# Video Stabilization Using Point Feature Matching

**1 author:**

Asal Rouhafzay
Université du Québec en Outaouais
**2** PUBLICATIONS   **1** CITATION

SEE PROFILE

**Some of the authors of this publication are also working on these related projects:**

Video Stabilization View project

# Video Stabilization Using Point Feature Matching

**Asal Rouhafzay**

Submitted to the
Institute of Graduate Studies and Research
in partial fulfillment of the requirements for the Degree of

Master of Science
in
Electrical and Electronic Engineering

Eastern Mediterranean University
January 2015
Gazimağusa, North Cyprus

Approval of the Institute of Graduate Studies and Research

_____
Prof. Dr. Elvan Yılmaz
Director

I certify that this thesis satisfies the requirements as a thesis for the degree of Master of Science in Electrical and Electronic Engineering.

_____
Prof. Dr. Hasan Demirel
Chair, Department of Electrical
and Electronic Engineering

We certify that we have read this thesis and that in our opinion, it is fully adequate in scope and quality, as a thesis of the degree of Master of Science in Electrical and Electronic Engineering.

_____
Assoc Prof. Dr. Erhan A. İnce
Supervisor

Examining Committee
_____

1. Prof. Dr. Sener Uysal            _____

2. Prof. Dr. Runyi Yu              _____

3. Assoc. Prof.  Dr. Erhan A. İnce    _____

# ABSTRACT

In the last decade utilization of handheld video cameras have become quite popular however the videos captured by unprofessional users or by fixed and vehicle mounted cameras have resulted in shaky and unclear videos. In this work we aim to use a video stabilization algorithm using point feature matching technique to reduce the vibrations in acquired video sequences.

The thesis presents motion estimation techniques, motion models, feature detection techniques, robust sampling consensus and mainly the RANSAC paradigm. Implementation of the feature points matching based stabilization algorithm was done using the MATLAB platform and applied to three different videos with jitter. The quality improvement in the video sequences after stabilization are demonstrated by comparing the mean of stabilized and unprocessed shaky videos, the normalized sum of absolute differences (NSAD), an singular value decomposition (SVD) based image quality metric, peak signal to noise ratio (PSNR) and translation in $x$ and $y$ directions. Results indicate that the stabilization of the videos would improve PSNR, NSAD and M-SVD values and help reduce the amount of translation in $x$ and $y$-directions. After stabilization it was observed that PSNR values would improve on average by 5.3dB. Similarly NSAD and M-SVD values were respectively improved by 32.11 %, and 37.88 %. Finally the displacements in $x$ and $y$ directions were respectively reduced by 91.21 % and 92.39%.

**Keywords**: RANSAC, SVD, feature detection, robust sampling consensus, motion estimation, normalized sum of absolute differences, PSNR.

# ÖZ

Geçen yüzyıldan beri kişisel video kameraların kulanımı oldukça artmıştır. Fakat profesyonel olmayan kişilerin, sabit kameralar ve araçlara monte edilmiş kameralar ca yakalanan birçok video sarsıntı ve bulanıklıga tabi kalmaktadır. Bu çalışmadaki hedefimiz öznitelik noktalasını çakıştıran bir video sabitleme algoritmesi kullanarak yakalanmış videolardaki ve bulanık oranlarını mümkün olduğunca azaltmaktadır.

Bu tezde hareket kestirim teknikleri, hareket modelleri, özyinelik çıkarma teknikleri, gürbüz örnekleme ve RANSAC paradigması hakkında bilgi verilmiş ve önerilen öznitelik noktalarına bağlı sabitleme algoritması kullanılarak üç farklı sallantılı video dizinindeki sarsıntlar mümkün olduğunca sabitlenmeye çalışılmıştır. Özyinelik noktalarına bağlı sabitleme algoritması MATLAB platforumunda gerçekleştirilmiştir. Sabitleme sonrası videolarda nasıl bir iyileşme olduğu sabitlenmiş ve sabitlenmemiş videoların ortalaması kıyaslanarak, düzgelenmiş mutlak farkların toplamına (NSAD) bakılarak , tekil değer ayrışım metriği (SVD), doruk sinyal gürültü oranı (PSNR) ve $x$ ve $y$ yönlerindeki ortalama kayma oranlarına bakılarak belirlenmiştir. Sabitlenme sonrası PSNR, NSAD ve M-SVD değerlerinde iyileşmeler ve $x$ ve $y$ yönlerindeki kaymalarda ise azalma gözlenmiştir. PSNR değerlerindeki iyileşme ortalamada 5.3 dB iken NSAD ve M-SVD değerleri 32.11% ve 37.88% oranlarında iyileşmiştir. Yatay ve dikey konumlardaki kaymaların ise sabitleme sonrası ortalamada 91.21% ve 92.39% azaldığı gözlemlenmiştir.

**Anahtar Kelimeler**: RANSAC, SVD, öznitelik sezimi, gürbüz örnekleme ve uzlaşma, hareket kestirimi.

Dedicated to

My parents Leili Razeghi and Dr. Farhad Rouhafzay

and my sisters Ghazal and Lael

# ACKNOWLEDGMENTS

I would like to express my deepest thanks and appreciation to my supervisor Assoc. Prof. Dr. Erhan A. İnce for encouraging my research and for being supportive of me whenever I needed his help.

I also would like to thank the academic staff at the Electrical and Electronic Engineering department for their help and support during my course of study.

A special thanks to my family. Particularly I am grateful to my mother, father, and sisters for the sacrifices that they've made on my behalf.

# TABLE OF CONTENTS

# LIST OF FIGURES

# LIST OF TABLES

# LIST OF ABREVIATIONS

CCD   Charge-Coupled Device

CS    Consensus Set

FG    Foreground

MLESAC  Maximum Likelihood Estimation Sample Consensus

MSAC   M-estimator SAmple and Consensus

MSS    Minimal Simple Sets

NADE   Normalized Absolute Difference Error

PSNR   Peak Signal-to-Noise Ratio

RANSAC  RANdom SAmple Consensus

SAD    Sum of Absolute Differences

SSD    Sum of Squared Differences

SUSAN   smallest uni-value segment assimilating nucleus

SVD    Singular Value Decomposition

VCM    Voice-Coil Motor

# Chapter1

# INTRODUCTION

Video stabilization is a technique which is used by many different fields in today's world to achieve a stable video sequence from a shaky video. Medicine, military and robotics are three main fields in which video stabilization is heavily used. For example, in endoscopy and colonoscopy videos need to be stabilized to determine the exact location and width of the problem. Videos captured by aerial vehicles on a reconnaissance flight need to be stabilized for localization, navigation, target tracking, etc. [1]. Furthermore utilization of digital cameras has always been popular and hence video stabilization has entered our daily life with the aim of removing shaky motions from videos captured by non-professional users. Different approaches to stabilize shaky videos as follows.

## 1.1 Different Approaches to Video Stabilization

There are mainly three different approaches to stabilize a shaky video. These include mechanical, optical and digital stabilization methods. In this section each approach is briefly discussed.

### 1.1.1 Mechanical Video Stabilization Technique

Mechanical image stabilization systems using the vibration feedback of the camera which is detected via special sensors like gyros accelerometers etc. are the earliest developed stabilization techniques [2]. In mechanical methods, accelerometer and gyros sensors are used for motion detection and then the camera is moved against the

movement direction. Figure 1.1 demonstrates a camera with mechanical stabilizer where a gyroscope is attached to the camera.



Figure 1.1: Camera with Mechanical Stabilizer [3]

### 1.1.2 Optical Video Stabilization Technique

Optical stabilization technique are developed few years after mechanical techniques. If instead of moving the whole camera just the pieces of the lens glass move, the stabilization technique is referred to as optical stabilization which is the most effective one and employs a moveable lens assembly that variably adjusts the path length of the light as it travels through the camera's lens system[4] . In this technique angle and speed of the camera shake is detected by two gyro sensors. According to the movement direction of the entire lens, the select lens elements should be moved so the image passing through the lens can be steady and sharp when it hits the imaging sensor. Figure 1.2 illustrates the function of optical image stabilizer when the lens is jerked downward. Due to the downward movement of the camera the

center of the image moves downward on the focal plane. Shifting the optical Image Stabilizer lens group downward, the light rays are refracted so that the image center returns to the center of the focal plane [5].
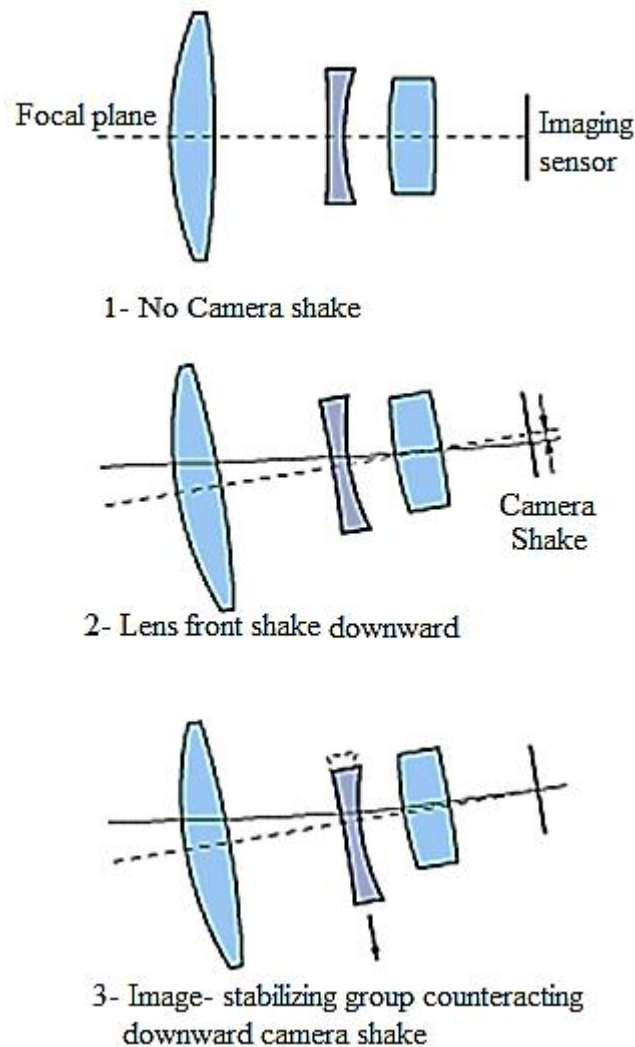


Figure 1.2: Optical Image Stabilizer Parallel Movement [5]

### 1.1.3 Digital Video Stabilization Technique

Digital video stabilization adopted by many companies for their products, is the least expensive and precise solution to remove unwanted motions from captured videos. In general, stabilizing a video by digital algorithms contains three main steps including motion estimation, motion smoothing and image composition. The transformation

parameters between two consecutive frames are derived in the first stage. The second stage filters out the unwanted motion and in the last stage the stabilized video will be reconstructed. In all video stabilization algorithms motion Estimation is the most important part which describes the transformation from one video frame to the subsequent one.

## 1.2 Literature Review

Many algorithms for Digital video stabilization have been proposed over the past two decades. Most of the proposed methods try to compensate for all motion producing a sequence with a motionless background [6] [7] [8]. In other techniques only the three dimensional motion of the camera is subtracted[9] [10]. The motion parameters between two frames of a video is frequently modeled using two dimensional affine or projective approaches [11] .Generally in two dimensional models, all the estimated affine motion parameters are compensated and subsequently the unwanted motion is removed from the input sequence [7] [8]. Three dimensional stabilization can be achieved by re-rotating the frame and generating a sequence which contains only the translational parameters. Duric *et al.* [9] uses a vehicle method to filter the high-frequency components of the rotational parameters. To estimate the rotational parameters he applies a flow-based motion estimator to the distant points, and the solution is recursively refined to obtain smoothed motion. Fast implementations of 2D stabilization algorithms are presented in [7] [8]. Hansen *et al.* [8] proposes an image stabilization algorithm using a mosaic-based registration technique. Burt *et al*. [7] introduces an algorithm using a multi-resolution, iterative process that estimates affine motion parameters between levels of Laplacian pyramid images.

One of the first researches to stabilize an amateur digital video has been carried out by Ratakonda [12]. He used a single large template window and a small search window and the algorithm was capable to stabilize only the mild translational motions. The work achieved a real-time performance on a low resolution video stream using profile matching and sub-sampling.

A fast and robust implementation of a digital video stabilization algorithm described in this thesis is based on the two dimensional model where we apply an affine transformation incorporating translation, rotation and scaling.

The developed algorithm is similar to the other algorithms based on the 2D rigid motion model [13].

## 1.3 Thesis Objectives

In this thesis a digital video stabilization algorithm is implemented using the MATLAB programming platform. The utilized feature based stabilization algorithm adopts RANdom SAmple Consensus (RANSAC) paradigm to estimate the motion model describing the displacement of points between consecutive frames. Feature points achieved by smallest uni-value segment assimilating nucleus (SUSAN) corner detection algorithm play a key role in motion estimation.

Three different videos are stabilized using the proposed algorithm and the mean of stabilized videos and unprocessed shaky ones are compared. The Singular Value Decomposition (SVD)-based Image Quality assessment method, Sum of Absolute differences between consecutive frames, PSNR and translation in $x$ and $y$ direction are also computed for fifty frames of each video and the results illustrated by pair processing graphs indicate a significant improvement in video quality.

## 1.4 Thesis Overview

After a brief introduction provided in Chapter 1, Chapter 2 discusses the motion estimation techniques and motion models. Subsequently in chapter 3 robust estimation methods are shown with a focus on the RANSAC paradigm. Salient features detection methods are introduced in Chapter 4. Chapter 5 provides an effective algorithm for video stabilization. Simulation results are discussed and evaluated in Chapter 6 and Chapter 7 concludes the thesis and gives direction to future works.

# Chapter 2

# MOTION ESTIMATION

Motion estimation is an important step for video stabilization algorithms. It is the attempt for estimating the displacement of points between two successive video frames. In video frame's motion is manifested as alteration in pixels intensity values which can be used to determine motion of objects.

Equation 2.1 presents a simple representation of the problem where $I(t)$ and $I(t + \Delta t)$ are two consecutive video frames. As depicted in Figure 2.1 $\Delta x$ and $\Delta y$ are the motion vector components.

$$I\,(x, y, t) = I(x + \Delta x, y + \Delta y, t + \Delta t) \qquad\qquad (2.1)$$
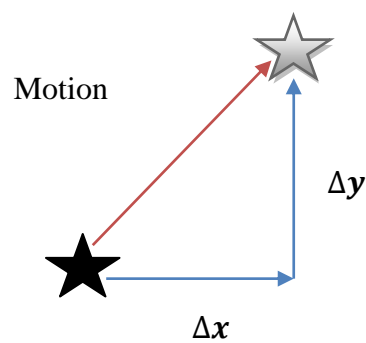
Figure 2.1: Motion Vector Component

In order to find $\Delta x$ and $\Delta y$ the following equation should be solved.

$$I\,(x, y, t) - I(x + \Delta x, y + \Delta y, t + \Delta t) = 0 \qquad (2.2)$$

However the existence of noise, camera displacements and light alterations can prevent the zero difference. Direct and Indirect motion estimation techniques are two different approaches to the problem. After introducing different motion models for two dimensional images, direct and indirect motion estimation techniques are discussed.

## 2.1 Principal Types of Motion Models

Mathematical equations describing the mapping procedure of pixel coordinates between two images are referred to as motion models. Any pixel coordinate in an image can be described as; $x = (x, y) \in R^2$ . For most transformations non-homogenous coordinates are sufficient however for perspective or projective transformations homogeneous transformations are needed. In what follows we give examples for various transformation types.

### 2.1.1 Translation transformation

Equation 2.3 describes a two dimensional translations [14]. This transformation preserves the orientation.

$$x' = x + t \qquad (2.3)$$

or

$$x' = [\,I \quad t\,]\bar{x} \quad, \text{ where } I \text{ is a } (2 \times 2) \text{ identity matrix}$$

8

### 2.1.2 Euclidean transformation

Euclidean transformation which is the union of Translation and Rotation transformations can be expressed as the following equation [14].

$$x' = [\ R \quad t\ ]\bar{x} \tag{2.4}$$

$$R = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix}$$

$$RR^T = I$$

$$|R| = 1$$

### 2.1.3 Similarity transformation

Equation 2.9 [14] describes similarity transform also known as scaled rotation. In this transformation angles between lines are preserved.

$$x' = [\ sR \quad t\ ]\bar{x} = \begin{bmatrix} a & -b & t_x \\ b & a & t_y \end{bmatrix}\bar{x} \tag{2.5}$$

Where $s$ is an arbitrary scale factor.

### 2.1.4 Affine transformation

Affine transformation described by the following equation preserves the parallelism between lines [14]. The parameter $A$ is an arbitrary $2 \times 3$ matrix.

$$x' = A\bar{x} = \begin{bmatrix} a_{00} & a_{01} & a_{02} \\ a_{10} & a_{11} & a_{12} \end{bmatrix}\bar{x} \tag{2.6}$$

### 2.1.5 Homography transformation

Homography transformation which is also referred to as perspective or projective transformation operates on homogenous coordinate and can be described by equation 2.7 [14].

$$\widetilde{x'} = \widetilde{H}\,\tilde{x} \tag{2.7}$$

$\widetilde{H}$ is an arbitrary $3 \times 3$ homogenous matrix. The resulting homogenous $\tilde{x}$ should be normalized in order to obtain an inhomogeneous result $x$ [14].

$$x' = \frac{h_{00}x + h_{01}y + h_{02}}{h_{20}x + h_{21}y + h_{22}} \tag{2.8}$$

$$y' = \frac{h_{10}x + h_{11}y + h_{12}}{h_{20}x + h_{21}y + h_{22}}$$
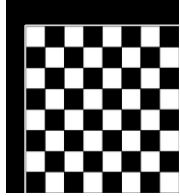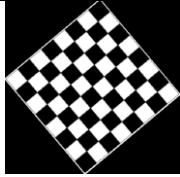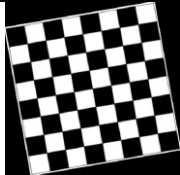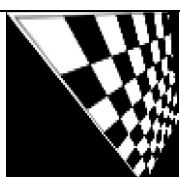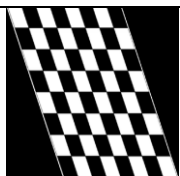
Table 2.1 represents an organized summary of different motion models.

## 2.2 Direct Motion Estimation Technique

In direct approach to estimate motion, all the pixels in the frame are in use to estimate the motion. Unlike feature based methods which are adopted in this research, in direct motion estimation methods unknown parameters are recovered directly from measurable image quantities such as intensity.

The advantage of using direct methods is increasing the sub-pixel accuracy. These methods are also more efficient in handling the data which does not fit the model (outliers). In direct methods spatial derivatives in each frame are compared using an affine model.

Table 2.1: Motion Models

| Transform | Preserved parameter | 2D coordinate transformations | Example |
|---|---|---|---|
| Translation | Orientation | $x' = x + t$ |  |
| Euclidean | Length | $x' = Rx + t$ |  |
| Similarity | Angles | $x' = sRx + t$ |  |
| Homography | Straight lines | $\underline{x'} = H\underline{x}$ |  |
| Affine | Parallelism | $x' = Ax + t$ |  |

The first step in most of direct methods is to determine brightness constancy constraint. Assuming $I$ and $J$ as two consecutive video frames we can write:

$$J(x,y) = I\left(x + u(x,y), y + v(x,y)\right) \tag{2.9}$$

Where $(u,v)$ represent pixel displacement between the frames. If $(u,v)$ are small enough and $I$ is linearized around $(x,y)$ the following constraint can be obtained

$$I_x u + I_y v + I_t = 0 \qquad\qquad (2.10)$$

$$I_t = I - J$$

In this equation $I_x$ and $I_y$ denote spatial derivatives of the brightness. There will be one such equation for every pixel in the frame.

In the second step of direct motion estimation methods another constraint describing the image motion variations in the total image is also defined. In most of the direct methods the affine motion model is described as follows[15].

$$u(x, y) = a_1 + a_2 x + a_3 y \qquad\qquad (2.11)$$

$$v(x, y) = a_4 + a_5 x + a_6 y$$

This model gives better results when the image depicts a distant scene. Substituting equation 2.11 in equation 2.10 we have.

$$I_x(a_1 + a_2 x + a_3 y) + I_y(a_4 + a_5 x + a_6 y) + I_t = 0 \qquad\qquad (2.12)$$

For each pixel of the image we have one constraint containing six parameters which are identical for all pixels so six constrains are adequate to solve the equation.

## 2.3 Indirect Motion Estimation Technique

In indirect motion estimation methods, image features are used with the purpose of estimating motion between frames. In these methods the first step is to find strong features of each frame. There are several methods to find feature points in an image. Harris and SUSAN corner detection are some examples. Generally corner points have higher chance to be in the next frame as well.

As each feature will have a distinct vector, a filter is required in indirect algorithms to filter out the outliners. RANSAC is a popular example.

The following steps constitute the indirect algorithm to compute a two dimensional homographic transformation between two frames.

1. Corner point features are computed in sub-pixel precision.

2. Considering the similarity and proximity of the neighborhood point intensity, a set of corner points matches is computed.

3. RANSAC robust assessment for $N$ samples

   - Selection of four random correspondences based on which the homography $H$ is calculated.

   - For all the assumed correspondences a geometric distance error should be computed.

   - Choosing Correspondences with the geometric distance error less than a threshold value based on which number of inliers consistent with $H$ is computed.

4. Optimal re-estimation of $H$ from the inliers.

5. Determination of more corner point correspondences based on the $H$ calculated in the previous step with the purpose of defining a search region around the transferred point position.

Figure 2.2 depicts corner points in two consecutive video frames

Figure 2.2: Corner Points for Two Consecutive Frames

# Chapter 3

# ROBUST SAMPLING CONSENSUS

Estimation of model parameters from an image is a very prevalent computational problem in vision. The name robust estimation is given to estimations which are tolerant to presence of outliers. In dictionary definition the word 'Outlier' means something that lies outside the main body or group that is a part of. In technical definition, if a data does not belong to the 'true' model defined by the 'true' set of parameters considering some threshold value, it will be referred to as outlier.

Robust estimation targets to find a set of inliers from the correspondences. Many robust estimation algorithms are introduced in the literature. In this chapter we discuss some distinct robust estimation algorithms and we mainly focus on RANSAC which is adopted in this research to stabilize video.

## 3.1 Random Sample Consensus (RANSAC)

RANdom SAmple Consensus abbreviated as RANSAC targets to estimate a mathematical model parameters based on some observations including outliners. This paradigm has been developed by M. A. Fischler and R. C. Bolles [16] with the purpose of fitting a model to experimental data. In most of robust estimation techniques in order to obtain an initial solution, it's attempted to gather as much of the data as possible but in contrast RANSAC uses the minimum feasible data set and when possible tries to enlarge this set with consistent data. RANSAC paradigm is composed of two main steps repeated iteratively, hypothesize and test. First minimal sample sets (MSSs) are randomly selected from the input dataset and the model

parameters are computed using only the elements of the MSS. Then the RANSAC paradigm searches for consensus set (CS) containing elements of the dataset which are consistent with the model instantiated with the parameters estimated in the first step. [17].

Consider $D = \{d_1, \dots, d_N\}$ is the input data set through which we want to estimate a model. If $\theta(\{d_1, \dots, d_N\})$ is the parameter vector estimated by $\{d_1, \dots, d_h\}$, and $h$ is greater than the minimum number of elements required to estimate the model, the model space $\mathcal{M}$ of the parameter vector $\theta$ will be as follows [17].

$$\mathcal{M}(\theta) = \{d \in \mathcal{R}^d : \mathcal{F}_{\mathcal{M}}(d; \theta) = 0\} \tag{3.1}$$

Where $\theta$ is a parameter vector and $\mathcal{F}_{\mathcal{M}}$ is a smooth function. Considering the condition that $\mathcal{F}_{\mathcal{M}}$ is equal to zero, it will contain all points fitting the model.

The distance from the datum $d$ to the model $\mathcal{M}$ represents the error $e_M$ [17]. Datum with an error value greater than a certain threshold value are not consistent with the model.

$$e_M(d, \theta) = \min_{d' \in \mathcal{M}(\theta)} dist(d, d') \tag{3.2}$$

So the CS is defined as:

$$S(\theta) = \{d \in D : e_M(d; \theta) \leq \delta\} \tag{3.3}$$

Where $\delta$ is a threshold value which can be determined based on the nature of the automatically based on some hypothesis.

$$\mathcal{M}(\theta) \overset{\text{def}}{=} \{d \in \mathbb{R}^d : f_{\mathcal{M}}(d; \theta) = 0\}$$
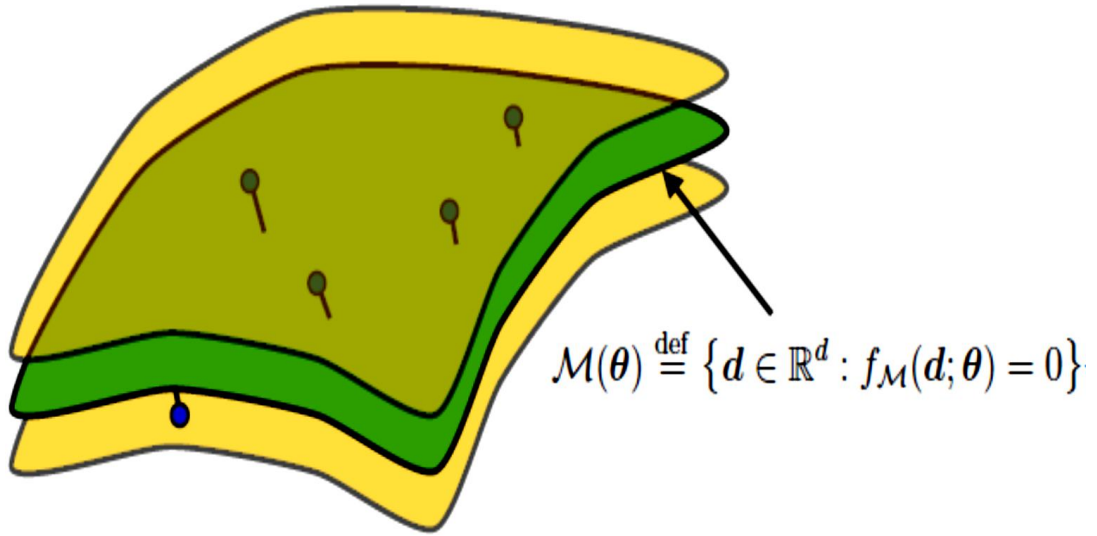
Figure 3.1: Model *M* and Inlier Boundries [17]

In Figure3.1 the model $\mathcal{M}$ is depicted as the green surface and yellow surfaces show the threshold value determining the inliers boundaries. Blue dots represent some inliers.

The RANSAC paradigm can be summarized as the following.

1. In the first step of RANSAC paradigm a sample of s random data points, Minimal Simple Sets (MSSs) are chosen from which the model should be investigated.

2. In second step the paradigm introduces $S_i$ as the Consensus Set (CS) of the sample. In fact $S_i$ contains the inliers of S. Inliers selection is conducted based on a distance threshold value t of the model.

3. In third step of the paradigm another threshold value T is introduced. If there are more inliers than the value of T, all the points in $S_i$ should be in use to re-approximate the model.

4. If there is less number of inliers than the value of T, a new subset should be selected and the previous steps will be repeated.

The procedure will continue for N trials and finally the largest consensus set $S_i$ is chosen. All the points in this set are again in use to re-approximate the model.

Figure 3.2 demonstrates different steps of RANSAC.
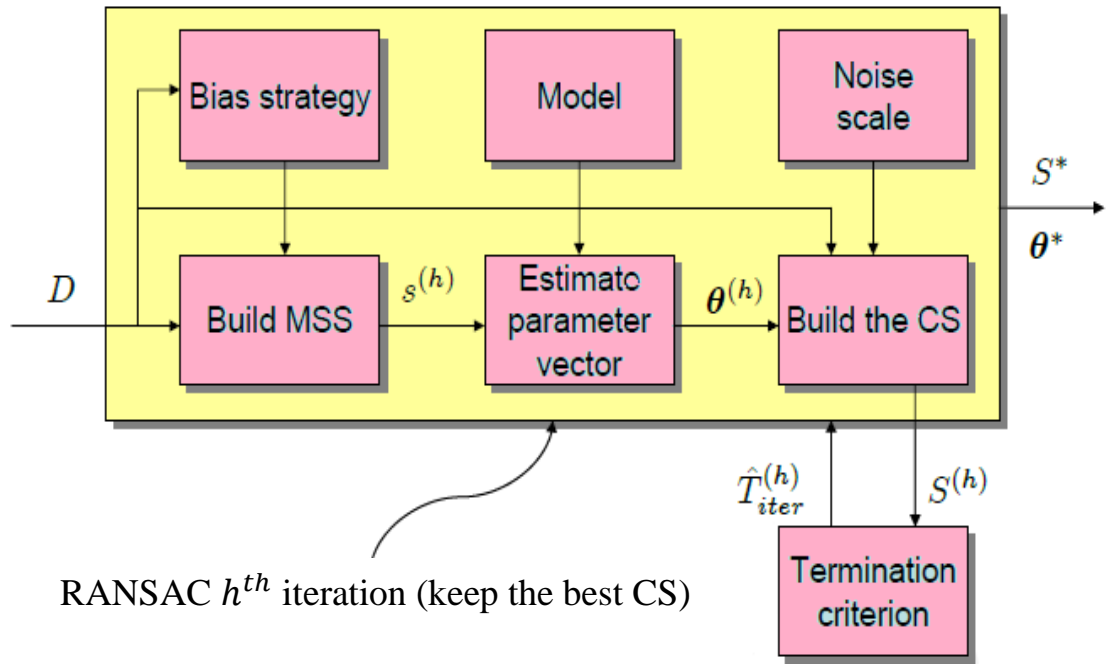


Figure 3.2: Fundamentals of RANSAC Iteration [17]

### 3.1.1 Number of Iterations to estimate the true model

Assume that sampling a MSS $s$ which can result a precise approximation of model parameters, has the probability $q$, therefore the probability of sampling a MSS containing at least one outliner will be 1-$q$. Sampling $h$ MSSs, $(1-q)^h$ will be the probability that all of them contain outliers. The preference is to choose number of iterations large enough to reduce the probability $(1-q)^h$ less than a threshold value [17].

18

$$h \geq \left\lceil \frac{\log \varepsilon}{\log 1 - q} \right\rceil \qquad (3.4)$$

So the threshold value for iterations can be set to:

$$\widehat{T_{iter}} = \left\lceil \frac{\log \varepsilon}{\log 1 - q} \right\rceil \qquad (3.5)$$

If the probabilities of selecting each elements of the dataset are equal then the probability of constructing a MSS containing just inliers is given by the following equation.

$$q = \frac{\binom{N_I}{k}}{\binom{N}{k}} = \frac{N_I! \, (N - k)!}{N! \, (N_I - k)!} \qquad (3.6)$$

In (3.6) the total number of inliers is presented by $N_I$. In order to calculate $q$ we need to have $N_I$.

### 3.1.2 Example of Using RANSAC

In this section we present an example of using RANSAC paradigm to find the best line fitting some points as in Figure 3.3 (a). The minimum number of required points to form a line is two, so in first step the algorithm selects two random points as MSS and estimates the line passing them as shown in Figure 3.3(b). Then the error function in calculated and points inside a certain threshold value are selected. This loop will be repeated for some iteration until the model containing the most number of inliers is found.

19
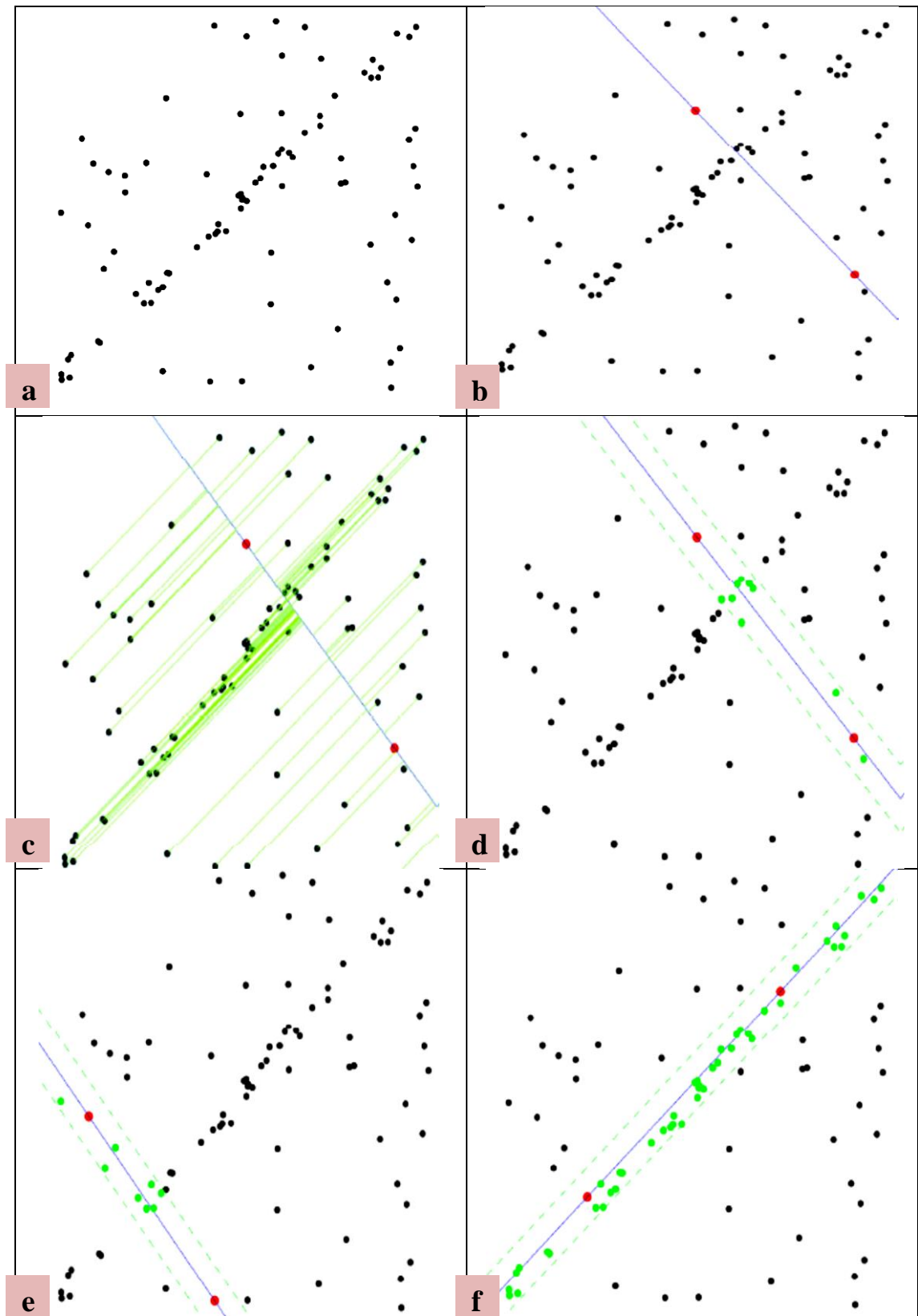
Figure 3.3 : Example of Using RANSAC to Find a Line Passing a Finite Number of Points

## 3.2 Maximum Likelihood Estimation Sample Consensus (MLESAC) and M-estimator Sample Consensus (MSAC)

If in RANSAC algorithm the threshold value determining inliers is considered very high, the robust estimation will be poor. Improving the quality of the consensus set is main idea of introducing MSAC and MLESAC algorithms[18].

### 3.2.1 Maximum Likelihood Estimation in the Presence of Outliers

Considering two images corrupted by zero mean Gaussian noise with standard deviation $\sigma$ the probability density function of data will be as follows. [18]

$$P_r(D|M) = \prod_{i=1...n} \left(\frac{1}{\sqrt{2\pi\sigma}}\right)^n e^{\left(\sum_{j=1,2}\left(\underline{x_i^j}-x_i^j\right)^2+\left(\underline{y_i^j}-y_i^j\right)^2\right)/(2\underline{\sigma}^2)} \tag{3.7}$$

Where $D$ represents the matches set, number of correspondences is denoted by $n$, and $M$ is the transformation between the two images. The following equation represents the negative logarithm of likelihood for all correspondences. [18]

$$-\log\left(Pr\left(x_i^{1,2}|M,\sigma\right)\right) = \sum_{i=1...n}\sum_{j=1,2}\left(\underline{x_i^j}-x_i^j\right)^2+\left(\underline{y_i^j}-y_i^j\right)^2 \tag{3.8}$$

Defining the function $C$ as a cost function, RANSAC algorithm finds the minimum value.

$$C = \sum_i \rho(e_i^2) \tag{3.9}$$

Where $\rho$ is

$$\rho(e^2) = \begin{cases} 0 & e^2 < T^2 \\ constant & e^2 \geq T^2 \end{cases}$$

So increasing the value of $T^2$ there will be more solutions with the same value of $C$ which results in poor estimation. Choosing $T$ large enough all the matches will be inliers. We can minimize a new cost function instead of minimizing $C$.

$$\rho_2(e^2) = \begin{cases} e^2 & e^2 < T^2 \\ T^2 & e^2 \geq T^2 \end{cases} \tag{3.10}$$

We target to minimize the negative logarithm likelihood using the mixing parameter $\gamma$. In order to estimate $\gamma$, the Expectation Maximization algorithm is used. We introduce the parameter $\eta_i$ which is equal to zero if the $i^{th}$ correspondence is an outlier and if it is an inlier the parameter will be equal to one. Firstly we consider some value for $\gamma$ and using this value an expectation of $\eta_i$ is estimated. The estimated $\eta_i$ is again in use to re-estimate the value of $\gamma$. This procedure is repeated until convergence[18].

$$\Pr(\eta_i = 1 \mid \gamma) = \frac{p_i}{p_i + p_o} = z_i \tag{3.11}$$

Where, $p_i$ is the likelihood of a datum to be inlier and $p_o$ is the likelihood of a datum to be outlier.

$$p_i = \left(\frac{1}{\sqrt{2\pi\sigma}}\right)^k exp\left(-\left(\sum_{j=1,2}\left(x_i^j - x_i^j\right)^2 + \left(y_i^j - y_i^j\right)^2\right)/(2\underline{\sigma}^2)\right) \tag{3.12}$$

$$p_o = (1 - \gamma)\frac{1}{v} \tag{3.13}$$

$$\gamma = \frac{1}{n}\sum_i z_i \qquad (3.14)$$

The following steps should be followed in MLESAC and MSAC algorithms by replacing $\rho$ with $\rho_2$.

1. Using a corner detection method corner features are detected.

2. Using cross correlation and proximity, Corner points are matched.

3. The previous steps are repeated for 500 iterations

   - A minimal sample set (MSS) of correspondence is selected randomly

   - The consistent Image relation with the MSS is estimated.

   - For each datum the error is calculated.

   - $c_2$ is calculated for MSAC and $\gamma$ is calculated for MLESAC

4. The best solution among all samples is selected, i.e., the $\min(c_2, -L)$. The MSS leading solution should be stored.

5. The cost function $C$ will be minimized for all correspondences.

# Chapter 4

# SALIENT POINTS OF IMAGE

In general, points containing the dominant information of an image are referred to as salient points. As mentioned in previous chapter the first step of any robust estimation technique is detecting the salient points. Corner points and edges of an image are the best candidates for salient points. In this chapter applied algorithms to detect salient points are discussed.

## 4.1 Corner points

The intersection of two edges is referred to as a corner point. Corner detector algorithms are widely used in applications like image registration, object recognition, motion estimation etc. A large number of corner detector algorithms have been introduced in the literature. Some representative ones are as follow.

### 4.1.1 Moravec corner detection algorithm

Moravec corner detector algorithm [19] developed in 1977 is one of the first techniques to find corner points. In this algorithm corner points are defined as points with enormous intensity alternation in all directions. Considering each pixel location as $(x, y)$ and its intensity as $I(x, y)$, Moravec algorithm runs as follow.

1.  The intensity variation for each pixel from the neighborhood pixel is calculated by equation 4.1 where $a$ and $b$ are the window size.

$$V_{u,v}(x,y) = \sum \left( I(x+u+a, y+v+b) - I(x+a, y+b) \right)^2 \qquad (4.1)$$

2.  Cornerness measure is calculated for each pixel by the following equation.

$$C(x, y) = min\left(V_{u,v}(x, y)\right) \qquad (4.2)$$

3. All $C(x, y)$ less than a certain threshold values are set to zero.

4. In order to find local maxima non-maximal suppression is performed.

Finally all the remaining non-zero points are considered as corners.

### 4.1.2 Harris corner detection algorithm

In Harris and Stephens's corner detection algorithm [20] which is an improved version of Moravec algorithm, rather than using shifted patches the differential of corner score with respect to the direction is considered. The corner score also referred to as autocorrelation is presented by equation 4.3 for the given shift $(x, y)$. In this equation $(x_i, y_i)$ is the corresponding point in the window centered at $(x, y)$ and $I$ is the image function.

$$C(x, y) = \sum_{W} [I(x_i, y_i) - I(x_i + \Delta x, y_i + \Delta y)]^2 \qquad (4.3)$$

Using truncated Taylor expansion $I(x_i + \Delta x, y_i + \Delta y)$ can be approximated as follow.

$$(4.4)$$
$$I(x_i + \Delta x, y_i + \Delta y) \approx \left[I(x_i, y_i) + \left[I_x(x_i, y_i)\; I_y(x_i, y_i)\right]\right]\begin{bmatrix}\Delta x\\ \Delta y\end{bmatrix}$$

$I_x$ and $I_y$ are partial derivatives.

The auto-correlation matrix can be introduced as

$$M = \begin{bmatrix} A & C \\ C & B \end{bmatrix}$$

$$A = \left(\frac{\partial I}{\partial x}\right)^2 \otimes w \qquad (4.5)$$

$$B = \left(\frac{\partial I}{\partial y}\right)^2 \otimes w$$

$$C=\left(\frac{\partial I}{\partial x},\frac{\partial I}{\partial y}\right)^2 \otimes w$$

The corner score $C(x,y)$ can be written as:

$$C(x,y)=det(M) - k\big(trace(M)\big)^2$$
$$det(M) = \lambda_1\lambda_2 = AB - H^2 \tag{4.6}$$
$$trace(M) = \lambda_1+\lambda_2 = A + B$$

The corner will be detected only if both $\lambda_1$ and $\lambda_2$ are large enough positive values these values are determined empirically.

### 4.1.3 Noble corner detection algorithm

In Noble corner detector algorithm [21] the corner score C is defined as a function of matrix M. This algorithm neglects the parameter $k$ previously introduced in Harris algorithm and suggests the following equation as corner score

$$C=\frac{det(M)}{trace(M)+\varepsilon} \tag{4.7}$$

The constant $\varepsilon$ has entered the equation to prevent singularity if $trace(M)$ is equal to zero.

### 4.1.4 SUSAN corner detection algorithm

SUSAN corner detector algorithm [22] firstly introduced by Smith and Brady uses a circular mask to detect corner points. In this algorithm the intensity of the nucleus of mask is compared with all other pixels in the mask and the area of mask with similar intensity as nucleus called USAN (Uni-value Segment Assimilating Nucleus) is chosen. The white area of each mask in Figure 4.1 presents USAN. Assuming $\vec{m}$ is a point in the mask, $\vec{m_0}$ is the nucleus and $t$ is the radius, the comparison function and the area of USAN can be presented as following.

$$C(\vec{m}) = e^{\frac{(I(\vec{m}) - I(\vec{m_0}))^6}{t}}$$

$$n(M) = \sum_{\vec{m} \in M} c(\vec{m})$$

(4.8)

If $c$ is the rectangular function, then $n$ is the number of pixels in the mask which are within $t$ of the nucleus. The response of the SUSAN operator is given by equation 4.9.

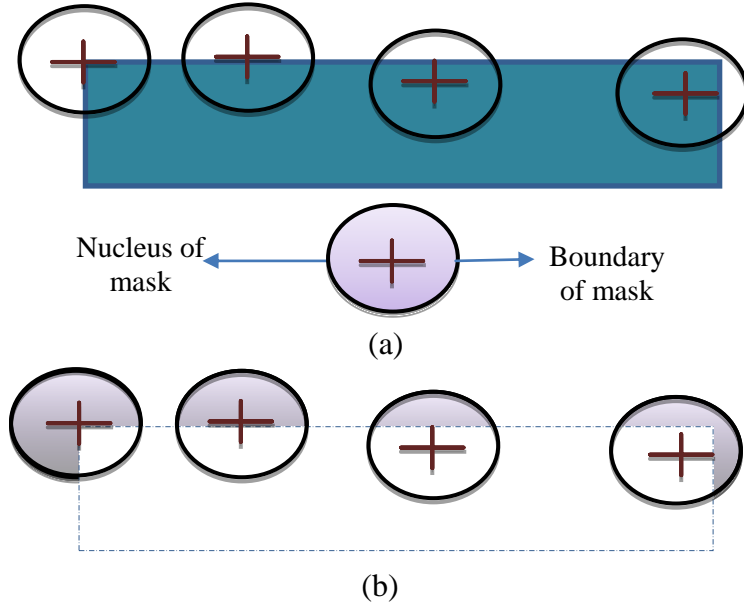$$R(M) = \begin{cases} g - n(M) & n(M) < g \\ 0 & Otherwise \end{cases}$$

(4.9)



Figure 4.1: SUSAN Corner Detection Algorithm

a)  Four circular masks at different locations in a sample image

b)  USANs are shown as the white parts of the mask

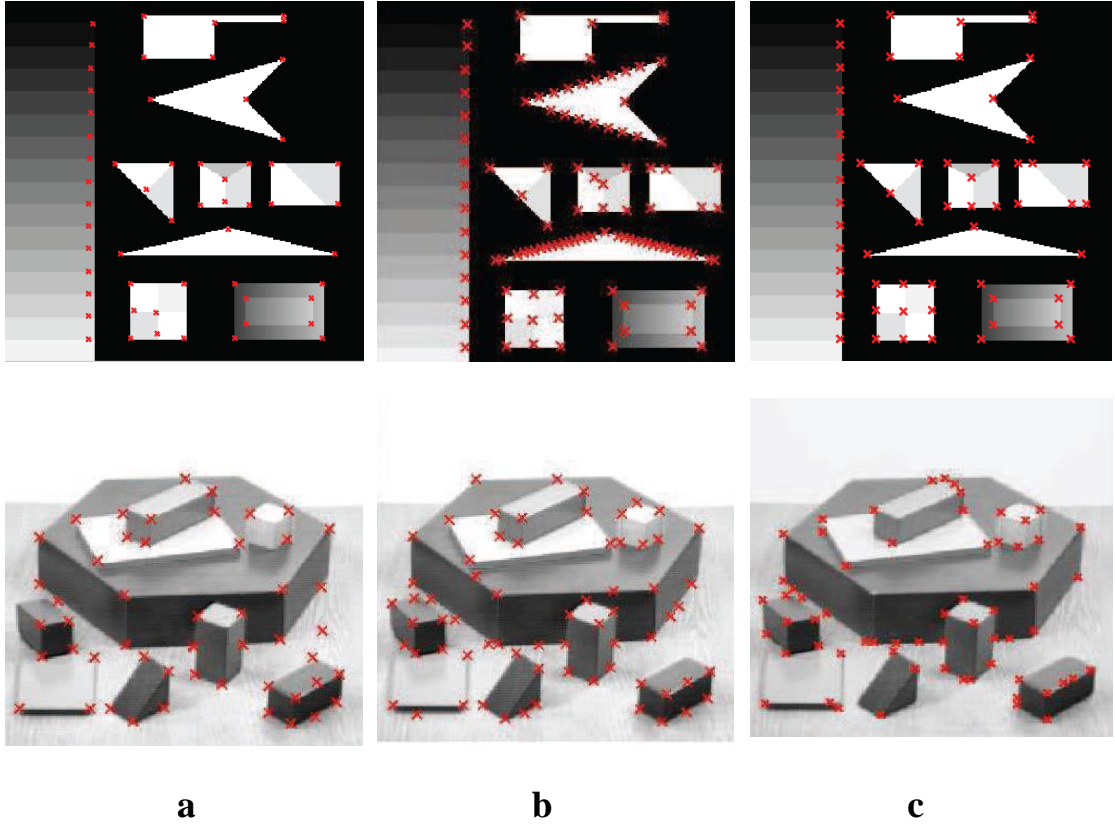Figure 4.2 compares results of different corner detection algorithms.



|  a  |  b  |  c  |

Figure 4.2: Corner Detection Algorithms a) Harris b) Noble c) SUSAN [23]

## 4.2 Edge points

In a digital image edges are points where the intensity sharply changes. Finding edge points is an essential step for many image processing applications like pattern recognition and feature extraction.

Many methods have been proposed in the literature for edge detection. Most of them can be classified in two major categories namely, search-based and zero-crossing based. In search-based methods first of all a measure for edge strength is defined and then estimating the local orientation of the edge it will be checked if the

pixel is local maximum along gradient direction. In zero-crossing based methods zero crossing in Laplacian of image is searched to find edges.

Applying small modifications in many corner detection algorithms can change them to an edge detector. For example in previously explained Harris corner detector algorithm if $\lambda_1 \approx 0$ and $\lambda_1$ has a positive value the detected point is an edge or in SUSAN corner detector if the geometrical threshold $g$ is chosen large enough the algorithm will work as an edge detector.

## 4.3 Blob points

In a digital image points with different properties such as different colors and brightness are referred to as blobs. Blob detection algorithms can be classified in two categories. Differential and local extrema based methods. Differential methods work using the function derivatives considering the position and local extrema based methods try to find the local minima and maxima of the function.

# Chapter 5

# VIDEO STABILIZATION VIA POINT FEATURE MATCHING

This thesis adopts RANSAC paradigm to stabilize a shaky video sequence. The input video frames are modified with the purpose of maintaining a stable image. The implemented framework presented in Figure 5.1 will be discussed in this chapter.
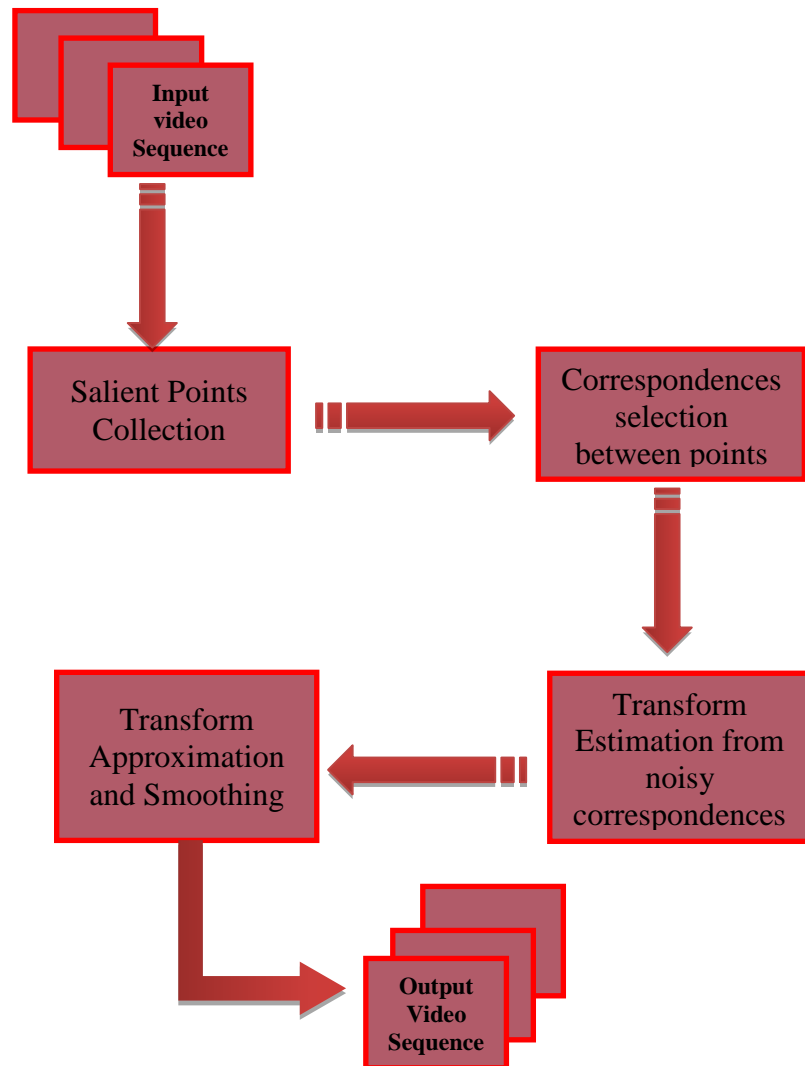


Figure 5.1: Video Stabilization Procedure

## 5.1 Reading Video Frames

The first step of video stabilization algorithm is to read the first two consecutive frames (Frame A and Frame B) of the video as grayscale images.



Figure 5.2: The First Two Frames of the Video

## 5.2 Salient Points Collection

The next step is to find salient points of each frame where SUSAN corner detection algorithm is used. The following matrix is the circular mask introduced in previous chapter.

$$
\begin{bmatrix}
0 & 0 & 1 & 1 & 1 & 0 & 0 \\
0 & 1 & 1 & 1 & 1 & 1 & 0 \\
1 & 1 & 1 & 1 & 1 & 1 & 1 \\
1 & 1 & 1 & 1 & 1 & 1 & 1 \\
1 & 1 & 1 & 1 & 1 & 1 & 1 \\
0 & 1 & 1 & 1 & 1 & 1 & 0 \\
0 & 0 & 1 & 1 & 1 & 0 & 0
\end{bmatrix}
\qquad (5.1)
$$

31

The circular mask will be placed over all pixels in image to test whether the point is a corner point. Figure 5.2 depicts corner points detected by SUSAN algorithm in the two consecutive frames.
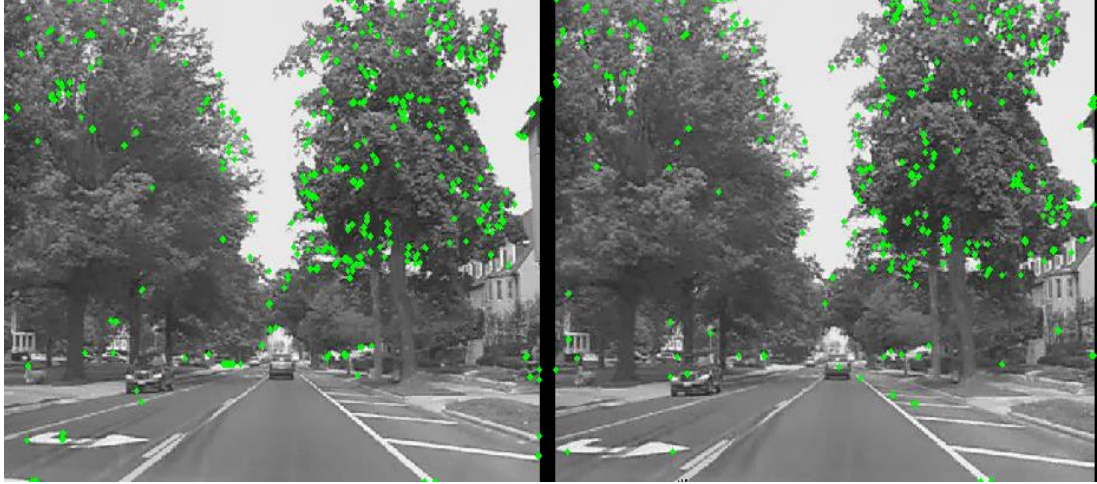


Figure 5.3: Feature Points in Frame A and B

## 5.3 Correspondences Selection between Points

In order to stabilize a video sequence we mainly need to find a transformation which reduces the amount of distortion between frames. In this step the likely correspondences between the derived points of interest are selected. In order to find the correspondences between feature points we extract a $9 \times 9$ block centered on each point. Sum of Squared Differences (SSD) is then adopted as the matching cost between respective points.

For two images $f(x, y)$ and $g(x, y)$ SSD can be defined as following.

$$\text{SSD}(d_1, d_2) = \sum_{i=-n_1}^{n_1} \sum_{j=-n_2}^{n_2} \left( f(x + i, y + j) - g(x + i - d_1, y + j - d_2) \right)^2 \qquad (5.2)$$

Where the summation extends over a region of size $(2n_1 + 1) \times (2n_2 + 1)$ and as we have chosen a $9 \times 9$ block so $n_1 = n_2 = 4$.

There exists one point in Frame B which corresponds to the points in Frame A. When finding all possible matching costs the algorithm searches to find the lowest one which means the best cost.
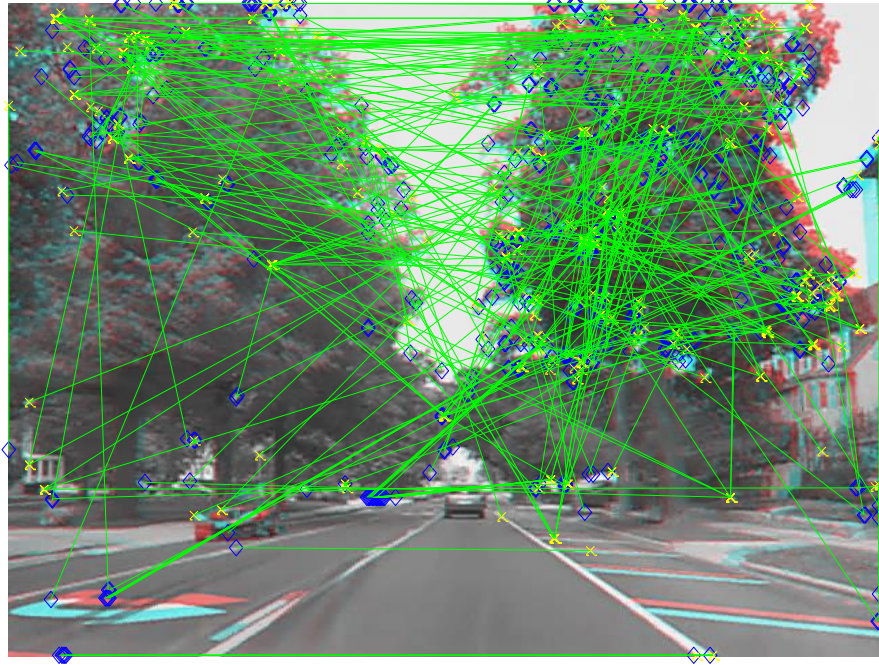


Figure 5.4: Initial Correspondences Between Frames A and B

## 5.4 Transform Estimation from Noisy Correspondences

Numerous correspondences achieved in the previous step are not acceptable. Using the Random Sample Consensus (RANSAC) algorithm, a robust estimate of transformation between Frame A and Frame B can be derived. Receiving the point correspondences from the previous step, the video stabilization algorithm searches to find effective inlier correspondences and afterward it derives the affine transformation mapping the inliers in Frame A to Frame B. This transformation is only capable to alter the image plane.

As mentioned in Chapter 2 the affine transform is a matrix of the following form.

$$\begin{bmatrix} a_1 & a_3 & t_x \\ a_2 & a_4 & t_y \\ 0 & 0 & 1 \end{bmatrix} \tag{5.3}$$

Where $t_x$ and $t_y$ are translation parameters and $a_1, a_2, a_3$ and $a_4$ describe sheering effect, rotation and scale. The affine transform targets to overlay the correspondence points on each other by warping the image.

This geometric transformation is estimated several times and for each result a cost is calculated based on the Sum of Absolute Differences (SAD) between frame A and B. The best transform which minimizes the cost is selected. This procedure increases the robustness.

SAD is the most commonly used algorithm which measures the distortion between two images by evaluating the similarity between image blocks. Equation 5.4 defines the SAD between elements in two image blocks.

$$\text{SAD} = \sum_{i=1}^{N} \sum_{j=1}^{N} \left| c_{i,j} - r_{i,j} \right| \qquad (5.4)$$

Where $r_{i,j}$ represent elements in the first frame and $c_{i,j}$ are the elements in the second one.

In the color composite Figure 5.5 the re-projected Frame B is laminated on Frame A. As illustrated in the figure the inlier correspondences strongly match. The centers of the images are aligned where the red-cyan color composite is almost black and white.
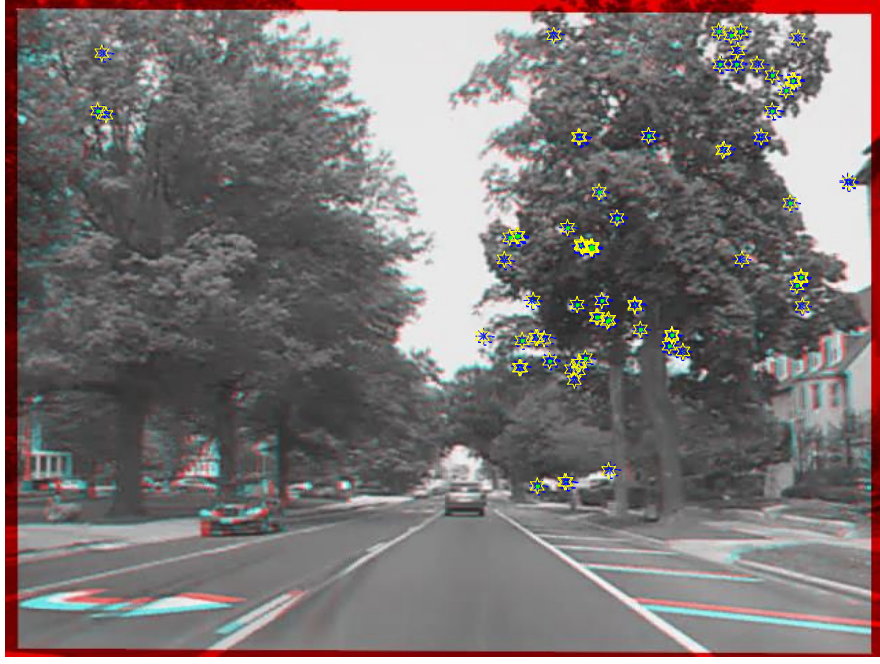
Figure 5.5: Correct Correspondences Based on RANSAC Paradigm

## 5.5 Transform Approximation and Smoothing

Steps 1 to 4 can be used to estimate the distortion between two consecutive frames as affine transformations $H_i$ in a complete video sequence. The product of all $H_i$s as explained by equation 5.5 is the cumulative distortion of Frame $i$ compared to the first one.

$$H_{cumulative,i} = \prod_{j=0}^{i-1} H_i \qquad (5.5)$$

Kalman filtering and numerical optimization are two different ways to smooth the mentioned cumulative transform of images transforms.

Convolution of the time sequence of $H_{cumulative}$ parameters with a Gaussian filter can be a simpler approach for smoothing. This convolution can remove high-frequency noise referred to as camera jitters.

For stability and numerical simplicity, the affine transform matrix 5.3 is replaced with the simpler following matrix containing scale, rotation, translation parameters.

$$H_{sRt} = \begin{bmatrix} s \times \cos(ang) & s \times -\sin(ang) & t_x \\ s \times \sin(ang) & s \times \cos(ang) & t_y \\ 0 & 0 & 1 \end{bmatrix} \tag{5.6}$$

Where $s$ is the scale factor, $t_x$ and $t_y$ are the two translation parameters and the parameter *ang* is the angle describing the rotation. This matrix contains two translation factors, one angle and one scale. In order to show that the error of replacing the transform $H$ with the equivalent transform given as equation 5.6 is minimal, we re-projected the two processed Frame B on each other as a red-cyan composite which is depicted in Figure 5.6. The pixel-wise difference between images can be neglected and the image appears nearly black and white.

## 5.6 Running the Full Video

The last step of video stabilization algorithm is to run the above procedure in a loop for all frames in a video sequence.

The transform $H$ is calculated between consecutive frames in each step and the result is smoothed by fitting $H$ as an s-R-t transform. Then the result in each loop is combined with the $H_{cumulative}$ describing the entire camera motions. This transform is estimated in every loop.

Figure 5.6: Color Composite of Affine and s-R-t Transform Outputs

# Chapter 6

# SIMULATION RESULTS

In this work three shaky videos from different scenes are stabilized using point feature matching technique presented in Chapter 5. The stabilization technique is implemented using the MATLAB platform. The improvement in the quality of the stabilized videos are evaluated using mean of video frames, the normalized sum of absolute difference between consecutive frames and SVD based grayscale image quality assessment metric. The translation parameter in $x$ and $y$ directions are also computed for 50 frames of each video sequence.

## 6.1 Comparison between Mean of Video Frames for Stabilized Video and Unprocessed Shaky Video

Means for stabilized sequence and unprocessed shaky videos have been computed once the tested video sequences are stabilized using the point feature matching technique. Results obtained for three different videos have been compared in Figure 6.1– 6.3 where the subfigure (a) depicts the mean of raw inputs and the subfigure (b) depicts the mean of the stabilized video sequences.

Video A taken by a car shows a straight road where some other cars are moving. The difference between video frames is mainly caused by the camera vibration so the stabilization algorithm is rather efficient and the mean of stabilized video frames has almost no distortion.

(a)



(b)

Figure 6.1: Video Sequence A

   a)   Mean of Unprocessed Shaky Video

   b)   Mean of Stabilized Video Sequence

Video B from the entrance of the Eastern Mediterranean University is distorted only by the camera vibration and the stabilization algorithm is quite efficient resulting a clear video mean.



(a)



(b)

Figure 6.2: Video Sequence B

a) Mean of Unprocessed Shaky Video

b) Mean of Stabilized Video Sequence

Video C from a pathway is distorted only by the camera vibration and the stabilization algorithm is quite efficient resulting in a clear video mean where the cobblestone lines become visible.



(a)



(b)

Figure 6.3: Video Sequence C

a) Mean of Unprocessed Shaky Video

b) Mean of Stabilized Video Sequence

## 6.2 Comparison between Normalized Sum of Absolute Difference between Consecutive Frames for Stabilized Video and Unprocessed Shaky Video

In this section we estimate the normalized sum of absolute difference Error (NSAD) between consecutive frames of unprocessed shaky video and stabilized video sequences. The results are then compared using pair processing graphs. NSAD is calculated using equation 6.1.

$$NSAD = \frac{\sum_{i=1}^{N} \sum_{j=1}^{M} \left| c_{i,j} - r_{i,j} \right|}{N \times M}$$

(6.1)

Where $c$ and $r$ are two consecutive frames and $M$ and $N$ represent the image size in horizontal and vertical directions. Figures 6.4 - 6.10 illustrate the NSAD for video sequence A-C respectively.

These diagrams show higher NSAD values for unprocessed shaky videos which means that the video frames in unprocessed videos are more different to each other. The video stabilization algorithm decreases the NSAD value between consecutive frames. As depicted in Figures 6.6- 6.10 in videos B and C the different between consecutive frames is considerably reduced and there is no interaction between NSAD graph of stabilized and shaky videos. The reason is that in these two videos the camera does not have any forward movement and only shakes in its position.
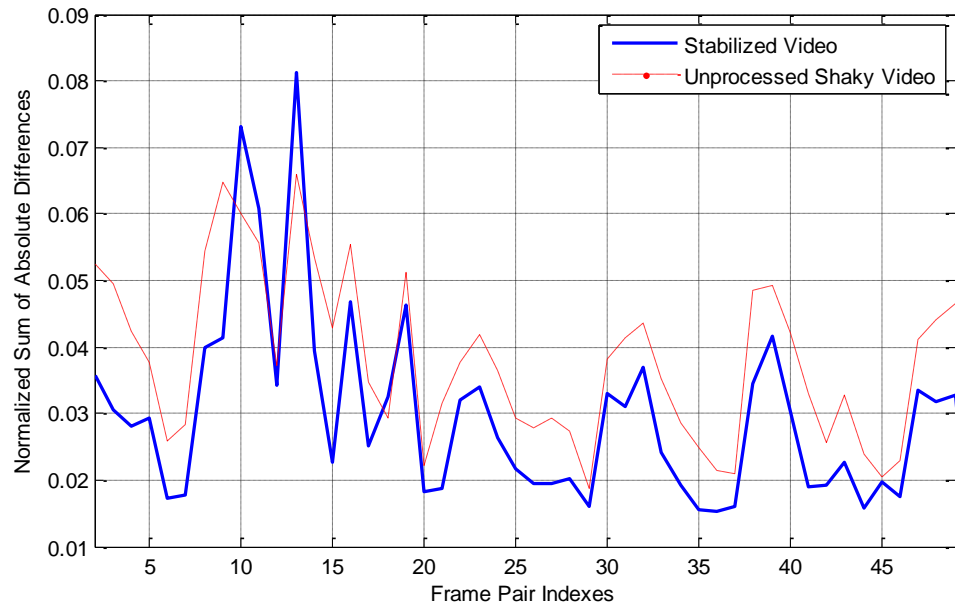
Figure 6.4: Normalized Sum of Absolute Difference for Video-A

In Video A the mean value of the Normalized Sum of Absolute Differences for 50 frame is reduced from 0.0366 to 0.0288.
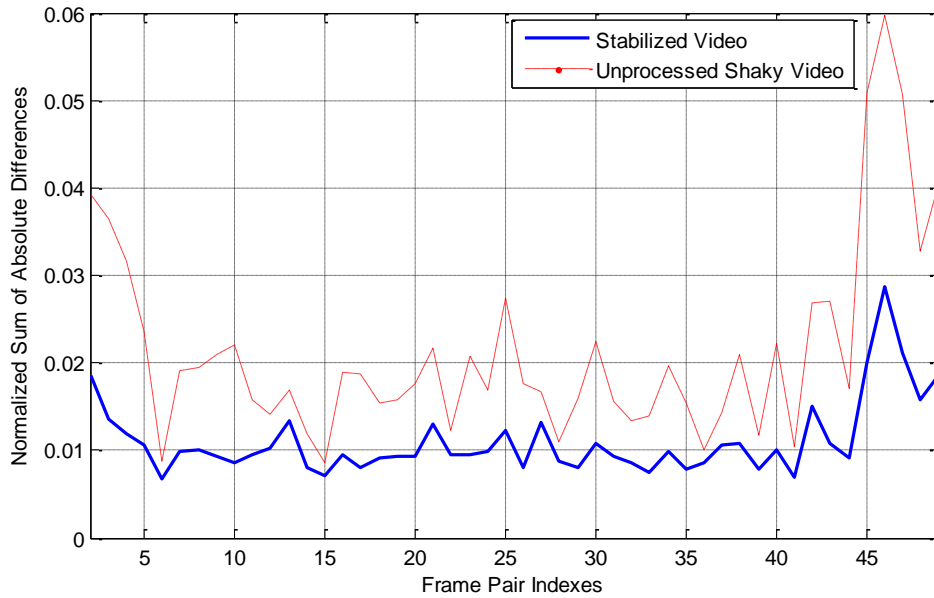


Figure 6.5: Normalized Sum of Absolute Differences for Video-B

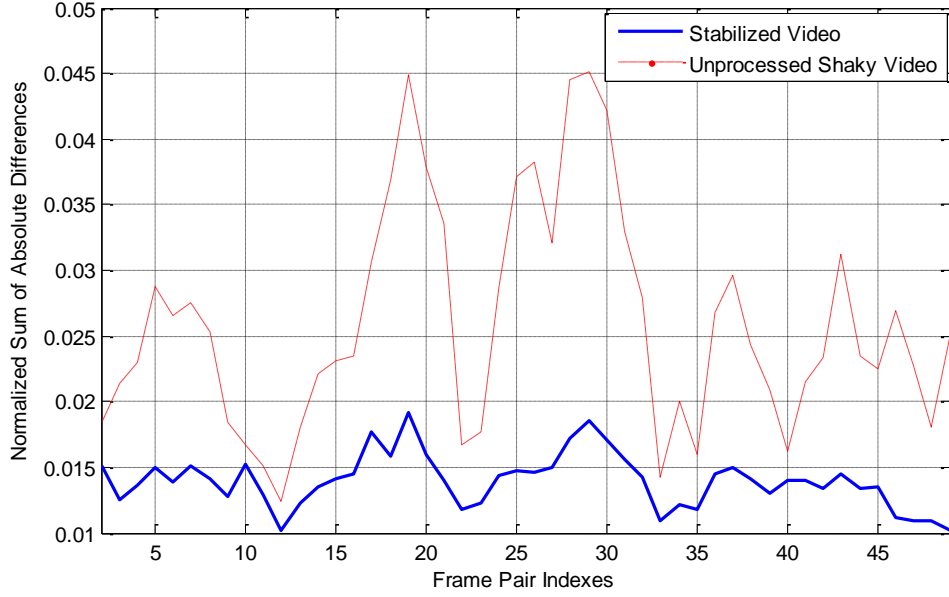In Video B the mean value of the NSAD for 50 frames are reduced from 0.0206 to 0.0106.

Figure 6.6: Normalized Sum of Absolute Difference for Video-C

In Video C the mean value of the Normalized Sum of Absolute Differences for 50 frame is reduced from 0.0250 to 0.0134.

## 6.3 SVD Based Image Quality Assessment

Most of the video stabilization algorithms mainly target to satisfy human perception. However in this work the SVD based grayscale image quality assessment method is also adopted to evaluate the quality of the output video.

The quality of distorted videos and images can be expressed using the recently developed measurement technique based on SVD. This method presented by Shnayderman introduces both a scalar value measurement called $M - SVD$ and a two dimensional graphical measurement to determine the image quality [24].

As we know any real Matrix A can be decomposed as follows.

44

$$A = USV^T$$

Where,

$$U^T U = I \tag{6.2}$$

$$VV^T = I$$

$$S = \text{diag}(s_1, s_2, \dots)$$

## 6.3.1 SVD-Based Graphical Measure between Consecutive Frames for Stabilized Video and Unprocessed Shaky Video

In graphical measure technique the gray scale image of the first frame and last frame are decomposed in smaller blocks (in this work $8 \times 8$) and for each block the singular value is computed. Then according to equation 6.3 the distance between the singular values is measured.

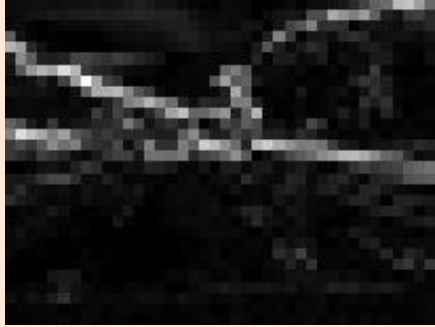$$D_i = \left[ \sum_{i=1}^{n} (s_i - \hat{s}_i)^2 \right]^{\frac{1}{2}} \tag{6.3}$$

Where $n$ is the block size. Considering an image of size $k \times k$ number of blocks can be obtained as $(k/n) \times (k/n)$.

We compute all $D_i s$ related to each block. The obtained $D_i s$ will form a new matrix of size $(k/n) \times (k/n)$ which introduces the Graphical Measurement of video. When there is less distortion between two images the distance between the singular values will be less. If we obtain the graphical measure of an image with itself the result is a zero matrix of size $(k/n) \times (k/n)$ i.e. a black image.

Table 6.1 provides the SVD based Graphical measurement between the first two frames of the three selected videos for unprocessed shaky and stabilized videos. As the stabilized videos have less distortions between the consecutive frames, the graphical measure has less values which means lower intensities.

Table 6.1: SVD-Based Graphical Measurement

| | Stabilized Video | Unprocessed Shaky Video |
|---|---|---|
| Video A |  |  |
| Video D |  |  |
| Video E |  |  |

### 6.3.2 SVD-Based Numerical Measure between Consecutive Frames for Stabilized Video and Unprocessed Shaky Video

The numerical value known as $M - SVD$ is derived from the previously presented graphical method. The $M - SVD$ value is calculated using equation 6.4.

$$M - SVD = \frac{\sum_{i=1}^{(k/n) \times (k/n)} |D_i - D_{mid}|}{(k/n) \times (k/n)} \tag{6.4}$$

Where $D_{mid}$ is the midpoint of sorted $D_i$ values.

M-SVD values between consecutive frames are estimated before and after stabilization for the three selected videos and the results are compared in Figure 6.7-Figure 6.9. The M-SVD values for stabilized videos are less than shaky ones which shows less distortion between frames. As also indicated by the mean of video frames video B and C where the difference between video frames are produced only by camera vibration the stabilization algorithm is more efficient and the quality of stabilized video is highly improved.
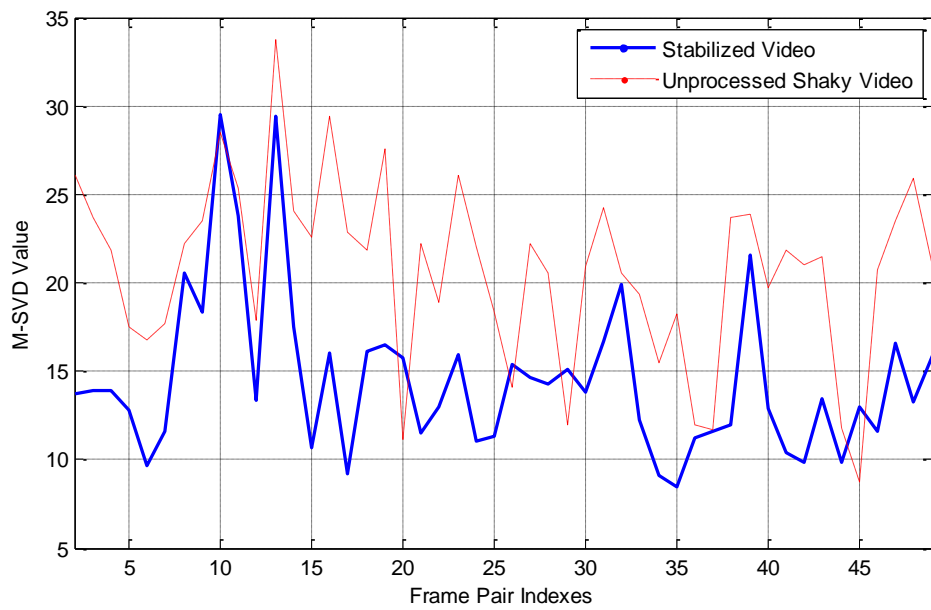


Figure 6.7: M-SVD Measure for Consecutive Frames of Video-A

47

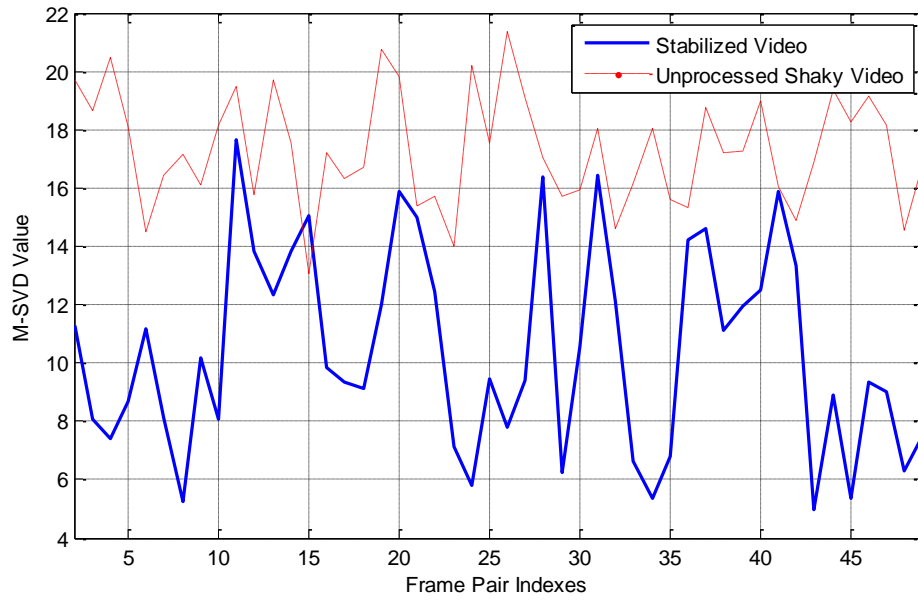In Video A the mean of estimated M-SVD values is reduced from 19.9301 to 13.9512.



Figure 6.8: M-SVD Measure for Consecutive Frames of Video-B

In Video-B the mean of estimated M- SVD values is reduced 16.6378 to 9.9928.
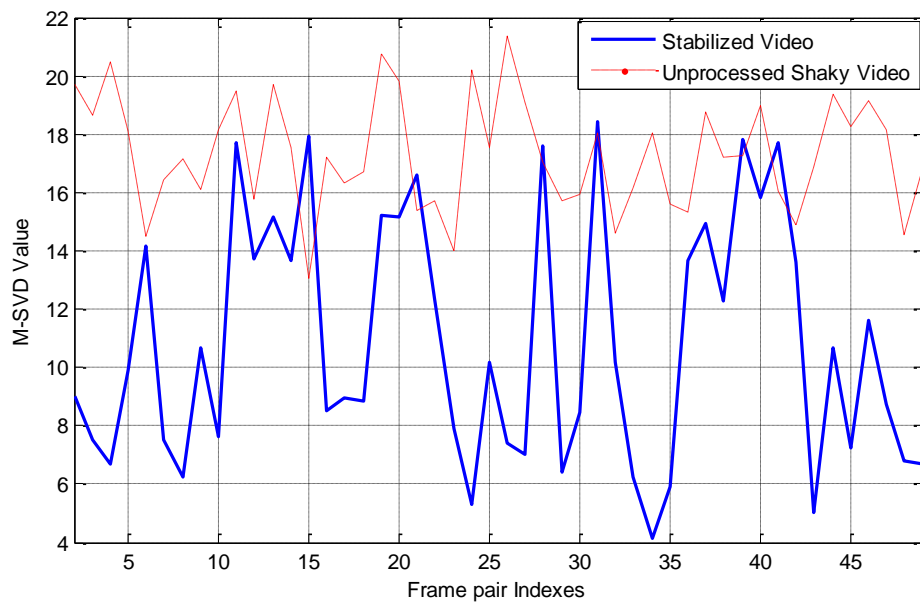


Figure 6.9: M-SVD Measure for Consecutive Frames of Video-C

In Video-C the mean of estimated M- SVD values is reduced 15.3051 to 8.3011.

## 6.4 Peak Signal-to-Noise Ratio Improvement for Stabilized Video

As the results confirm the video stabilization algorithm has the highest efficiency for videos A, D and E where the differences between frames are resulted only by the camera vibration. In this section the Peak Signal-to-Noise Ratio (PSNR) are computed for these videos and the results are presented in Figures 6.16- 6.19.

The PSNR between consecutive frames can be considered as a measure of the departure from the optimal case, or as a measure of the overlap between two frames. The PSNR value which is maximized for identical video frames is computed using equation 6.5 [25].

$$\text{PSNR}(I_1, I_0) = 10 \log \frac{255^2}{MSE((I_1, I_0))} \tag{6.5}$$

Where 255 is the maximum intensity for grayscale images. $I_1$ and $I_0$ are two consecutive frames and the Mean Squared Error (MSE) is calculated using equation 6.6.

$$\text{MSE} = \frac{1}{MN} \sum_{n=1}^{M} \sum_{m=1}^{N} (I_1 - I_0)^2 \tag{6.6}$$
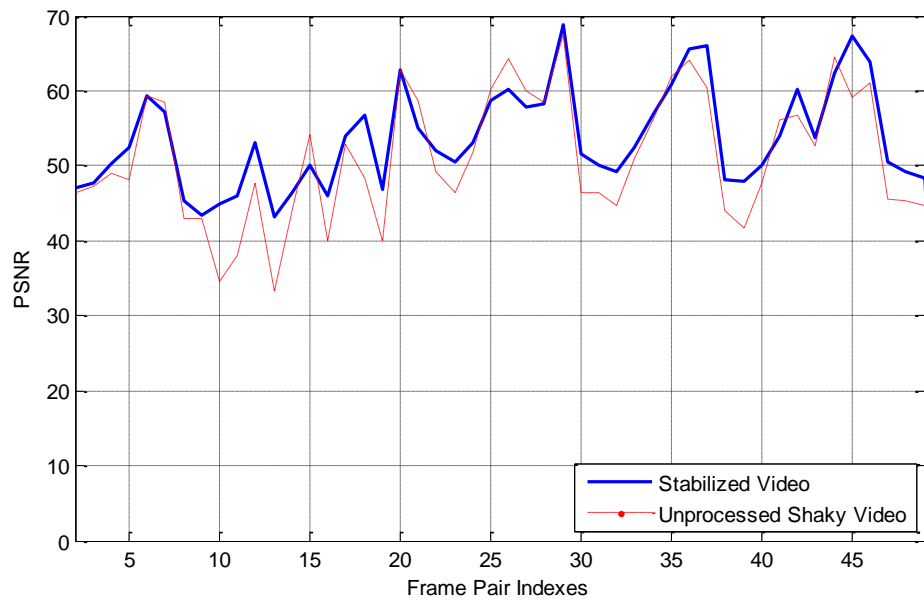
*M* and *N* are the image size.

Figure 6.10: PSNR for Consecutive Frames of Video-A

In video A the mean value of PSNR between consecutive frames is increased from 49.1601 to 51.5108.
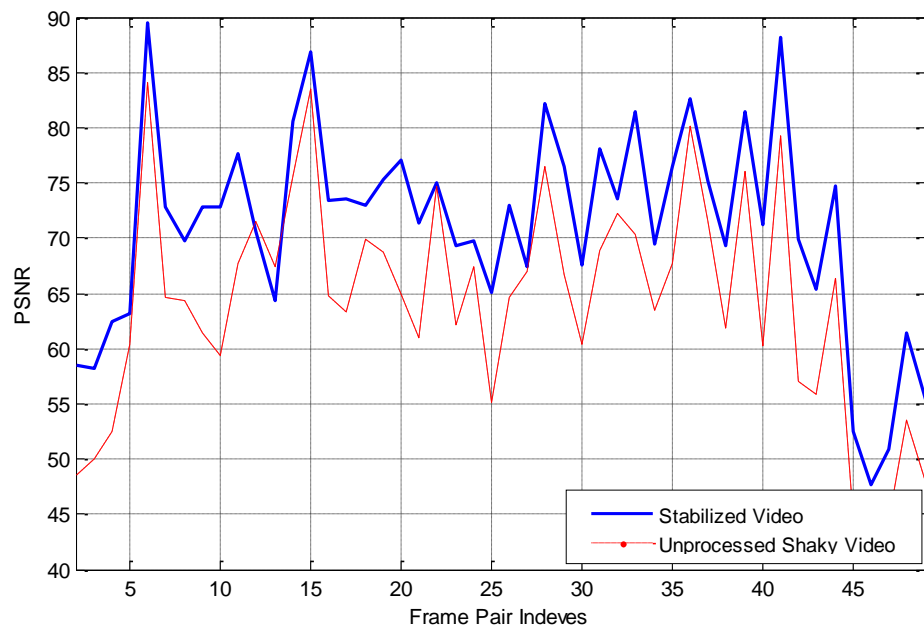


Figure 6.11: PSNR for Consecutive Frames of Video-B

50

In video B the mean value of PSNR between consecutive frames is increased from 61.6273 to 68.2914.
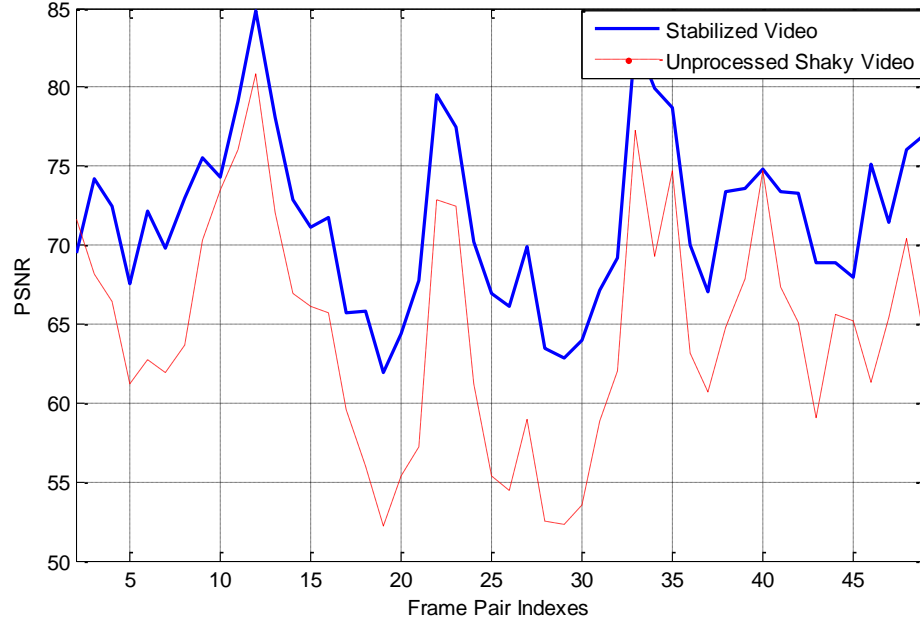


Figure 6.12: PSNR for Consecutive Frames of Video-C

In video C the mean value of PSNR between consecutive frames is increased from 61.9572 to 68.8203.

## 6.5 Translation Parameters in *x* and *y* Directions Between Consecutive Frames for Stabilized Video and Unprocessed Shaky Video

The video stabilization algorithm removes the undesired translations in *x* and *y* direction. The scale, rotation, translation matrix described in Equation 5.6 is used to find $t_x$ and $t_y$ between consecutive frames for shaky videos and stabilized ones and the results are provided in Figure 6.13-6.18. The results indicate a significant reduction in translation parameters in stabilized videos.

Figure 6.13: Translation in *x* Direction for Video-A

In Video-A the variance of translation parameter in *x* direction is reduced from $5.4030 \times 10^3$ to 38.5628 and the deviation from zero is reduced from 68.9897 to 2.9443.



Figure 6.14: Translation in *y* Direction for Video A

In Video-A the variance of translation parameter in $y$ direction is reduced from $1.4147\times 10^3$ t 13.6498 and the deviation from zero is reduced from 39.3373 to 1.5470.



Figure 6.15: Translation in $x$ Direction for Video-B

In Video-B the variance of translation parameter in $x$ direction is reduced from $1.4938\times 10^5$ to $1.3631\times 10^3$ and the deviation from zero is reduced from 243.3358 to 35.5892.
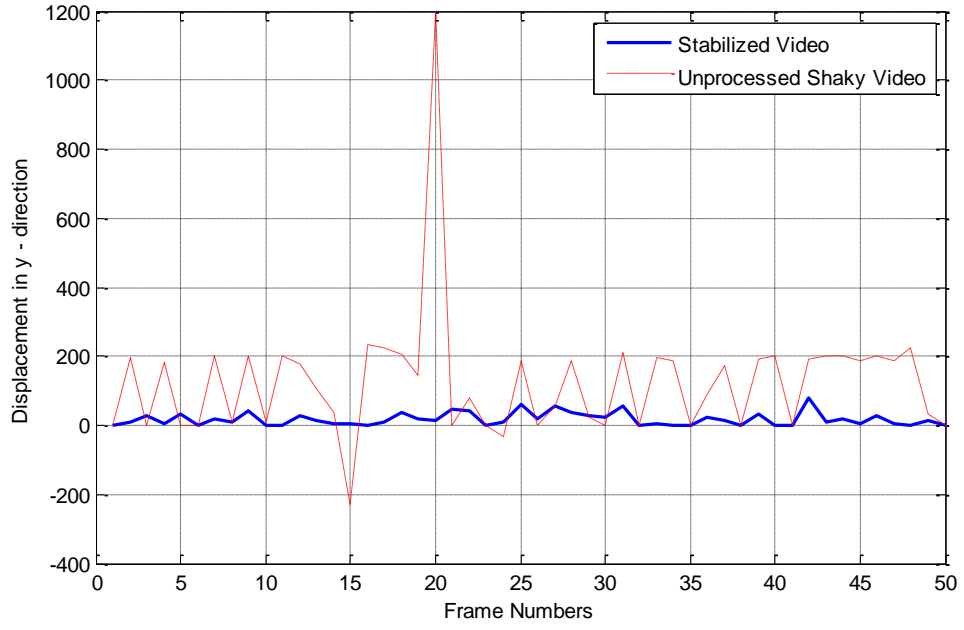
Figure 6.16: Translation in *y* Direction for Video-B

In Video-B the variance of translation parameter in y direction is reduced from $3.4720\times 10^4$ to 354.8653 and the deviation from zero is reduced from 138.8844 to 18.4567.
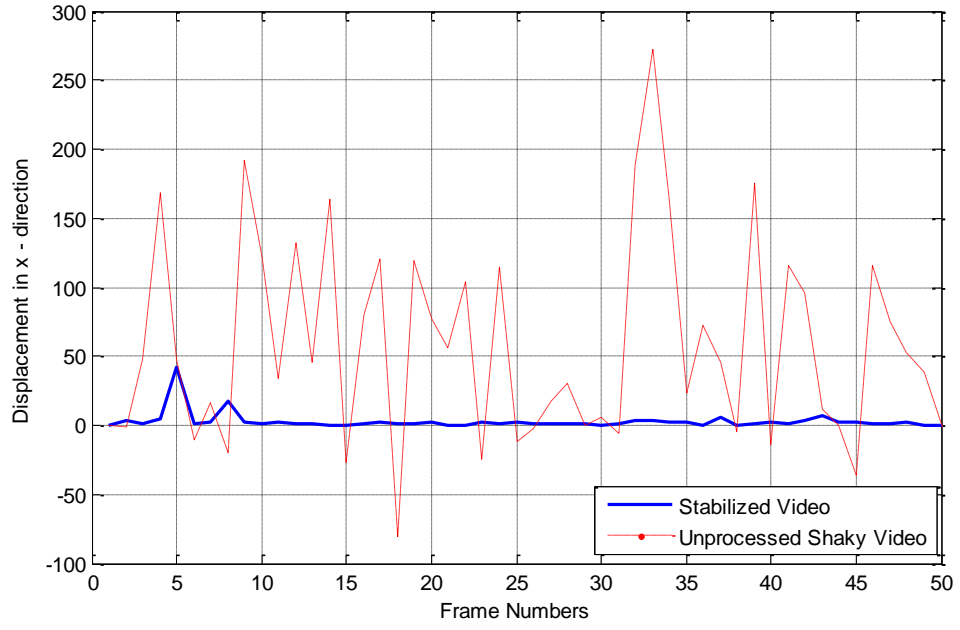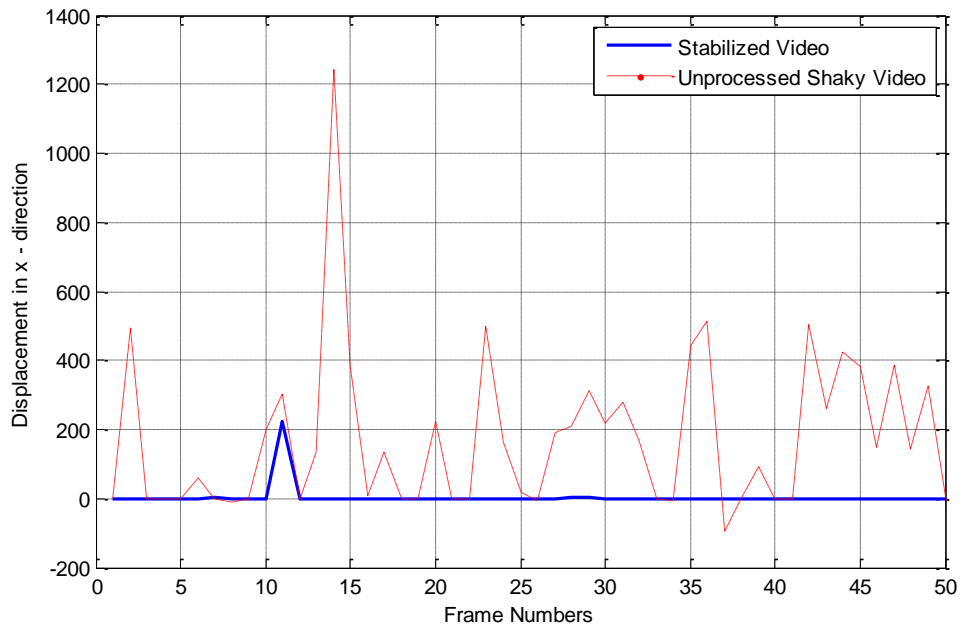


Figure 6.17: Translation in *x* Direction for Video C

In Video-C the variance of translation parameter in $x$ direction is reduced from $5.4288 \times 10^4$ to 992.2415 and the deviation from zero is reduced from 183.5995 to 5.0150.
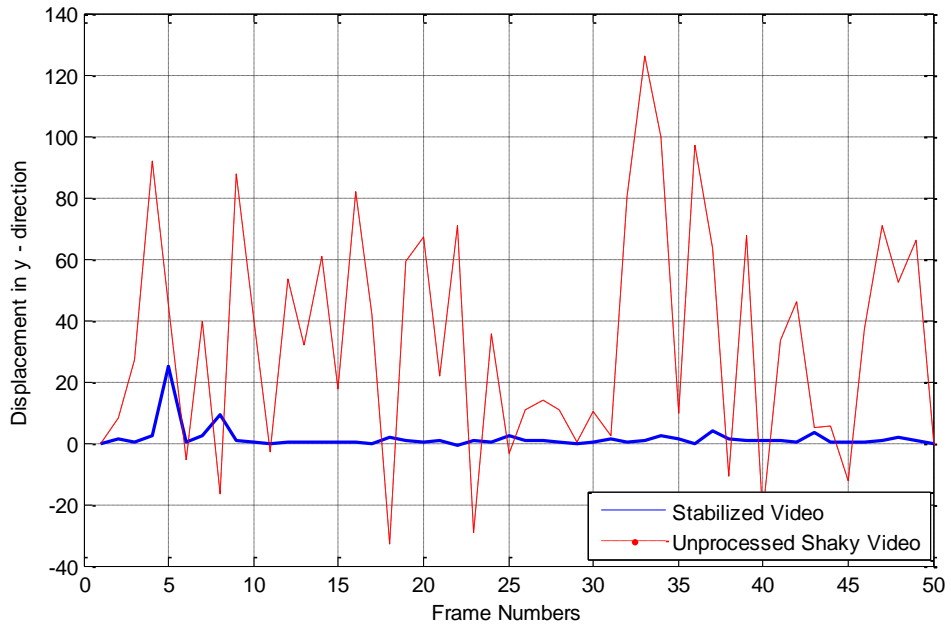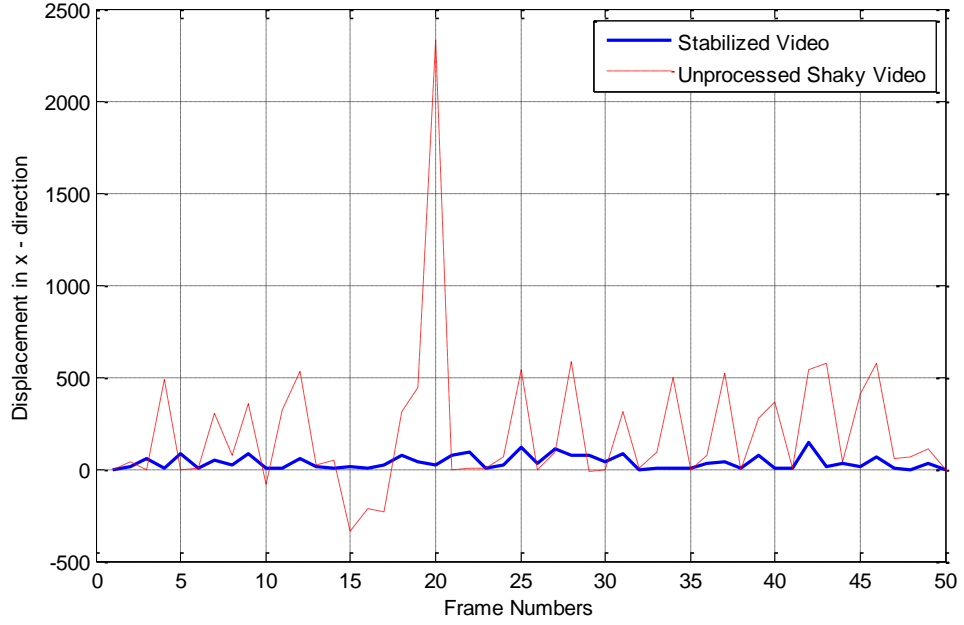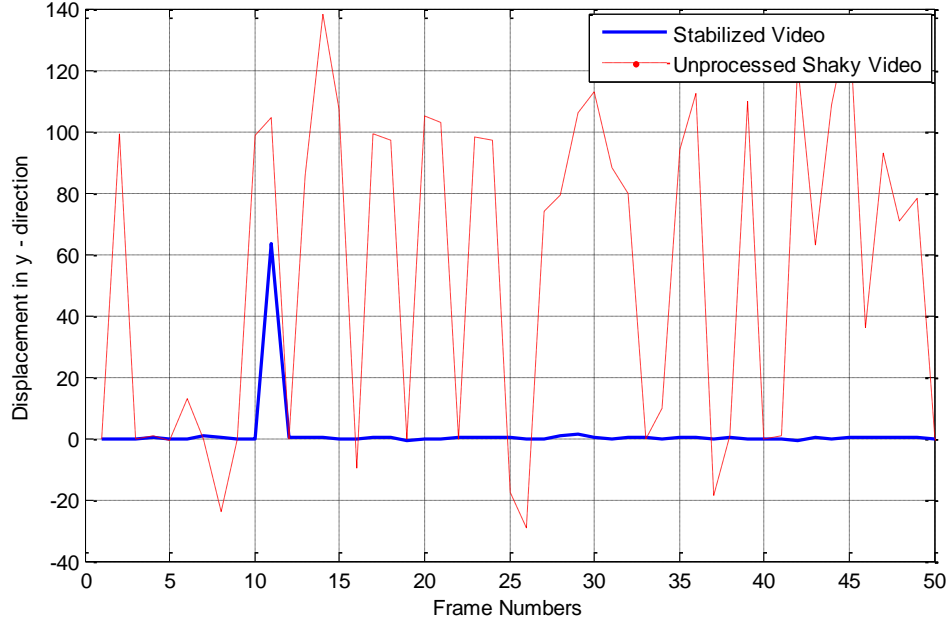


Figure 6.18: Translation in *y* Direction for Video-C

In Video-C the variance of translation parameter in $y$ direction is reduced from $2.7828 \times 10^3$ to 80.8687 and the deviation from zero is reduced from 59.6336 to 1.5227.

Table 6.2 provides NASD, M-SVD, PSNR measurements and specifies amounts of translations in *x* and *y* directions for the stabilized and shaky videos used in this study. The table indicates that the stabilization algorithm improves the PSNR value on average by 5.3dB. NSAD and M-SVD were also improved by 32.11 % and 37.88. Finally it can be observed that translations in *x* and *y* directions were on average reduced 91.21 % and 92.39% respectively.

Table 6.2: Comparison Between Stabilized and Shaky Videos

| | Video A | | Video B | | Video C | |
|---|---|---|---|---|---|---|
| | Processed Video | Shaky Video | Processed Video | Shaky Video | Processed Video | Shaky Video |
| **NSAD** | 0.0288 | 0.0366 | 0.0106 | 0.0206 | 0.0134 | 0.0250 |
| **M-SVD** | 13.9512 | 19.9301 | 9.9928 | 16.6378 | 8.3011 | 15.3051 |
| **PSNR** | 51.5108 | 49.1601 | 68.2914 | 61.62 | 68.8203 | 61.9572 |
| **x Translation (Deviation from zero)** | 2.9443 | 68.9897 | 35.5892 | 243.3358 | 5.0150 | 183.5995 |
| **y Translation (Deviation from zero)** | 1.5470 | 39.3373 | 18.4567 | 183.8844 | 1.5227 | 59.6336 |

# Chapter 7

# CONCLUSIONS AND FUTURE WORK

## 7.1 Conclusion

In this work a point feature matching technique based on RANSAC paradigm has been adopted to stabilize shaky videos. Finding the feature points using SUSAN corner detection algorithm in each frame we estimated the motion between the subsequent frames and then video frames have been warped to remove the jitters. 50 frames of three different video sequences are stabilized using the explained algorithm. Mean of stabilized videos and unprocessed shaky ones are compared and for all videos the image core has less distortion than foreground objects.

The NSAD, the SVD based graphical and numerical measurements, the peak signal to noise ratio and translation in $x$ and $y$ directions have been utilized to evaluate the quality of stabilized videos. NSAD diagram indicated a 32.11% improvement in stabilized videos. M-SVD based graphical measurement resulted in darker images for stabilized video which shows less distortion between consecutive frames. The numerical measurements were also lower in stabilized sequences for 37.88%. The PSNR is improved on average by 5.3 dB and the translations in $x$ and $y$ directions were reduced by 91.21 % and 92.39% respectively.

Comparing the complete stabilized and shaky video also confirmed that the processed videos highly satisfy the human perception. Results indicate a remarkable elimination of high jittery from shaky videos.

## 7.2 Future Works

As future work we can carry out background estimation to detect only the moving foreground objects. Then select the salient points for the moving foreground objects and determine the correspondence points between those FG objects in two consecutive frames. This will help reduce the computational complexity and speed up the computations.

Also, when the transformations in different frames are corrected the outside borders of each frame will become black due to corrections in rotation angle and inevitably when the video is reconstructed its outside borders are not perfect. In the future it is possible to address this problem through use of image inpainting which we believe will further improve PSNR values.

# REFERENCES

[1] R. Kumar, H. Sawhney, S. Samarasekera, S. Hsu, H. Tao, Y. Guo, K. Hanna, A. Pope, R. Wildes, D. Hirvonen, M. Hansen, and P. Burt., "Aerial video surveillance and exploitation," *Proceedings of the IEEE,* vol. 89, no. 10, pp. 1518-1539,, 2001.

[2] M. Oshima, T. Hayashi, S. Fujioka, T. Inaji, H. Mitani, J. Kajino, K. Ikeda, and K. Komoda., "VHS camcorder with electronic image stabilizer," *IEEE Transactions Consumer Electronics,* vol. 35, no. 14, p. 749–758, 1989.

[3] "Camera Stuff Review," [Online]. Available: http://www.camerastuffreview.com/. [Accessed 14 January 2015].

[4] K. Sato, S. Ishizuka, A. Nikami, and M. Sato, "Control techniques for optical image stabilizing system," *IEEE Transactions Consumer Electronics,* vol. 39, no. 3, p. 461–466, 1993.

[5] "Canon," [Online]. Available: http://www.usa.canon.com/. [Accessed 14 January 2015].

[6] C. Morimoto, R. Chellappa, "Fast electronic digital image stabilization," in *Proceedings of the 13th International Conference on Pattern Recognition*, Vienna, August 1996.

[7] P. Burt and P. Anandan, "Image Stabilization by registration to a reference mosaic," in *Proc. DARPA Image Understanding Workshop*, Monterey, CA, November 1994.

[8] M. Hansen, P. Anandan, K. Dana, G. van der Wal, and P.J. Burt., "Realtime

scene stabilization and mosaic construction," in *In Proc. DARPA Image Understanding Workshop*, Monterey, CA, November 1994.

[9] Z. Duric and A. Rosenfeld, "Stabilization of image sequences," Center for Automation Research, University of Maryland, College Park, 1995.

[10] P. B. a. R. C. Y. Yao, "Fast Electronic image stabilization," in *Proc. International Conference on Image*, Washington, D.C., October 1995.

[11] Matsushita Y., Ofek E., Tang X., and Shum H.Y, "Full-frame Video Stabilization with Motion Inpainting," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. 28, no. 7, pp. 1150 - 1163, July 2006.

[12] K. Ratakonda, "Real-time digital video stabilization for multi-media application," in *Proceedings of the 1998 IEEE International Symposium on Circuits and Systems,*, Monterey, CA, USA, 1998.

[13] Andrew Litvin, Janusz Konrad, William C. Karl, "Probabilistic video stabilization using Kalman filtering and mosaicking," in *IS&T/SPIE Symposium on Electronic Imaging, Image and Video Communications and Proc*, 2003.

[14] R. Szeliski, Computer Vision: Algorithms and Applications, Springer, October 19, 2010.

[15] M. Irani and P. Anandan, "All about direct methods," The Weizmann Institute of Science, Israel and Microsoft Research of Redmond, 1999.

[16] Fischler M. A., and Bolles R. C, "Random Sample Consensus: A Paradigm for Model Fitting with Apphcatlons to Image Analysis and Automated Cartography," *Communications of the ACM,* vol. 24, pp. 381-395, 1981.

[17] Zuliani M, RANSAC for Dummies, 2008.

[18] Torr P.H.S., and Zisserman A, "MLESAC: A New Robust Estimator with

Application to Estimating Image Geometry," *Computer Vision and Image Understanding,* vol. 78, no. 1, p. 138–156, 2000.

[19] H. P. Moravec, "Towards Automatic Visual Obstacle Avoidance," in *Proc. 5th International Joint Conference on Artificial Intelligence*, 1977.

[20] Harris, C., Stephens, M., "A combined corner and edge detector," in *Proceeding of 4th Alvey Vision Conference*, 1988.

[21] J. Noble, "Finding corners," *Image Vision Computing,* vol. 6, no. 2, pp. 267-274, 1988.

[22] Smith S. M., and Brady J. M, "SUSAN − a new approach to low level image processing," *International Journal of Computer Vision,* vol. 23, no. 1, p. 45 − 78, May 1997.

[23] Liu JJ., Jakas A., Al-Obaidi A., and Liu Y., "A comparative study of different corner detection methods," in *IEEE International Symposium on Computational Intelligence in Robotics and Automation (CIRA)*, Daejeon, 2009.

[24] Shnayderman A., Gusev A., and Eskicioglu A. M., "An SVD-Based Grayscale Image Quality Measure for Local and Global Assessment," *IEEE Transactions on Image Processing,* vol. 15, no. 2, pp. 422-429, February 2006.

[25] C.Morimoto and R.Chellappa, "Evaluation of image stabilization algorithms," in *Proceedings of the 1998 IEEE International Conference on Acoustics, Speech and Signal Processing*, Seattle, WA, 1998.

[26] Puglisi G. , and Battiato S., "A Robust Image Alignment Algorithm for Video Stabilization Purposes," *IEEE Transactions on Circuits and Systems for Video Technology,* vol. 21, no. 10, pp. 1390 - 1400, 2011.

[27] Luo Q. , and Khoshgoftaar T.M., "An Empirical Study on Estimating Motions in

Video Stabilisation," in *IEEE International Conference on Information Reuse and Integration*, Las Vegas, IL, 2007.

[28] Shapiro L. S., Affine Analysis of Image Sequences, Cambridge, UK: Cambridge University Press, 1995.

[29] Y. Matsushita, E. Ofek, W. Ge, X. Tang, and H.-Y. Shum, "Full-frame video stabilization with motion inpainting," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 28, no. 7, pp. 1-14, 2006.

[30] F. Liu, M. Gleicher, H. Jin, and A. Agarwala, "Content-preserving warps for 3d video stabilization," in *ACM Transactions on Graphics. (Proc. of SIGGRAPH),*, 2009.

[31] Tordoff B., and Murray D. W, "Guided Sampling and Consensus for Motion Estimation," in *7th European Conference on Computer Vision*, Copenhagen, Denmark, 2002.

[32] Wang Z., Bovik A. C., Sheikh H. R., and Simoncelli E. P, "Image Quality Assessment: From Error Visibility toStructural Similarity," *IEEE Transactions on Image Processing,* vol. 13, no. 4, pp. 600 - 612, April 2004.

[33] Censi A., Fusiello A., and Roberto A, "Image stabilisation by features tracking," in *International Conference on Image Analysis and Processing, 1999. Proceedings.*, Venice, 1999.

[34] S. H.W, "Least-squares estimation: from Gauss to Kalman," *Spectrum, IEEE,* vol. 7, no. 7, pp. 63 - 68, July 1970.

[35] Lee K. Y., Chuang Y. Y., Chen B. Y., and Ouhyoung M, "Video Stabilization using Robust Feature Trajectories," in *IEEE 12th International Conference on Computer Vision, 2009*, Kyoto, 2009.

[36] Abdullah L.M., Tahir N. M., and Samad M, "Video Stabilization Based on Point Feature Matching Technique," in *Control and System Graduate Research Colloquium (ICSGRC), 2012 IEEE*, Shah Alam, Selangor, 2012.

[37] Liu F., Gleicher M., Wang J., Jiin H., and Agarwala A, "Subspace Video Stabilization," *ACM Transactions on Graphics,* vol. 30, no. 1, pp. 1-10, 2011.

[38] Choi S., Kim T., Yu W, "Robust video stabilization to outlier motion using adaptive RANSAC," in *International Conference on Intelligent Robots and Systems*, St. Louis, MO, 2009.

[39] Huber P.J, Robust Statistics, New York: John Wiley and Sons, 1985.

[40] Rousseeuw P.J., and A. M. Leroy, Robust Regression and Outlier Detection, New York: John Wiley & Sons, 1987.

[41] Torr P.H.S., Zisserman A., and Maybank S.J, "Robust detection of degenerate configurations for the fundamental matrix," in *Fifth International Conference on Computer Vision*, Cambridge, MA, 1995.

[42] Ben-Ezra M., Peleg S., and Werman M, "Real-time motion analysis with linear-programming," in *The Proceedings of the Seventh IEEE International Conference on Computer Vision*, Kerkyra, 1999.

[43] Awrangjeb M., Lu G., and Fraser C.S, "Performance Comparisons of Contour-Based Corner Detectors," *IEEE Transactions on Image Processing,,* vol. 21, no. 9, pp. 4167 - 4179, 2012.

[44] Chikkerur Sh., Sundaram V., Reisslein M., and Karam L.J, "Objective Video Quality Assessment Methods: A Classification, Review, and Performance Comparison," *IEEE Transactions on Broadcasting ,* vol. 57, no. 2, pp. 165-182, 2011.

[45] Hartley R., and Zisserman A, Multiple View Geometry in Computer Vision, New York: Cambridge University Press, 2000.

[46] Shnayderman A., Gusev A., and Eskicioglu A, "Multidimensional image quality measure using singular value decomposition," *Proceedings of SPIE image quality and system performance,* vol. 5294, no. 1, pp. 82-92, 2003.

[47] Wang Z., and Bovik A. C, "Mean Squared Error: Love It or Leave It," *IEEE Signal Processing Magazine,* vol. 26, no. 1, pp. 98-117, January 2009.