

Image Restoration of Degraded Images

(Super Resolution of Color Image from Multiple Low Resolution Images)



Author

Nasir Mahmood

2011-NUST-Ms PhD-Mts-28

Supervisor

Dr. Umar Shahbaz Khan

DEPARTMENT OF MECHATRONICS ENGINEERING
COLLEGE OF ELECTRICAL & MECHANICAL ENGINEERING
NATIONAL UNIVERSITY OF SCIENCES AND TECHNOLOGY
ISLAMABAD
JUNE, 2015

Image Restoration of Degraded Images
Super Resolution of Color Image from Multiple Low Resolution
Images

Author

Nasir Mahmood

2011-NUST-Ms PhD-Mts-28

A thesis submitted in partial fulfillment of the requirements for the degree of
M.Sc. Mechatronics Engineering

Thesis Supervisor:

Dr. Umar Shahbaz Khan

Thesis Supervisor Signature: _____

DEPARTMENT OF MECHATRONICS ENGINEERING
COLLEGE OF ELECTRICAL & MECHANICAL ENGINEERING
NATIONAL UNIVERSITY OF SCIENCES AND TECHNOLOGY,
ISLAMABAD
JUNE, 2015

Declaration

I certify that research work titled “*Image Restoration of Degraded Images - Super Resolution of Color Image from Multiple Low Resolution Images*” is my own work. The work has not been presented elsewhere for assessment. Where material has been used from other sources it has been properly acknowledged / referred.

Signature of Student

Nasir Mahmood

2011-NUST-Ms PhD-Mts-28

Language Correctness Certificate

This thesis has been read by an English expert and is free of typing, syntax, semantic, grammatical and spelling mistakes. Thesis is also according to the format given by the university.

Signature of Student

NASIR MAHMOOD

2011-NUST-Ms PhD-Mts-28

Signature of Supervisor

Copyright Statement

- Copyright in text of this thesis rests with the student author. Copies (by any process) either in full, or of extracts, may be made **only** in accordance with instructions given by the author and lodged in the Library of NUST College of E&ME. Details may be obtained by the Librarian. This page must form part of any such copies made. Further copies (by any process) of copies made in accordance with such instructions may not be made without the permission (in writing) of the author.
- The ownership of any intellectual property rights which may be described in this thesis is vested in NUST College of E&ME, subject to any prior agreement to the contrary, and may not be made available for use by third parties without the written permission of the College of E&ME, which will prescribe the terms and conditions of any such agreement.
- Further information on the conditions under which disclosures and exploitation may take place is available from the Library of NUST College of E&ME, Rawalpindi.

Acknowledgements

I am thankful to you, my Creator Allah Subhana-watala that you guided me throughout this work at every step and for every new thought which you setup in my mind to improve it. Indeed I could have done nothing without your priceless help and guidance, not even I breathe without your intention let alone other endeavors. Whosoever helped me, whether my parents or any other individual was your will, so indeed none be worthy of praise and you be worthy of all praise and thanks.

I am profusely thankful to my beloved parents who raised me when I was not capable of walk and continue to support me throughout in every department of my life.

Special Thanks to my supervisor Dr. Umar Shahbaz Khan for his help throughout my thesis. Thanks to Dr. Robina Ashraf, Dr. Umer Izhar and Dr. Khurram Kamaal for being on my thesis guidance and evaluation committee.

Finally, I would like to express my gratitude to all the individuals who have rendered valuable assistance to my study.

Dedicated to my exceptional Parents and Wife

*Their tremendous support and cooperation led me to this wonderful
accomplishment*

Abstract

The use of digital imaging systems is widespread in several applications. These applications include taking images for astronomical and meteorological uses, consumer photography and images taken aerially. It is important to restore distorted images taken for such applications. Theoretically and practically this process of restoration of images poses an interesting problem of image processing. There are many types of distortion: 1) Blurring - due to incorrect focus and movement 2) Zooming in – Due to zooming of images etc. In zooming problems we have to find the missing image points and that is a challenging problem.

Imaging systems often provides us low resolution images of the same scene. Every LR image got some information. Super resolution approach gathers this information from multiple LR images and combines it to get an HR image. Super resolution approach gathers this information from multiple LR images and combines it to get an HR image. Basic assumption is the availability of multiple images of the same scene to obtain different information in each LR image, such that some relative motion must exist between LR images or video sequence. The main objective of Super-Resolution (SR) is to recover a high resolution (HR) image from multiple Low resolution (LR) images such that zooming problem is solved.

Super-Resolution (SR) image reconstruction is the procedure of joining several spatially misaligned low resolution (LR) images into a single high resolution (HR) image. In this route accurate image registration is the key step and homography serves as basis for image registration. Homography estimation between LR images was done using Harris corner detector and Random Sampling Consensus (RANSAC). Harris corner detector finds the corner points of LR images and RANSAC finds the transformation matrix by filtering out mismatched points. A new SR approach is proposed in which LR images are then projected to HR grid using transformation matrix. New SR approach avoids iterative process for convergence making computation faster for better results. Remaining unfilled spaces in HR image are filled with interpolation.

KeyTerms: Super Resolution, Correlation, Homography, Image Registration, Harris corner detector, RANSAC, Bi-cubic interpolation.

Table of Contents

Declaration	iii
Language Correctness Certificate	iv
Copyright Statement	v
Acknowledgements	vi
Abstract	viii
List of Figures	xii
List of Tables	xiii
1 INTRODUCTION	1
1.1 Background – Image Registration.....	2
1.2 Background - Super Resolution	3
1.3 Basic observation model	6
1.4 Methodology	6
1.5 Objectives.....	7
1.6 Cataloging	7
2 HOMOGRAPHY	9
2.1 Geometric Transformations.....	9
2.1.1 Translation	10
2.1.2 Rotation.....	10
2.1.3 Scaling.....	10
2.1.4 Shearing	10
2.1.5 Similarity Transform.....	11
2.1.6 Affine Transformation	11
2.1.7 Projective Transformation	11
3 Image Registration	16
3.1 Harris corner detector.....	16

3.1.1	Basic idea	16
3.1.2	Mathematical equations	17
3.1.3	Summary	18
3.1.4	Implementation	20
3.2	Correlation Catching	21
3.3	RANSAC.....	22
3.4	Homography.....	23
4	Super Resolution Reconstruction.....	24
4.1	Projection onto Convex Sets (POCS).....	24
4.2	Iterated Back Projection (IBP)	25
4.3	Proposed Method.....	26
5	Experiments and Results.....	29
6	CONCLUSION	41
6.1	Future work	42
	References	44
	Appendix A	47

List of Figures

FIGURE 1.1: SUPER RESOLUTION DESCRIPTION 1

FIGURE 1.2: BASIC OBSERVATION MODEL 6

FIGURE 1.3: BASIC METHODOLOGY 7

FIGURE 2.1: HIERARCHY OF 2D TRANSFORMATIONS 12

FIGURE 3.1: HARRIS DETECTOR BASIC PRINCIPLE 16

FIGURE 3.2: ELLIPSE SHOWING DIRECTION OF CHANGE 18

FIGURE 3.3: CORNER DETECTION EXAMPLE 19

FIGURE 3.4: CORNER DETECTION CRITERIA 19

FIGURE 3.5: REFERENCE IMAGE CORNERS DETECTED 20

FIGURE 3.6: TEST IMAGE CORNERS DETECTED 20

FIGURE 3.7: AFTER CORRELATION MATCHING CORNER POINTS IN REF IMAGE 21

FIGURE 3.8: AFTER CORRELATION MATCHING CORNER POINTS IN TEST IMAGE 21

FIGURE 3.9: AFTER RANSAC SELECTED CORNER POINTS IN REFERENCE IMAGE 22

FIGURE 3.10: AFTER RANSAC SELECTED CORNER POINTS IN TEST IMAGE 23

FIGURE 4.1: ITERATIVE BACK PROJECTION METHOD 25

FIGURE 4.2: PROPOSED METHOD SCHEMATICS 26

FIGURE 5.1: LENA IMAGE 30

FIGURE 5.2: TEXT IMAGE 31

FIGURE 5.3: PROCESSOR FAN IMAGE 32

FIGURE 5.4: HARD DISK IMAGE 33

FIGURE 5.5: SWING IMAGE 34

FIGURE 5.6: PRAM IMAGE 35

FIGURE 5.7: TOWER IMAGE 36

FIGURE 5.8: LANDSCAPE IMAGE 37

FIGURE 5.9: HOUSE IMAGE 38

FIGURE 5.10: LINES IMAGE 39

FIGURE 5.11: TRUCK IMAGE 40

List of Tables

TABLE 1-1: IMAGE REGISTRATION COMPARISON	2
TABLE 1-2: SR MAIN CLASSIFICATION	4
TABLE 1-3: DIFFERENT SPATIAL DOMAIN SR APPROACHES	5
TABLE 5-1: LENA IMAGE STATISTICS	31
TABLE 5-2: FAN IMAGE STATISTICS	33
TABLE 5-3: HARD DISK IMAGE STATISTICS	34
TABLE 5-4: SWING IMAGE STATISTICS.....	34
TABLE 5-5: PRAM IMAGE STATISTICS.....	36
TABLE 5-6: CASING IMAGE STATISTICS	38
TABLE 5-7: TRUCK IMAGE STATISTICS	39
TABLE 6-1: PROPOSED METHOD COMPARISON.....	42

1 INTRODUCTION

Image restoration of degraded images is a very challenging and wide field of research area. Sometimes one has many low resolution samples of same scene but due to lack of resolution it cannot be magnified to great extent. Therefore our research work is to increase the resolution of an image from multiple images and called as Super Resolution reconstruction. The Figure 1.1 is shown below depicting Super Resolution process.

Image super resolution is a very challenging field and has many problems that need to be addressed. The main issue is the sensor size that limits the image resolution; it can be addressed by Super resolution [1, 2]. A large sensor size makes capacitance high and low pixel size increases noise due to low light availability.

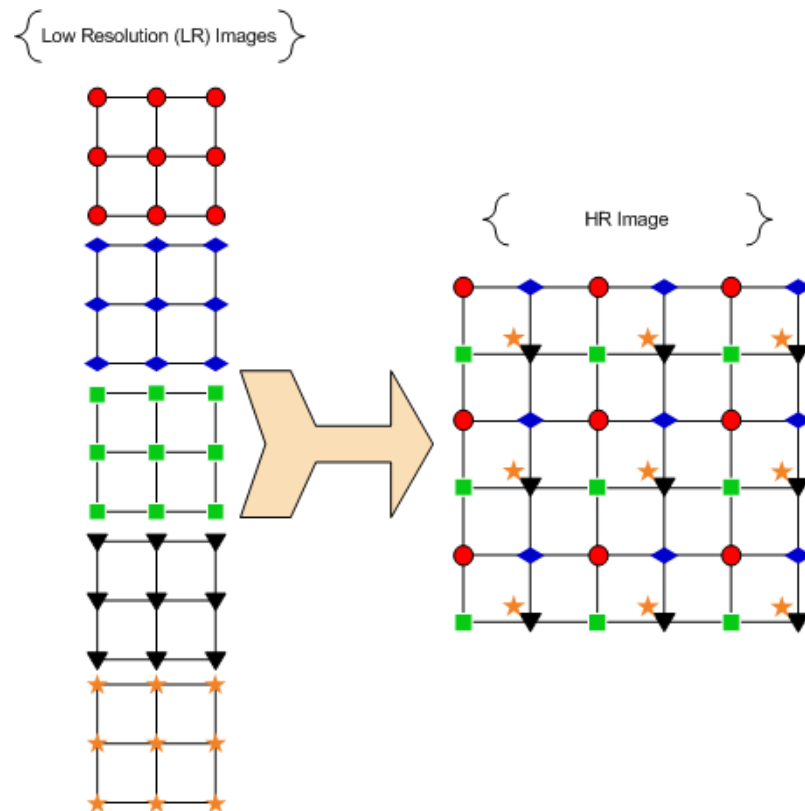


Figure 1.1:Super Resolution Description

The research work is mainly composed of two parts. First part is to obtain transformation matrix that brings spatial alignment between two images and called as Image registration. In this regard corners which are features of an image were selected using Harris corner detector and Random Sampling Consensus (RANSAC) finds the transformation matrix by filtering out mismatched points.

Second part is to obtain High Resolution (HR) image from multiple low resolution images and called as Super-Resolution (SR). The current research work includes a new SR method that registers two images using homography. After registering the LR images are projected to HR grid using projection matrix and zooming factor. The empty spaces in HR grid are then filled using Bi-cubic interpolation.

1.1 Background – Image Registration

Image registration is a very vast field and many algorithms have been developed in it. But it has two main types, first is area based and second is feature based [3]. Both have their advantages and disadvantages. Area based methods matches the image features rather their extraction e.g NCC (Normalized cross correlation). And feature based methods first extract features of an image e.g corners, edges etc. Then these features are used to find registration. Feature based methods have many advantages over area based methods like low memory requirement, higher computational efficiency and higher system flexibility. A comparison is given in Table 1-1. So a feature based image registration method using corners and projective transformation is used.

Table 1-1: Image Registration Comparison

Parameters	Method	
	Area Based Method	Feature Based Method
Robust to noise	High	Medium
Accuracy of Registration	High	High
Computational efficiency	High	Low
Memory Usage	High	Low
System Flexibility	Low	High

Corner detection main purpose is to calculate true corners instead of false corners also known as detection rate. As our method is based on corner detection, our first priority is to calculate true corners.

There are many different corner detection algorithms and the earliest is Moravec [4] corner detection. This method is based on self-similarity and corner is a point having low resemblance. This similarity is calculated by the sum of squared differences (SSD) among different patches. Trajkovic and Hedley [5] also proposed a corner detector, but its detection rate is less than 50 %.

In some cases computing the corner in an image with sub pixel accuracy is necessary. For approximate solution, the Förstner [6] algorithm derives it for the closest point to the entire lines tangent to the corner in a window by least square method. Simple solution for corner detection can be done by using correlation, but can be computationally expensive. There are many comparisons of corner detectors performed in [7, 8]. These comparisons show that Harris corner detector proposed by Harris and Stephens [9] finds the corner points of an image with more than 90% detection rate and is excellent for finding transformations due to its good repeatability rate. Therefore it is selected for this project work.

1.2 Background - Super Resolution

SR methods are broadly classified into two main categories i) Example based Super-Resolution (SR) and ii) Classical multi images Super-Resolution (SR) reconstruction. In example based SR, high and low resolution patches are learned after resemblance from a database of high and low resolution image pairs [10]. In this method, tiles which are combination of pixel are used instead of each pixel. In case of larger noise this method treats noise as signal and image degrades instead of improvement. Secondly a training set is required which contains low and high resolution patches. Many different improvements have come in the past.

Kim and Kwon [11] improved above method by including high frequency details using Bi-cubic interpolation and KKR (kernel ridge regression) instead of nearest neighbor. It smooths the artifacts in image due to regularization. Glasner and Irani [12] proposed a method that utilized both constraints from single image SR and classical SR reconstruction. But this approach has some assumptions that are suitable for natural image. Finally Yang and Lin [13] proposed a fast regression base model which learns from external database as well as self-similarity.

Work has been done in the past on classical SR reconstruction. Tsai and Huang [14] used frequency domain approach, shifting property of Fourier transform to find relation between LR and HR image. Though this technique is implemented in spatial domain, it is a frequency domain approach. This method is limited to translation motion and degrade model should be space invariant. Meanwhile Irani and Peleg [15] implemented the iterative back projection method but have no distinctive solution due to inverse problem, also for registration approach proposed by Keren [16] was valid for small displacements. Elad and Feuer [17] used sparse representation and proposed a hybrid method involving POCS and IBP. They used sparse representation to make it computationally efficient. Hsieh and Huang [18] continued the iterative back projection approach (IBP) initially proposed by Irani and Peleg but used Normalized cross correlation (NCC) as registration technique [19] instead of affine model proposed by Keren which was valid for small displacements. Keren model was formulated for small displacements by excluding nonlinear terms. They also extended the work for video super resolution but again it has no distinctive solution due to inverse problem. Secondly this is an iterative and slow method. The main concept of IBP is explained in chapter 4 and compared with proposed method.

Most of the methods that have been described are in spatial domain. In Table 1-2, comparison is given between SR frequency domain method and spatial domain method. Each method has its own advantages and disadvantages depending upon the scenario.

Table 1-2: SR Main Classification

Parameters	Frequency Domain Approach	Spatial Domain Approach
Observation model	Frequency Domain	Spatial Domain
Motion models	Global	Independent
SR Mechanism	Low	High
Computational efficiency	High	Low
A-priori information	Limited	Excellent
Versatility	Limited	Wide
Applicability	Limited	Unlimited

Projection onto convex sets (POCS) is another iterative approach first submitted by Stark and Oskoui [20] and used it to construct medical images and then continued by Tekalp et al.

[21]. They improved the above method computationally by introducing a blurring function in an efficient manner. This statistical method has slow convergence and high computational cost. Xie and Zhang [22] extended the work of Ozkan and Tekalp by imposing new constraints on Projection onto convex sets (POCS) which is widely used. In POCS method the main difficulty is to set proper projection function for convergence therefore it has slow convergence, high computational cost and non-uniqueness of the solution. Furthermore for image registration affine model is used that is limited to small displacements between images. Xiaoqing and Li [23] proposed SR reconstruction method that is based on POCS and also used sparse representation for learning self-similarity. Finally in chapter 4 POCS method is described in detail and compared with the proposed method.

Stochastic SR approach like Maximum Likelihood (ML) has also been used, but requires prior information for the solution of ill-natured problems. Elad and Feuer [24] suggested ML-POCS approach which is a Hybrid SR reconstruction technique. In POCS methods the main difficulty is to set proper projection function for convergence. Filip and Gabriel [25] proposed a unified approach to Super-Resolution and multichannel blind de-convolution based on energy minimization principle but here in case of large number of images HR image instead of improving can even worsen. Some other sophisticated SR approaches were also introduced.

Finally in Table 1-3 a comparison of some of SR techniques is given for better view. One can see clearly that MAP method requires prior PDF and it has its own complications.

Table 1-3: Different Spatial SR Approaches

Parameters	Bayesian (MAP)	IBP	POCS
A-priori information	Prior PDF	Back projection constant	Convex sets
SR solution	Unique	Non-unique	Non-unique
Computation requirement	Medium	Iterative – Medium	High
Convergence	Good	Medium	Medium
Optimization	High	Medium	Low
Complications	Non-convex priors Optimization	Back projection operator	Projection functions

Jian and Zongben [26] used gradient profile prior for Super-Resolution but under noisy conditions this method is inaccurate. Finally Nasir and Stankovic [27] proposed new registration method based on RANSAC and SIFT (Scale Invariant Feature Transform) and BP (Belief Propagation) and used super resolution algorithm of Elad and Milanfer [28]. This is stochastic approach based on ML estimators which must have prior knowledge and involves complexity. Various other SR techniques have been discussed [29].

1.3 Basic observation model

First step for thorough analysis of the SR image reconstruction is to devise an observation model that connects the HR image and LR images. Consider desired HR image X of dimension $M \times N$ and k number of LR images as Y having dimension $m \times n$. In common imaging system an original scene during capture suffers from warping due to scene motion or change of camera position M , motion and sensor blur B and down sampling factor D with noise n gives us LR image. This is the basic Observation model and is shown as a block diagram in Figure 1.2.

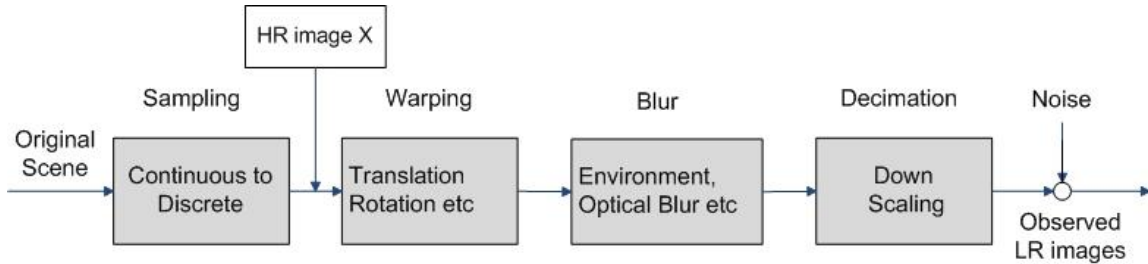


Figure 1.2:Basic Observation Model

In equation form it can be written as

$$Y_k = DB_k M_k X + n_k \quad 1.1$$

1.4 Methodology

As mentioned earlier, work has mainly two parts. First part is image registration and second part is acquiring HR image. Basic methodology is shown in Figure 1.3. In image registration Harris corner detector finds the corners and their correspondence is found by

correlation matching. Then by least square formulation and RANSAC, the projective matrix is found. This matrix is found for each test image. If there are five LR images then there are four projective matrixes. RANSAC selects the inliers and rejects outliers.

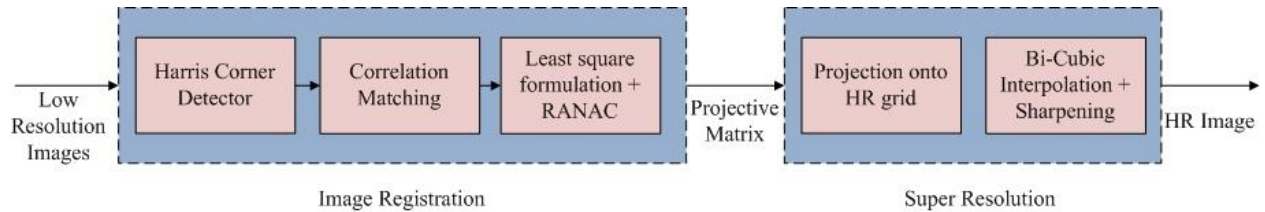


Figure 1.3:Basic Methodology

In second part, a new proposed method is used to project pixels from LR images to HR space. After that partial filled HR space is filled completely by Bi-Cubic interpolation. In the end sharpening is done to eliminate the softening effect of Bi-Cubic interpolation. In chapter proposed method is described in detail

1.5 Objectives

The following objectives in this research work will be completed.

1. To Obtain the High Resolution image (HR) from multiple Low resolution (LR) blurred and noisy images.
2. To do the feature based image registration between LR images by Homography.
3. Bring some novelty aspect in the proposed SR method that can ensure better performance of proposed method.
4. MATLAB implementation of the proposed SR method.
5. To verify the results by comparing with some of the existing techniques.
6. Quality analysis of the results on the basis of statistical measures.

1.6 Cataloging

In the second chapter basic image geometric transformations are discussed with homography and their mathematical modeling is shown. In third chapter image registration between reference and test images is done. First of all Harris corner detection mathematical model is derived in detail and implementation in MATAB is done. Then corners are further

classified into inliers and outliers using RANSAC. Finally projective matrixes are found with the equations from chapter two.

In fourth chapter, this projective matrix is used with our novel super reconstruction technique to yield Super Resolution image. IBP and POCS are also briefly explained. In chapter five experiments are conducted in MATLAB and their results are shown in correspondence to the IBP, POCS, bilinear and Bi-cubic interpolation. Some qualitative statistical analysis is carried out on obtained results. Conclusion is given in the end to describe the achievements and the future work.

2 HOMOGRAPHY

Homography is a direct mapping between two projection planes with the same center of projection. Image of planar objects taken by offset cameras are also related by it [30]. It has many applications in computer vision and used in mosaics, panoramas, removing perspective distortion from images and rendering textures.

The obtained LR images from the hardware have some transformations from each other in form of translation, rotation and shear etc. Sometimes knowledge of these transformations is not known. Purpose of homography is to find these transformation matrixes. In current chapter basic concept behind homography has been established in order to understand the problem. Furthermore two dimensional (2D) transformations have been discussed.

2.1 Geometric Transformations

Geometric transformations change the spatial pixels coordinates in an image. There are four types of geometric transformations in computer vision.

1. 2D to 2D (Image transformation between two images).
2. 3D to 2D (Camera model capturing world to image).
3. 2D to 3D (3D reconstruction from 2D).
4. 3D to 3D (Transformation between two different locations).

The images between them have image transformations (1) and our problem falls in this category. It may be a single transformation matrix or combination of two or more transformation matrixes. Main 2D image transformations are listed below. For simplicity only one pixel is assumed but in reality all the pixels in the image are transformed equally that result in global shifting.

2.1.1 Translation

Most basic form of transformation is translation. Consider a pixel original location is given by x and y and then translated by d_x and d_y in x and y direction respectively. This pixel new location x' and y' can be given by Equation 2.1.

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & d_x \\ 0 & 1 & d_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \quad 2.1$$

2.1.2 Rotation

Next form of image transformation is rotation and θ is the rotation angle. Consider an image pixel original location is given by x and y and then rotated by θ from origin. This pixel new location x' and y' can be given by Equation 2.2.

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} \cos \theta & -\sin \theta & 0 \\ \sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \quad 2.2$$

2.1.3 Scaling

Scaling affects the size of an image and is given by Equation 2.3. Where s_x and s_y scaling along x and y axis.

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} s_x & 0 & 0 \\ 0 & s_y & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \quad 2.3$$

2.1.4 Shearing

If all the pixels are not translated with the same magnitude then this produces shearing in an image. It may be in x or y direction.

Shearing in x -direction: x -coordinate shifts with magnitude proportional but not equal to the y -coordinate and given by Equation in 2.4.

Shearing in y -direction: y -coordinate shifts with magnitude proportional but not equal to the x -coordinate and given by Equation in 2.5.

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ e_x & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \quad 2.4$$

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & e_x & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \quad 2.5$$

2.1.5 Similarity Transform

Similarity transform is the concatenation of scaling, rotation and translation in any order of multiplication. We can join these transformations into a one transformation matrix. The equation 2.6 shows such transformation where rotation is followed by scaling and then translation. Similarity transform preserves the angles in an image.

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & d_x \\ 0 & 1 & d_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} s_x & 0 & 0 \\ 0 & s_y & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos \theta & -\sin \theta & 0 \\ \sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \quad 2.6$$

2.1.6 Affine Transformation

In affine transformation shearing is also a part of concatenation with scaling, rotation and translation. Affine transformation preserves the parallel lines but not the angles in an image.

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \quad 2.7$$

2.1.7 Projective Transformation

Consider an image pixel original location is x and y and its new location x' and y' then homography can be given by Equation 2.8. The homography relates the pixel coordinates in the reference and test images, when it is applied to each and every pixel the new image becomes original image's warped version. It is also called as projective transform. In next step, projective transform is solved and at least four point pairs are required [31].

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \quad 2.8$$

There are nine numbers from a_{11} to a_{33} that means this matrix has nine degree of freedoms. In equation form x' and y' can be written as Equation 2.9 and 2.10. Figure 2.1 shows some 2D transformations.

$$x' = \frac{a_{11}x + a_{12}y + a_{13}}{a_{31}x + a_{32}y + a_{33}} \quad 2.9$$

$$y' = \frac{a_{21}x + a_{22}y + a_{23}}{a_{31}x + a_{32}y + a_{33}} \quad 2.10$$






Transformation	Matrix	# DoF	Preserves	Icon
translation	$\left[\mathbf{I} \mid \mathbf{t} \right]_{2 \times 3}$	2	orientation	
rigid (Euclidean)	$\left[\mathbf{R} \mid \mathbf{t} \right]_{2 \times 3}$	3	lengths	
similarity	$\left[s\mathbf{R} \mid \mathbf{t} \right]_{2 \times 3}$	4	angles	
affine	$\left[\mathbf{A} \right]_{2 \times 3}$	6	parallelism	
projective	$\left[\tilde{\mathbf{H}} \right]_{3 \times 3}$	8	straight lines	

Figure 2.1: Hierarchy of 2D Transformations

It can be enforced to eight degree of freedom by two approaches. One approach is to set $a_{33} = 1$. Second approach is to impose a unit vector constraint written in Equation 2.11.

$$a_{11}^2 + a_{12}^2 + a_{13}^2 + a_{21}^2 + a_{22}^2 + a_{23}^2 + a_{31}^2 + a_{32}^2 + a_{33}^2 = 1 \quad 2.11$$

In first case Equation 2.9 and 2.10 can be modified to 2.12 and 2.13

$$x' = \frac{a_{11}x + a_{12}y + a_{13}}{a_{31}x + a_{32}y + 1} \quad 2.12$$

$$y' = \frac{a_{21}x + a_{22}y + a_{23}}{a_{31}x + a_{32}y + 1} \quad 2.13$$

Multiplying through by denominator and after rearranging Equation 2.14 and 2.15 are obtained.

$$a_{11}x + a_{12}y + a_{13} - a_{31}xx' - a_{32}x'y = x' \quad 2.14$$

$$a_{21}x + a_{22}y + a_{23} - a_{31}xy' - a_{32}yy' = y' \quad 2.15$$

There are two equations of each point pair. As in projective transform at least four points are required so there are total eight equations and in matrix form it can be written as in 2.16.

$$\begin{bmatrix} x_1 & y_1 & 1 & 0 & 0 & 0 & -x_1x_1' & -y_1x_1' \\ 0 & 0 & 0 & x_1 & y_1 & 1 & -x_1y_1' & -y_1y_1' \\ x_2 & y_2 & 1 & 0 & 0 & 0 & -x_2x_2' & -y_2x_2' \\ 0 & 0 & 0 & x_2 & y_2 & 1 & -x_2y_2' & -y_2y_2' \\ x_3 & y_3 & 1 & 0 & 0 & 0 & -x_3x_3' & -y_3x_3' \\ 0 & 0 & 0 & x_3 & y_3 & 1 & -x_3y_3' & -y_3y_3' \\ x_4 & y_4 & 1 & 0 & 0 & 0 & -x_4x_4' & -y_4x_4' \\ 0 & 0 & 0 & x_4 & y_4 & 1 & -x_4y_4' & -y_4y_4' \end{bmatrix} \begin{bmatrix} a_{11} \\ a_{12} \\ a_{13} \\ a_{21} \\ a_{22} \\ a_{23} \\ a_{31} \\ a_{32} \end{bmatrix} = \begin{bmatrix} x_1' \\ y_1' \\ x_2' \\ y_2' \\ x_3' \\ y_3' \\ x_4' \\ y_4' \end{bmatrix} \quad 2.16$$

It is formulated as a general case of matrix solution in Equation 2.17.

$$a = A^{-1}b \quad 2.17$$

If there are more than four points then we take pseudo inverse of matrix A instead of inverse as shown in equation 2.18.

$$a = (A^t A)^{-1}(A^t b) \quad 2.18$$

But if $a_{33} \sim 1$, answer obtained is incorrect, so second constraint is used in our experiments $\| a \| = 1$.

Now equation 2.8 and 2.9 can be modified to Equation 2.19 and 2.20.

$$a_{11}x + a_{12}y + a_{13} - a_{31}xx' - a_{32}x'y - a_{33}x' = 0 \quad 2.19$$

$$a_{21}x + a_{22}y + a_{23} - a_{31}xy' - a_{32}yy' - a_{33}y' = 0 \quad 2.20$$

In matrix form its general equation can be written as Equation 2.21. The difference between 2.16 and 2.21 is also in its size. In first case size of matrix A is $2n \times 8$ and in second case size of matrix A is $2n \times 9$, where n is the number of points and minimum requirement is four.

$$\begin{bmatrix} x_1y_1 & 1 & 0 & 0 & 0 & -x_1x'_1 & -y_1x'_1 & -x'_1 \\ 0 & 0 & 0 & x_1y_1 & 1 & -x_1y_1 & -y_1y'_1 & -y'_1 \\ & & & & & \vdots & & \\ & & & & & \cdot & & \\ & & & & & \cdot & & \\ x_ny_n & 1 & 0 & 0 & 0 & -x_nx'_n & -y_nx'_n & -x'_n \\ 0 & 0 & 0 & x_ny_n & 1 & -x_ny_n & -y_ny'_n & -y'_n \end{bmatrix} \begin{bmatrix} a_{11} \\ a_{12} \\ a_{13} \\ a_{21} \\ a_{22} \\ a_{23} \\ a_{31} \\ a_{32} \\ a_{33} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ 0 \end{bmatrix} \quad 2.21$$

The Equation 2.22 can be modified to 2.23

$$A h = 0 \quad 2.22$$

$$(A^t A) h = 0 \quad 2.23$$

$$\text{Svd}(A^t A) = U V U^t \quad 2.24$$

In final step shown in Equation 2.24 by singular value decomposition of $A^t A$ we get matrix h , Let h be the column of U (unit eigenvector) associated with the smallest Eigen value

in V . If there are only 4 points, that Eigen value will be 0. From equation 2.24 final results are obtained. After further rearranging, matrix h of size 9×1 is converted to 3×3 matrix. This final matrix is used in our calculations.

In the next chapter formulation derived above is used. How to calculate the control points and find their correspondences is discussed in detail. Furthermore image registration by above mentioned method is done using RANSAC.

3 Image Registration

In this chapter, image registration technique has been discussed in connection to the homography. This chapter explains the procedure for obtaining projective matrix (homography) that is used for projecting LR images to HR grid. First of all, Harris Corner Detector was used as feature extraction and then putative matches were found between different images. By using RANSAC, outliers were rejected. Next step is to find two dimensional image transformations. The real intention is to find out how they are connected with homography.

Corner detection is a method used for the extraction of features of an image. A corner is the junction of two edges. An interest point in an image has its own position and also can be detected robustly. This means; corner is an interest point or a curve point where curvature has local maxima.

3.1 Harris corner detector

It is the simple and effective corner detection method. Its detail is given below.

3.1.1 Basic idea

Basic idea behind corner detection is very simple. Place a small window on a point in an image and translate or shift it in all directions. If one finds no change, it is planar area and if change is in all directions, it is a corner. An edge gives no change in edge direction. This is shown in Figure 3.1.

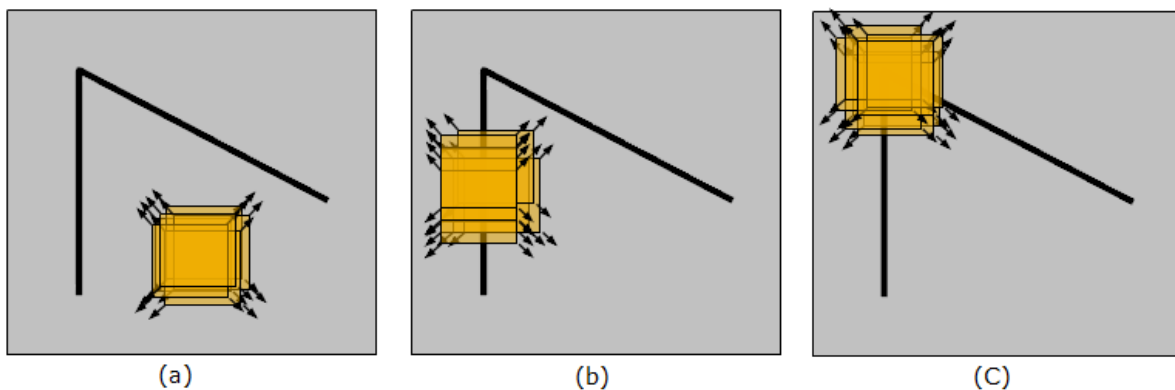


Figure 3.1: Harris Detector Basic Principle

3.1.2 Mathematical equations

In this section mathematical model of Harris corner detector is explained. Let W is a window and shifted by u, v . Then compare each pixel before and after by summing up the squared differences (SSD) of all the pixels under window. And $E(u, v)$ is the change after shifting window as shown in Equation 3.1.

$$E(u, v) = \sum_{(x,y) \in W} [I(x + u, y + v) - I(x, y)]^2 \quad 3.1$$

This equation is further simplified by Taylor series expansion.

$$I(x + u, y + v) = I(x, y) + \frac{\partial I}{\partial x} u + \frac{\partial I}{\partial y} v + \text{higher order terms} \quad 3.2$$

As small motion of window is considered, higher order terms can be neglected,

$$I(x + u, y + v) \approx I(x, y) + \frac{\partial I}{\partial x} u + \frac{\partial I}{\partial y} v \quad 3.3$$

In matrix form it can be written as

$$I(x + u, y + v) \approx I(x, y) + [I_x I_y] [u \ v]' \quad 3.4$$

Putting Equation 3.4 into Equation 3.1 results

$$E(u, v) \approx \sum_{(x,y) \in W} [I(x, y) + [I_x I_y] [u \ v]' - I(x, y)]^2 \quad 3.5$$

$$E(u, v) \approx \sum_{(x,y) \in W} [[I_x I_y] [u \ v]']^2 \quad 3.6$$

This can be rewritten as

$$E(u, v) = \sum_{(x,y) \in W} [u \ v] \begin{bmatrix} I_x^2 & I_x I_y \\ I_y I_x & I_y^2 \end{bmatrix} [u \ v]' \quad 3.7$$

From here we get

$$H = \begin{bmatrix} \sum I_x^2 & \sum I_x I_y \\ \sum I_y I_x & \sum I_y^2 \end{bmatrix} \quad 3.8$$

Matrix H is the covariance matrix of the gradient in the window and it can be used to differentiate between edges, corners and plain areas in an image.

By moving the window value of $E(u, v)$ changes and will result in the smallest and largest values. Its direction can be found by finding eigenvectors of H from Equation [32]. Shifts are dependents on $E(u, v)$ value and can be defined as largest or smallest change.

λ_+ = amount of increase in direction x_+

λ_- = amount of increase in direction x_-

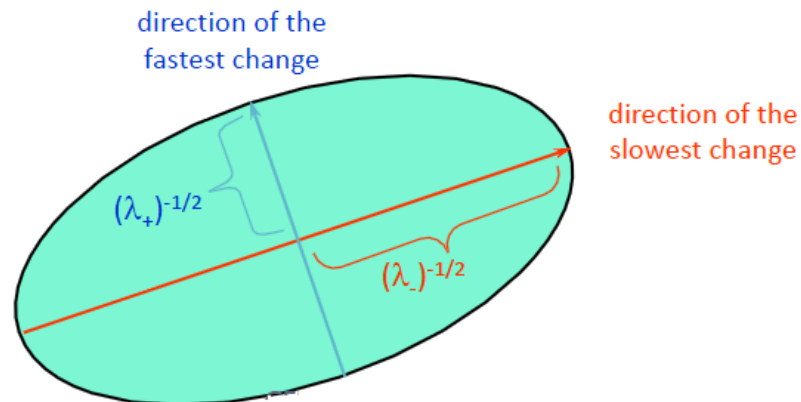


Figure 3.2: Ellipse showing direction of change

3.1.3 Summary

So we have to do these steps in order to find the corners

- Calculate the gradients of the image in x and y directions.
- Create the H matrix from the window in the gradient at each point.
- Compute the Eigen values.
- Find those points having greater response ($\lambda_- > \text{threshold}$).
- Select only points where λ_- is a local maximum.

Their depiction is also shown in the Figure

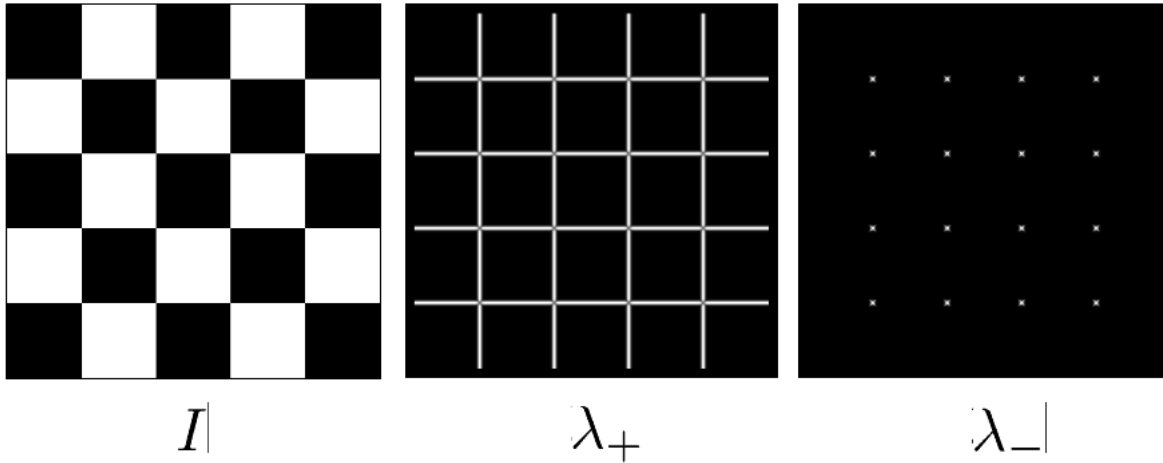


Figure 3.3:Corner Detection Example

Finally measure of corner response can be used by the Equation. Value of R decides whether a point is a corner, edge or flat area as shown in Figure 3.4.

$$R = \det(H) - K(\text{trace}H)^2 \quad 3.9$$

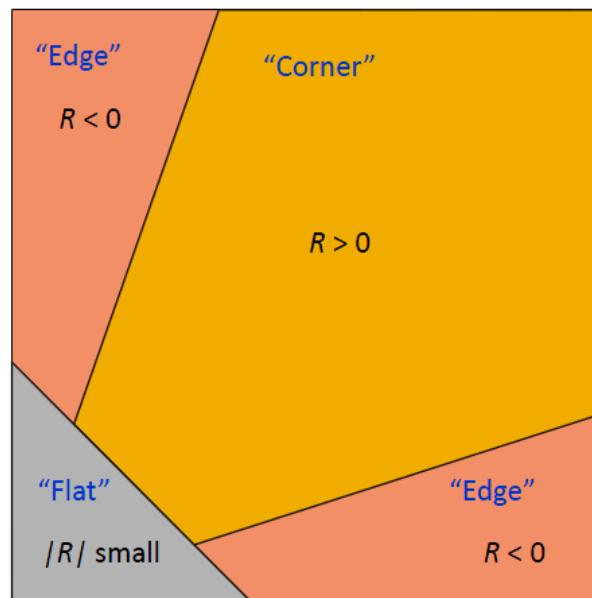


Figure 3.4:Corner Detection Criteria

3.1.4 Implementation

Grayscale images are used here for the corner detection. Two images are shown below in Figure 3.5 as reference image and Figure 3.6 as test image marked with “+” signs to show their corners. These images have translational offsets. Their corners are different and not equal in number. Harris corner detection was implemented in MATLAB. Code is given in APPENDIX A.

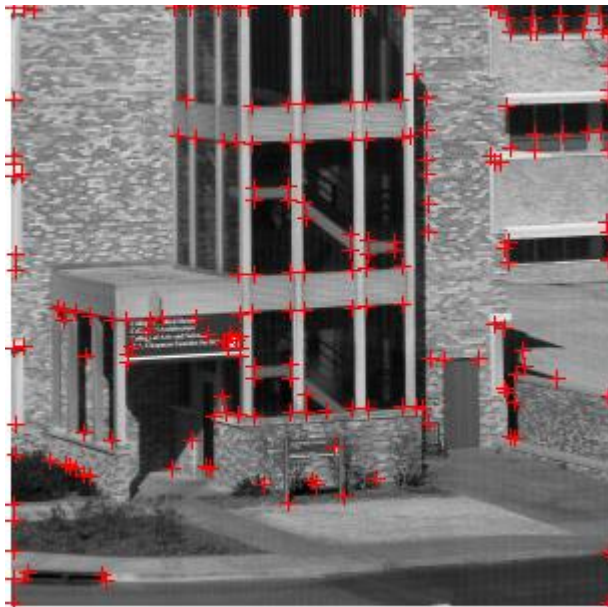


Figure 3.5:Reference image corners detected



Figure 3.6:Test image corners detected

3.2 Correlation Catching

To find the homography between these two images, those corners are required which are common in both images. Therefore correlation matching is done to have equal number of matched points. Circles in Figure 3.7 and Figure 3.8 are common points from Figure 3.5 and Figure 3.6 respectively.

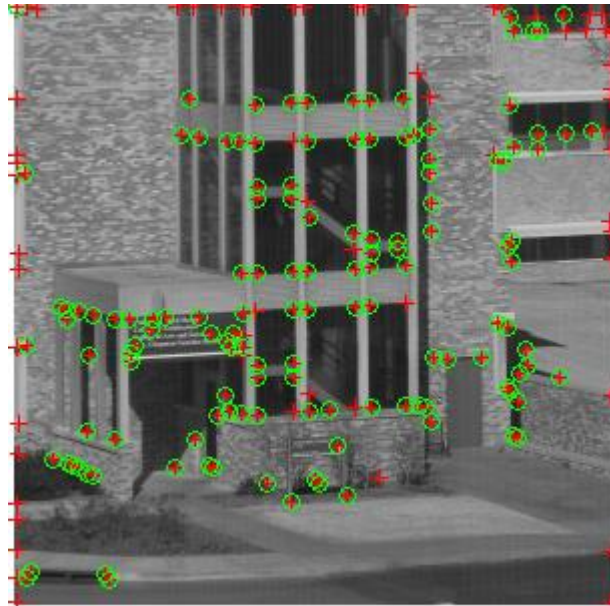


Figure 3.7:After correlation matching corner points in Ref image

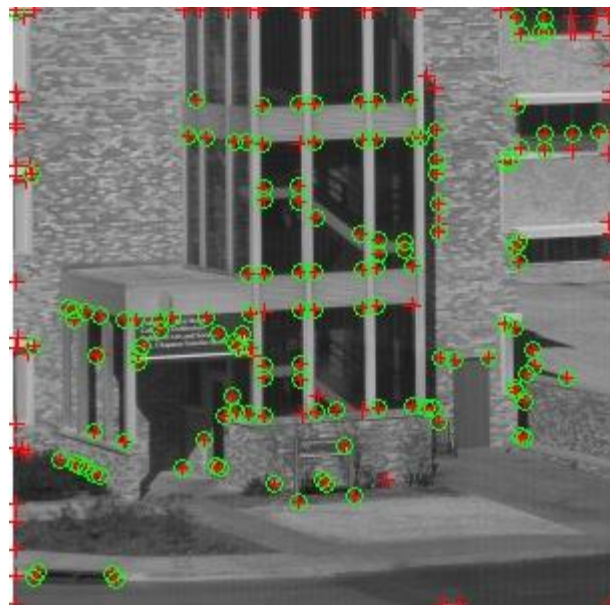


Figure 3.8:After correlation matching corner points in Test image

3.3 RANSAC

After correlation matching, next step is to further sort selected corner points using RANSAC. RANSAC algorithm randomly selects any four points then calculates the projective matrix by Equation 2.24. After that, their norm distance is calculated using projective matrix from line passing through those points. A distance threshold is self-selected and kept as 0.01. Points which are near to line and less than distance 0.01 are considered as inliers. Apart from this, number of iterations and minimum number of correct points or inliers are also self-selected. One thousands iteration were used and the above mentioned steps are carried out to find iteration having maximum number of inliers. In Figure 3.9 and 3.10 blue rectangles show inliers, while points in red “+” and green “o” are rejected points or outliers. These outliers are robustly rejected by RANSAC and thus making calculation easier and accurate.

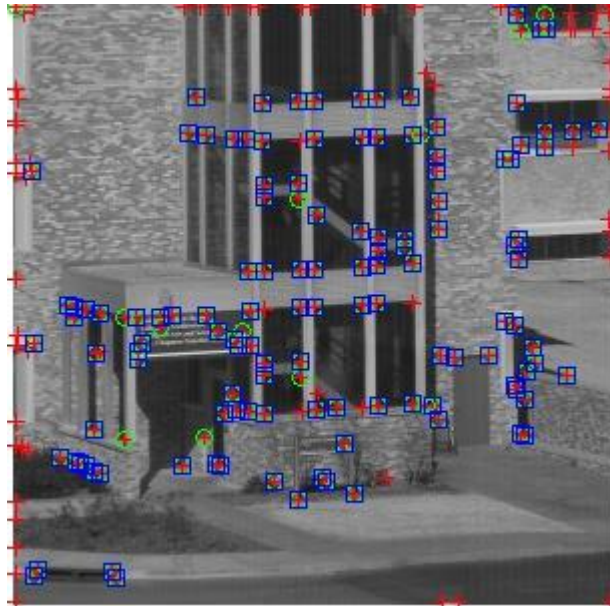


Figure 3.9:After RANSAC selected corner points in Reference image

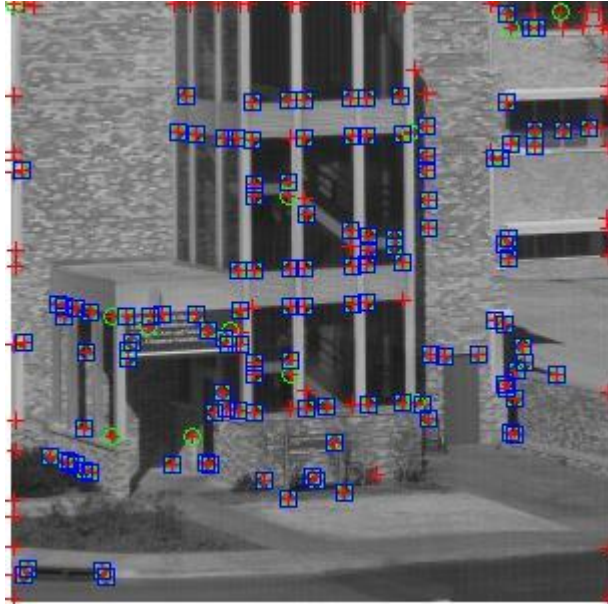


Figure 3.10:After RANSAC selected corner points in Test image

3.4 Homography

After getting iteration with four points having maximum number of inliers, once again use Equation 2.24 but this time with all the inliers instead of four points. This will give the projective matrix with maximum number of correct points also called as homography. All the images are registered with the reference image using this method. As homography is variant to scale, projective matrix for one frame is found twice. First at normal image scale and second by zooming image two times using Bi-cubic interpolation. Then from these two matrices, nearly scale invariant homography is found.

In the next chapter this projective matrix is used for image super resolution. The proposed method is described in detail in the next chapter.

4 Super Resolution Reconstruction

In this chapter SR reconstruction has been discussed in correspondence with homography. First of all, POCS (Projection onto convex sets) and IBP (Iterative back projection) are briefly described and then proposed method is discussed in the detail. As discussed in the previous chapter, after getting projective transform next step is to project each LR image to HR grid. This is where proposed method is different from other SR reconstruction methods.

4.1 Projection onto Convex Sets (POCS)

This method obtains the SR image from the intersection of convex set constraints such that HR image satisfies these constraints. The POCS method is an approach that incorporates solution prior knowledge for SR reconstruction. This method solves interpolation and restoration simultaneously for the estimation of SR image. According to this method a prior knowledge of the solution can be used to restrict the solution as a member of the closed convex sets, which are defined to satisfy particular property [20]-[22]. Any solution that is consistent with the prior convex sets is a reasonable solution.

In other words first of all convex sets are defined using various constraints, then starting values are mapped with iterative scheme and their intersection which is also a convex set is found. In equation form it can be written as Equation 4.1 and 4.2

$$C = \bigcap_{i=1}^m C_i \quad 4.1$$

$$Y^{n+1} = P_m P_{m-1} \dots P_2 P_1 Y^n \quad 4.2$$

Here Y is a starting point and P is a projection operator that projects Y onto convex sets C_i ($i = 1, 2, \dots, m$). In super resolution many constraints can be included such as data consistency, pixel values, bounded energy etc. Equation 4.2 is an iterative process that uses projection operators assuming that motion information or image registration is accurate. It is simple method that uses spatial model but suffers from uniqueness of the solution and slow convergence speed, therefore having high computational cost.

4.2 Iterated Back Projection (IBP)

Iterative Back Projection is an iterative method for super resolution reconstruction. Its main process is shown in Figure 4.1. Suppose we have observed images of low resolution. After getting an initial guess of High Resolution (HR) image, low resolution images are simulated from it. Error is found between original LR images and simulated LR images. Then error is minimized by back projecting each difference value in corresponding field in initial guess. After every iteration, guessed HR image is modified such that difference between simulated and observed images is minimum. The iterative scheme is shown in Equation 4.3. $f^{(n+1)}(x)$ is the HR image after n iterations.

$$f^{(n+1)}(x) = f^{(n)}(x) + \sum_{y \in U_k Y_{k,s}} (g_k(y) - g_k^{(n)}(y)) \frac{(h_{xy}^{BP})^2}{c \sum_{y' \in U_k Y_{k,s}} h_{x,y'}^{BP}}, \quad 4.3$$

This method is iterative so it can diverge also leading to distorted image. Results of proposed method are compared with this method also.

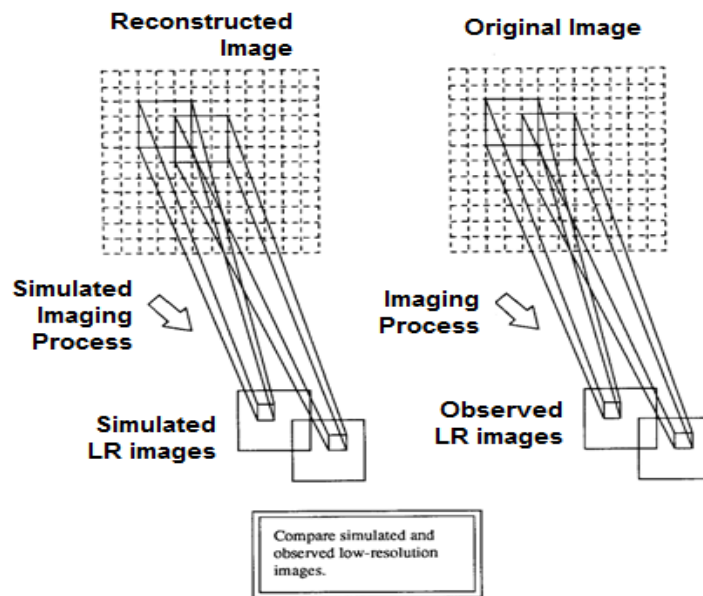


Figure 4.1: Iterative Back Projection Method

4.3 Proposed Method

After obtaining projective transform as described in chapter 3, next step is to fill HR grid from LR images. For this purpose a simple novel approach that is computationally inexpensive has been used. Pixel locations (indexes) and zooming factor are used as a reference for projection. For this cause accurate projective matrix calculation is crucial. Proposed Scheme is shown in the Figure 4.2.

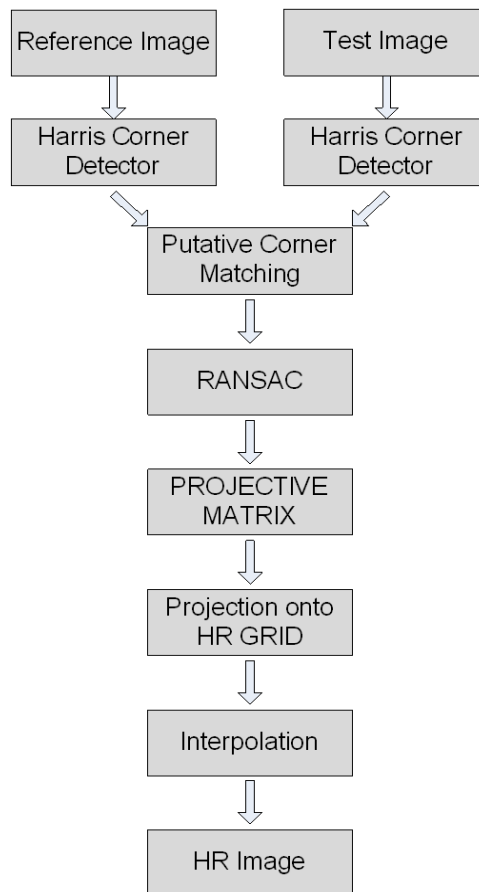


Figure 4.2:Proposed Method Schematics

Equation 4.4 illustrates the projective matrix calculated from Equation 2.24 for one test image and reference image. Similarly all the test images will have their own projective matrixes and their pixel location will be different as well.

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & -4 \\ 0 & 0 & 1 \end{bmatrix} \quad 4.4$$

In first step, reference image pixel locations matrix is multiplied with each corresponding LR image projective matrix to obtain new pixel locations of each LR image. This shifting is global so that all the image pixel locations of an image are shifted with the same magnitude. All these pixel locations are then multiplied by zooming factor and HR image pixel locations are obtained for all the test images. The only purpose to obtain these pixel locations is to get original pixel values.

In second step, each LR color image is projected to HR color image grid using HR pixel locations obtained in first step. Gray scale images are taken for homography calculation but real images are colored. That means all the color channels are projected with same relation. So all three color channel values (RGB) are projected separately to HR image corresponding color channels. Some location may be empty but others can have more than one pixel. For this purpose we take medians of two or more pixels mapping to same location in HR grid, if there are more images, it will reduce the effect of noise and blurring. Locations in the image are integer values but the new pixel locations after above process can be float values. For the purpose of keeping original pixel values, instead of interpolating pixel values to integer locations, pixel location nearest to integer locations with the threshold $\epsilon = 0.2$ is found. If a new pixel location is (1.1, 5.2) we consider it (1, 5) and if a new pixel location is (1.4, 5.3) it is rejected.

After the projection of pixels onto HR grid, the HR space is not filled completely. There are some empty spaces. In order to fill these pixel locations Bi-cubic interpolation is done at locations where no pixel is mapped by finding unfilled locations. This is the main advantage of the proposed method that we project original values that leads to better results instead of projecting interpolated values. Secondly, there is no iterative process for convergence which makes computation low and makes our method robust. As Bi-cubic interpolation softens the

image slightly we finally do some sharpening with Laplacian filter [33] as a post processing of HR image using un-sharp masking method. Here the filter is 3x3 square matrix as written in Equation 4.5. This filter is convolved with HR image and the resultant image is subtracted from the original HR image. After subtraction we get sharp image. This is the Final HR image. Its complete code is given in APPENDIX A.

$$w = \begin{bmatrix} 0 & 0.5 & 0 \\ 0.5 & -2 & 0.5 \\ 0 & 0.5 & 0 \end{bmatrix} \quad 4.5$$

As one can see in first two methods, first initial guess is calculated and then SR reconstruction is based on that guess. So, before the start of SR reconstruction original pixel values are changed.

5 Experiments and Results

In this section we verify our proposed method by comparing results with some other commonly used techniques. Simulation was run in the MATLAB using zoom factor 4. We also use some common statistical parameters to estimate image quality. These estimates include RMSE (root mean square error), PSNR (peak signal to noise ratio) and SSIM (structure similarity index) [34]. RMSE lesser and PSNR higher value represents good results. Similarly SSIM value is 1 if images are identical, SSIM values close to and less than 1 represents better quality.

MSE can be given by Equation 5.1 where $f(i, j)$ is the original image, $F(i, j)$ is the test image and r, c represents rows and columns respectively. RMSE is the root square of MSE. PSNR can be given by Equation 5.3 and it is in decibels. PSNR sometimes can be poor in perceptual quality therefore SSIM is used as counter check and their MATLAB code is given in APPENDIX A.

$$MSE = \frac{1}{r * c} \sum_{i=1}^r \sum_{j=1}^c [f(i, j) - F(i, j)]^2 \quad 5.1$$

$$RMSE = \sqrt{MSE} \quad 5.2$$

$$PSNR = 20 \log_{10} \left(\frac{255}{RMSE} \right) \quad 5.3$$

Ten cases were considered for the experiments. A both indoor and outdoor image data set has been selected for experiments. First was Lena image as ideal case with no noise and blur. Four images of 256 x 256 were generated from a 512 x 512 image by selecting alternate pixels. For POCS method these LR images were registered by affine method and we used homography as registration in our method to see the effect of image registration. Then HR image 1024 x 1024 was constructed by these methods.



Figure 5.1:Lena Image

We compared the results of our method Figure 5.1(e) with Bi-linear interpolation Figure 5.1(a), Bi-cubic interpolation Figure 5.1(b), Iterated Back Projection (IBP) Figure 5.1(c) and Projection onto convex sets Figure 5.1(d) with original image 5.1(f). Results are shown below. We can see that in Figure 5.1(e) image registration is correct and far better than Figure 5.1(d). Secondly SR image constructed by Figure 5.1(e) has more detail and stronger edges which are

clearly visible e.g. eye lashes and hat in Figure 5.1(e). Similarly image in Figure 5.1(e) is sharper and brighter than all other images and closest to original image.

Table 5-1 shows statistical results of LENA image. Proposed method has minimum RMSE and higher PSNR value that shows proposed method is better than other methods. Similarly SSIM index value is higher than competitors.

Table 5-1: Lena Image Statistics

	BL	BC	IBP	POCS	PROPOSED
RMSE	5.44	5.60	8.59	8.86	4.67
PSNR	33.45	33.20	29.46	29.19	34.80
SSIM	0.71	0.70	0.38	0.36	0.73

In second case, ten offset images of 300x300 were taken. These images have translational offsets between each other. These images include atmospheric and camera sensor blurring and slight noise as well. In this case we used homography as registration for POCS, IBP and our method to see the efficacy of the proposed method. SR image was created using all above mentioned methods and then compared with our method. Results are shown below.

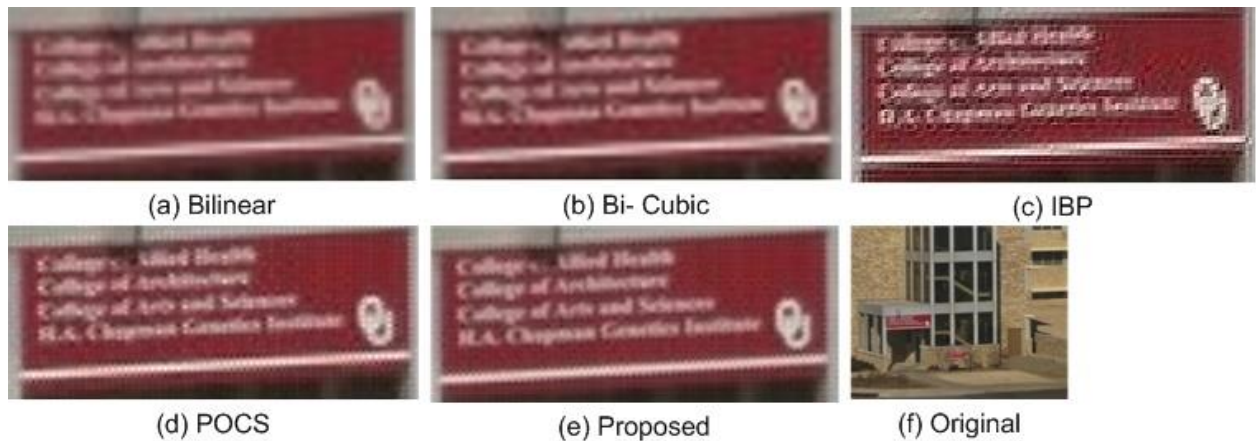


Figure 5.2:Text Image

Our method Figure 5.2(e) outperforms Figure 5.2(d) and 5.2(c) in detail and sharpness. Due to minimum noise and artifacts in Figure 5.2(e) it has clearer edges and color than 5.2(d). Results given in Figures below demonstrate that our proposed method recovers the image

sharpness and detail with minimum artifacts. Text in Figure 5.2(e) is also clear than other methods.

In third case, five images of Processor fan were taken and their results are shown in Figure 5.3. One can clearly see that proposed method Figure 5.3(e) is close to original image and noise is also under control. Furthermore statistical analysis given in Table 5-2 suggests that in this scenario IBP could not perform. But in Figure 5.4 IBP result is better than POCS, and Table 5-3 confirms it.

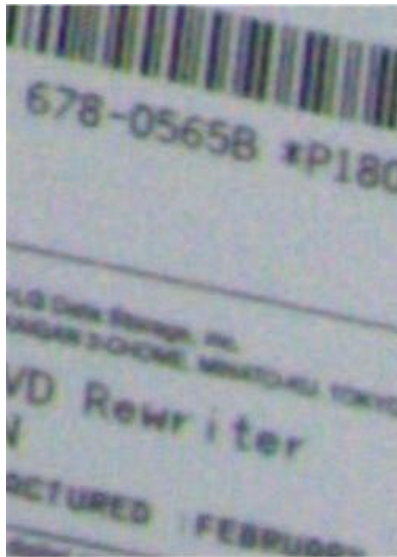


Figure 5.3:Processor Fan Image

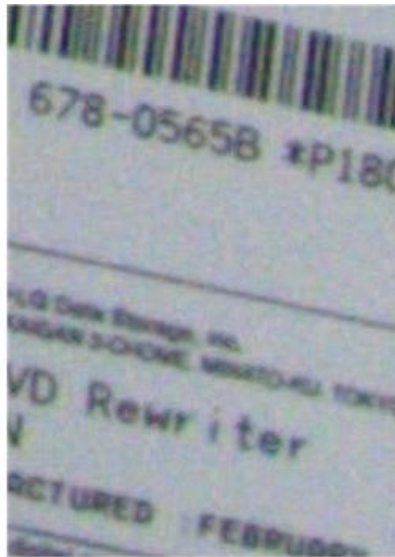
Table 5-2 depicts statistical performance of proposed method and shows superior results.

Table 5-2: Fan Image Statistics

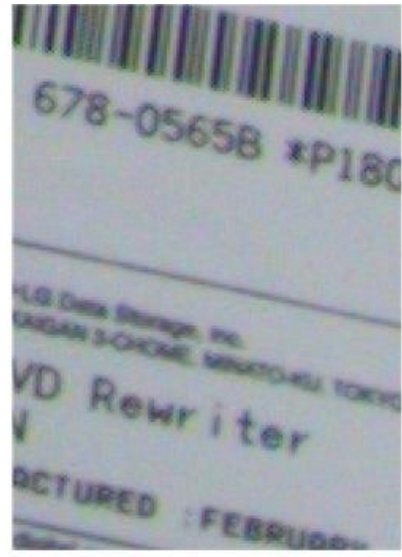
	BL	BC	IBP	POCS	PROPOSED
RMSE	4.36	4.26	6.13	4.05	3.20
PSNR	35.51	35.75	32.40	36.22	38.18
SSIM	0.78	0.79	0.57	0.80	0.86



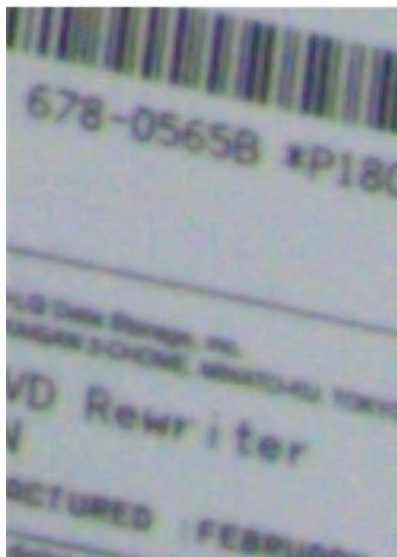
(a) Bilinear



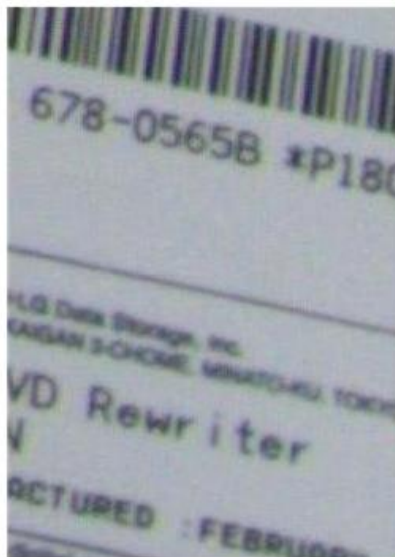
(b) Bi-Cubic



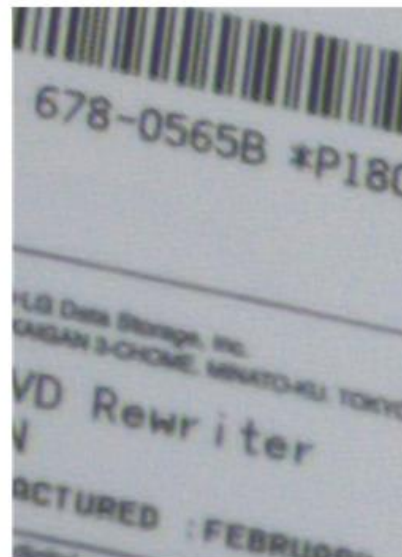
(c) IBP



(d) POCS



(e) Proposed



(f) original

Figure 5.4: Hard Disk Image

Table 5-3: Hard disk Image statistics

	BL	BC	IBP	POCS	PROPOSED
RMSE	7.43	7.21	6.99	7.73	6.96
PSNR	30.75	31.01	31.32	30.41	31.33
SSIM	0.82	0.85	0.86	0.83	0.86

In next sample seven snaps of a child swing has been taken under normal condition and their results are compared with original image. Our method again out performs other methods and closest to original image. Results are shown in Figure 5.5 and statistical analysis is given in Table 5-4. Statistical analysis and visual quality shows advantage.

Table 5-4: Swing Image Statistics

	BL	BC	IBP	POCS	PROPOSED
RMSE	8.69	8.75	7.62	7.09	6.89
PSNR	29.37	29.30	30.49	31.12	31.37
SSIM	0.53	0.52	0.58	0.61	0.62

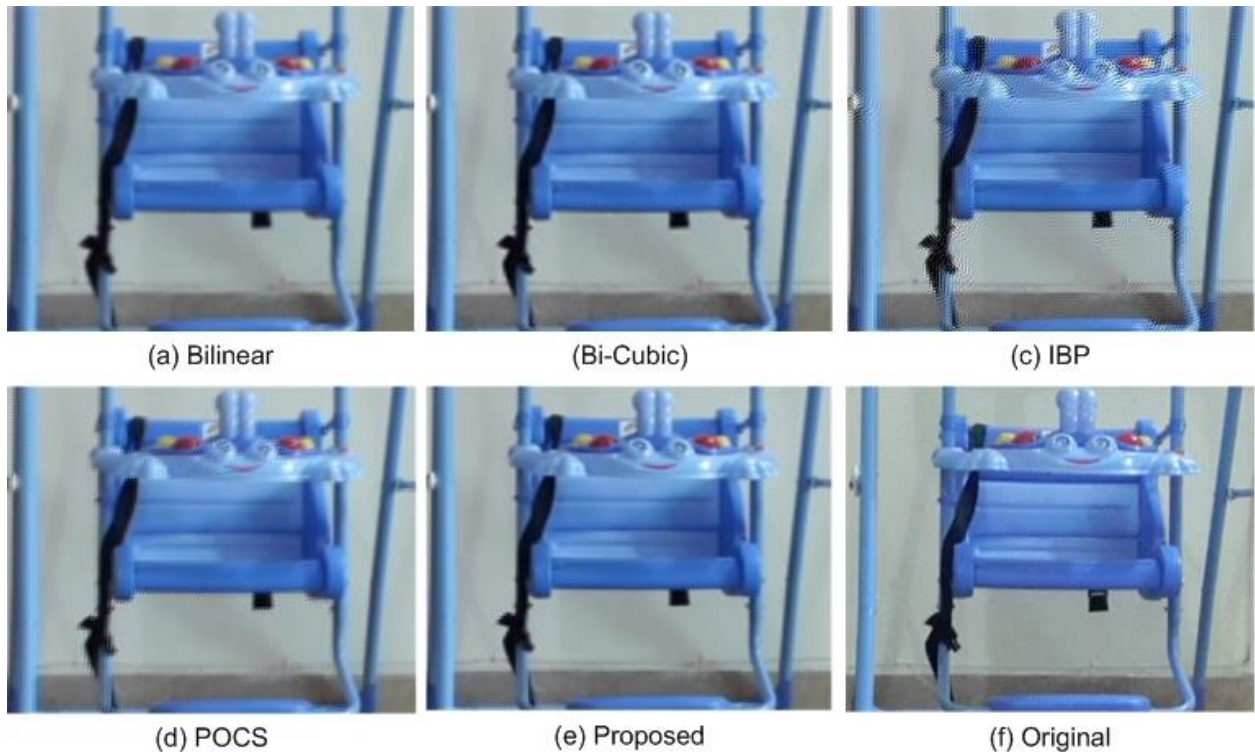


Figure 5.5:Swing Image

Similarly in the next example six snaps of a child pram were taken and all the above mentioned algorithms were applied. Results are shown in Figure 5.6, images obtained by both interpolation techniques are very softer and there are artifacts in images obtained by POCS and IBP. Proposed method is closest to original image; their statistical data given in Table 5. 5 suggest the same.

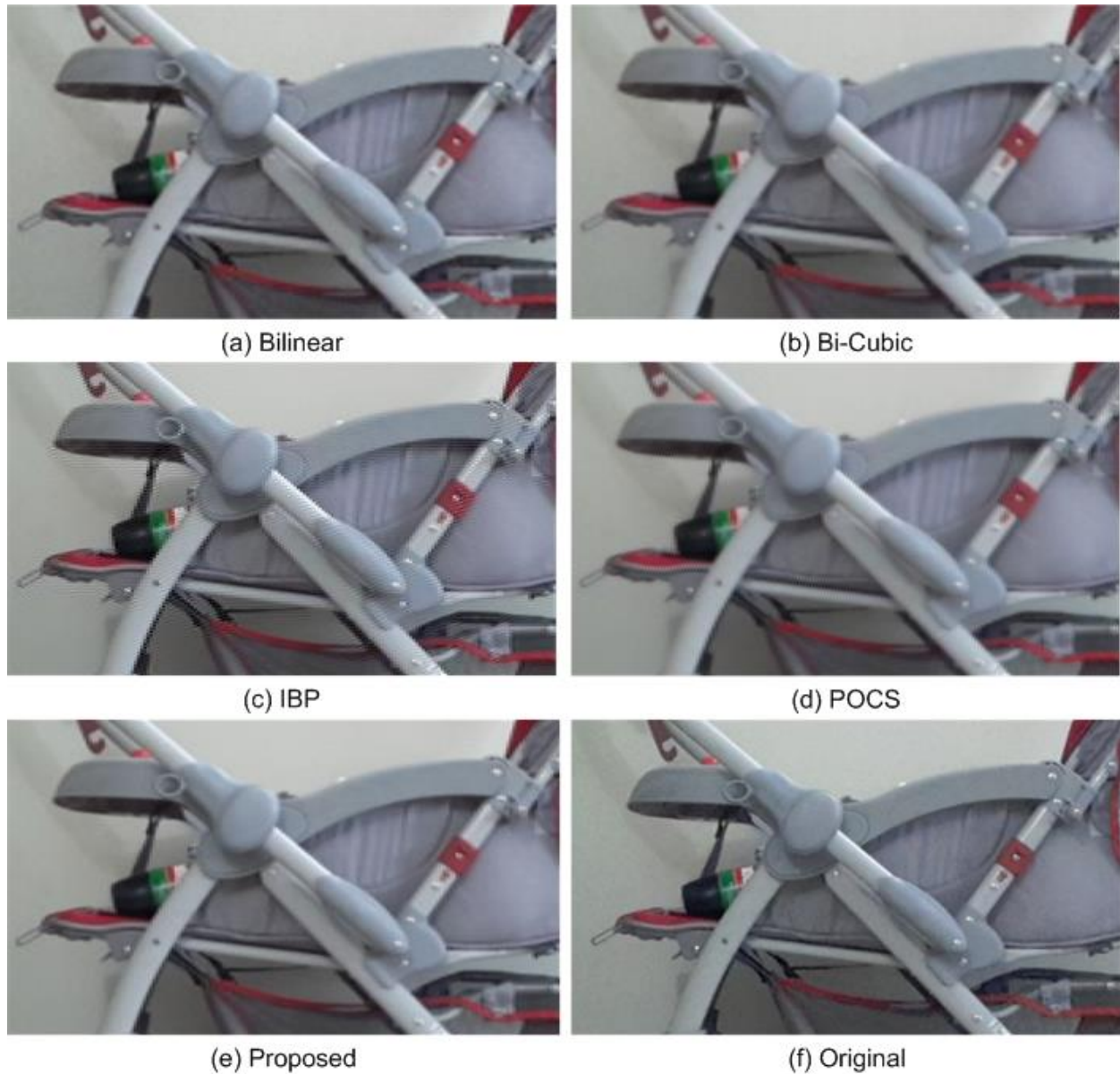


Figure 5.6: Pram Image

Table 5-5: Pram Image Statistics

	BL	BC	IBP	POCS	PROPOSED
RMSE	4.59	4.72	5.48	4.70	4.51
PSNR	34.90	34.66	33.36	34.59	35.01
SSIM	0.60	0.60	0.57	0.68	0.70

In next experiment four images of a castle were taken to conduct experiment. In this case the more noticeable thing is the presence of some high frequencies in each low resolution image which only cannot be recovered from one image by interpolation. This is also known as aliasing effect. SR reconstruction techniques tend to recover high frequency details because they gather information from all LR images.

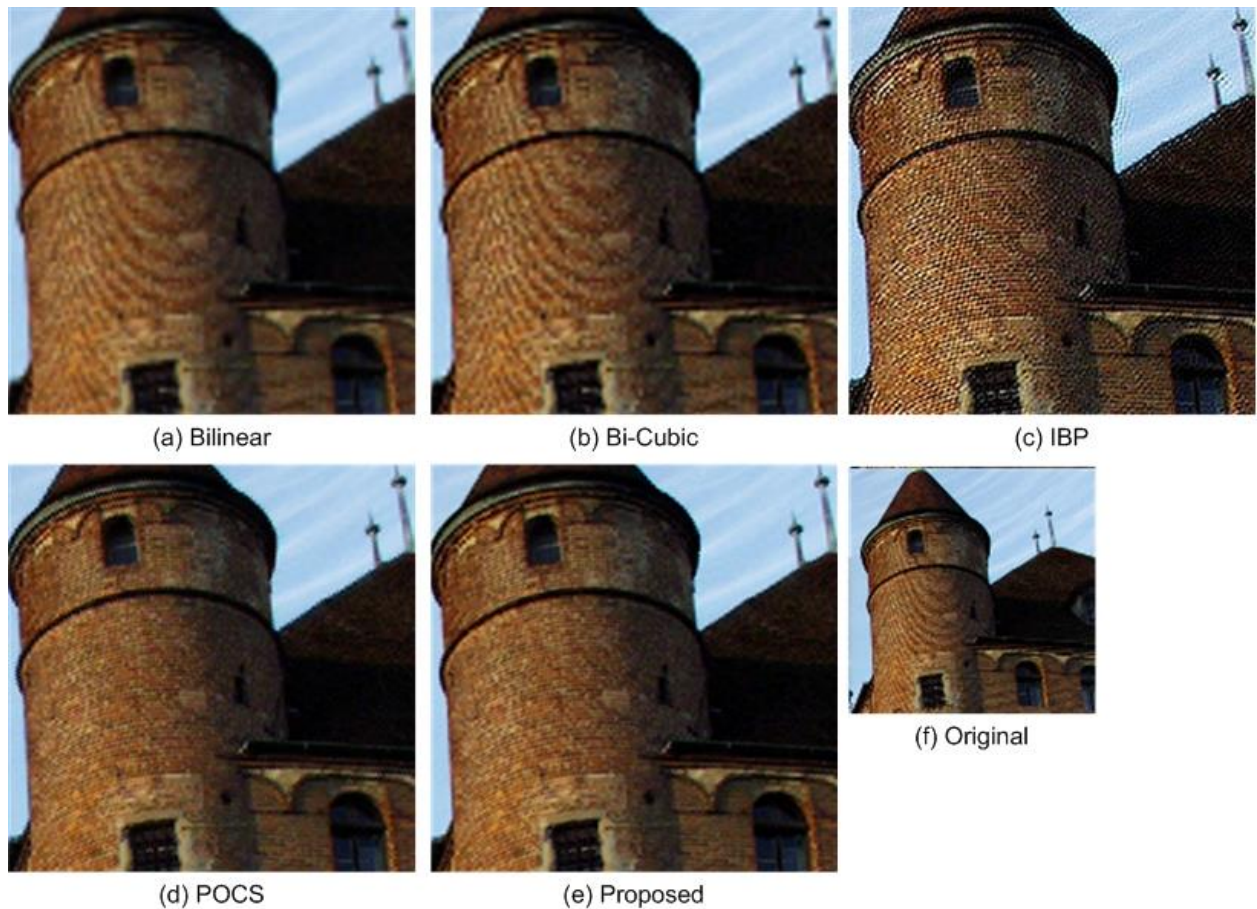


Figure 5.7: Tower Image

As far as results are concerned IBP method in this case produces high noise image and closest match is between POCS and proposed method. But POCS method in this scenario could not produce sharp edges that can be seen in Figure 5.7. Also due to Gaussian operator POCS is slightly blur. The circles in the Figure 5.7 (a) and (b) on the tower are visible because of the absence of high frequencies.

Some outdoor images are included to show the efficacy of proposed method. Six images were taken in these scenarios and results are shown in Figure 5.8 and Figure 5.9. These images were self-taken by mobile phone of a landscape view. In this case proposed and POCS method are closely related in image quality. Our proposed method performs slightly better by keeping noise under control while POCS suffers on edges. Images produced by interpolation are very soft and lacks detail that is clearly visible in the text.



Figure 5.8: Landscape Image

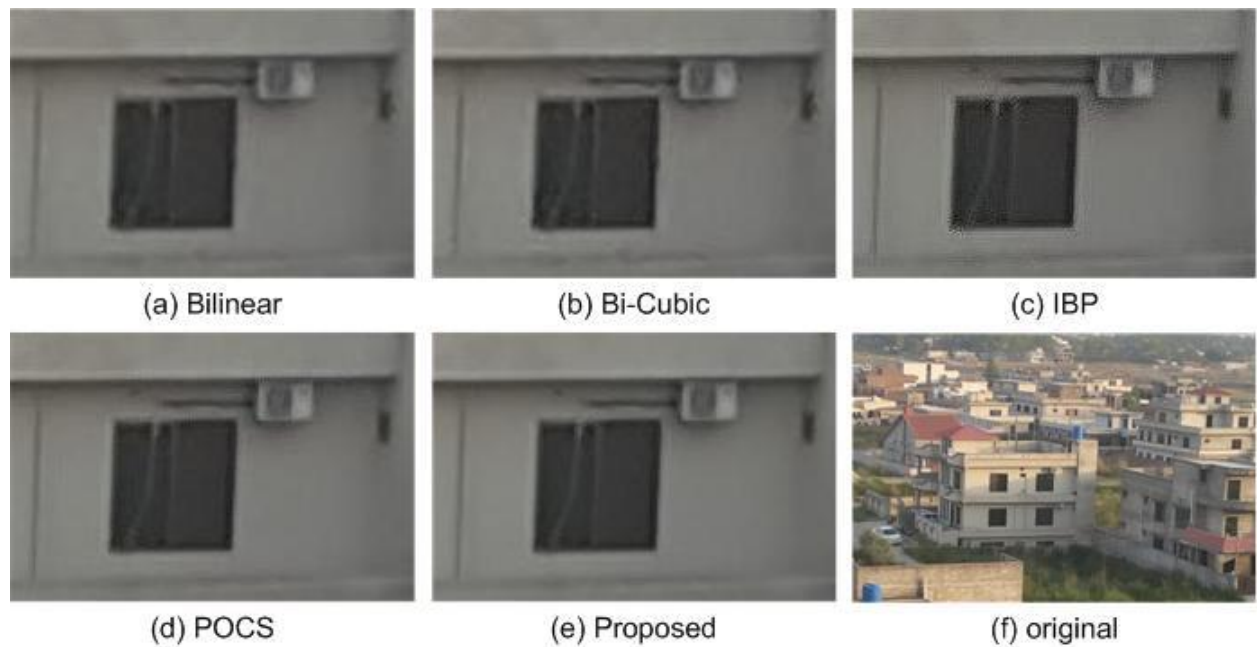


Figure 5.9:House Image

In next case five images were taken of inner part of casing and the cropped results are shown in Figure 5.10. Statistical table is given in Table 5-6. The vertical lines and text in Figure 5.10 (e) is much clearer than all counterparts. Similarly noise is less and edges are sharp and clearer. RMSE is smaller and PSNR is higher in the proposed method.

Table 5-6: Casing Image Statistics

	BL	BC	IBP	POCS	PROPOSED
RMSE	7.40	7.51	7.20	7.25	7.01
PSNR	30.76	30.63	30.98	30.92	31.23
SSIM	0.69	0.68	0.72	0.71	0.74

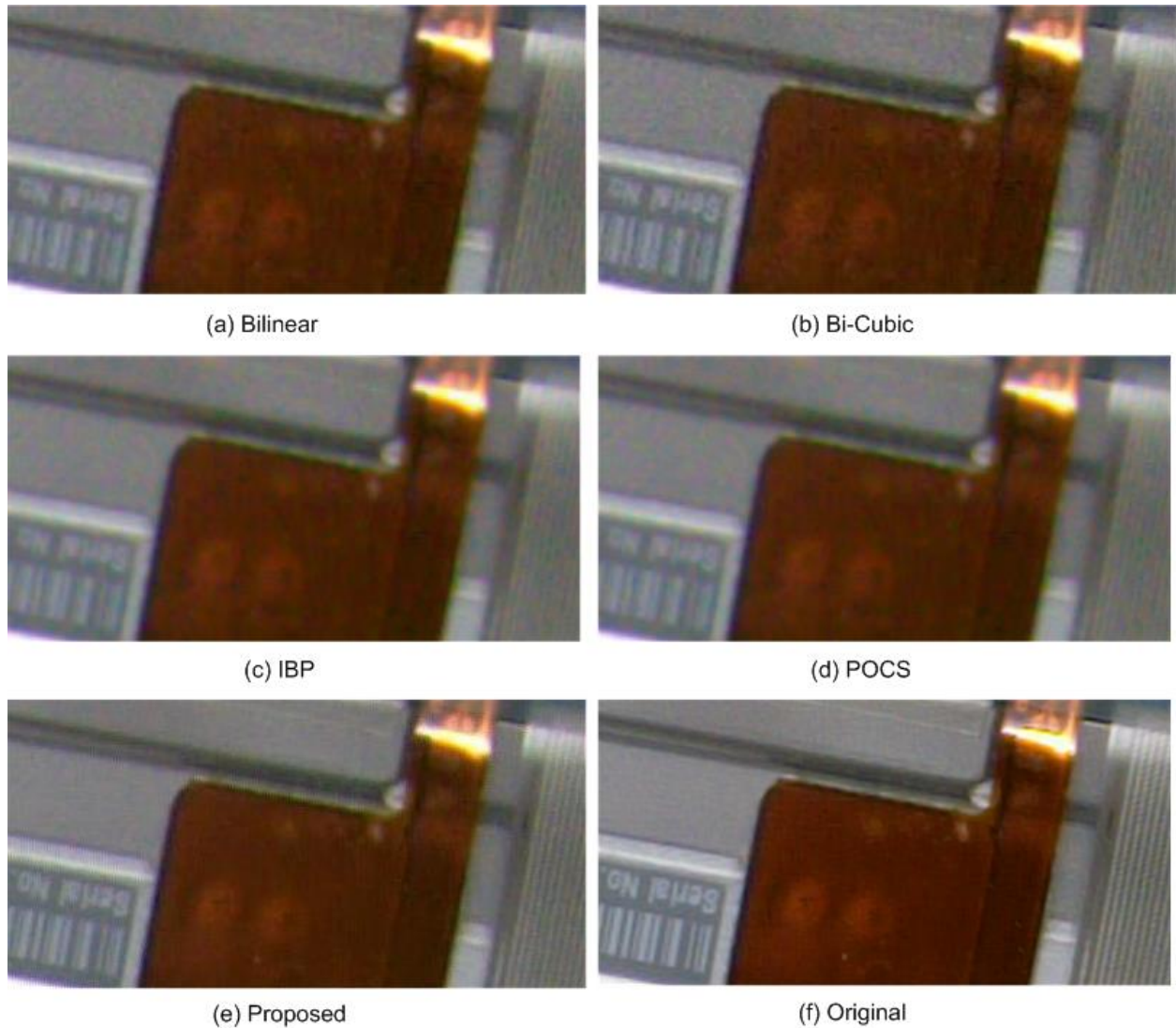


Figure 5.10:Lines Image

In last case we have taken four images of an outdoor scene and results from different methods are obtained and shown in Figure 5.11 and their analysis in Table 5-7

Table 5-7: Truck Image Statistics

	BL	BC	IBP	POCS	PROPOSED
RMSE	5.34	5.33	5.25	5.20	5.14
PSNR	33.61	33.63	33.73	33.81	33.91
SSIM	0.83	0.84	0.84	0.85	0.85

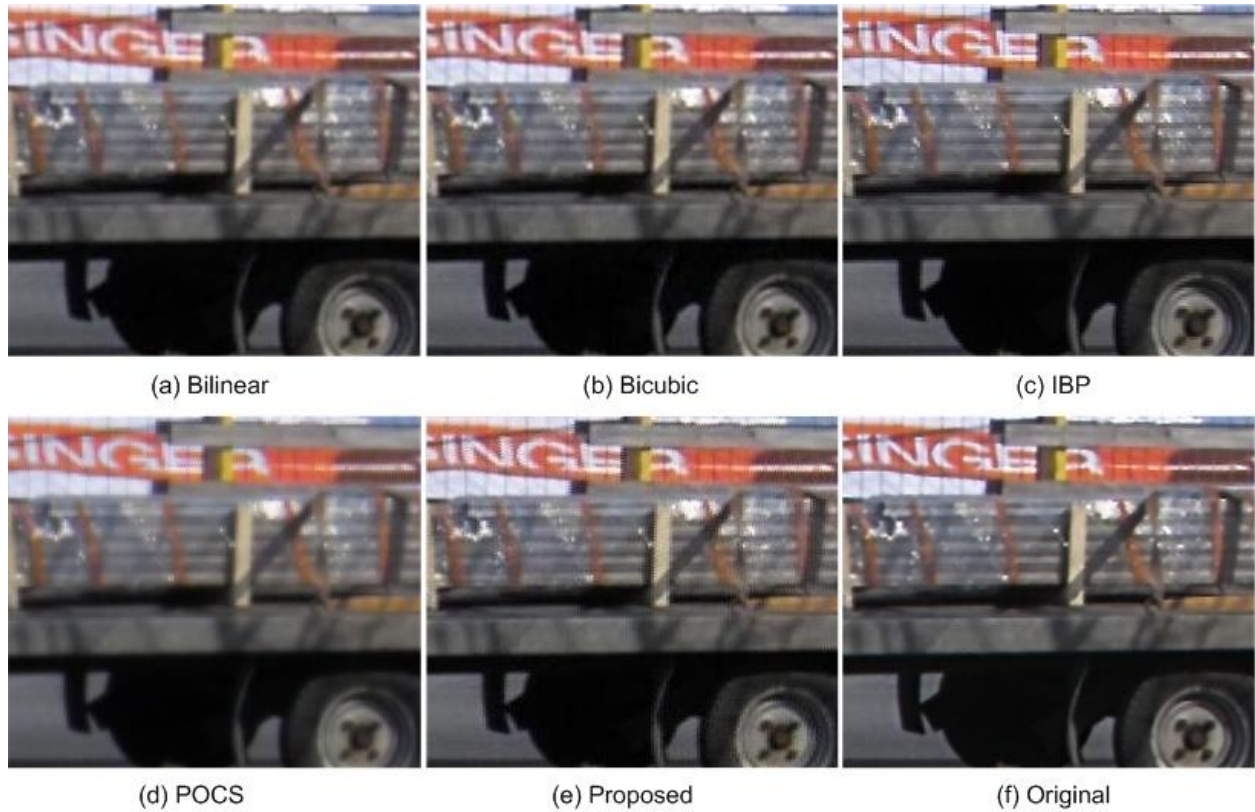


Figure 5.11:Truck Image

More than ten datasets were experimented for comparison between different above mentioned methods with proposed method. In some scenarios IBP performed better than POCS and vice versa. But in a few cases both methods showed inferior results to interpolation. Both visual quality and statistical data suggested that propose method is better than all other mentioned methods.

6 CONCLUSION

In the current paper, a new SR method is proposed that uses Homography as registration technique between multiple images and can be used for both grayscale and color images. First of all, all the images were registered with the reference image using homography. It was necessary to see the efficacy of this registration technique and it was found better because it also caters rotation, scaling and translation. Thus, homography was used to obtain transformation matrix that brings spatial alignment between two images and called as Image registration. This method is based on corners that are features of an image. Different corners detectors were analyzed for different criteria like detection rate, speed efficiency and accuracy of transformation matrix due to good repeatability. After that, Harris corner detector was used for its better detection rate and excellent performance for finding affine transformation. Random Sampling Consensus (RANSAC) finds the transformation matrix by filtering out mismatched points.

Secondly, High Resolution (HR) image from multiple low resolution images was constructed using a novel approach. After registering, the LR images are projected to HR grid using projection matrix and zooming factor. The empty spaces in HR grid are then filled using Bi-cubic interpolation. After interpolation, softness in HR image is sharpened using Laplacian filter.

HR image obtained by proposed method was then compared with POCS, IBP and two commonly used interpolation techniques. Similarly some common qualitative measures like PSNR, RMSE and SSIM were used to access the HR images quality obtained from all above mentioned methods. Results clearly showed improvement in statistical measures, visual quality and detail department.

Most of the traditional SR techniques like IBP and POCS get their initial HR image guess by interpolation and after that their algorithm starts projecting pixels from other LR images. In this proposed method, first of all information obtained in the form of projective transformation matrix is used to project pixels from LR images to HR space. Maximum information from LR images is conveyed to HR space in this way. Because original information from LR images is conveyed first, this technique in projection method makes a big difference in image quality.

Another main difference is computational efficiency; IBP and POCS are iterative schemes, they need at least two to three iterations to converge solution properly. Proposed method is non iterative and HR image is created in single iteration. Therefore proposed method is 10-15 % computationally efficient than counterparts. Table 6.1 shows simplicity of proposed method in comparison with IBP and POCS.

Table 6-1: Proposed Method Comparison

Parameters	PROPOSED	IBP	POCS
A-priori information	No	Back projection constant	Convex sets
SR solution	Unique	Non-unique	Non-unique
Computation requirement	Non-iterative	Iterative – Medium	High
Convergence	High	Medium	Possibly low
Optimization	Medium	Medium	Low
Complications	Simple	Back projection operator	Projection functions
Implementation	Easy	Medium	Easy

6.1 Future work

All the image dataset that were used has planar images and involve global motion; all the pixels in an image have same motion translation, rotation etc. similarly the image registration technique that was used only deal with planar homography. In the future, work will be extended in the registration department by adding complex motion; all the pixels in an image could have independent motion.

Similarly proposed SR algorithm is better in quality as concluded in the previous chapter but still can be improved to some extent in terms of computational efficiency by optimization. As mentioned above, with the involvement of complex motion image registration method and proposed SR method both need to be addressed. Once these issues are addressed then this work can be used for video super resolution.

In video super resolution, sequence of video frames is available that can be used as LR images. Then these frames can be used to create HR image and HR video. At each frame, next four to six frames are used to create HR image. Then these HR frames are joined to create the HR video.

After working with 2D - 2D image transformations, this work can be further extended to 2D – 3D transformations also known as 3D reconstruction of image. This is used for capturing the real shape of different objects [35]. Furthermore, homography can be used to improve and create the 3D panoramas.

References

- [1]. Farsiu, Sina, et al. "Advances and challenges in super-resolution." *International Journal of Imaging Systems and Technology* 14.2 (2004): 47-57.
- [2]. Tian, Jing, and Kai-Kuang Ma. "A survey on super-resolution imaging." *Signal, Image and Video Processing* 5.3 (2011): 329-342.
- [3]. Neha Prakash and Prof. K.T.Talele. Comparison of various Image Registration Techniques with the Proposed Hybrid System, *Proc. of Int. Conf. on Advances in Recent Technologies in Electrical and Electronics* 2013.
- [4]. H. Moravec (1980). "Obstacle Avoidance and Navigation in the Real World by a Seeing Robot Rover". Tech Report CMU-RI-TR-3 Carnegie-Mellon University, Robotics Institute.
- [5]. M. Trajkovic and M. Hedley. Fast Corner Detection. *Image and Vision Computing*, Vol.16 (2), pp. 75-87, 1998.
- [6]. Förstner, W; Gülch (1987 1987). "A Fast Operator for Detection and Precise Location of Distinct Points, Corners and Centres of Circular Features". ISPRS.
- [7]. C. Schmid, R. Mohr, and C. Bauckhage. Comparing and Evaluating Interesting Points. *International Conference on Computer Vision*, pp. 230-235, 1998.
- [8]. F. Mohanna and F. Mokhtarian. Performance Evaluation of Corner Detection Algorithms under Similarity and Affine Transforms. *BMVC 2001*.
- [9]. C.G. Harris and M.J. Stephens. "A combined corner and edge detector", Proceedings Fourth Alvey Vision Conference, Manchester. pp 147-151, 1988.
- [10]. W. Freeman, T. Jones, and E. Pasztor. Example-Based Super-Resolution. *IEEE Computer Graphics and Applications*, 22(2):56–65, 2002.
- [11]. K. Kim and Y. Kwon. Example-based learning for single image SR and JPEG artifact removal. *MPI-TR*, (173), 08.
- [12]. D. Glasner, S. Bagon, and M. Irani. Super-resolution from a single image. In *IEEE International Conference on Computer Vision*, 2009.
- [13]. Fast Image Super-resolution Based on In-place Example Regression. Jianchao Yang, Zhe Lin, Scott Cohen, *cvpr* 2013.

- [14]. 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.
- [15]. M. Irani and S. Peleg, "Improving resolution by image registration" *CVGIP: Graphical Models and Image Proc.*, vol. 53, pp. 231-239, May 1991.
- [16]. D.Keren, S. Peleg and R.Brada, Image sequence enhancement using sub-pixel displacements, in *IEEE conference on Computer Vision and Pattern Recognition*, pp. 742-746, Ann Arbor, MI, June 1988.
- [17]. Elad, M., and A. Feuer. "Super-resolution reconstruction of an image." *Electrical and Electronics Engineers in Israel, 1996. Nineteenth Convention of. IEEE*, 1996.
- [18]. Chen Chiung Hsieh, Yo-Ping Huang, Yu-Yi Chen and Chiou-ShannFuh. Video Super-Resolution by Motion Compensated Iterative Back-Projection Approach, *Journal of Information Science and Engineering* 27, 1107-1122 (2011).
- [19]. Zhao, Feng, Qingming Huang, and Wen Gao. "Image matching by normalized cross-correlation." *Acoustics, Speech and Signal Processing, 2006. ICASSP 2006 Proceedings. 2006 IEEE International Conference on*. Vol. 2. IEEE, 2006.
- [20]. H. Stark and P. Oskoui, "High-resolution image recovery from imageplane arrays, using convex projections," *J. Opt. Soc. Amer. A, Opt. Image Sci., Vis.*, vol. 6, no. 11, pp. 1715–1726, 1989.
- [21]. A. M. Tekalp, M. K. Ozkan, and M. I. Sezan, "High-resolution image reconstruction from lower-resolution image sequences and space-varying image restoration," presented at the *IEEE Int. Conf. Acoustics, Speech, Signal Processing (ICASSP)*, San Francisco, CA, 1992.
- [22]. W. Xie, F. Zhang, H. Chen and Q. Qin. Blind Super-resolution Image Reconstruction Based on POCS Model, *International Conference on Measuring Technology and Mechatronics Automation* 2009.
- [23]. Multi-frame Image Super-resolution Reconstruction based on Sparse Representation and POCS. Xiaoqing Su, Shutao Li, *International Journal of Digital Content Technology and its Applications*. Volume 5, Number 8, August 2011.

- [24]. M. Elad and A. Feuer, "Restoration of a single superresolution image from several blurred, noisy, and undersampled measured images" *IEEE Trans. Image Process*, vol. 6, no. 12, pp. 1646–1658, Dec. 1997.
- [25]. F. Šroubek, G. Cristobal, and J. Flusser. A unified approach to superresolution and multichannel blind deconvolution. *IEEE Trans. Image Processing*, 16(9):2322–2332, Sept. 2007.
- [26]. J. Sun, Z. Xu, and H. Shum. Image super-resolution using gradient profile prior. In *CVPR*, 2008.
- [27]. H.Nasir, V.stankovic and S.Marshall. Image registration for super-resolution using scale invariant feature transform, belief propagation and random sampling consensus, 18thEuropean Signal Processing Conference (EUSIPCO-2010).
- [28]. S. Farsiu, D. Robinson, M. Elad, and P. Milanfar, "Fast and robust multi-frame super resolution," *IEEE Trans. Image Processing*, vol. 13, no. 10, Oct. 2004.
- [29]. S. C. Park, M. K. Park, and M. G. Kang. Super-resolution image reconstruction: a technical overview. *Signal Proc Magazine, IEEE*, 20(3):21 – 36, 2003.
- [30]. Hartley, R., and Zisserman, A. *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2000.
- [31]. Bolles, R. C., and Fischler, M. A. A RANSAC-based approach to model fitting and its application to finding cylinders in range data. In *Proceedings of the 7th International Joint Conference on Artificial Intelligence, IJCAI 1981 (1981)*, pp. 637–643.
- [32]. Richard Szeliski "Computer Vision: Algorithms and Applications" May 17, 2010.
- [33]. Rafael C. Gonzalez and Richard E. Woods "Digital Image Processing" , 3rd Edition, Pearson International Edition.
- [34]. Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: From error visibility to structural similarity," *IEEE Trans. Image Process.*, vol. 13, no. 4, pp. 600–612, Apr 2004.
- [35]. Varady, Tamas, Ralph R. Martin, and Jordan Cox. "Reverse engineering of geometric models—an introduction." *Computer-Aided Design* 29.4 (1997): 255-268.

Appendix A

MATLAB Code

```
function Ireg4i

clc
clearall;
closeall;
numberofimages=10;
zoomfactor=4;
%
fori = 1:numberofimages,

    %% Reading the images
    if (i< 10);
        cim{i} = imread(strcat('E:\11-711\Restoration\ThesisCode1\setf\IMG_0',num2str(i),'.JPG'));
        imc{i}=cim{i};
        im{i}=rgb2gray(cim{i});

    else
        cim{i} = imread(strcat('E:\11-711\Restoration\ThesisCode1\setf\IMG_',num2str(i),'.JPG'));
        im{i}=rgb2gray(cim{i});
        imc{i}=cim{i};
    end
end

im{1} =double(im{1});
%*****
```

```

imshow(im{1},[]);

%% selecting portion to zoom in an image

k = waitforbuttonpress;
point1 = get(gca,'CurrentPoint'); %button down detected
rectregion = rbbox; %%%return figure units
point2 = get(gca,'CurrentPoint'); %%%button up detected
point1 = point1(1,1:2); %%% extract col/row min and maxs
point2 = point2(1,1:2);
lowerleft = min(point1, point2);
upperright = max(point1, point2);
ymin = round(lowerleft(1));
ymax = round(upperright(1));
xmin = round(lowerleft(2));
xmax = round(upperright(2));
rectangle('Position',[ymin,xmin,ymax-ymin,xmax-xmin])

disp('1 finding projective transformation')
tm=imc;

fori=1:numberofimages
imm{i} = imresize(im{i},2,'bicubic');
end

[cim{1}, r{1}, c{1}] = harris(imm{1} , 1,10e3,1, 1);

w = 11; % Window size for correlation matching
dmax =250; % Maximum search distance for correlation matching
% tic

```

```

for pt=2:numberofimages
    [cim{pt}, r{pt}, c{pt}] = harris(imm{pt} , 1,10e3,1, 1);

[m{1},m{pt}] = matchbycorrelation(imm{1} , [r{1}';c{1}'],imm{pt} , [r{pt}';c{pt}'], w, dmax);
[H corrPtIdx{pt}] = findHomography3(m{pt},m{1});
H(1:2,3)=H(1:2,3)/2;
XX{pt-1}=H;
end

disp('2) creating indexes')

for i=1:numberofimages
    im{i}=im{i}(xmin:xmax,ymin:ymax);
    tm{i}=imc{i}(xmin:xmax+1,ymin:ymax+1,:);
    imc{i}=imc{i}(xmin:xmax,ymin:ymax,:);
end

szs=size(im{1});
xmax=szs(1,1);
ymax=szs(1,2);

szs=size(im{1});

sz=size(im{1});
Z=(1:sz(1,1)*sz(1,2))';
[I,J]=ind2sub([sz(1,1) sz(1,2)],Z);

ind2=cell(1,numberofimages);

for i=1:numberofimages

```

```

ind2{i}=ones(sz(1,1)*sz(1,2),3);
end
% ind{1}=[I,I,ones(length(I),1)];
ind2{1}=[I,J,ones(length(I),1)];
% ind3{1}=[I,J,ones(length(I),1)];

Y=cell(1,numberofimages);
fori=1:numberofimages-1
for k=1:sz(1,1)*sz(1,2)

ind2{i+1}(k,:)=XX{i}*ind2{1}(k,:);

end
Y{i}=imc{i}(:);
Y{i}=reshape(Y{i},szs(1,1)*szs(1,2),3);
end
Y{i+1}=imc{numberofimages}(:);
Y{i+1}=reshape(Y{i+1},szs(1,1)*szs(1,2),3);
fori=1:numberofimages
ind2{i}=(ind2{i}*zoomfactor-(zoomfactor-1));
end

disp('3 starting SR stage-1')
e=0.5;
F=zeros(sz(1,1)*zoomfactor,sz(1,2)*zoomfactor,3);
F(1:zoomfactor:end,1:zoomfactor:end,:)=imc{1};
F(1:zoomfactor:end,end,:)=tm{1}(1:end-1,end,:);
F(end,1:zoomfactor:end,:)=tm{1}(end,1:end-1,:);
F(end,end,:)=tm{1}(end,end,:);
BC = imresize(imc{1},zoomfactor,'bicubic');

```

```

BCD=double(BC);
DD=F;
warningoff;
flagg=0;
% tic
PMLN=[];
for xx=1:sz(1,1)*zoomfactor
for yy=1:sz(1,2)*zoomfactor
if F(xx,yy,')==0
xstep=1;
    a=[];
    b=[];
    F1=zeros(numberofimages,3);

while xstep~=numberofimages+1
    [a,b]=find(ind2{xstep}(:,1)>xx-e & ind2{xstep}(:,1) <xx+e& ind2{xstep}(:,2)>yy-e &
ind2{xstep}(:,2)<yy+e);
if isempty(a)==0
    PMLN= [PMLN;xx,yy,ind2{xstep}(a(1),2),Y{xstep}(a(1),:)]

F1(xstep,:)=Y{xstep}(a(1),:);

else
F1(xstep)=0;
end
xstep=xstep+1;
end
if max(F1)==0
F(xx,yy,')==0;
else
F(xx,yy,1)=median(nonzeros(F1(:,1)));

```



```

F(xx,yy,2)=median(nonzeros(F1(:,2)));
F(xx,yy,3)=median(nonzeros(F1(:,3)));
end
else
F(xx,yy,:)=F(xx,yy,:);
end
end
end
figure,imshow(BC,[])
disp('4) starting SR stage-2');
Z=F;
G=F;
[xt,yt]=size(Z(:,1));
[xt1,yt1]=size(Z(:,1));
rrr=0;
ii=[];
jj=[];
fori=1:xt
if length(nonzeros(Z(i,1)))==0
ii=[ii,i];
end
end
Z(ii,,:)=[];
fori=1:yt
if length(nonzeros(Z(:,i)))==0
jj=[jj,i];
end
end
Z(:,jj)=[];
while rrr < 3
forrgb=1:3

```

```

ROW(:,1)=Z(1,:,'rgb');
ROW(:,2)=Z(end,:,'rgb');
COL(:,1)=Z(:,1,'rgb');
COL(:,2)=Z(:,end,'rgb');
for cor=1:2
    [tt,pp]=find(ROW(:,cor)==0);
    [tt1,pp1]=find(ROW(:,cor)~=0);
    xf=nonzeros(ROW(:,cor));
    yf=interp1(tt1,xf,tt,'linear');
    ROW(tt,cor)=yf;
    [tt,pp]=find(COL(:,cor)==0);
    [tt1,pp1]=find(COL(:,cor)~=0);
    xf=nonzeros(COL(:,cor));
    yf=interp1(tt1,xf,tt,'linear');
    COL(tt,cor)=yf;
end
Z(1,:,'rgb')=ROW(:,1);
Z(end,:,'rgb')=ROW(:,2);
Z(:,1,'rgb')=COL(:,1);
Z(:,end,'rgb')=COL(:,2);
rrr=rrr+1;
end
end

Z1=Z;

while length(nonzeros(Z))~=xt*yt*3
for rgb=1:3
    [xt,yt]=size(Z(:, :, rgb));

```

```

fori=1:xt
length_row(i)=length(nonzeros(Z(i,:,rgb)));
iflength_row(i)==yt
length_row(i)=0;
end
end
[ab,ab_index]=max(length_row/yt);
fori=1:yt
length_col(i)=length(nonzeros(Z(:,i,rgb)));
iflength_col(i)==xt
length_col(i)=0;
end
end
[bc,bc_index]=max(length_col/xt);
if ab>bc
    PPPP=Z(ab_index,:,rgb)';
    PPPB=BC(ab_index,:,rgb)';
else
    PPPP=Z(:,bc_index,rgb);
    PPPB=BC(:,bc_index,rgb);
end
% xf=nonzeros(PPPP);
if PPPP(1) ==0
PPPP(1)=PPPB(1);
elseif PPPP(end) ==0
PPPP(end)=PPPB(end);

```

```

end
    [tt,pp]=find(PPPP==0);
    [tt1,pp1]=find(PPPP~=0);
xf=nonzeros(PPPP);

yf=interp1(tt1,xf,tt,'linear');

PPPP(tt)=yf;

if ab>bc
Z(ab_index,.,rgb)=PPPP;
else
Z(:,bc_index,rgb)=PPPP;

end
end
end
G=Z;
G1=double(Z1);
G3=double(Z1);
for k=1:3
fori=2:xt-1
for j=2:yt-1
    G2=[G1(i-1,j-1,k) G1(i-1,j,k) G1(i-1,j+1,k) G1(i,j+1,k) G1(i+1,j-1,k) G1(i+1,j,k)
G1(i+1,j+1,k) G1(i,j-1,k) G1(i,j+1,k)];
RR(i,j,k)=length(nonzeros(G2));
if G3(i,j,k)==0

G3(i,j,k)=median(nonzeros(G2));

```

```
end
end
end
end
```

```
zoomm=yt1/yt;
% XXX=imresize(Z,zoomm,'bicubic');
% h = fspecial('gaussian', [3 3], 0.4) ;
% Z=imfilter(G3,h);
Z=G3;
XXX=imresize(Z,zoomm,'bicubic');
figure,imshow(uint8(XXX))
w=[0 0.5 0;0.5 -2 0.5;0 0.5 0];
R=im2double(XXX);
r1 = imfilter(R(:,:,1),w,'corr');
r2 = imfilter(R(:,:,2),w,'corr');
r3 = imfilter(R(:,:,3),w,'corr');
Ye(:,:,1)=R(:,:,1)-r1;
Ye(:,:,2)=R(:,:,2)-r2;
Ye(:,:,3)=R(:,:,3)-r3;
figure,imshow(uint8(Ye))

% end
% HARRIS - Harris corner detector
%
% Usage:          cim = harris(im, sigma)
%                [cim, r, c] = harris(im, sigma, thresh, radius, disp)
%                [cim, r, c, rsubp, csubp] = harris(im, sigma, thresh, radius, disp)
%
```

```

% Arguments:
%     im   - image to be processed.
%     sigma - standard deviation of smoothing Gaussian. Typical
%           values to use might be 1-3.
%     thresh - threshold (optional). Try a value ~50
%     radius - radius of region considered in non-maximal
%             suppression (optional). Typical values to use might
%             be 1-3.
%     disp  - optional flag (0 or 1) indicating whether you want
%            to display corners overlaid on the original
%            image. This can be useful for parameter tuning. This
%            defaults to 0
%
% Returns:
%     cim  - binary image marking corners.
%     r    - row coordinates of corner points.
%     c    - column coordinates of corner points.
%     rsubp - If five return values are requested sub-pixel
%     csubp - localization of feature points is attempted and
%            returned as an additional set of floating point
%            coords. Note that you may still want to use the integer
%            valued coords to specify centres of correlation windows
%            for feature matching.
%
% If thresh and radius are omitted from the argument list only 'cim' is returned
% as a raw corner strength image. You may then want to look at the values
% within 'cim' to determine the appropriate threshold value to use. Note that
% the Harris corner strength varies with the intensity gradient raised to the
% 4th power. Small changes in input image contrast result in huge changes in
% the appropriate threshold.
%

```

```
% Note that this code computes Noble's version of the detector which does not
% require the parameter 'k'. See comments in code if you wish to use Harris'
% original measure.
%
% See also: NONMAXSUPPTS, DERIVATIVE5
```

```
% References:
% C.G. Harris and M.J. Stephens. "A combined corner and edge detector",
% Proceedings Fourth Alvey Vision Conference, Manchester.
% pp 147-151, 1988.
```

```
function [cim, r, c, rsubp, csubp] = harris(im, sigma, thresh, radius, disp)
```

```
error(nargchk(2,5,nargin));
```

```
if nargin == 4
```

```
disp = 0;
```

```
end
```

```
if ~isa(im,'double')
```

```
im = double(im);
```

```
end
```

```
subpixel = nargin == 5;
```

```
% Compute derivatives and elements of the structure tensor.
```

```
[Ix, Iy] = derivative5(im, 'x', 'y');
```

```
Ix2 = gaussfilt(Ix.^2, sigma);
```

```
Iy2 = gaussfilt(Iy.^2, sigma);
```

```
Ixy = gaussfilt(Ix.*Iy, sigma);
```

```
% Compute the Harris corner measure. Note that there are two measures
```

```

% that can be calculated. I prefer the first one below as given by
% Nobel in her thesis (reference above). The second one (commented out)
% requires setting a parameter, it is commonly suggested that k=0.04 - I
% find this a bit arbitrary and unsatisfactory.

%   cim = (Ix2.*Iy2 - Ixy.^2)./(Ix2 + Iy2 + eps); % My preferred measure.
    k = 0.04;
cim = (Ix2.*Iy2 - Ixy.^2) - k*(Ix2 + Iy2).^2; % Original Harris measure.

if nargin > 2 % We should perform nonmaximal suppression and threshold

ifdisp % Call nonmaxsuppts to so that image is displayed
if subpixel
    [r,c,rsubp,csubp] = nonmaxsuppts(cim, radius, thresh, im);
else
    [r,c] = nonmaxsuppts(cim, radius, thresh, im);
end
else % Just do the nonmaximal suppression
if subpixel
    [r,c,rsubp,csubp] = nonmaxsuppts(cim, radius, thresh);
else
    [r,c] = nonmaxsuppts(cim, radius, thresh);
end
end
end

% DERIVATIVE5 - 5-Tap 1st and 2nd discrete derivatives
%
% This function computes 1st and 2nd derivatives of an image using the 5-tap
% coefficients given by Farid and Simoncelli. The results are significantly
% more accurate than MATLAB's GRADIENT function on edges that are at angles

```



```

% other than vertical or horizontal. This in turn improves gradient orientation
% estimation enormously. If you are after extreme accuracy try using DERIVATIVE7.
%
% Usage: [gx, gy, gxx, gyy, gxy] = derivative5(im, derivative specifiers)
%
% Arguments:
%         im - Image to compute derivatives from.
%         derivative specifiers - A comma separated list of character strings
%         that can be any of 'x', 'y', 'xx', 'yy' or 'xy'
%         These can be in any order, the order of the
%         computed output arguments will match the order
%         of the derivative specifier strings.
% Returns:
% Function returns requested derivatives which can be:
%   gx, gy - 1st derivative in x and y
%   gxx, gyy - 2nd derivative in x and y
%   gxy - 1st derivative in y of 1st derivative in x
%
% Examples:
%   Just compute 1st derivatives in x and y
%   [gx, gy] = derivative5(im, 'x', 'y');
%
%   Compute 2nd derivative in x, 1st derivative in y and 2nd derivative in y
%   [gxx, gy, gyy] = derivative5(im, 'xx', 'y', 'yy')
%
% See also: DERIVATIVE7

% Reference: Hany Farid and Eero Simoncelli "Differentiation of Discrete
% Multi-Dimensional Signals" IEEE Trans. Image Processing. 13(4): 496-508 (2004)

% Copyright (c) 2010 Peter Kovesi

```

```

function varargout = derivative5(im, varargin)

varargin = varargin(:);
varargout = cell(size(varargin));

    % Check if we are just computing 1st derivatives. If so use the
    % interpolant and derivative filters optimized for 1st derivatives, else
    % use 2nd derivative filters and interpolant coefficients.
    % Detection is done by seeing if any of the derivative specifier
    % arguments is longer than 1 char, this implies 2nd derivative needed.
secondDeriv = false;
for n = 1:length(varargin)
    if length(varargin{n}) > 1
        secondDeriv = true;
        break
    end
end

if ~secondDeriv
    % 5 tap 1st derivative coefficients. These are optimal if you are just
    % seeking the 1st derivatives
    p = [0.037659 0.249153 0.426375 0.249153 0.037659];
    d1 = [0.109604 0.276691 0.000000 -0.276691 -0.109604];
else
    % 5-tap 2nd derivative coefficients. The associated 1st derivative
    % coefficients are not quite as optimal as the ones above but are
    % consistent with the 2nd derivative interpolator p and thus are
    % appropriate to use if you are after both 1st and 2nd derivatives.
    p = [0.030320 0.249724 0.439911 0.249724 0.030320];
    d1 = [0.104550 0.292315 0.000000 -0.292315 -0.104550];
    d2 = [0.232905 0.002668 -0.471147 0.002668 0.232905];
end

```

```
end
```

```
% Compute derivatives. Note that in the 1st call below MATLAB's conv2  
% function performs a 1D convolution down the columns using p then a 1D  
% convolution along the rows using d1. etc etc.
```

```
gx = false;
```

```
for n = 1:length(varargin)
```

```
    if strcmpi('x', varargin{n})
```

```
        varargout{n} = conv2(p, d1, im, 'same');
```

```
        gx = true; % Record that gx is available for gxy if needed
```

```
        gxn = n;
```

```
    elseif strcmpi('y', varargin{n})
```

```
        varargout{n} = conv2(d1, p, im, 'same');
```

```
    elseif strcmpi('xx', varargin{n})
```

```
        varargout{n} = conv2(p, d2, im, 'same');
```

```
    elseif strcmpi('yy', varargin{n})
```

```
        varargout{n} = conv2(d2, p, im, 'same');
```

```
    elseif strcmpi('xy', varargin{n}) | strcmpi('yx', varargin{n})
```

```
        if gx
```

```
            varargout{n} = conv2(d1, p, varargout{gxn}, 'same');
```

```
        else
```

```
            gx = conv2(p, d1, im, 'same');
```

```
            varargout{n} = conv2(d1, p, gx, 'same');
```

```
        end
```

```
    else
```

```
        error(sprintf("%s" is an unrecognized derivative option', varargin{n}));
```

```
    end
```

```
end
```