Department of Electronics and Communication Engineering
National Institute of Technology, Rourkela
Odisha, India- 769008

A Thesis on

IMAGE MOSAICING OF PANORAMIC IMAGES

Under the supervision
of
Prof. U. C. Pati
Associate Professor

Submitted by:

Ishan Kumar Sarangi
110EI0150
B.Tech 4th year
ishan.sipun1@gmail.com

Sudarshan Nayak
110EC0175
B.Tech 4th year
110ec0175@nitrkl.ac.in

IMAGE MOSAICING OF PANORAMIC IMAGES

A THESIS SUBMITTED IN PARTIAL FULFILLMENT OF
THE REQUIREMENTS FOR THE DEGREE OF
Bachelor of Technology
in
Electronics and Instrumentation Engineering

By
Ishan Kumar Sarangi
110EI0150
B.Tech 4th year
ishan.sipun1@gmail.com

&
Bachelor of Technology in
Electronics and Communication Engineering

By
Sudarshan Nayak
110EC0175
B.Tech 4th year
110ec0175@nitrkl.ac.in

Project Guide
Prof. U. C. Pati
Associate Professor
Dept. of Electronics & Communication Engineering
National Institute of Technology, Rourkela

NATIONAL INSTITUTE OF TECHNOLOGY, ROURKELA
Rourkela,
Sundargarh-769008
Website- www.nitrkl.ac.in

National Institute of Technology

ROURKELA

CERTIFICATE

This is to certify that the thesis entitled, “Image Mosaicing on Panoramic Images” submitted by Ishan Kumar Sarangi (Roll No.: 110EI0150) and Sudarshan Nayak (Roll No.: 110EC0175) in partial fulfillment of the requirements for the award of Bachelor of Technology in Electronics and Communication Engineering at the National Institute of Technology, Rourkela is an authentic work carried out by them under my supervision and guidance.

To the best of my knowledge, the matter embodied in the thesis has not been submitted to any other University/ Institute for the award of any Degree or Diploma.

Place : Rourkela

Date : 10-05-2014

Prof. U. C. Pati

Dept. of Electronics and Communication Engineering

National Institute of Technology

Rourkela – 769008

ACKNOWLEDGEMENT

We would like to thank the Director, NIT Rourkela for giving us the opportunity to use the resources of the institute and work in such a challenging environment.

We would like to express our deepest gratitude and sincere thanks to our supervisor Prof. U. C. PATI for the inspiration, encouragement, guidance and efforts taken throughout the entire course of this work. His constructive criticism, timely help, and efforts made it possible to present the work contained in this Thesis. We would like to convey our profound gratitude and indebtedness to Prof. L. P. ROY for teaching “Digital Image Processing” in such depth and building up a strong fundamental base in Image Processing that helped us to achieve this success. We would also like to thank all faculty members and Ph.D. scholars of the Department of Electronics and Communication Engineering, N.I.T.Rourkela for their extreme help throughout course.

Last but not the least our sincere thanks to all our friends, who have patiently extended all sorts of help for accomplishing this undertaking.

Sudarshan Nayak	Ishan Kumar Sarangi
(Roll No. : 110EC0175)	(Roll No. : 110EI0150)

ABSTRACT

Image mosaicing is combining or stitching several images of a scene or object taken from different angles into a single image with a greater angle of view. This is practised a developing field. Recent years have seen quite a lot of advancement in the field. Many algorithms have been developed over the years.

Our work is based on feature based approach of image mosaicing. The steps in image mosaic consist of feature point detection, feature point descriptor extraction and feature point matching. RANSAC algorithm is applied to eliminate variety of mismatches and acquire transformation matrix between the images. The input image is transformed with the right mapping model for image stitching.

Therefore, this paper proposes an algorithm for mosaicing two images efficiently using Harris-corner feature detection method, RANSAC feature matching method and then image transformation, warping and by blending methods.

Keywords: *image mosaicing, Harris-corner method, RANSAC, feature detection*

CONTENTS

INTRODUCTION

Image Mosaicing stitches multiple correlated images to obtain an image of greater field of view (FOV). General cameras, which have low FOV can't generate images with higher FOV while mosaicing can help us achieve it. It is a special case of scene reconstruction through which images are related by planar homography. Two or more images can be stitched with each other uniquely without loss of information in any images with a greater FOV.

Numerous mosaicing algorithms have been proposed. Applications of the algorithms proposed is based on the quality of the results we obtain. It depends upon human perception (how much aesthetic the generated picture is) as well as machine perception (these can be used for other processing where some data extraction is required from the image.)

This paper proposes a unique algorithm for mosaicing two or a number of images. Input images are taken and features are detected using Harris-corner detection method. RANSAC is applied to find feature correspondences between images. Images are then projected in a plane and blended together. The whole method is implemented using MATLAB software.

LITERATURE REVIEW

.

Work by Samy Ait-Aoudia [2] was focused on mosaicing of satellite images or aerial images. It was using SIFT future correspondence for feature detection. Thus finding the relevant ones and stitching the images. Debabrata Ghose worked on quantitative evaluation of image mosaicing methods [3]. An algorithm was developed to determine the performance matrix for different methods i.e. RANSAC, SIFT etc., thus determining the correlation and errors between the outputs and taking the best of the results among those from the created performance matrix. Richard Szeliski [4] has done an extended research on the topics and had found many novel algorithms for registration and stitching. There are unique methods developed for extracting large 2-D textures from image sequences based on image registration and compositing techniques. After a review of related work and of the basic image formation equations led to the development of method for registering pieces of a flat (planar) scene, which is the simplest interesting image mosaicing problem. Then it was seen how the same method can be used to mosaic panoramic scenes attained by rotating the camera around its centre of projection. Finally, we conclude with a discussion of the importance of our results.

In computer graphics, compositing multiple image streams together to create greater format (Omnimax) images is discussed in [Greene and Heckbert, 1986]. However, in this application, the relative position of the cameras was known in advance. The registration methods developed are related to image warping [Wolberg, 1990] since once the images are registered, they can be warped into a common reference frame before being composited. [3] While most current methods require the manual specification of feature correspondences, several new methods [Beymer et al., 1993] as well as the methods developed can be used to automate this process. Combinations of local image warping and compositing are now commonly used for special effects under the general rubric of morphing [Beier and Neely, 1992].

Chapter 1

Planar Image Mosaicing

Planar Image Mosaicing

Planar Image mosaicing is a developing area in the field of Computer Vision. As the development in computer vision has demanded stitching of images, the field has grown its own significance perpetually. Images that are too big to be captured by a camera i.e. images with big Field of View (FOV) can't be captured with a single camera that are available in present day markets, can be processed by image mosaicing algorithms. Obtaining the whole view from different FOVs and from different places at different times, we can efficiently blend those to produce an image with greater FOV which would contain more information and will be more aesthetic. It can be used not only in panorama generation, but also in telereality applications. Sometimes, it can be used even to generate a model with 360 FOV. Cylindrical Mosaicing concentrates on this part.

Through this thesis, we have developed methods for Image stitching for panorama generation which is planar which goes through many processes i.e. Motion modelling, Feature detection and matching, warping and blending.

Chapter 2 discusses image motion models in detail. Chapter 3 is dedicated solely to feature detection part. Chapter 4 discusses a method (RANSAC) for feature matching and correspondence. Chapter discusses image warping and blending in short; just the basics are mentioned. Chapter 6 shows the result of our work, provides a conclusion to the whole process and discusses further development that can be applied to the algorithms developed and other modifications.

Chapter 2

Motion Models

Motion Models

We have to know the basics of the mathematical relationships which will be helpful in mapping pixel coordinates from one pixel to a different co-ordinate system before registering and aligning the images. A range of parametric motion models are available. These can be applied for straight-forward 2D transformation to 3D rotations, to planar perspective models to non-planar surfaces. These models are described below for convenience of the reader.

2.1 2D (Planar) Motions

Taking the basic Cartesian co-ordinate systems, the transformation of co-ordinates can be visualised as follows.

Translation: It can be represented as x' = x + t or

$$\boldsymbol{x'} = [I \quad t]\, \tilde{x} \qquad \text{… eq. 2.1}$$

I is the 2*2 identity matrix and $\tilde{x}$= (x,y,1) is the homogenous 2D co-ordinate

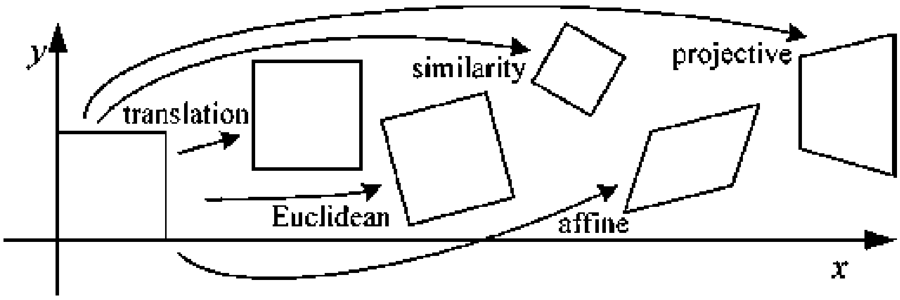


Fig 2.1 Basic set of 2D planar transformation

Scaled Rotation : It can be called 'Similarity Transform' too. If x is the image matrix taken as before, the expression for scaled rotation is represented as x' = sRx + t. (s is taken as an arbitrary scale factor). This expression can alsi be represented as

$$\boldsymbol{x'} = [sR \quad t]\, \tilde{x} = \begin{bmatrix} a & -b & t_x \\ b & a & t_y \end{bmatrix} \qquad \text{…eq. 2.2}$$

Where $\boxed{a^2 + b^2 = 1}$ is no longer a requirement. In similarity transformation, angle between lines are never altered.

Rotation and Translation : Alternatively, it can be called 2D Euclidean transformation as the Euclidean distances are preserved in this type of transformation. Here, x’ = Rx + t or

$$\boldsymbol{x'} = [I \quad t]\,\tilde{x}$$

$$R = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \qquad \text{…eq. 2.3}$$

R is an orthonormal rotation matrix whose determinant value is 1.

Affine This can be represented as x’ = A$\check{x}$. A is an arbitrary 2*3 matrix.

$$\boldsymbol{x'} = \begin{bmatrix} a_{00} & a_{01} & a_{02} \\ a_{10} & a_{11} & a_{12} \end{bmatrix} \tilde{x} \qquad \text{…eq. 2.4}$$

Parallel lines remain parallel under affine transformation.

Projective Also known as perspective transform, it can be operated on homogenous coordinates represented as $\check{x}$ and $\tilde{x}$’ as follows

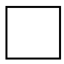
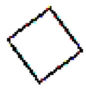
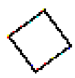
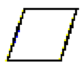
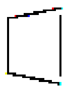
$$\tilde{x} \sim \widetilde{H}\tilde{x},$$

Where ~ denotes equality upto scale and H is an arbitrary 3*3 matrix. H is a homogenous matrix which is defined upto a scale. The homogenous coordiane obtained should be normalized to get inhomogenous coordinate x’ represented as

$$\boldsymbol{x'} = \frac{h_{00}x + h_{01}y + h_{02}}{h_{20}x + h_{21}y + h_{22}} \quad \text{and} \quad \boldsymbol{y'} = \frac{h_{10}x + h_{11}y + h_{12}}{h_{20}x + h_{21}y + h_{22}} \qquad \text{…eq. 2.5}$$

This transformation never alters the orientation of straight lines.

Table 2.1 Hierarchy of 2D coordinate transformations.

Name	Matrix	Number of d.o.f.	Preserves	Icon
Translation	$\begin{bmatrix} \boldsymbol{I} \mid \boldsymbol{t} \end{bmatrix}_{2\times 3}$	2	Orientation $+\cdots$	
Rigid (Euclidean)	$\begin{bmatrix} \boldsymbol{R} \mid \boldsymbol{t} \end{bmatrix}_{2\times 3}$	3	Lengths $+\cdots$	
Similarity	$\begin{bmatrix} s\boldsymbol{R} \mid \boldsymbol{t} \end{bmatrix}_{2\times 3}$	4	Angles $+\cdots$	
Affine	$\begin{bmatrix} \boldsymbol{A} \end{bmatrix}_{2\times 3}$	6	Parallelism $+\cdots$	
Projective	$\begin{bmatrix} \tilde{\boldsymbol{H}} \end{bmatrix}_{3\times 3}$	8	Straight lines	

Chapter 3

Feature Detection and Matching

Feature Detection and Matching

Feature Detection and feature correspondence between images is one of the basic steps in Mosaicing of Images. As we have to align and blend the images, we need to have feature correspondence between the images to blend them properly. How to detect and correlate the features is described briefly below.

3.1 Feature

Before detecting the features, the first thing that comes into one's mind is what is a 'Feature'? Feature is any combination of data points or any part of the image that can be taken as an identity point/ it can be used as a candidate point for further processing.

What kind of points serve as a better set of feature points is another question that one should consider. Certainly, those points where pixel value variation is smooth or rather, the change in pixel value is small should not be taken as feature points as they won't help much in obtaining correlation between images. Generally, the points where pixel value variation is large are taken as candidate points. An extraction from [1] is shown below to make the point clear.

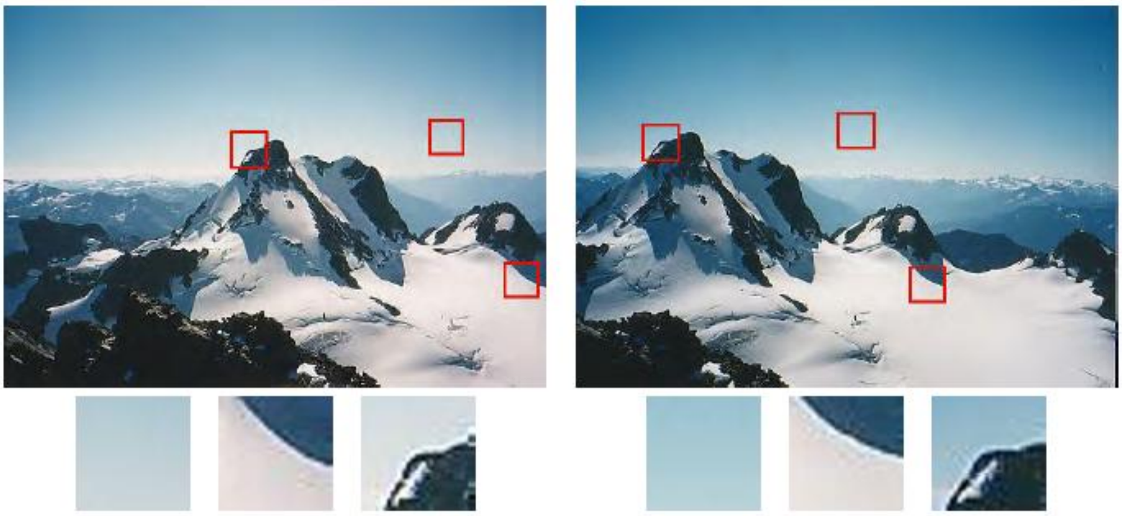


Fig. 3.1

As shown in fig 3.1, intuitively one will consider the edges or corners as good candidate points. Here, a brief description about the type of features is presented.

3.1.1 Points

Points can be taken as features. But it is not worth taking points as features because there are too many points in an image and processing would be time consuming and memory requirement would be quite high.

3.1.2 Edges

Edges are generally considered as the boundary points of the images. Edges can be junctions or any sort of structure between pixel values that are different to their neighbourhood pixel values. Gradient between image pixels is high compared to the neighbourhood derivative values. Edges are considered as good features for low level processing. Edges are one dimensional feature points.

3.1.3 Corners

Corners are two-dimensional feature points and refer to feature points in images which produce gradients of greater values in both the dimensions. Corners show rapid changes in pixel values. Algorithms are developed over time to detect corner points using edge detection algorithms that were developed previously; it is just applied in both the directions. Sometimes, some points which are traditionally not the corner points are chosen as the 'corner' points i.e. small dark patches in a white background. Although, intuitively these should not be considered as the interest points, still we take them as corner points.

3.1.4 Blobs

While corners provide feature that are point like, Blobs provide the description of feature points that is complementary and the structures are region-type. Blob detectors work on the concept of taking a centre point first which works as a preferred point(it is generally a local maxima of the operator response) and all the other points are compared with the centre point. Thus, it works on a patch of image rather than certain localized points and helps in detecting some interest points which are rather difficult to be recognised by corner detection algorithms.

When a patch is smooth in an image and the image is shrinked in another process, the smooth patch may be converted to a point of interest. Thus a scale should also be taken while processing the images. Due to some matching properties between the images, The algorithms like LoG and DoG, although they work on blob detection algorithm, they are included in Corner detection algorithm.

Next, we shall see how to take and evaluate feature points for image mosaicing operations.

3.2 Feature Processing

In general Image Processing applications, four types of feature processing are done. First, one have to consider the features (extract) those one wants to process. This part is called ‘feature extraction/detection’. Through this process, each image is searched to find out the points which have more likelihood of matching with the points of another image considered.

After this comes ‘feature description’ stage. The points that we have extracted are detected key points which are converted to a more suitable descriptor.

‘Feature matching’ stage comes next; which matches or correlates the key points described above between the images taken. ‘Feature tracking’ is another kind of feature processing tasks, which is similar to feature matching, but it works on small consecutive patches between the images and thus is good for video processing.

In this section (3), feature detection is considered while next section (4) explains feature matching.

3.3 Feature Detection

As explained in sec 3.1, feature detection is intuitive yet a basic and primary steps in image mosaicing through which we find candidate points for further processing. Through feature detection, the points which can be easily correlated are chosen. There are various methods, developed through decades for feature detection. These feature detectors are described below in brief, just to present an idea how they are used to detect features.

Before going into the brief description of feature description methods, It is further emphasised here that the candidates points are chosen in such a way that the pixel value variation at candidate points are rather high and easily detectable.

3.3.1 Edge Detection

CANNY

Canny edge detection works on the simple formula of finding the edge using gradient values. Developed by Canny, this method convolutes the image with a predefined gradient matrix. The values can be modified using adaptive techniques. Because of its popularity and use in various fields, MATLAB has included a predefined function for it.

SOBEL

While canny edge detector convolutes one patch over the image, sobel does the same using two patches; one in x direction and another in y-direction to obtain a better and faster result. It can also be modified using adaptive filtering techniques. Due to its huge applications in image processing, MATLAB has also included a predefined function for it.

3.3.2 Corner Detection

HARRIS-STEPHENS

It is one of the oldest method of corner detection. It finds gradient along both the directions in the image. The maximum values along the directions are considered as corner points. Sometimes, due to reduction of eigen value to zero, error is taken. Our method is solely based on this method; thus it is explained in details in further sections.

SUSAN

SUSAN is an abbreviation for 'Smallest Univalue Segment Assimilating Nucleus'. Through this method, a spherical mask is placed over the pixel that is to be verified. Every pixel within a certain radius is compared with the centre pixel value through an exponential function. The accumulated value is taken finally and compared with a predefined threshold. Thus, the candidate points are selected.

FAST

FAST stands for Features from Accelerated Segment Test. This is a machine learning technique that is adaptive in processing which takes some points within a range and processes on them and rejects the point that fall outside the range of interest.

Our technique is based on Harris-Stephens corner algorithm that takes into account Non-Maximal Suppression as well. This adaptive algorithm and its implementation is discussed in details.

2.4 Harris-Corner Detection

Simplest criterion one can think of to detect the relationship between two images is to compare different patches on them.

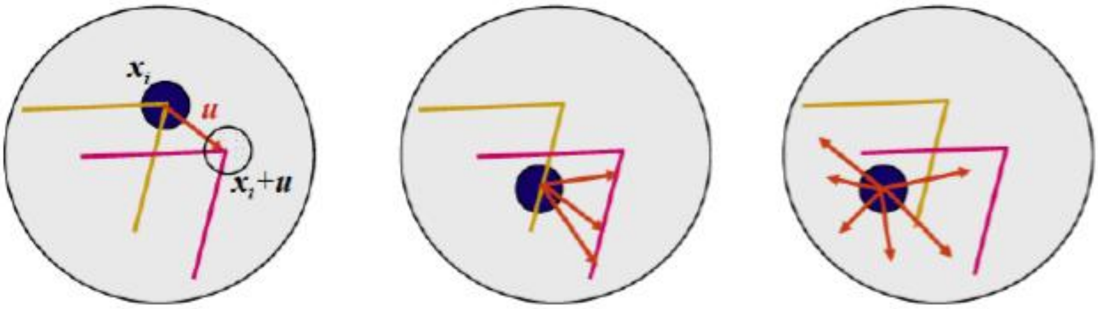


Fig 3.2 – Harris – Corner Concept

As shown in fig 3.2, a window can be taken. The window should be moved from one place to other place and calculations should be done over the window. The radius and value of the window constant can be varied to get different results. The equation can be obtained as follows:

$$E_{\mathrm{WSSD}}(u) = \sum_i w(x_i)[I_1(x_i + u) - I_0(x_i)]^2 , \qquad \ldots \text{ eq. } 3.1$$

The equation above compares the image patches between two images i.e. Image I1 and I0. In corner detection that we are using in our method to mosaic two images need to detect the corners of a single image (the method can be further applied to any number of images before getting the correlation between different images.). Thus, the window taken can be moved over a single image to obtain the corner points via auto-correlation.

Harris corner method applies the same concept. It moves a window to find out the corners by taking the derivatives in both directions simultaneously. The auto correlation is done with the neighbourhood points through convolution of the image with the window taken and thus the output is produced. The Equation for autocorrelation is presented below for convenience and the method of Harris-corner detection is presented clearly throughout the section.

$$E_{\mathrm{AC}}(\Delta u) = \sum_i w(x_i)[I_0(x_i + \Delta u) - I_0(x_i)]^2 \qquad \ldots \text{ eq. } 3.2$$

The concept of Harris- Corner Detection

As described previously, we can convolute the image with the window taken to find corners. This is a general method taken for corner detection. Now, the question that arises is how to take the window and what values should be assigned to the weights of the window. To know those values, let's go back to the basic of Harris –Corner Detection and understand it properly.

Whenever the image pixel values shows a better gradient value in its neighbourhood, the patch is taken as feature point. But for better processing, we need good features yet less in number. Corners satisfy such a criterion.

Next, how to detect corners? It is quite simple. We can take derivative over an image from both directions and the window taken which does this act and shows a large variation is obviously a corner point. Thus, as described above, we need to assign some values to the window which helps in optimizing the above process.

In our work, we have followed the paper by farid [14] which optimally defines the value of P, D1, and D2 for finding derivatives.

In our work, the derivatives obtained are further filtered using Gaussian functions; just to take the required feature points and to minimize the number of these feature points so that further processing would be simple yet effective.

The General Equation of Harris – Corner Detection

$$S(x, y) = \sum_u \sum_v w\,(u\,,v)\ (I(u+x, v+y) - I\,(u\,,v))^2 \qquad \text{... eq 3.3}$$

As shown above, eq 3.1 is a simplified version of the above mention auto-correlation function.

It can be expanded using Taylor's series to get

$$I\,(u + x, v + y) \approx I(u\,,v) + I_x(u\,,v)x + I_y(u\,,v)y \qquad \text{... eq 3.4}$$

which on further simplification produces

$$S(x, y) = \sum_u \sum_v w\,(u\,,v)\ (I_x(u\,,v)x + I_y(u\,,v)y)^2 \qquad \text{... eq 3.5}$$

Equation 3.3 can be converted to the matrix form

$$S(x,y) \approx (x \quad y) A \begin{pmatrix} x \\ y \end{pmatrix} \qquad \text{... eq 3.6}$$

A is the structure tensor given by (obtained after simplification)

$$A = \sum_u \sum_v w\ (u, v) \begin{pmatrix} I_x{}^2 & I_x I_y \\ I_x I_y & I_y{}^2 \end{pmatrix} = \begin{pmatrix} \langle I_x{}^2 \rangle & \langle I_x I_y \rangle \\ \langle I_x I_y \rangle & \langle I_y{}^2 \rangle \end{pmatrix} \qquad \text{... eq 3.7}$$

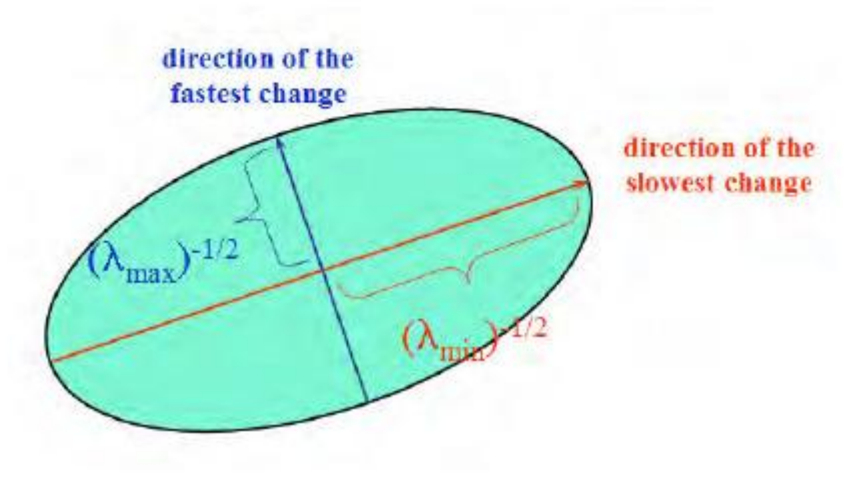
This matrix is a Harris matrix, and angle brackets denote averaging (i.e. summation over (u , v)). If a circular window (or circularly weighted window, such as a Gaussian) is used, then the response will be isotropic.

A corner is detected by large variation in both the directions of S. Analysis of matrix A shows that the matrix should produce large Eigen values at the corner points.

Following conclusions can be derived from the magnitude of Eigen values obtained

1. For $\lambda_1 \approx 0$ and $\lambda_2 \approx 0$, the pixel taken is no feature of interest.
2. For $\lambda_1 \approx 0$ and λ_2 has some large positive value, then an edge is found.
3. If λ_1 and λ_2 have large positive values, then a corner is found.

Computation of the values is quite cumbersome as it requires the computation of a square root.



Anandan had shown in his work that some eigen values obtained are not applicable for use, thus approximate values can be taken and errors should be added, within the bound of uncertainty. We can use the following function

$$M_c = \lambda_1 \lambda_2 - k(\lambda_1 + \lambda_2)^2 = \det(A) - k\ \text{trace}^2(A) \qquad \text{….. eq 3.8}$$

Where K is tunable sensitive parameter.

The formula shows that rather than finding the eigen values, we can take the determinant and trace of the matrix and find the corner points through approximation of the eigen values.

The Shi–Tomasi corner detector directly computes $\min(\lambda_1, \lambda_2)$ because under certain assumptions, the corners are more stable for tracking.

The covariance matrix for the corner position is A^{-1}, i.e.

$$\frac{1}{\langle I_x{}^2\rangle\langle I_y{}^2\rangle - \langle I_x{}^2 I_y{}^2\rangle}\begin{pmatrix} \langle I_y{}^2\rangle & -\langle I_x I_y\rangle \\ -\langle I_x I_y\rangle & \langle I_y{}^2\rangle \end{pmatrix} \quad \text{... eq 3.9}$$

Adaptive Non-Maximal Suppression

Non-maximal suppression is applied to the feature points obtained previously. It is an adaptive process that we have applied in our process to suppress the points that are not needed for processing purposes. The figure below taken from Szeliski [5] shows a nice demonstration of ANMS.

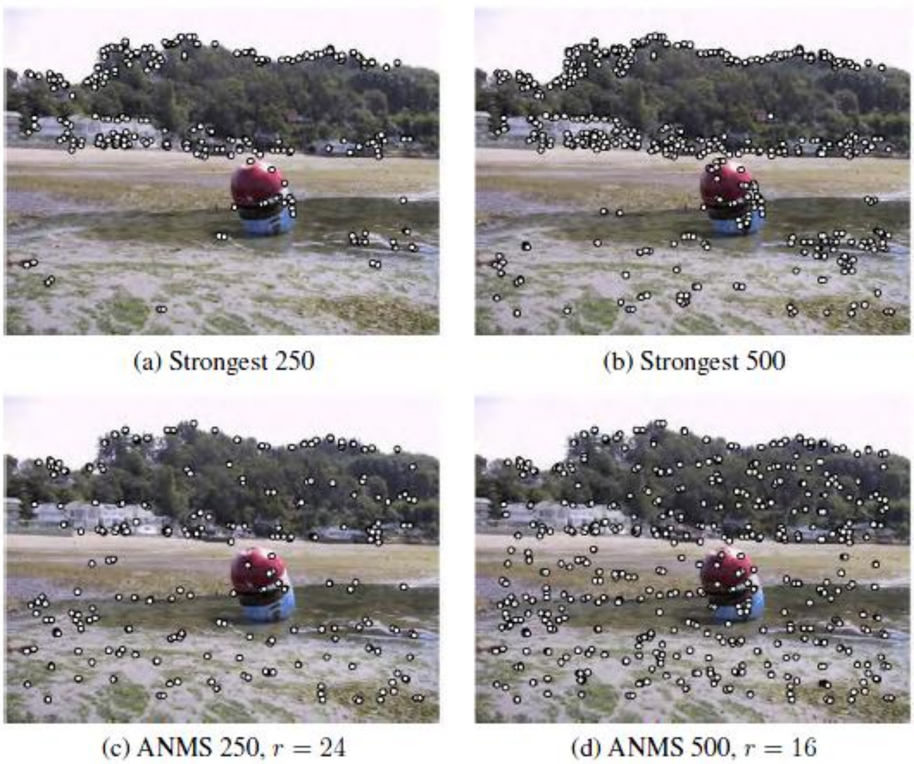


Fig 3.3

Application of the ANMS is the last process in our feature detection part. The features obtained are taken for further processing and applied for feature correspondence.

The results of the above processes are shown in chapter.6

Chapter 4

Feature Matching

Feature Matching

One should match the features (key points) detection, i.e., confirm that features are from the corresponding locations from completely distinct pictures. As feature points may not be situated precisely, an accurate matching has to be done by progressive incremental motion refinement, that is time consuming, and might result in decrease in performance (Brown et al. 2005)[13].

Tracking features over larger image sequences may cause larger variations in their appearances. In such case, one needs to compare appearances with the help of an affine motion model. Shi and Tomasi (1994) compared patches using a translational model among neighbouring frames, then used the calculable location to initialize an affine registration among the patch in the present frame and the base frame where a feature is detected first [13]. In reality, features are solely identified occasionally, that is, solely in region where the tracking fails.

Usually, a region round the current foretold feature's location is looked for with a progressive registration technology. This algorithm is known as "detect then track approach", as detection occurs rarely. It favours video sequences where the expected feature point's locations can be easily found. Among the native descriptors that Mikolajczyk and Schmid matched, they established that David Lowe's Scale Invariant Feature Transform (SIFT) typically executed the best, followed by freewoman and Adelson's steerable filters and then cross-correlation [13].

Differential invariants, whose descriptors are unresponsive to moderations in alignment by design, failed to do as expected.

First a local orientation employing a bar chart or histogram of the local gradient orientations is found to compute SIFT features that is more precise than just the typical orientation. Once the local frame is made, gradients are copied into other orientation planes, and blurred and faded re-sampled versions of these pictures are used as the features [13]. This offers some insensitiveness to few feature localization errors and geometric distortions to the descriptor.

Steerable filters are mixtures of derivative of Gaussian filters which permits fast computation of even and odd edge-like and corner-like features at all potential alignments. They're additionally insensitive to localization and orientation errors as they use broad Gaussians.

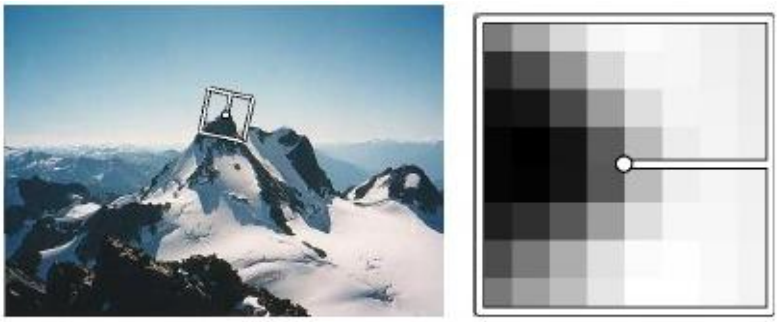


Figure 4.1: MOP descriptors are formed using an 8 × 8 sampling

4.1 RANSAC

After the computation of an initial set of feature correspondences we would like to seek out and retrieve a collection that may generate a high accuracy alignment. In several cases, it's good to choose a better initial set of inliers, i.e., points that are all steady with certain specific motion estimate.

One excessively used way out to the present drawback is RANSAC. RANSAC stands for "RANdom SAmple Consensus" [9]. This procedure was 1st issued by Fischler and Bolles at SRI International in 1981 [10]. RANSAC is preferably suitable for applications in automated image examination where understanding is made based on the data given by feature detectors that are error-prone as it is capable of interpreting data that consists of substantial percentage of total errors. The algorithm is able to produce results with nice estimates from data contaminated by a large (> 50%) fraction of outliers in the entire point cloud.

Properties of the algorithm are:

- Robustness even in the presence of many outliers and a high degree of noise.
- Generality.
- Simplicity.

The algorithm is composed of two steps that are being iterated:

- Hypothesize. First a randomly selected input dataset is needed. Then parameters of a minimal sample set (MSS) are calculated. To determine the model parameters, the cardinality of the MSS is the smallest sufficient required.
- Test. In this step, elements of the whole dataset and model created with the parameters established in the hypothesize step are being checked for compatibility. The set of such elements is known as consensus set (CS).

A certain threshold is defined in order to find the best ranked CS. If the probability of discovering a better ranked CS drops below it, RANSAC stops its work. As we can see, the algorithm of Ransac can have errors. But with the improvement of some steps of the general Ransac, it will be stable and effective. RANSAC produces a sensible result with a probability in the same way as the more iterations it has, the higher the probability; therefore it is called a non-deterministic algorithm.

4.2 Overview

The RANSAC algorithm inputs a set of data values, a parameterized model that explains or fits to the observations, and a few confidence parameters.

RANSAC succeeds by iteratively choosing a random set of the initial data. These data are known as inliers and this hypothesis is then tested as follows:

1. The sample of theoretical inliers are fitted with a model, i.e. from this sample all free parameters of the model are fitted.

2. The fitted model then tests all other and, the points are taken as portion of the consensus set that match the projected model perfectly.
3. The projected model is fairly good if adequately several points are classified as a fragment of the consensus set.
4. Then by means of all the associates of the present consensus set the scheme is developed by re-estimating it.
5. Lastly, error estimation of the inliers comparative to the model is done to calculate the model.

This procedure is done again for a few more iterations giving out two types of models: 1- a scheme that is excluded as very less points are included as far as the set is concerned, 2- a developed scheme along with a provided consensus size of set. When we consider the second case, we generally keep the developed scheme when the provided consensus set is excess as compared to the previously saved model.

This technique initiates by choosing (at random) a subset of k correspondences, that is then utilised to get p (motion estimate).The residuals of the full set of correspondences are then computed as

$$r_i = \widetilde{x_i}' (x_i ; p) - \widetilde{x_i}', \quad \text{… eqn 4.1}$$

where $\widetilde{x_i}'$ are estimated (mapped) locations.

The process of random selection is iterated S times to obtain the sample set of largest numbers of inliers or the set which contains the smallest median residual. This set is kept as the final solution. The initial parameter taken as p or the set of inliers that are computed is passed to the next stage of data fitting. In a recently developed algorithm of RANSAC known as PROSAC (PROgressive Sample Consensus), random samples are taken initially from 'confident' matches that are available which helps in speeding he process of finding better set of inliers.

Let us say that a number of trials S must be carried out for finding good set of inliers. If p be the probability of any given correspondence to be valid, P be taken as the total probability of success after those S trials. The likelihood for one trial that k random samples be inliers is p^k.

Therefore, the likelihood that S such trials will all fail is

$$1 - P = s\,(1 - p^k)^S \,, \qquad \text{... eq 4.2}$$

and the required minimum number of trials is

$$S = \frac{\log(1-P)}{\log(1-p^k)} \qquad \text{... eq 4.3}$$

4.4 Advantages and disadvantages

Advantages

- Even if a substantial number of outliers exist in the data set, RANSAC is able to perform a strong assessment of the model parameters very accurately. Robustness is a major property of this algorithm.

Disadvantages

- One of disadvantages is that no upper bound is defined on the time for the algorithm to run and compute parameters that we need. The results we calculate form a solution, but it is limited and not optimal. It means that it may not be one that best fits the data. In this occasion RANSAC offers another solution: with the increasing repetitions the probability of a practical model being generated also increases.
- A certain threshold has to be defined.
- RANSAC is restricted to estimation of one model for a particular data set. When two (or more) model instances exist for any one-model approach, RANSAC may fail to point out any one. When more than one model instance exist, Hough transform, an alternative robust estimation technique, may be useful. Both methods can't be totally effective when working with a high usage of memory.

Chapter 5

Warping and Blending

Warping and Blending

Warping methods are already discussed in transformation section. After warping, the images are mapped and blended with each other in the same plane, thus producing the final output.

Image warping digitally manipulates an image to distort it significantly, with a purpose to blend images or with other views at hand. It can be used to distort and map images from one frame to another frame keeping the pixel value unchanged, yet modifying the position of the image points so that they can be blended. It can also be used for video processing. In our method, we have used an existing code to warp the images to a generalized plane so that they could be blended properly.

After warping the images to a common plane, they are blended i.e. mixed with each other to create the final image with greater FOV than the existing images. The pixel values are averaged and histogram equalization applied for better output.

Chapter 6

Results and Discussions

6.1 Results

6.1.1 Harris Corner Detection

Test Case -1

Fig. 6.1.1 Input Image

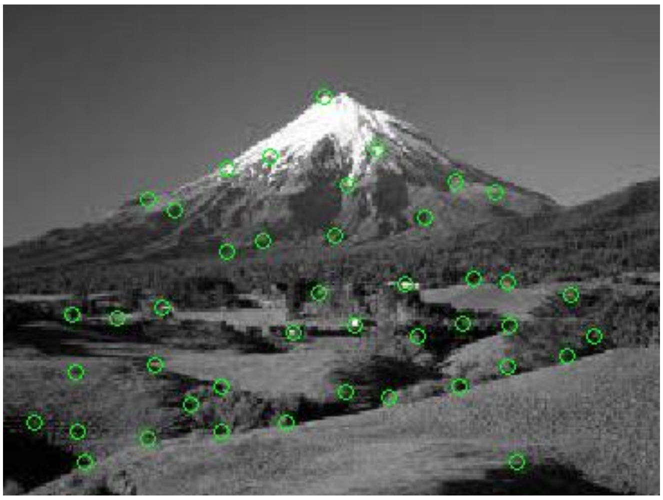


Fig. 6.1.2 Output Image with threshold = 20

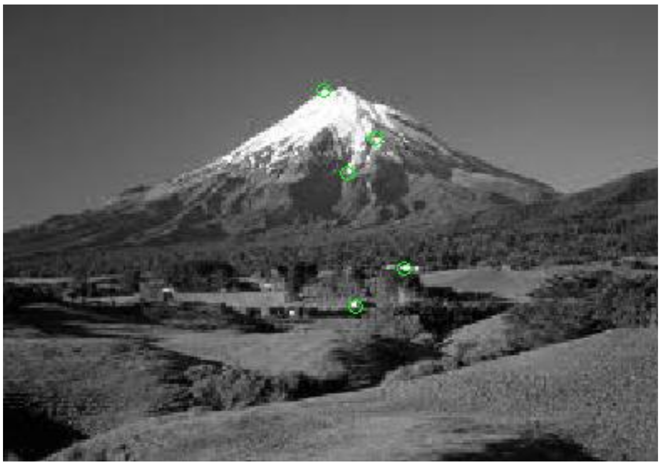


Fig. 6.1.3 Output Image with threshold = 200

Test Case 2

Fig. 6.1.4 Input Image

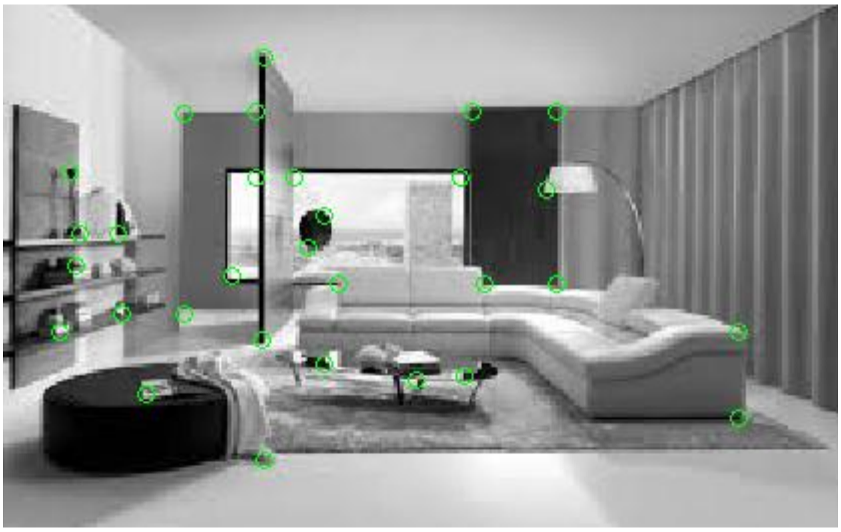


Fig. 6.1.5 Output image with Threshold = 100

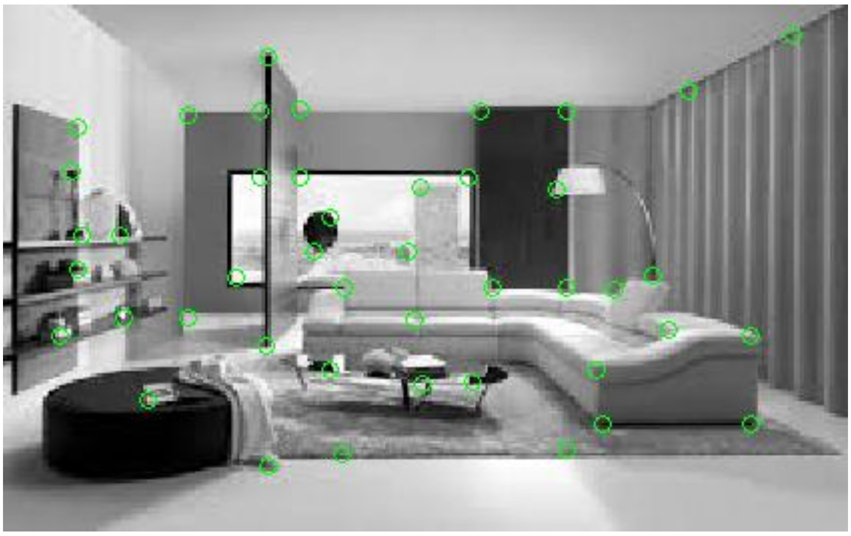


Fig. 6.1.6 Output image with threshold = 20

6.1.2 Mosaicing

TEST CASE 1

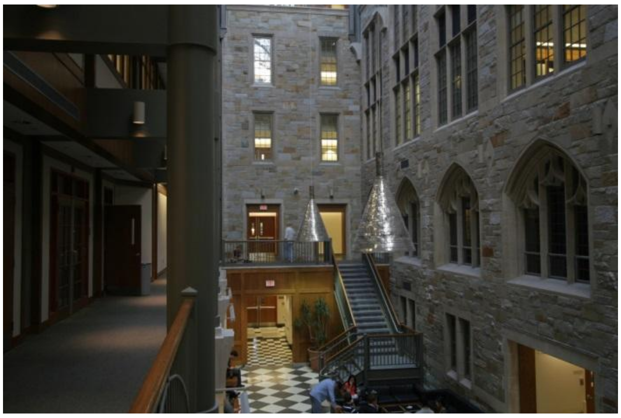


Fig. 6.1.7 Input Image- 1

Fig. 6.1.8 Input Image 2

HARRIS CORNER DETECTION

Fig. 6.1.9 -Image 1

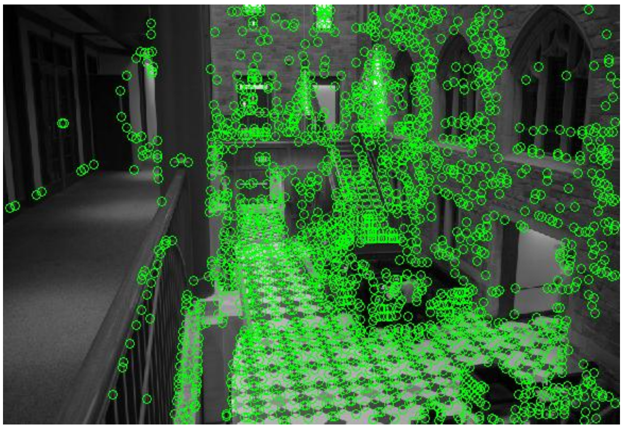


Fig. 6.1.10 -Image 2

Fig. 6.1.11 RANSAC OUTPUT

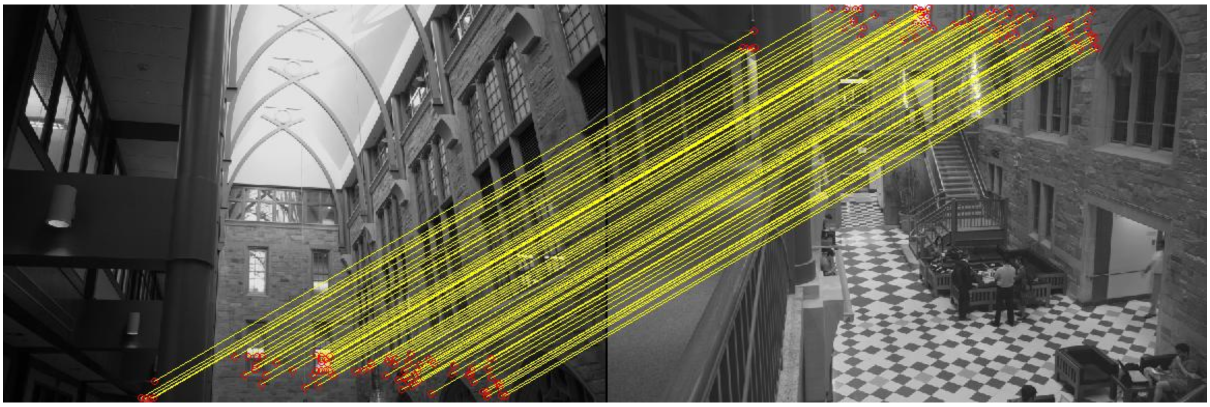
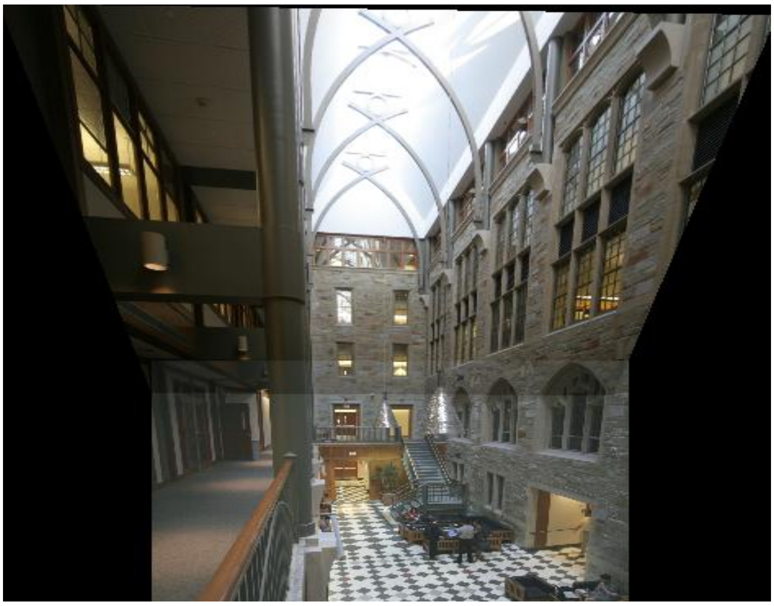


Fig. 6.1.12 FINAL OUTPUT



TEST CASE 2

Fig. 6.1.13 Image 3

Fig. 6.1.14 Image 4

After Harris Corner Detection

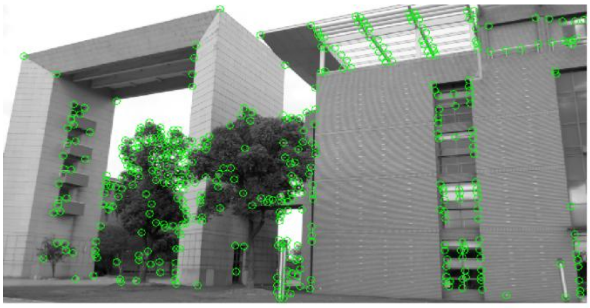


Fig. 6.1.15 Image 3

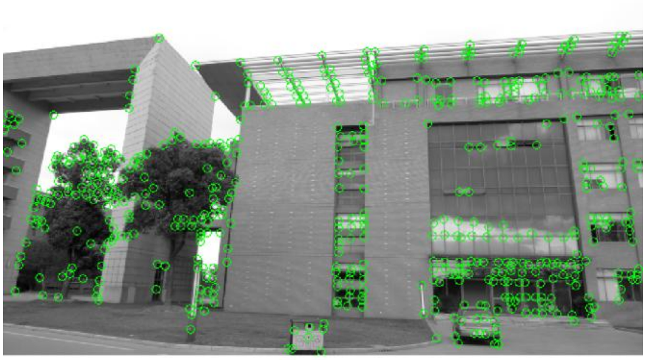


Fig. 6.1.16 Image 4

After RANSAC

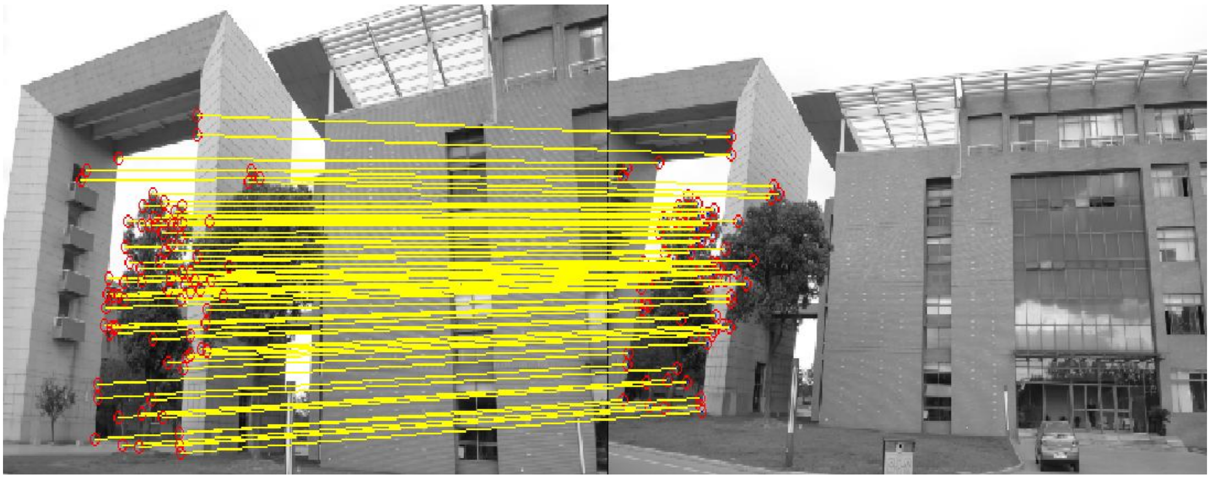


Fig. 6.1.17 RANSAC Output

OUTPUT IMAGE

Fig. 6.1.17 Final Output

TEST CASE 3

Fig 6.1.18 – Input Image 1

Fig 6.1.19 – Input Image 2

HARRIS CORNER DETECTION

Fig 6.1.20 – Output 1

Fig. 6.1.21– Output 2

RANSAC OUTPUT

Fig 6.1.22 – Output (RANSAC)

FINAL OUTPUT

Fig 6.1.23 - Output

6.2 Conclusion

The programs for Harris Corner Detection were implemented properly on various input images and tested. Variation of threshold provides the required outputs and it can be further manipulated to get good corner points. The images obtained through Corner detection algorithms are further processed through RANSAC algorithm to obtain greater correlation between images; these were warped and blended properly. The outputs obtained for various input images were quite satisfactory.

6.3 Future Works and Developments

Future works can be done on removing the aliasing (Image feathering) that were introduced as noise. For removal of noise, Poisson or Laplacian distribution method can be used. Images with non-overlapping regions can be taken into consideration for mosaicing purposes. RANSAC algo can be further developed. Image mosaicing for document mosaicing should be taken into account.

As shown in Section 6.1.2 Test Case 3, some images are not quite well blended, which needs further modifications in programming. The main reason being that a lot of feature points have been detected because of which RANSAC algorithm fails. So we need to work upon other techniques which can map these points or we should find some another way to reduce these feature points.

REFERRENCES

[1]Gonzalez and Woods- “Digital Image Processing”, Pearson Educations, 2010,

[2]Satellite and aerial Image mosaicing – a comparative insight- Samy Ait-Aoudia, Ramdane Mahiou, Hamza Djebli, El – Hachemi Guerrout [IEEE,2012]

[3]Quantitative evaluation of Image Mosaicing in multiple scene catagories- Debabrata Ghose, Sangho Park, Naima Kaabouch and William Semke [IEEE,2012]

[4] Image Mosaicing for Telereality application by Richard Szelisky.

[5] Image Allignment and Stitching by Richard Szelisky, Springer [2011,]

[6] Martin A. Fischler and Robert C. Bolles (June1981). "Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography". Comm. of the ACM 24 (6): 381–395. Doi: 10.1145/358669.358692.

[7] David A. Forsyth and Jean Ponce (2003). Computer Vision, a modern approach. Prentice Hall. ISBN 0-13-085198-1.

[8] P.H.S. Torr and D.W. Murray (1997). "The Development and Comparison of Robust Methods for Estimating the Fundamental Matrix". International Journal of Computer Vision24 (3): 271–300. doi:10.1023/A:1007927408552.

[9]Ondrej Chum (2005). "Two-View Geometry Estimation by Random Sample and Consensus". PhD Thesis.

[10] Sunglok Choi, Taemin Kim, and Wonpil Yu (2009). "Performance Evaluation of RANSAC Family". In Proceedings of the British Machine Vision Conference (BMVC).

[11] Anders Hast, Johan Nysjö, Andrea Marchetti (2013). "Optimal RANSAC – Towards a Repeatable Algorithm for Finding the Optimal Set". Journal of WSCG. 21(1): 21–30.

[12]Hossam Isack, Yuri Boykov (2012). "Energy-based Geometric Multi-Model Fitting". International Journal of Computer Vision. 97(2): 1: 23–147.

[13] , N. et al., editors, Handbook of Mathematical Models in Computer Vision, pages 273–292, Springer, 2005