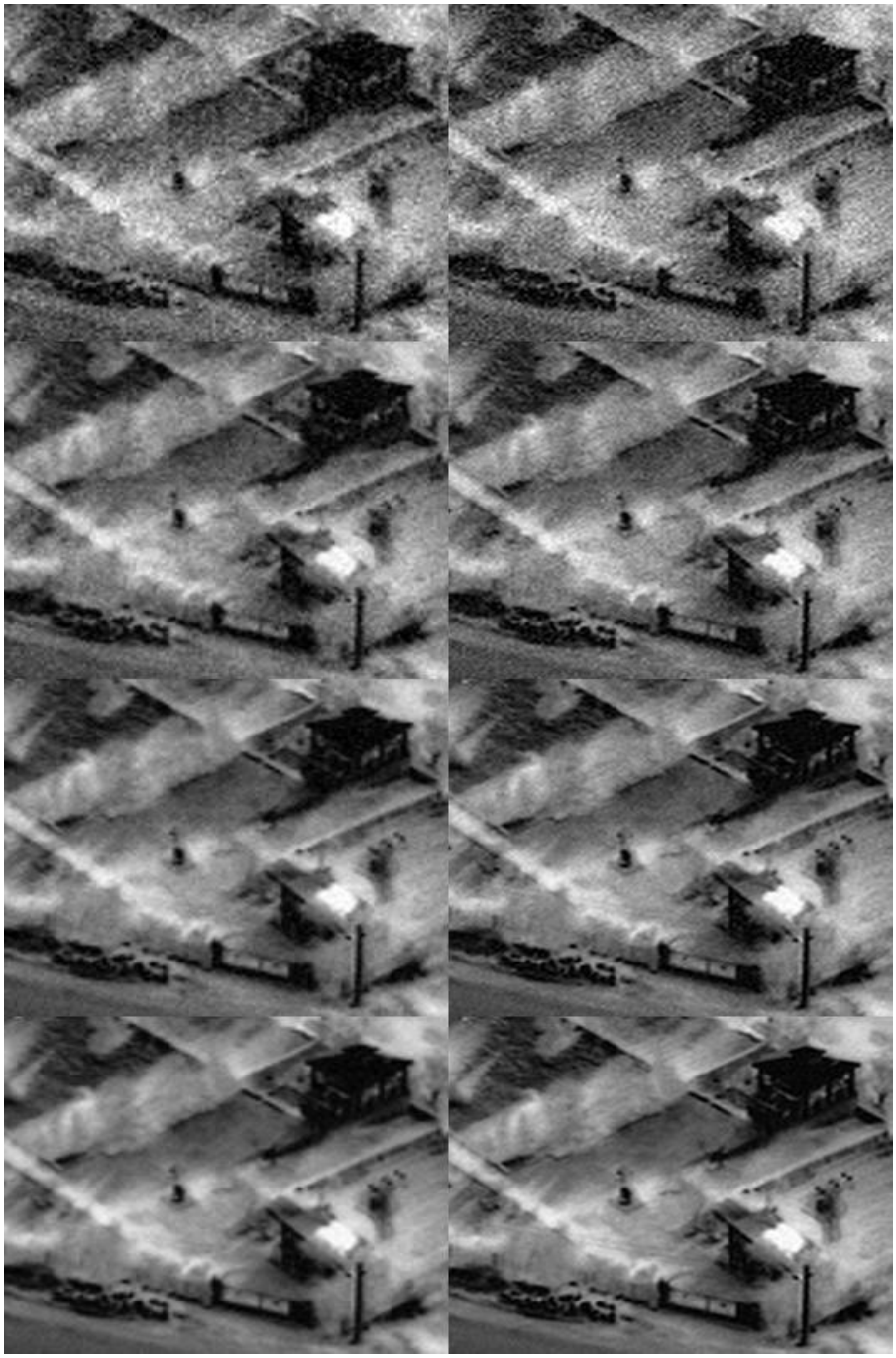
The resulting images of FPGA algorithm with bilinear interpolation and single image bicubic interpolation are shown in Figure 29. The images at left column correspond to resulting images of bicubic interpolation while SNR of input images vary from 10 DB to 45 DB with 5 DB steps. The images at right column correspond to resulting images of FPGA implementation with bilinear interpolation which will be implemented on real time system. As it is seen from the figures, while SNR of input images increase, the affect of super resolution becomes clearer.

10 DB

15 DB

20 DB

25 DB



(a) (b)

Figure 29 Images Reconstructed by a) Bicubic Interpolation b) FPGA Implementation Algorithm with Bilinear Interpolation

30 DB

35 DB

40 DB

45 DB

(Figure Continued)

4.4 FPGA Implementation

4.4.1 Video Timings

The implementation is performed on a real time infrared imaging system. The system has an infrared camera which produces video sequence with 640x512 resolution. The format of produced video is interlaced and frame rate is 30 Hz. The timing diagrams of video signals are given in Appendix A.

4.4.2 Duration of Iterations

In FPGA structure of high resolution image reconstruction stage, iterations are performed during a single frame period in order to reach solution. In a real time system, the frame period is generally determined and could not be changed. So, total amount of iterations depend on duration of single iteration. In our application, duration of single iteration depends on resolution of HR interested region since iterations update HR interested region.

Table 4 Total Number of Iterations with respect to HR interested region resolution

HR Interested Region Resolution (Horizontal resolution x Vertical Resolution)	**160x120**	**320x240**	**640x480**
Total Horizontal Active Data (Clock Cycles)	160	320	640
Total Vertical Active Data (Lines)	120	240	480
Duration of Single Iteration(ms)	0,387	1,485	5,814
Total Number of Iterations	86	22	5

Duration of single iteration is calculated by following manner:

Duration of Single Iteration = (Total Horizontal Active Data) x (Total Vertical Active Data) x (1/Pixel Clock Frequency)

So, total number of iterations during a frame period is calculated by

Maximum Number of Iterations = Frame Period / Duration of Single Iteration

where frame period is 33.333 ms since frame rate is 30Hz.

Related values with respect to the resolution of HR interested region are shown in Table 4. As it is seen from the table, when resolution of HR interested region increases, total number of iterations that can be performed during a frame period decreases.

For FPGA implementation, the resolution of HR interested region is chosen as 320x240. So, total 22 iterations can be performed during a frame period. Since first iteration is performed for initial estimation, total of 21 iterations can be performed in order to reach final solution.

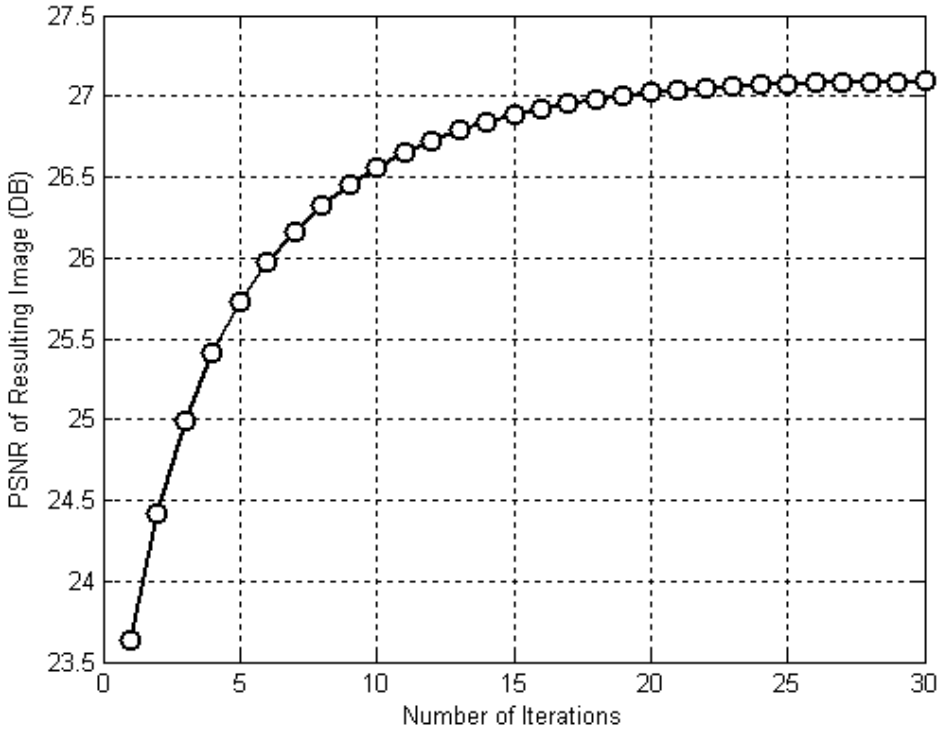


Figure 30 PSNR values of LMS with respect to number of iterations

Figure 30 shows the effect of number of iterations performed during a single frame period. When the number of iterations is increased, the resulting image will get closer to final solution. As seen from the graph, 21 iterations may be assumed to be sufficient in order to reach final solution. In this experiment, the step size (λ) is selected as 0.1.

4.4.3 FPGA Simulations

After comparisons made in Section 4.2, LMS algorithm is selected to be implemented in FPGA. FPGA blocks are designed by using VHDL coding. Then, the codes are simulated by Modelsim simulation tool. This tool shows all signal flow as it is in FPGA. Since the same codes are used in order to program FPGA, the simulation gives exact real time performance of the implementation.

Synthetic LR images obtained from real HR images are used for simulations by using image acquisition model. Produced images are combined to create a video sequence. Since Modelsim accepts text files as data inputs, obtained video sequence is converted to a text file by using Matlab. The text file is read by Modelsim and video signals are created according to real time video timings as described in Section 4.4.1. Simulation results are given in Appendix B.

The output of simulation is written to a text file by Modelsim. This file is read by Matlab and the results of Matlab and Modelsim are compared numerically in order to validate written VHDL codes.

4.4.4 Resource Usage of FPGA Implementation

The design is synthesized by Xilinx ISE 12.4. The tool gives exact usage of resources. Total resource usages for *Motion Estimation*, *High Resolution Image Reconstruction* and *Iteration Timing Generator* blocks are shown individually in Table 5, Table 6 and Table 7 respectively.

Table 5 Resource Usage of Motion Estimation Block

Resources	**LUTs**	**Flip Flops**	**RAMs**
Counts	3899	7728	16 KB

Table 6 Resource Usage of High Resolution Image Reconstruction Block

Resources	**LUTs**	**Flip Flops**	**RAMs**
Counts	2500	2503	300 KB

Table 7 Resource Usage of Iteration Timing Generator Block

Resources	**LUTs**	**Flip Flops**	**RAMs**
Counts	784	399	0 KB

4.4.5 Results of FPGA Implementation

The system used in the tests contains an infrared camera and a usable board for super resolution process which has a FPGA on it as a processing unit. After the design is synthesized by Xilinx ISE 12.4, the resulting design file is downloaded to this FPGA. The output of the system is recorded and the resulting images are demonstrated in Figure 31. Since only bilinear interpolation is implemented for FPGA, the visual comparisons of the system output are given between the output images of single image bilinear interpolation algorithm and super resolution algorithm. The images at left column shows the resulting images when only bilinear interpolation is activated on the system. On the other hand, the images at right column correspond to the resulting images when SR is activated.

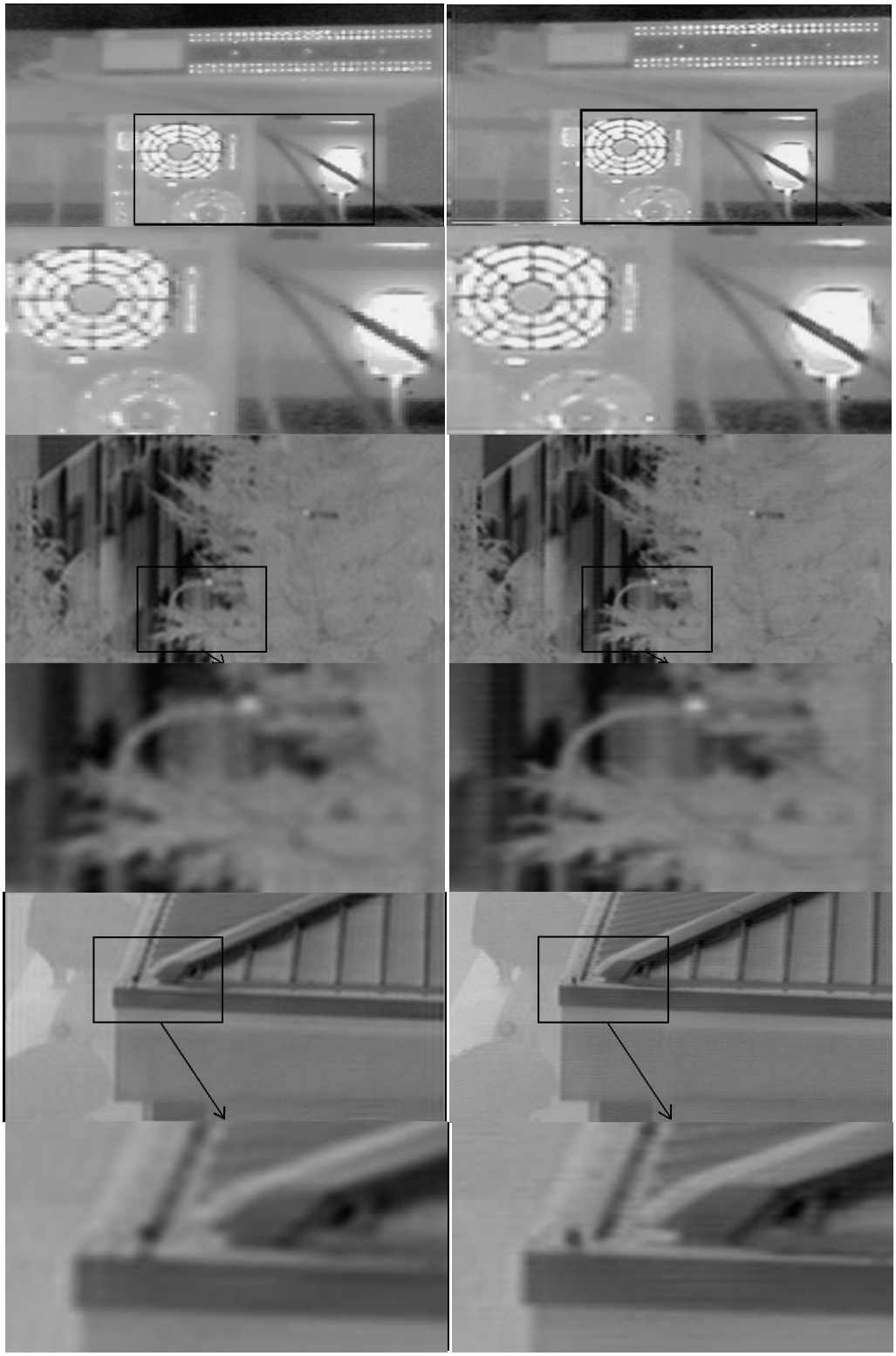


Figure 31 Real time system output with a) Bilinear Interpolation b) Implemented super resolution

As seen in the resulting images, implemented algorithm gives better results visually than single image bilinear interpolation on real time system as expected.

CHAPTER 5

CONCLUSIONS AND FUTURE WORKS

5.1 Summary

In this study, real time video super resolution is considered as main problem. The literature is searched and PM algorithm for motion estimation and POCS, MLE, MAP and LMS algorithms for HR image reconstruction are selected as candidates in order to be implemented. FPGA structures of these algorithms are proposed. The blocks used in the structures are designed so that they can be used commonly. After that, a comparison between high resolution image reconstruction algorithms is performed with respect to PSNR values, resource usage and timing considerations. As a result of comparison, LMS algorithm is selected to be implemented for high resolution image reconstruction. Then, FPGA codes are written for motion estimation algorithm and LMS algorithm in VHDL. The VHDL codes are simulated and the results are compared with MATLAB implementations. After validation of VHDL codes, the design is compiled in order to program FPGA. The result is observed from output of the real time infrared imaging system.

5.2 Discussion

At present, the infrared cameras are used in many areas such as astronomy, measurement devices and military areas. The infrared imaging systems are generally produce low quality images because of the fabrication of the cameras and the optics. So, it is an important issue to improve the image quality. Super resolution approaches use low quality images captured by imaging systems and produce better quality images. These processes require long computational time and high computational load which makes them hard to use in real time applications.

The recent advances in hardware elements, such as FPGAs, can handle high computational load. Since FPGAs can handle parallel processing, they are preferable in real time applications. So, it makes FPGAs attractive for implementation of super resolution. Due to these reasons, FPGA is selected as processing unit in this study.

As observed from the output of super resolution implementation, FPGA implementation works well as expected. It is seen that implemented algorithms PM and LMS are good candidates for real time implementation. They give good results visually. Furthermore, the implementation and results show that FPGAs are suitable units for performing real time digital video super resolution.

5.3 Future Works

In FPGA implementation, interpolation method is chosen as bilinear because of low resource requirements, low computational complexity and sufficient performance as discussed in Section 4.3. As a feature work, bicubic interpolation can be implemented to improve the performance. MATLAB results show that bicubic interpolation leads to better performance than bilinear interpolation as given in Section 4.3.

In the implementation, built-in RAMs of FPGA are used for memory requirements. An external RAM can be used instead of these RAMs. By this way, on-chip memory of FPGA can be emptied which is a desired situation. As external RAM, high speed DDR RAMs (Double Data Rate RAMs) or SRAMs (Synchronous RAMs) can be chosen for this application.

The motion model is assumed to be translational in this work. Because of low computational complexity and low resource usage, a cheap solution is applied for translation estimation. This motion model covers most of situations in real world when the camera is fixed or just translating without rotation and a stationary far scene is observed. However, in order to cover more situations, the motion model should be extended to cover rotation, scale, shear etc. For this purpose, as a future work, a motion estimation algorithm which can estimate affine motion information with sufficient accuracy can be implemented. This will make implementation applicable for wide range of fields.

REFERENCES

[1] S.P. Kim, N.K. Bose, and H.M. Valenzuela, "Recursive reconstruction of high resolution image from noisy undersampled multiframes," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. 38, pp. 1013-1027, June 1990.

[2] S. C. Park, M. K. Park, and M. G. Kang, "Super-Resolution Image Reconstruction A Technical Overview", IEEE Signal Processing Mag., vol. 20, pp. 21-36, May 2003.

[3] R.C. Hardie, K.J. Barnard, J.G. Bognar, E.E. Armstrong, and E.A. Watson, "High-resolution image reconstruction from a sequence of rotated and translated frames and its application to an infrared imaging system," Opt. Eng., vol. 37, no. 1, pp. 247-260, Jan. 1998.

[4] L.G. Brown, "A survey of image registration techniques," ACM Comput. Surveys, vol. 24, no. 4, pp. 325-376, Dec. 1992.

[5] B. Zitova, J. Flusser, "Image registration methods: a survey", Image and Vision Computing 21 (2003) 977–1000

[6] D. Keren, S. Peleg, and R. Brada, "Image sequence enhancement using sub-pixel displacements", In Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (1988), pp. 742-746.

[7] D. Capel and A. Zisserman, "Computer vision applied to super resolution," IEEE Signal Processing Magazine, vol. 20, no. 3, pp. 75–86, May 2003.

[8] C. J. Harris and M. Stephens. "A combined corner and edge detector" In Proc.Alvey Vision Conf., pages 147–151, 1988.

[9] P. Vandewalle, L. Sbaiz, M. Vetterli, and S. Susstrunk, "Super-resolution from highly undersampled images," in IEEE International Conference on Image Processing, September 2005.

[10] B. Marcel, M. Briot, and R. Murrieta, "Calcul de translation et rotation par la transformation de Fourier," Traitement du Signal, vol. 14, no. 2, pp. 135–149, 1997.

[11] L. Lucchese and G. M. Cortelazzo, "A noise-robust frequency domain technique for estimating planar roto-translations," IEEETransactions on Signal Processing, vol. 48, no. 6, pp. 1769–1786, June 2000.

[12] H. Foroosh, J. B. Zerubia, and M. Berthod, "Extension of phase correlation to subpixel registration," IEEE Transactions on Image Processing, vol. 11, no. 3, pp. 188–200, March 2002.

[13] P. Vandewalle, S. Susstrunk and M. Vetterli, "A Frequency Domain Approach to Registration of Aliased Images with Application to Super-resolution", EURASIP Journal on Applied Signal Processing Volume 2006, Article ID 71459, Pages 1–14.

[14] Y. B. Sert, "An examination of super resolution methods," M. Sc. dissertation, The Graduate School of Natural and Applied Sciences, Middle East Technical University, Ankara, Turkey, 2006.

[15] Puglisi, G., Battiato, S., "Fast block based local motion estimation for video stabilization", Computer Vision and Pattern Recognition Workshops (CVPRW), 2011 IEEE Computer Society Conference on, On page(s): 50 – 57.

[16] Yih-Chuan Lin, Shen-Chuan Tai, "Fast full-search block-matching algorithm for motion-compensated video compression", Communications, IEEE Transactions on, On page(s): 527 - 531 Volume: 45, Issue: 5, May 1997

[17] J. S. Kim and R. H. Park, "A fast feature-based block matching algorithm using integral projections," IEEE J. Select. Areas Commun., vol. 10, no. 5, pp. 968-971, June 1992.

[18] Sauer, K., Schwartz, B., "Efficient block motion estimation using integral projections", Circuits and Systems for Video Technology, IEEE Transactions on, On page(s): 513 - 518 Volume: 6, Issue: 5, Oct 1996.

[19] R.Y. Tsai and T.S. Huang, "Multipleframe image restoration and registration" in *Advances in Computer Vision and Image Processing*. Greenwich, CT: JAI Press Inc.,

1984, pp. 317-339.

[20] S.P. Kim and W.Y. Su, "Recursive high-resolution reconstruction of blurred multiframe images," *IEEE Trans. Image Processing*, vol. 2, pp. 534-539, Oct. 1993.

[21] N.K. Bose, H.C. Kim, and H.M. Valenzuela, "Recursive implementation of total least squares algorithm for image reconstruction from noisy, undersampled multiframes," in *Proc. IEEE Conf. Acoustics, Speech and Signal Processing*, Minneapolis, MN, Apr. 1993, vol. 5, pp. 269-272.

[22] S.H. Rhee and M.G. Kang, "Discrete cosine transform based regularized high-resolution image reconstruction algorithm," *Opt. Eng.*, vol. 38, no. 8, pp. 1348-1356, Aug. 1999.

[23] J. J. Clark, M R. Palmer, and P.D. Laurence, "A transformation method for the reconstruction of functions from nonuniformly spaced samples," IEEE Trans. Acoust., Speech, Signal Processing, vol. ASSP-33, pp. 1151-1165, 1985.

[24] S.P. Kim and N.K. Bose, "Reconstruction of 2-D bandlimited discrete signals from nonuniform samples," *Proc. Inst. Elec. Eng.*, vol. 137, pt. F, pp. 197-204, June 1990.

[25] A. Papoulis, "Generalized sampling theorem," *IEEE Trans. Circuits Syst.* vol. 24, pp. 652-654, Nov. 1977.

[1] M.S. Alam, J.G. Bognar, R.C. Hardie, and B.J. Yasuda, "Infrared image registration and high-resolution reconstruction using multiple translationally shifted aliased video frames," IEEE Trans. Instrum. Meas., vol. 49, pp. 915-923, Oct. 2000.

[27] N. Nguyen and P. Milanfar "An efficient wavelet-based algorithm for image superresolution," in *Proc. Int. Conf. Image Processing*, vol. 2, 2000, pp. 351-354.

[28] A. Zomet and S. Peleg, "Efficient Super-Resolution and Applications to Mosaics", ;in Proc. ICPR, 2000, pp.1579-1583.

[29] A. Zomet, A. Rav-Acha, and S. Peleg, Robust Super-Resolution, Proceedings international conference on computer vision and pattern recognition (CVPR), I-645 - I-650 vol.1, 2001.

[30] S. Farsiu, M.D. Robinson, M. Elad, and P. Milanfar, "Fast and robust multiframe super resolution", ;presented at IEEE Transactions on Image Processing, 2004, pp.1327-1344.

[31] N.K. Bose, S. Lertrattanapanich, and J. Koo, "Advances in superresolution using L-curve," in Proc. Int. Symp. Circuits and Systems, vol.2, 2001, pp. 433-436.

[32] B.C. Tom and A.K. Katsaggelos, "Reconstruction of a high-resolution image by simultaneous registration, restoration, and interpolation of low-resolution images," *Proc. 1995 IEEE Int. Conf. Image Processing*, vol. 2, Washington, DC, Oct. 1995, pp. 539-542.

[33] R.R. Schulz and R.L. Stevenson, "Extraction of high-resolution frames from video sequences," *IEEE Trans. Image Processing*, vol. 5, pp. 996-1011, June 1996.

[34] R.C. Hardie, K.J. Barnard, and E.E. Armstrong, "Joint MAP registration and high-resolution image estimation using a sequence of undersampled images," *IEEE Trans. Image Processing.*, vol. 6, pp. 1621-1633, Dec. 1997.

[35] P. Cheeseman, B. Kanefsky, R. Kraft, J. Stutz, and R. Hanson, "Super-resolved surface reconstruction from multiple images," NASA Ames Research Center, Moffett Field, CA, Tech. Rep. FIA-94-12, Dec. 1994.

[36] C. A. Segall, R. Molina, and A. K. Katsaggelos, "High-resolution images from low-resolution compressed video," IEEE Signal Process. Mag., vol. 20, no. 3, pp. 37–48, May 2003.

[37] H. Shen, L. Zhang, B. Huang, and P. Li, "A MAP approach for joint motion estimation, segmentation, and super resolution," IEEE Trans. Image Process., vol. 16, no. 2, pp. 479–490, Feb. 2007.

[38] Keller, S.H.; Lauze, F.; Nielsen, M.;, "Video Super-Resolution Using Simultaneous

Motion and Intensity Calculations," Image Processing, IEEE Transactions on , vol.20, no.7, pp.1870-1884, July 2011.

[39] H. Stark and P. Oskoui, "High resolution image recovery from image- plane arrays, using convex projections," J. Opt. Soc. Am. A, vol. 6, pp. 1715-1726, 1989.

[40] A.M. Tekalp, M.K. Ozkan, and M.I. Sezan, "High-resolution image reconstruction from lower-resolution image sequences and space varying image restoration," in Proc. IEEE Int. Conf. Acoustics, Speech and Signal Processing (ICASSP), San Francisco, CA., vol. 3, Mar. 1992, pp. 169-172.

[41] A.J. Patti, M.I. Sezan, and A.M. Tekalp, "Superresolution video reconstruction with arbitrary sampling lattices and nonzero aperture time," IEEE Trans. Image Processing, vol. 6, no. 8, pp. 1064-1076, Aug. 1997.

[42] P.E. Eren, M.I. Sezan, and A.M. Tekalp, "Robust, object-based high-resolution image reconstruction from low-resolution video," *IEEE Trans. Image Processing*, vol. 6, no. 10, pp. 1446-1451, Oct. 1997.

[43] A.J. Patti and Y. Altunbasak, "Artifact reduction for set theoretic super resolution image reconstruction with edge adaptive constraints and higher-order interpolants," *IEEE Trans. Image Processing*, vol. 10, no. 1, pp. 179-186, Jan. 2001.

[44] B.C. Tom and A.K. Katsaggelos "An iterative algorithm for improving the resolution of video sequences," in *Proc. 1996 SPIE Conf. Visual Communications and Image Processing*, Orlando, FL, Mar. 1996, pp. 1430-1438.

[45] H. Irani and S. Peleg, "Improving Resolution by Image Registration", Graphical Models and Image Processing Vol. 53, No. 3, May, pp. 231-239, 1991.

[46] S. Mann and R.W. Picard, "Virtual bellows: Constructing high quality stills from video," in *Proc. IEEE Int. Conf. Image Processing*, Austin, TX, Nov. 1994, pp. 13-16.

[47] M. Irani and S. Peleg, "Motion analysis for image enhancement resolution, occlusion, and transparency," *J. Visual Commun. Image Represent.*, vol. 4, pp. 324-335, Dec. 1993.

[48] M. Elad and A. Feuer, "Superresolution restoration of an image sequence: adaptive filtering approach," IEEE Trans. Image Processing, vol. 8, pp. 387-395, Mar. 1999.

[49] S. Farsiu, M. Elad and P. Milanfar, "Video-to-Video Dynamic Super-Resolution for Grayscale and Color Sequences", EURASIP Journal of Applied Signal Processing, Special Issue on Superresolution Imaging, Volume 2006, Article ID 61859, Pages 1-15. Processing Volume 2006, Article ID 61859, Pages 1–15.

[50] G. M. Callico, S. Lopez, J. F. Lopez, R. Sarmiento, and A. Nunez. "Low-cost implementation of a super-resolution algorithm for real- time video applications". In IEEE International Symposium on Circuits and Systems (ISCAS), 2005.

[51] O. Bowen and C. S. Bouganis. Real-time image super resolution using an FPGA. In International Conference on Field Programmable Logic and Applications (FPL), pages 89–94, September 2008.

[52] M. E. Angelopoulou, C.-S. Bouganis, P. Y. K. Cheung, and G. A. Constantinides. "FPGA-based Real-time Super-Resolution on an 285 Adaptive Image Sensor". In International Workshop on Applied Reconfigurable Computing (ARC), pages 125–136, March 2008.

[53] M. E. Angelopoulou, C.-S. Bouganis, P. Y. K. Cheung, and G. A. Constantinides. "Robust Real-Time Super-Resolution on FPGA and an Application to Video Enhancement". ACM Transactions on Reconfigurable Technology and Systems, to appear, 2009.

[54] P. Miklós , "Image Interpolation Techniques," 2nd Serbian-Hungarian Joint Symposium, 2004.

[55] A. Zomet and S. Peleg, "Super-Resolution from Multiple Images Having Arbitrary Mutual Motion", Super-Resolution Imaging (2001), pp. 195-209.

[56] A.N. Tikhonov and V.Y. Arsenin. Solutions of Ill-Posed Problems. V.H. Winston &

Sons, John Wiley & Sons, Washington D.C., 1977.

[57] Sina Farsiu, Michael Elad, and Peyman Milanfar, Senior Member, IEEE, "Multiframe Demosaicing and Super-Resolution of Color Images", IEEE Trans. on Image Processing, vol. 15, no. 1, pp. 141-159, January 2006.

[58] L. Rudin, S. Osher, and E. Fatemi, "Nonlinear total variation based noise removal algorithms," Phys. D, vol. 60, pp. 259–268, Nov. 1992.

[59] M. Elad and A. Feuer, "Super-resolution reconstruction of image sequences," IEEE Trans. Pattern Anal. Machine Intelli., vol. 21, no. 9, pp. 817-834, Sept. 1999.

[60] M. Elad, and A. Feuer, "Superresolution Restoration of an Image Sequence: Adaptive Filtering Approach", IEEE Transactions on Image Procesing, VOL. 8, No. 3, March 1999.

[61] J. Chase, B. Nelson, J. Bodily, Z. Wei, and D.-J. Lee, "Real-time optical flow calculations on FPGA and GPU architectures: A comparison study," in Proc. 16th Int. Symp. Field-Program. Custom Comput. Mach. (FCCM), 2008, pp. 173–182.

[62] Altera Corporation White Paper, "FPGA and DSP Design Reliability and Maintenance", ver 1.1, May 2007.

[63] Xilinx, "Virtex-5 FPGA User Guide", v5.4, March 2012.

APPENDIX A

TIMING DIAGRAM OF VIDEO SIGNALS

The timing diagrams of video signals are shown in Figure 32.

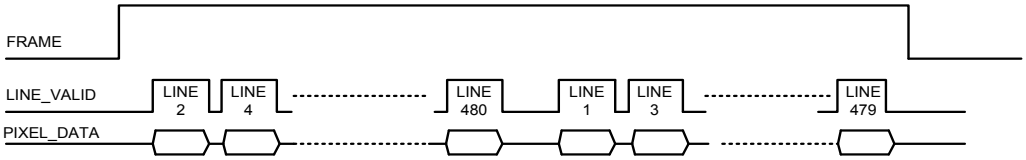


Figure 32 Input video timings for Super resolution Block

Input signals of super resolution are *FRAME*, *LINE_VALID* and *PIXEL_DATA*. The explanations of these signals are given below:

FRAME: This signal covers whole frame period. The rising edge of this signal points out the starting of a new frame. The period is 30Hz and it covers odd and even lines of interlaced video.

LINE_VALID: This signal covers a line of a frame. It is an active high signal. It will be high during 640 pixel clock where 640 is horizontal resolution. This signal is active at 480 times during a single frame period where 480 is vertical resolution. When a new frame starts, even lines will flow firstly. After that, odd lines will flow as shown in Figure **32**.

PIXEL DATA: This signal indicates corresponding pixel intensity values when *LINE_VALID* is active. The bit width of this signal is 8 bit which means the range of pixel intensity values is 0 to 255.

APPENDIX B

SIMULATION RESULTS

Input video timings of super resolution block are shown in Figure 33.

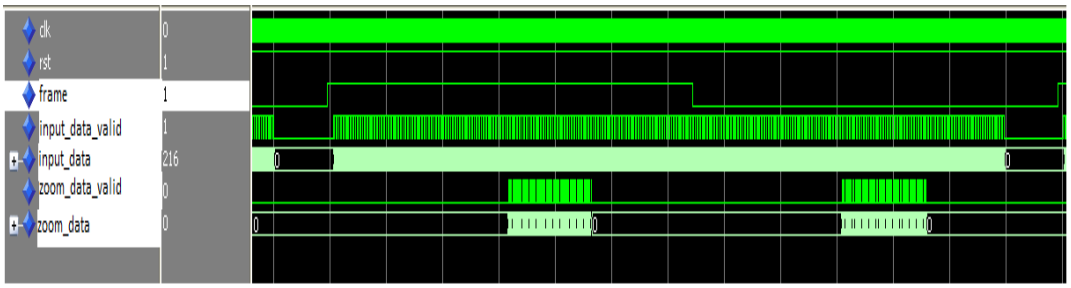


Figure 33 Input video timings

The signal *clk* is pixel clock which has frequency of 54 MHz and signal *rst* is system reset which is an active low signal. The signal *frame* covers a LR frame and rising edge of this signal points out that a new frame is starting. The signal *input_data_valid* covers a line of the frame and the signal *input_data* corresponds to the pixel intensity values which have 8-bit resolution.

Since LR interested region of input LR frame is used, this region is extracted and written to a LR RAM. The signal *zoom_data_valid* shows the lines of LR interested region. Since the input frame is interlaced, there are two parts (even and odd lines) to be extracted from the frame as shown in Figure 33.

In parallel, input LR frame is used to calculate shift values as explained in Section 3.2.2. In the implementation, limited box is used. So, the shift values are calculated before whole frame ends. After the region determined by the limited box flows, the shift values are starting to calculate as shown in Figure 34.

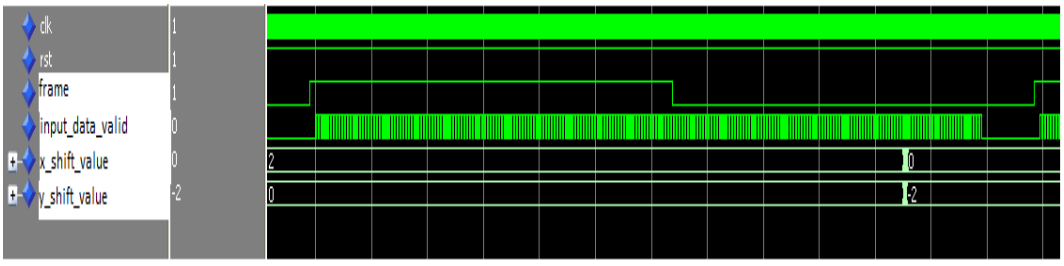


Figure 34 Shift values calculation in FPGA

The signals *x_shift_value* and *y_ shift_value* shows calculated shift values in x and y direction between current and previous frame respectively. In the implementation, the projection vectors are

upsampled by factor 2. The signals show the actual shift values multiplied by this factor. So, values *0* and *-2* in the figure mean that the estimated shift values are *0* and *-1* pixels between current and previous frames.

The iteration timings in Section 3.2.1 are shown in Figure 35. In the simulation, four iterations are used in order to reduce simulation time. The first one is performed to estimate initial HR interested region. The other three iterations are performed to update estimated HR interested region. The signal *hr_iteration_data_valid* indicates the lines of related iteration frame and the signal *hr_iteration_data* indicates the estimated HR interested region pixel intensity values. The pixel values at last iteration is written to output HR RAM as explained in Section 3.2.3.5.

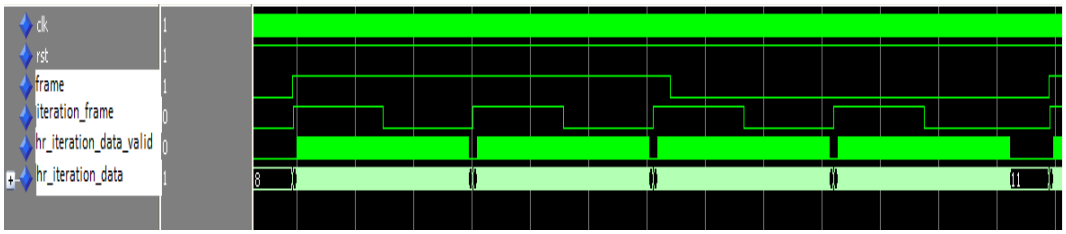


Figure 35 Iteration Timings

HR interested region estimated at previous frame is read from Output HR RAM at beginning of next frame period. This region is placed to the center of output grid and the other regions are filled with black as shown in Figure 36. Value *0* corresponds to black visually. The signal *sr_output_data_valid* shows the output lines of HR frame. It has same timing characteristics with signal *input_data_valid* except some pipeline delays are added due to pipeline delays of the blocks used in the structure. The signal *sr_output_data* shows the pixel values of HR frame. Since HR interested region is stored as progressive format in output HR RAM, it is converted to interlace format while being read.

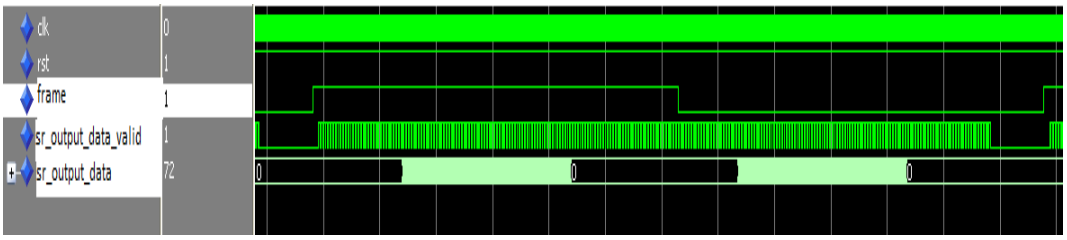


Figure 36 SR Output Timings