VIDEO STABILIZATION: DIGITAL AND MECHANICAL APPROACHES

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

BY

SERHAT BAYRAK

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

NOVEMBER 2008

Approval of the thesis:

VIDEO STABILIZATION: DIGITAL AND MECHANICAL APPROACHES

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

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

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

Assist. Prof. Dr. İlkay Ulusoy _____
Supervisor, **Electrical and Electronics Engineering**

Prof. Dr. Uğur Halıcı _____
Co-Supervisor, **Electrical and Electronics Engineering**

Examining Committee Members:

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

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

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

Prof. Dr. Kemal Leblebicioğlu _____
Electrical and Electronics Engineering Dept., METU

Emre Turgay, MSc. in ECE _____
MGEO, ASELSAN

Date : 26.11.2008

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

Name, Last name : Serhat BAYRAK

Signature :

ABSTRACT

VIDEO STABILIZATION: DIGITAL AND MECHANICAL APPROACHES

Bayrak, Serhat

M.Sc., Department of Electrical and Electronics Engineering

Supervisor : Assist. Prof. Dr. İlkay Ulusoy

Co-Supervisor : Prof. Dr. Uğur Halıcı

November 2008, 87 pages

General video stabilization techniques which are digital, mechanical and optical are discussed. Under the concept of video stabilization, various digital motion estimation and motion correction algorithms are implemented. For motion estimation, in addition to digital approach, a mechanical approach is implemented also. Then all implemented motion estimation and motion correction algorithms are compared with respect to their computational times and accuracies over various videos. For small amount of jitter, digital motion estimation performs well in real time. But for big amount of motion, digital motion estimation takes very long time so for these cases mechanical motion estimation is preferred due to its speed in estimation although digital motion estimation performs better. Thus, when mechanical motion estimation is used first and then this estimate is used as the initial estimate for digital motion estimation, the same accuracy as digital estimation is obtained in approximately the same time as mechanical estimation. For motion correction Kalman and Fuzzy filtering perform better than lowpass and moving average filtering.

Keywords : video stabilization, image registration.

ÖZ

GÖRÜNTÜ SABİTLEME: SAYISAL VE MEKANİK YAKLAŞIMLAR

Bayrak, Serhat

Yüksek Lisans, Elektrik ve Elektronik Mühendisliği Bölümü

Tez Yöneticisi : Yrd. Doç. Dr. İlkay Ulusoy

Ortak Tez Yöneticisi : Prof. Dr. Uğur Halıcı

Kasım 2008, 87 sayfa

Genel görüntü sabitleme yöntemleri olan sayısal, mekanik ve optik sabitleme incelenmiştir. Görüntü sabitleme kavramı altında değişik sayısal hareket kestirme ve hareket düzeltme yöntemleri uygulanmıştır. Hareket kestiriminde sayısal yaklaşıma ek olarak mekanik bir yaklaşım da uygulanmıştır. Küçük miktardaki görüntü bozukluklarında, sayısal hereket kestirimi gerçek zamanlı olarak iyi performans vermektedir. Fakat, büyük miktardaki görüntü bozukluklarında, sayısal hareket kestirimi daha iyi sonuç vermesine rağmen çok zaman aldığı için hız nedeniyle mekanik hareket kestirimi tercih edilir. Mekanik hareket kestirimi ilk olarak kullanılıp daha sonra buradan elde edilen kestirim sayısal hareket kestirimi için ilk tahmin olarak alınırsa, sayısal kestirimle aynı hassaslık mekanik kestirimle yaklaşık aynı zaman içerisinde elde edilir. Hareket düzeltmesi açısından, Kalman ve Fuzzy süzgeçleri Lowpass ve Moving Average süzgeçlerinden daha iyi performans verirler.

Anahtar Kelimeler : görüntü sabitleme, imge düzeltme.

To My Beloved Family…

ACKNOWLEDGEMENTS

I would like to express my appreciation to my supervisor Assist. Prof. Dr. İlkay ULUSOY for her wisdom and guidance throughout this study and I would like to thank my family for their love, encouragement and full support.

TABLE OF CONTENTS

LIST OF TABLES

TABLES

LIST OF FIGURES

FIGURES

CHAPTER 1

INTRODUCTION

1.1. MOTIVATION

Video enhancement techniques such as superresolution, surveillence, compression and other high level operations like tracking, mosaicing, etc., have become very important with the increasing usage of digital visual media. Like the others, video stabilization, sometimes called as image sequence stabilization, can also be considered as one of the most important video enhancement techniques. Video stabilization is a process which aims to remove annoying shaky motions called jitter from the video sequence and to increase the spatial resolution of the video. If we think a video sequence with two frames, video stabilization can be thought as if image registration which is a fundamental task for most of the video enhancement and image processing operations.

In real life, there are lots of application areas in which video stabilization is used. Handy cams are one of the most popular and well known application areas. Furthermore, video stabilization is used in a mobile video transmission to reduce the bitrate of the video [18], or is used in aerial imagery to detect moving objects in a video [24], etc. Mainly, any application having a video captured from an unstabilized platform can be thought as the interest area of the video stabilization. Besides, stabilization is expected to work in different conditions such as over the real time videos or over the videos which contain high jitter. Considering the application, a

suitable stabilization method, algorithm and parameters are selected to obtain a better performance.

Let's think a person taking a video with a handy cam while walking. Since it is difficult to hold the camera as stabilized, captured video is exposed to some distortions. On the other hand, there is not only distortion but also there are desired motions on the video such as panning or zooming. The important point here is to prevent distortions which have generally high frequency characteristics while preserving the desired motions which have generally smooth and low frequency characteristics. Video stabilization removes these undesired motions while preserving the desired motions. High frequency distortions may also cause blurring on the video. If the sensors of the camera have not enough speed to capture the scene completely within a shutter time, corresponding frame cannot carry the correct information about the scene and then blur occurs on the frame. Both blurring and high frequency variations over the frames cause degradation in video quality.

The above problem can be generalized for all platforms on which a camera is mounted. Since there is no ideal frictionless environment, vibrations and shocks always occur on the camera while platform is moving. Even if there is no movement, camera or platform may be exposed to some distortions while standing because of the other effects such as wind. All of these effects cause degradation on the quality of the video.

Let's think a segmentation or an object detection application over the video which is captured from a mobile platform. If unstabilized video is used, the results of these applications will certainly fail. Therefore, video stabilization must be the first step if there is an application including a series of image/video processing operations. Consequently, video stabilization increases the quality of a video by removing the undesired motions. Actually, having a high quality video is useful for both human perception and other image processing operations which are implemented over the video.

1.2. SCOPE OF THE THESIS

In this study, it is aimed to propose a video stabilization solution and implement it as suitable for a mobile robot in which a series of image/video processing operations in addition to video stabilization will be performed in real-time. Since image/video processing operations are very time-consuming and exhaustive operations, it is very challenging to use them for real-time applications in which both performance and accuracy are required. Therefore, selection of the algorithms for video stabilization is a critical issue depending on the application.

Under the scope of this thesis, three video stabilization methods which are taken a part in the literature have been summarized [16, 31]. Considering the implementation area, which is mobile robot, two of three methods, called digital and mechanical video stabilization, are examined in more detail. For the digital video stabilization case, various motion estimation and motion correction algorithms are examined and compared with respect to their performances and accuracies. For the mechanical video stabilization, the original mechanical stabilization is not the case. Instead of this, a composite solution in which estimation is realized as mechanical stabilization and correction is realized as digital stabilization has been examined. This approach decreases the computation time in motion estimation which is the most time-consuming part in digital stabilization. In addition to decrease in computation time, it also decreases the cost in motion correction that is the most costly part in mechanical video stabilization. Furthermore, some experiments have been performed in which digital and mechanical motion estimation techniques have been used together to obtain a better performance.

1.3. OUTLINE OF THE THESIS

This study has been composed of five Chapters. The First Chapter consists of the introduction, problem statements and the goal of the research. Futhermore, scope and outline of the thesis have been given in this chapter. In the Second Chapter, video

stabilization methods which are taken a part in the literature have been summarized. In Chapter Three, details of the implemented video stabilization process have been given and all steps of video stabilization have been mentioned. Furthermore, this Chapter also covers theoretical background of the algorithms and techniques which have been handled in this study. All of the experiments conducted over the examined algorithms and evaluations of the algorithms have been given under the Chapter Four. Finally, Chapter Five contains the conclusion part about the study and possible future works in order to improve the performance and accuracy of video stabilization. References and appendixes are given at the end of the thesis.

CHAPTER 2

VIDEO STABILIZATION METHODS

Since video stabilization is a fundamental and one of the most commanly used image processing operations, different methods have been developed for different applications where stabilization is required. Mechanical video stabilization, optical video stabilization and digital video stabilization are currently available three methods taken a part in the literature.

Each of these methods has different motion estimation, motion correction and image correction parts. These parts can be considered as the main parts of a general video stabilization. Motion estimation is the process where global motions over the frames of the video are obtained. On the other hand, motion correction is the process where intentional motions are extracted from obtained global motions which are composed of intentional and unintentional motions. And, consequently, image correction is the process where stabilized video is produced using the estimated unintentional motions.

If video stabilization is required for an application, one of those methods with suitable motion estimation, motion correction and image correction techniques is chosen and implemented. But for some cases, it may be required to develop an application dependent solution using classical video stabilization methods. All methods have their own pros and cons and, therefore, a method should be chosen by evaluating all of the available methods which are suitable for the application. Details about these three methods are given in the following sections.

2.1. MECHANICAL VIDEO STABILIZATION

Mechanical stabilization is a kind of video stabilization method where stabilization is realized mechanically. That is, mechanical equipments are used to estimate and correct unintentional motions to obtain stabilized video.

In mechanical stabilization, motion is estimated by motion sensors. Depending on the application, type and number of used motion sensors may change. For example, if camera is exposed to motions only in the x direction, it is enough to use just one motion sensor to detect the motions in the x direction.

Since the distortions on the video are caused by undesired movements of the camera, the critical issue for motion estimation part is to find the movements of the camera accurately. In mechanical stabilization, camera movements are obtained by measuring the acceleration or velocity of the camera and manipulating a series of mathematical operations over these data. Acceleration and velocity are measured by accelerometers and gyros respectively which are the most commonly used inertial motion sensors for not only stabilization systems but also inertial navigation systems, automotives, etc.

Accelerometer is a kind of motion sensor which measure the linear acceleration in x, y and z directions. To obtain linear movements of the camera, acceleration data must be converted into displacement data. Acceleration data is translated into displacement data by double integration method using the following formula.

$$x(t_2) - x(t_1) = \int_{t_1}^{t_2}\int_{t_1}^{t_2} a(t)dt \tag{2.1}$$

But only double integration over the acceleration may not give the correct displacement. This is because of the imperfections of accelerometers. There may be a bias or a drift on the accelerometer outputs. Therefore, some preventive operations are needed to be applied in addition to double integration [30]. Accelerometers measure the acceleration only in one predetermined direction which is generally

indicated on the accelerometer. If it is located along x axis, acceleration along the x axis is obtained. Therefore, the number of accelerometer in a system depends on the number of required acceleration data in different directions. Figure below shows a kind of accelerometer.

Figure 2.1 : An example for accelerometer

As given above, another motion sensor commonly used for various applications is gyros. Gyro is a kind of sensor which measures the angular velocity in roll, pitch and yaw directions. Figure below shows a kind of gyro.

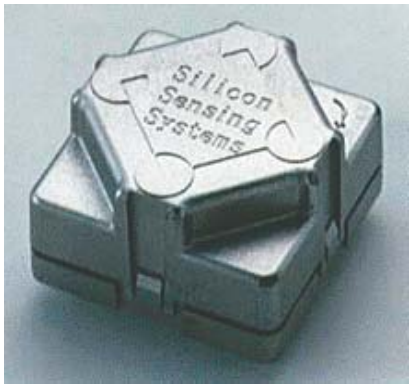

Figure 2.2: An example for gyro

Since velocity is the first derivative of displacement, displacement is obtained by taking one integration over the velocity using the following formula

$$x(\theta_2) - x(\theta_1) = \int_{\theta_1}^{\theta_2} \omega(\theta) d\theta \tag{2.2}$$

Like accelerometers, some undesired effects over the gyros such as drift and bias have to be considered for displacement calculations and some preventive operations have to be performed for gyro outputs also.

For mechanical video stabilization systems, generally gyros are used as the motion sensors and, therefore, rotational correction mechanism is generally realized. Actually the reason for the selection of gyros rather than accelerometers is that the movements of the camera in roll, pitch and yaw directions have much more effective on the video rather than the movements of the camera in x, y and z directions.

In mechanical video stabilization, it is aimed to keep the position of the camera stable with respect to its reference position. Therefore, all the estimated movements to which camera is exposed are taken as unintentional motions. Since stabilization is to remove only unintentional motions, there is no need to have motion correction part in mechanical stabilization which is different from digital video stabilization. Although there is no motion correction part, mechanical stabilization has very challenging image correction part. Generally different sensors like encoder data may be utilized in the control algorithm to increase the stabilization sensitivity as feedback.

Image correction is realized by a kind of mechanically controllable platform which contains motors and different mechanical components. Motors are controlled to obtain stabilized video using motion information coming from the motion sensors mounted on necessary points of the platform. In mechanical stabilization, frames of the video are manipulated before they come on to the video capturing plane (generally CCD or CMOS arrays) of the camera.

There are two kinds of stabilized platforms used for mechanical stabilization. Depending on the application, stabilized platform and camera may be mounted on the same mass or may be mounted on different masses. For the mounting on the

same mass type, since all mechanics are located altogether, complete system has to be stabilized to realize stabilization. Figure 2.3 below shows a military application of a system on which camera and platform are located on the same mass.

For the mounting on different masses type, since camera and stabilized platform are located on different bodies, there is no need to stabilize complete system. This kind of systems generally has a mirror as the stabilized platform which reflects the field of view on to the camera. Then a static camera captures the stabilized video. Figure 2.4 below shows a military application of a system on which camera and platform are located on different masses.

Although they have different mechanical structures, stabilization is performed with the same logic which is to give inverse movements to stabilized platform via motors by the amount of jitter.

As stated above, image correction is more challenging part for mechanical video stabilization systems. It requires designing a good controller algorithm for the motors which is very difficult job and needs exhaustive control theory. In addition to algorithm, image correction requires also a good mechanical structure.

Mechanical stabilization serves real time performance. It also serves the best stabilization accuracy among all the methods if reliable motion sensors are used and a good controller mechanism is developed.

Since there is no image operation performed over the captured video, no visual degradations such as black regions occurs on the video which is a considerable problem for digital stabilization. Furthermore, dynamic range of mechanical stabilization is much better than other stabilization methods. Dynamic range describes the amount of jitter that can be compensated by the system. For example, mechanical stabilization can compensate up to 50 pixels jitter, on the other hand, other stabilization methods can compensate up to 20 pixels jitter with the same performance criterion.

On the contrary, mechanical stabilization has considerable cost among the other methods. Therefore, it is generally used in military applications.

Figure 2.3 : Stabilized platform and camera are mounted on the same mass [30]

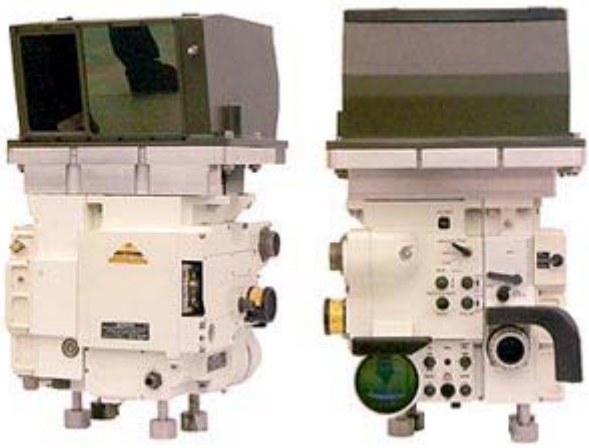

Figure 2.4 : Stabilized platform and camera are mounted on different masses [31]

2.2. OPTICAL VIDEO STABILIZATION

Optical stabilization can be thought as a kind of mechanical stabilization. The main difference between optical and mechanical stabilization is the image correction part which is realized by a kind of optical mechanism instead of a mechanical platform. In optical stabilization, it is aimed to remove the effects of relatively high frequency motions of the camera. Differentiation of relatively high frequency motions from the whole estimated motions is the task of motion correction part. Therefore, optical stabilization has a motion correction part which is different from mechanical stabilization.

In optical stabilization, motion sensors such as accelerometers or gyros are used to detect the motions of the camera. Since the dominant disturbances over the video are caused by the rotational movements, gyros are used as the motion sensor in optical stabilization systems. The details about gyros and manipulation of gyro data are given in mechanical stabilization section.

After motion estimation, motion correction and image correction parts are initiated respectively. First, the amount of unintentional motions among the whole estimated motions is determined then corresponding correction is given to the image correction system to overcome the disturbances. Image correction part of optical stabilization systems uses a group of floating lens to counteract the unwanted movements of the camera. This lens group is built in the system in such a way that each lens is able to shift itself through a plane perpendicular to the optical axis. In this structure, the focal length of the camera can be adjusted by moving related lens or lenses forward and backward. Changes in the focal length cause changes in the projection of the scene on to the capturing plane which is also perpendicular to the optical axis. Although camera is exposed to distortions from different directions, stabilization is performed to compensate the distortions in yaw and pitch directions. This is because the distortions in the directions different from yaw and pitch are negligible.

Because of their structure, floating lens group can not be given big amount of movements or high frequency movements. Therefore optical stabilization is effective to compensate low frequency disturbances caused by wind, hand movements etc.

Figure 2.8 below illustrates basically the operation of floating lens group in an optical stabilization system.

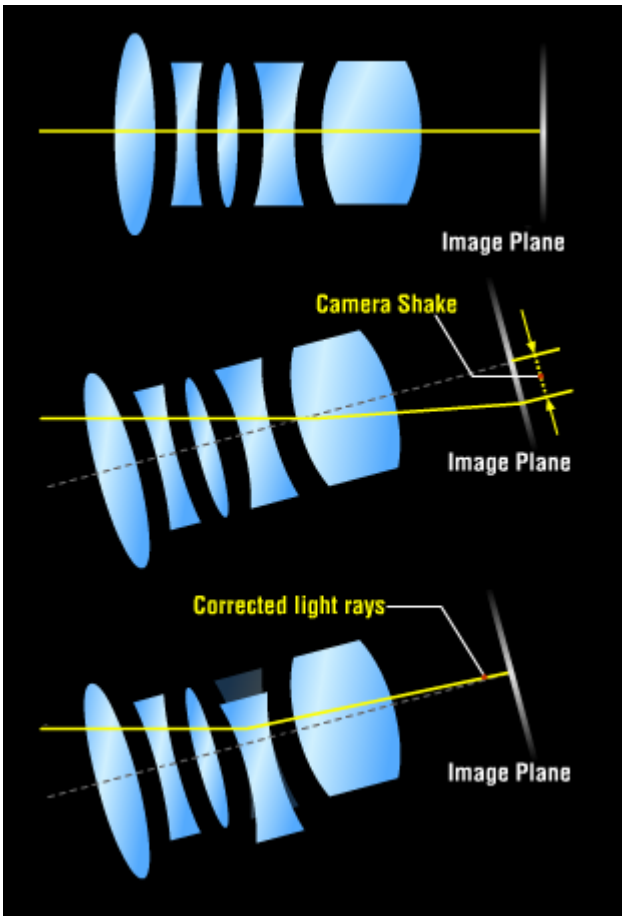

Figure 2.5: Floating lens group

There is another architecture used in optical stabilization systems. Instead of the floating lens group, floating CCD array may be used to realize image correction. The advantage of using floating CCD array rather than using floating lens group is to have lens independent stabilization.

Consequently, optical stabilization serves real time performance. Although it has worse accuracy and dynamic range with respect to mechanical stabilization, it can be preferred in civil applications such as for handy cams because of its reasonable cost and enough performance.

2.3. DIGITAL VIDEO STABILIZATION

Digital stabilization systems use completely electronic processing to control the image stability. That is, only software algorithms are used rather than hardware components such as motion sensors, actuators or floating lenses to compensate the disturbances. This makes digital stabilization more portable and cost effective among other methods.

In digital stabilization, interframe global motions are obtained by taking consecutive two frames of the video and performing a series of operations over the frames. Because of exhaustive image processing operations, motion estimation is the most time consuming and difficult part in digital stabilization. The output of motion esitmation is interframe global motions which contain not only unintentional motions but also intentional motions. After motion estimation, motion correction part differentiates intentional motions from unintentional motions. The last step is the alignment of the frames with respect to the estimated jitter. In this part, same amount of movements are given to the frames in the inverse direction with the jitter in order to obtain stabilized video sequence.

Digital stabilization causes some distortions over the stabilized video. Since interpolation is utilized to correct the frames, sharp edges and high frequency details of the frames are lost. Furthermore, movements cause also to loose some content of the frames. In addition to visual degradations, computation cost is another weakness of digital stabilization. But digital stabilization can be used for real time applications if the algorithms are optimized. On the other hand, while other stabilization methods can be used for real time applications only, digital stabilization can be used for both real time and off-line applications. This is an advantage of digital stabilization.

CHAPTER 3

VIDEO STABILIZATION

As mentioned in Chapter 2, there are three different techniques for video stabilization. But, digital stabilization is selected and implemented in this thesis because it has low cost and it is independent of hardware. Motion estimation, motion correction and image correction are the three main steps of digital stabilization as stated in the previous sections. These three steps can be thought as three independent steps and shown successively in the following figure.

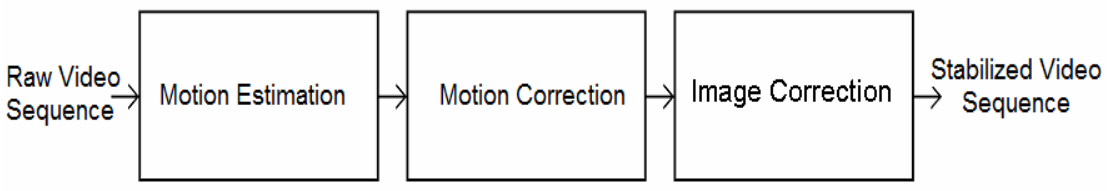

Figure 3.1 : Video Stabilization Process

There are various algorithms for each of these steps. But, since it is aimed in this thesis that implemented video stabilization must be able to work in real time with enough accuracy, algorithms are selected and optimized to overcome these requirements. Furthermore, in addition to algorithms, motion sensors are utilized in motion estimation which brings different approach to classical digital video stabilization.

Video stabilization aims to remove unwanted displacements of frames in a video sequence. Figure below shows basically the digital stabilization process.

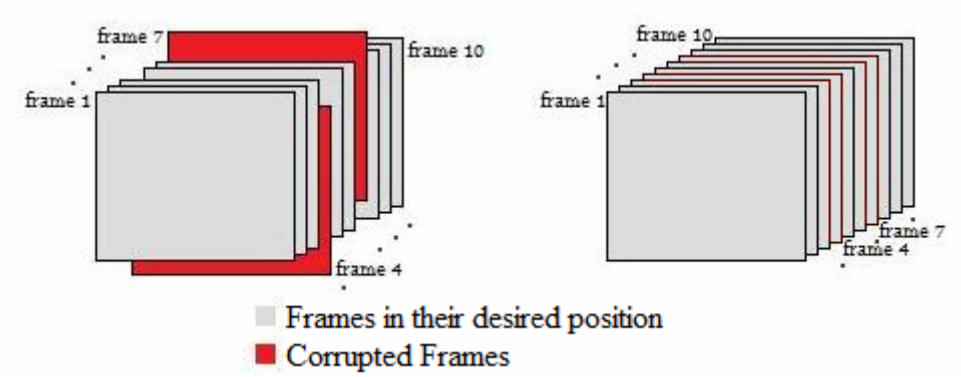

Figure 3.2: Sample frames of unstabilized (left) and stabilized (right)video

It is seen from the Figure 3.2 that frame 4 and frame 7 on left hand side have a global and unwanted displacement with respect to the complete image sequence before stabilization process is applied and sequence on the right hand side illustrates whole frames after stabilization is performed. Because of sudden scene differences occured on frame 4 and frame 7 (figure on the left hand side), some corruptions occur on the complete movements of the objects in the video. That is, static objects can be perceived as moving and places of objects can be perceived as different. As a result, quality of the video decreases. Stabilization corrects these corruptions and makes the whole video more meaningful.

The rest of this chapter examines all parts of digital stabilization individually and explains the theoretical backgrounds of all implemented algorithms.

3.1. MOTION ESTIMATION

For digital stabilization, motion estimation is the most time consuming part among all other processes. There are various types of digital motion estimation algorithms which have different types of theoretical backgrounds. But, all digital algorithms have considerable computational costs because of exhaustive image processing

operations. Therefore, in this thesis, a different approach is also examined and implemented for motion estimation in addition to suitable digital algorithms. This approach is to estimate interframe global motions via motion sensors like mechanical video stabilization. Following sections give details about both of digital and mechanical approaches in motion estimation.

3.1.1. DIGITAL APPROACH

There are various types of digital motion estimation algorithms in the literature each of which handles the estimation process differently. But, these algorithms can be mainly grouped with respect to their workspace as time (spatial) domain based motion estimation and frequency domain based motion estimation.

Frequency domain based motion estimation algorithms find the motions using phase information between the frames. Marcel [43], Vandewalle [44] and Block Based Phase Correlation [5] are the examples of frequency domain based motion estimation algorithms. On the other hand, time domain based motion estimation algorithms find the motions in spatial domain. They generally use local motions to obtain global motion which requires extra processsing load. Area Based Correlation [20], Lucas & Kanade [32], Feature Based Correlation [45], Keren [35] and Horn & Schunck [39] algorithms can be given as the examples of time domain based motion estimation algorithms.

All digital algorithms use successive two frames of the video while estimation. The critical point here is to process two frames within one frame duration if real time performance is the case. Therefore, all motion estimation algorithms have to be implemented to work fast enough or suitable algorithms have to be chosen which can be worked fast enough.

Under the concept of the thesis, Area Based Correlation, Lucas & Kanade and Block Based Correlation algorithms are examined and implemented because of their enough performances both in time and accuracy [38, 25].

3.1.1.1. AREA BASED CORRELATION ALGORITHM

Area based correlation algorithm, known as block matching algorithm, is the most common and simple motion estimation algorithm in terms of understanding and implementation. The idea behind block matching is to divide the images into macro blocks and then match each block in the reference image with a block in the current image. If we think a video sequence, reference image term is used for the previous frame or first coming frame and current image term is used for the next coming frame from the reference image. Motion estimation via block matching can be divided into two parts which are local motion estimation and global motion estimation. In this algorithm, first, a local motion vector is obtained for each block, and then, all local motion vectors are used to find a global motion between reference and current images. Since global motion estimation from local motions is a kind of matrix operation, local motion estimation is the most time consuming and dominant part of block matching algorithm.

Local Motion Estimation

First, reference image is divided into sub blocks. Figure 3.3 shows the reference image when it is divided into sub blocks.

Selection of sub block size is an important issue. It is effective on both computation time and accuracy of the algorithm. If sub block size is defined too small, the number of sub blocks in the reference image increases which results to increase the computation time. Furthermore, the possibility of finding wrong matches in the current image increases with small sub block size. On the other hand, if sub block size is defined too big, the number of sub blocks decreases which results to decrease in the computation time. In addition, big sub block size results also to have less number of local motion vectors. Having less number of local motion vector causes not to be able to determine the global motion accurately. After dividing the reference image into sub blocks, a correct match for each sub block is searched in the current image. Correct match, best correlated match, is not searched in the whole image. It is

searched in a predetermined region which is generally called search region in the current image. Figure 3.4 shows a search region for a randomly selected sub block.

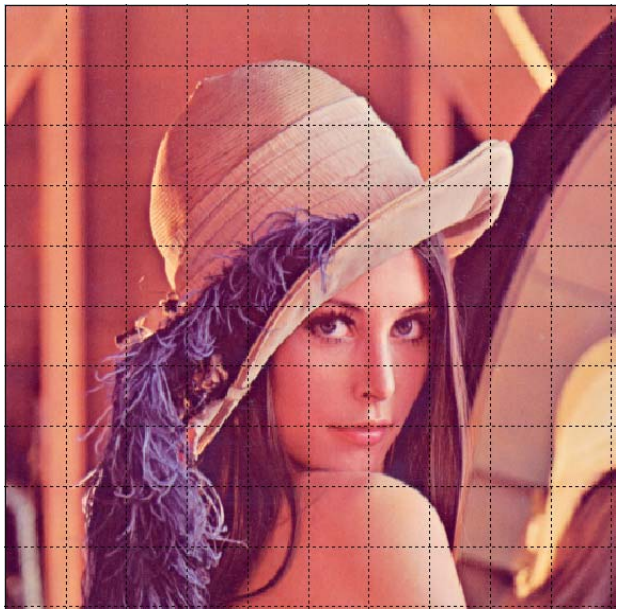

Figure 3.3 : Sub blocks in Reference Image

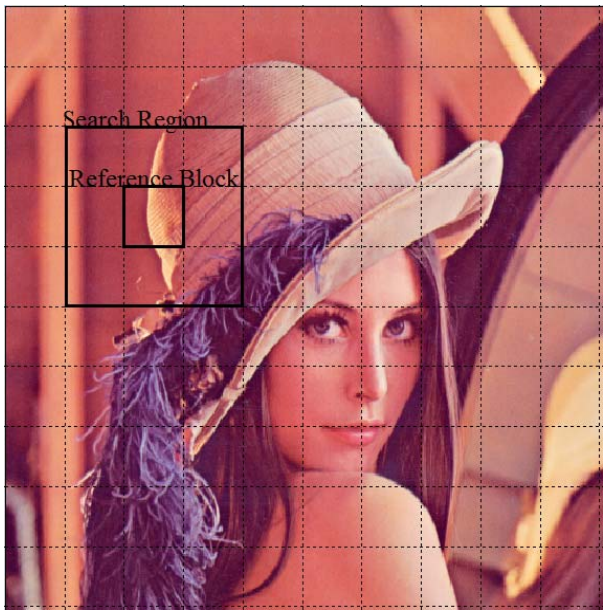

Figure 3.4 : Search region and sub blocks in Current Image

Like sub block size, the size of search region is also important. Search region size must be greater than the size of sub blocks and must be defined big enough to find the correct displacement vector between the images. If it is not defined big enough, the exact displacements of a sub blocks can not be covered so that the correct displacements can not be obtained. And, if search region is defined too big, computation cost of the algorithm and the possibility of finding more than one correlated sub block in the current image increases. Therefore the best way of defining search region size is to define it considering the motion characteristics of the video. There are different search strategies in the literature [47]. Three Step Search, Logarithmic Search, Four Step Search and Exhaustive Search are the most widely used search strategies for block matching algorithm. In this thesis Exhaustive Search is used. In Exhaustive Search, all possible locations where reference sub blocks can go are searched for the best match. That is, it is performed by looking all possible sub blocks which has the same size with reference sub block in the search region and which has just one pixel difference from the other blocks horizontally or vertically. Therefore, Exhaustive Search is the most accurate but computationally heavy algorithm among others.

In block matching algorithm, correlation is realized by using the intensity information of the sub blocks. There are various correlation criteria in the literature. In this thesis, Mean Absolute Difference (MAD) is used.

$$MAD = \frac{\sum_{x=0}^{N-1}\sum_{y=0}^{N-1}\left|E_{xy} - F_{xy}\right|}{N^2} \tag{3.1}$$

Here N is the size of sub blocks, E_{xy} and F_{xy} are the intensity values of a pixel at (x, y) location in the search region of current image and reference image respectively.

Following pseudo code shows the implementation of block matching algorithm for one block selected on the reference image.

BEGIN----------

Let blocksize is MxM and search size is NxN where N is M+2k+1, k is an integer

Let (r_x, r_y) be center point of the block at the reference image

Let (c_x, c_y) be center point of the block at the current image

Let (d_x, d_y) be displacement of the block between reference and current images

- Take a block from the reference image whose center is (r_x, r_y)

if standard deviation of reference block is greater than a predetermined threshold

- Set MAD value big enough to indicate that there is no initial correlation

for loop search in x direction, c_x = from $r_x - k$ to $r_x + k$

for loop search in y direction c_y = from $r_y - k$ to $r_y + k$

- Take a block from the current image whose center is (c_x, c_y)
- Calculate MAD value between reference and current blocks

if current MAD value is less than the previous

- Take (c_x, c_y) as the center point of the best correlated block
- Replace current MAD value with previous

end of for loop

end of for loop

- Calculate $(d_x, d_y) = (c_x, c_y) - (r_x, r_y)$

if minimum MAD value is less than a predetermined threshold

- Use (d_x, d_y) local displacements in the calculation of global displacement between reference and current image

else

- Do not use (d_x, d_y) local displacements in the calculation of global displacement between reference and current image

end of if

end of if

END----------

Above process is performed for all sub blocks. At the end a translational motion which is composed of a vector in the *x* direction and a vector in the *y* direction is found for all sub blocks in the reference image.

Global Motion Estimation

After obtaining a translational motion for each sub block, a global motion is estimated for the whole image. Global motion may be composed of translational motion, rotational motion, affine motion, projective motion etc. The property of global motion is determined by the stabilization requirements. In this thesis, global motion is accepted as the composition of translational and rotational motions which are the main contributors of degradation in the video quality. Global motion is found by using a transformation matrix. Transformation matrix uses the homograpy which corresponds to global displacement between reference and current images. Following figure shows the found local motion vectors.

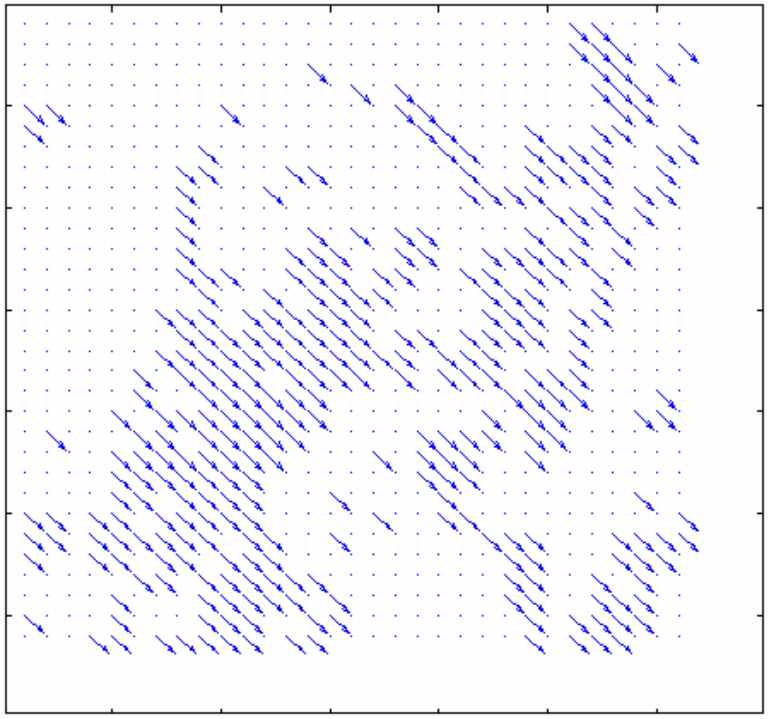

Figure 3.5 : Local motion vectors between Reference Image and Current Images

Consequently, transformation matrix produces a translational motion vector and a rotational motion vector. These motion vectors are the outputs of motion estimation part of digital video stabilization. Details of finding global motion from local motions are given in Appendix A.

For some conditions, block matching algorithm may fail. For example, if there is a region in the image having less texture, it is most likely to find wrong matches for sub blocks in that region. To prevent these wrong matches, standard deviation is used as a flag for each sub block. That is, if the standart deviation of the sub block is big enough which means that there is a high texture region, this flag is set to one and corresponding local motion is used in the calculation of global motion. Otherwise that local motion vector is not used in the calculation of global motion. There is another flag used to prevent the wrong matches. If found match does not exhibit high correlation value which means that there is no correct match, flag is set to zero and corresponding local motion is not used in the calculation of global motion. Usage of

correlation and standart deviation flags in the block matching algorithm increases the accuracy and also the computation time of the algorithm.

3.1.1.2. LUCAS AND KANADE ALGORITHM

Lucas and Kanade algorithm registers the images using optical flow information which is defined as the distribution of apparent velocities of movement of brightness patterns in an image [39]. Lucas and Kanade is one of the most widely used optical flow estimation algorithm because of its simplicity and performance.

Optical flow estimation relies on some assumptions and constraints [37, 39]. Therefore, Lucas and Kanade algorithm has also some assumptions and constraints [37]. Intensities of the pixels do not change over time is the first assumption. This assumption brings brightness constraint and is expressed by the following equation

$$\frac{dE}{dt} = 0 \tag{3.2}$$

where E is the intensity value of a particular pixel in the image and t is the time. Equation 3.2 is a fundamental equation for not only Lucas and Kanade algorithm but also all variants of optical flow estimation algorithms. If we indicate the location of the particular pixel in the image plane as (x, y), the intensity value E at time t can be expressed as $E(x, y, t)$, and, Equation 3.2 can be written in the following form

$$E(x, y, t) = E(x + \partial x, y + \partial y, t + \partial t) \tag{3.3}$$

where $E(x + \partial x, y + \partial y, t + \partial t)$ is the intensity value of the particular pixel at time $t + \partial t$. ∂x and ∂y are displacements of the particular pixel in x and y directions respectively. In Equation 3.3, if second term is expressed by its taylor series expansion, following equation is obtained.

$$E(x, y, t) = E(x, y, t) + \partial x \frac{\partial E}{\partial x} + \partial y \frac{\partial E}{\partial y} + \partial t \frac{\partial E}{\partial t} + \varepsilon \tag{3.4}$$

In the above equation, ε represents the second and higher order terms. If we neglect the second and higher order terms and divide the whole equations by ∂t, Equation 3.5 is obtained.

$$\frac{\partial E}{\partial x}\frac{dx}{dt}+\frac{\partial E}{\partial y}\frac{dy}{dt}+\frac{\partial E}{\partial t}=0 \tag{3.5}$$

In Equation 3.5, $\frac{dx}{dt}$ and $\frac{dy}{dt}$ represent the velocities of intensity of the particular pixel in x and y directions respectively. Let u and v represent the velocities in x and y directions and E_x, E_y, E_t represent the derivatives in x direction, y direction and time respectively, Equation 3.5 can be rewritten in the following forms.

$$E_x u + E_x v + E_t = 0 \tag{3.6}$$

or,

$$\begin{bmatrix} E_x & E_y \end{bmatrix}\begin{bmatrix} u \\ v \end{bmatrix} = -E_t \tag{3.7}$$

As seen from the final equations, 3.6 and 3.7, there are two unknown u and v, and only a single equation. It needs to have at least one more equation to find the velocity vectors u and v of the partical pixel. Therefore, Lucas and Kanade algorithm uses another assumption which accepts the velocity constant in a region in the small neighbourhood of the particular pixel. Using the last assumption, a region is defined on the image and velocity of all pixels in that region are taken as same. The number of equation depends on the number of pixels in the region. If the size of the region is mxm, m is an integer, Equation 3.7 can be rearranged in the following equation

$$\begin{bmatrix} E_{x1}E_{y1} \\ E_{x2}E_{y2} \\ ..... \\ E_{xm}E_{ym} \end{bmatrix}\begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} -E_{t1} \\ -E_{t2} \\ ..... \\ -E_{tm} \end{bmatrix} \tag{3.8}$$

or

$$E\,X = -T \tag{3.9}$$

where E_{x1}, E_{x2} … E_{xm} are the intensity derivatives of the pixels in the x direction, E_{y1}, E_{y2} … E_{ym} are the intensity derivatives of the pixels in the y direction, E_{t1}, E_{t2} … E_{tm} are the derivatives of the points with respect to time, u and v are the velocities of the block in x and y direction respectively. Since Equation 3.9 is an overdetermined system, it can be solved in the following form based on least square solution.

$$\left(E^T E\right) X = \left(E^T\right)\left(-T\right) \tag{3.10}$$

$$\left(E^T E\right)^{-1}\left(E^T E\right) X = \left(E^T E\right)^{-1}\left(E^T\right)\left(-T\right) \tag{3.11}$$

$$X = \left(E^T E\right)^{-1}\left(E^T\right)\left(-T\right) \tag{3.12}$$

If we expand Equation 3.12, Equation 3.13 is obtained

$$\begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} \sum_i E_{xi}^2 & \sum_i E_{xi}E_{yi} \\ \sum_i E_{xi}E_{yi} & \sum_i E_{yi}^2 \end{bmatrix}^{-1} \begin{bmatrix} \sum_i -E_{xi}E_{ti} \\ \sum_i -E_{yi}E_{ti} \end{bmatrix} \tag{3.13}$$

$$X = I^{-1}C \tag{3.14}$$

where $X = \begin{bmatrix} u \\ v \end{bmatrix}$, $I = \begin{bmatrix} \sum_i E_{xi}^2 & \sum_i E_{xi}E_{yi} \\ \sum_i E_{xi}E_{yi} & \sum_i E_{yi}^2 \end{bmatrix}$ and $C = \begin{bmatrix} \sum_i -E_{xi}E_{ti} \\ \sum_i -E_{yi}E_{ti} \end{bmatrix}$. Consequently, Lucas and Kanade algorithm divides the images into blocks and performs Equation 3.14 for each block. A translational motion vector is obtained for each block. Then, a global motion is obtained for the whole frame by manipulating the local motions like block matching algorithm. Appendix A contains the details of finding global motion from local motions.

Lucas and Kanade algorithm is based on the assumption that there is a small difference between the images. Therefore, algorithm fails for big differences. But the algorithm can be expanded to estimate big displacement differences using coarse to

fine approach. Coarse to fine approach builds an image pyramid whose level depends on the quantity of difference between the images. If there is a big difference, level must be chosen big enough to find the displacement accurately. Although coarse to fine approach increases the accuracy, it increases the computation time. Therefore, it is a critical issue to choose the level of the pyramid minimum while providing reasonable accuracy. Details of finding local motions and global motion are given in the rest of the section.

Local Motion Estimation

As stated above, coarse to fine approach is used in Lucas and Kanade algorithm. Since algorithm is performed over the blocks, coarse to fine approach is applied to all blocks. Size and number of the blocks are another important performance criterion for the algorithm in addition to the level of coarse to fine approach. If size is defined too big, constant velocity assumption in a small region starts to fail. On the other hand, number of the block has an inverse relation with on the computation time. But, increase in the number of block increases the accuracy in the estimation of global motion. In Lucas and Kanade algorithm, blocks can be determined with different ways. One is to obtain the blocks like block matching algorithm. Another way is to find some feature points on the image and use them as the centers of the blocks. Following figure shows an example of the application of 3-level coarse to fine approach on an MxN image.

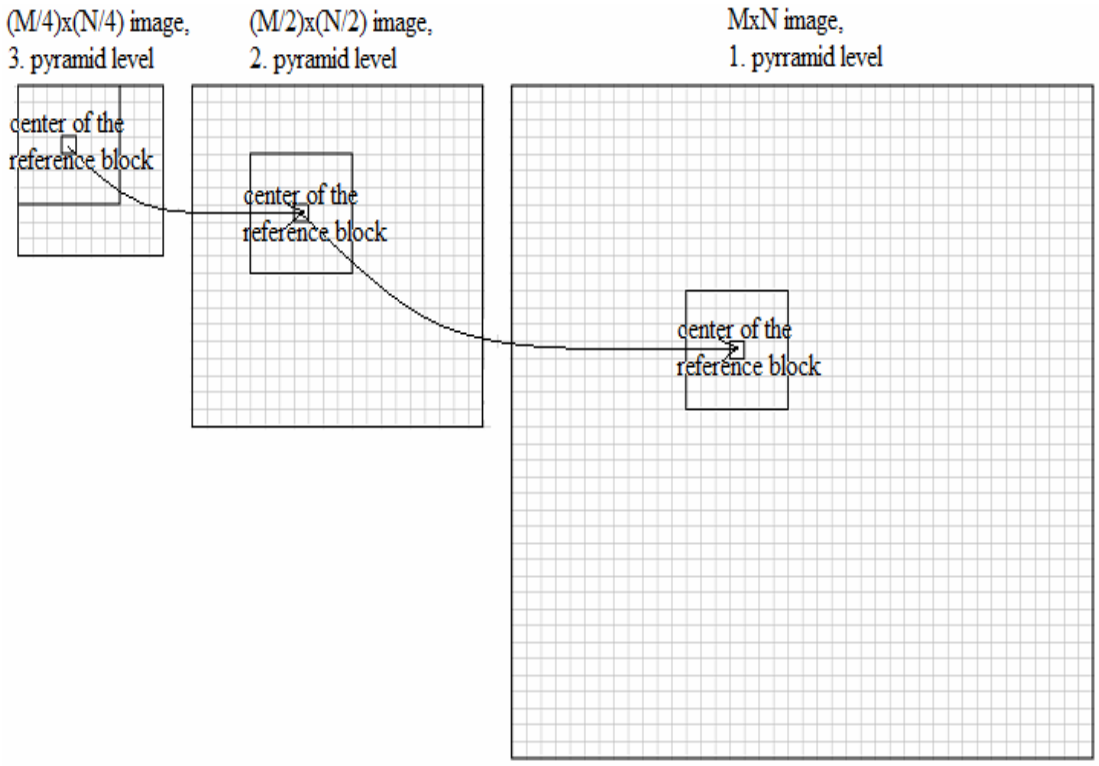

Figure 3.6 : Pyramidal structure of Lucas and Kanade algorithm

In the figure above, only some blocks are selected and shown to illustrate the locations of each block at all pyramid levels clearly.

In coarse to fine approach, first, optical flow is found for the lowest level (in above figure, the lowest level is 3. Level) and then it is used as an initial estimate for the upper level. This process is performed for all blocks up to the highest level (in above figure, the highest level is 1. Level).

Following pseudo code shows the implementation of Lucas and Kanade algorithm for one block selected on the reference image.

BEGIN----------

Let blocksize is MxM

Let (p_x, p_y) be center point of the block on the reference image at the lowest level

Let (r_x, r_y) be center point of the block on the reference image at the current level

Let (c_x, c_y) be center point of the block on the current image at the current level

Let (d_x, d_y) be displacement of the block between reference and current images

- Take the initial guess (d_x, d_y) for the displacement at Lowest Pyramid Level as (0, 0)

for loop Pyramid Level from Lowest Level to First Level

- Take the derivative of the reference image in x direction

$$E_x^{Pyramid\ Level} = \frac{\partial E^{Pyramid\ Level}}{\partial x}$$

- Take the derivative of the reference image in y direction

$$E_y^{Pyramid\ Level} = \frac{\partial E^{Pyramid\ Level}}{\partial y}$$

- Calculate the center point of the block at the current pyramid level

$$(r_x, r_y) = \frac{(p_x, p_y)}{2^{\text{Pyrmaid Level} - 1}}$$

- Take a block from $E_x^{Pyramid\ Level}$ and a block from $E_y^{Pyramid\ Level}$ whose centers are (r_x, r_y)
- Calculate I matrix in Equation 3.14
- Take a reference block from the reference image at the current Pyramid Level whose center is (r_x, r_y)

  $RB^{Pyrmaid\ Level}$

- Add the initial guess coming from the lowest pyramid level to the center location of the block at the current pyramid level

  $(d_x, d_y) = (2*d_x, 2*d_y)$

- Take the iteration guess (v_x, v_y) for the shifts at current level as (0, 0)

for loop iteration is from 1 to a predetermined value

- Calculate the center point of the current block at the current pyramid level

  $(c_x, c_y) = (r_x + d_x + v_x, r_y + d_y + v_y)$

- Take a reference block from the reference image at the current Pyramid Level whose center is (c_x, c_y)

  $CB^{Pyrmaid\ Level}$

- Take the derivative of the images with respect to time at the current Pyramid Level

  $E_t^{Pyramid\ Level} = RB^{Pyramid\ Level} - CB^{Pyramid\ Level}$

- Calculate C matrix in Equation 3.14
- Calculate X matrix in Equation 3.14 where X is (u_x, u_y)
- Refresh the iteration quess

$$(v_x, v_y) = (v_x + u_x, v_y + u_y)$$

end of for loop

- Refresh the estimation for the current Pyramid Level

$$(d_x, d_y) = (d_x + v_x, d_y + v_y)$$

end of for loop

END----------

Above process is performed for all sub blocks. At the end a translational motion which is composed of a vector in the *x* direction and a vector in the *y* direction is found for all blocks.

Global Motion Estimation

After obtaining a local motion for each block in the reference image, a global motion between reference and current images is computed. Global motion estimation in Lucas and Kanade algorithm is same as block matching algorithm. For the details, global motion estimation part of section 3.1.1.1 can be examined.

3.1.1.3. BLOCK BASED PHASE CORRELATION ALGORITHM

Block based phase correlation algorithm is blockwise implementation of classical frequency based motion estimation techniques. Classical frequency based motion estimation techniques use the principal that translational shifts in x and y directions cause only difference in frequency domain phase information of the image. In block

based phase correlation algorithm, this principal is used for each block. That is, first, local motions are found for each block like block matching and Lucas and Kanade algorithms and then a global motion is obtained by using all local motions. Therefore block based phase correlation algorithm can be examined under two sections which are local motion estimation and global motion estimation like the other methods.

Local Motion Estimation

In block based phase correlation algorithm, reference and current images are divided into sub blocks like block matching algorithm. As such in all blockwise motion estimation techniques, the size of the block is one of the most important performance and accuracy criterion. Blocksize must be chosen big enough to find the motion accurately. On the other hand, increase in block size is also effective on the computation time.

Let $E(m)$ represents a block in the reference image and $F(m)$ represents corresponding block in the current image. Assume that there is a translational shift Δm between $E(m)$ and $F(m)$.

$$F(m) = E(m + \Delta m) \tag{3.15}$$

where $m = \begin{bmatrix} x \\ y \end{bmatrix}, \Delta m = \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix}$, Δx represents the shift in x direction and Δy represents the shift in y direction. Let, the Fourier transforms of $E(m)$ and $F(m)$ are $FE(u)$ and $FF(u)$ respectively.

$$\begin{aligned} FE(u) &= \iint E(m) e^{-j2\pi u^T m} dm \\ FF(u) &= \iint F(m) e^{-j2\pi u^T m} dm \end{aligned} \tag{3.16}$$

By the definition, if we put $E(m + \Delta m)$ instead of $F(m)$ in the Fourier transform of $F(m)$ and proceed the equation, we obtain

$$
\begin{aligned}
FF(u) &= \iint F(m) e^{-j2\pi u^T m} dm \\
&= \iint E(m+\Delta m) e^{-j2\pi u^T m} dm \\
&= \iint E(m') e^{-j2\pi u^T (m'-\Delta m)} dm' \\
&= \iint E(m') e^{-j2\pi u^T m'} e^{j2\pi u^T \Delta m} dm' \\
&= e^{j2\pi u^T \Delta m} \iint E(m') e^{-j2\pi u^T m'} dm' \\
&= e^{j2\pi u^T \Delta m} FE(u)
\end{aligned}
\tag{3.17}
$$

It is seen from the above equation that shifted images or blocks have only phase difference in the frequency domain. We can obtain this phase difference component by calculating the cross spectrum of Fourier transforms of current and reference blocks.

$$
R(u) = \frac{FE(u)\, FF^*(u)}{\left| FE(u)\, FF^*(u) \right|} = e^{j2\pi\Delta m} \tag{3.18}
$$

where $R(u)$ represents the cross spectrum and $FF^*(u)$ represents the complex conjugate of fourier transform of the current block.

The shift between the blocks is the location of peak point in the inverse Fourier transform of the cross spectrum. Using the following formula, a translational shift is obtained for corresponding block.

$$
\begin{aligned}
\Delta m = &\arg\max_{(x,y)} (F^{-1}\{R(u)\}) \\
&\arg\max_{(x,y)} (F^{-1}\{e^{j2\pi\Delta m}\})
\end{aligned}
\tag{3.19}
$$

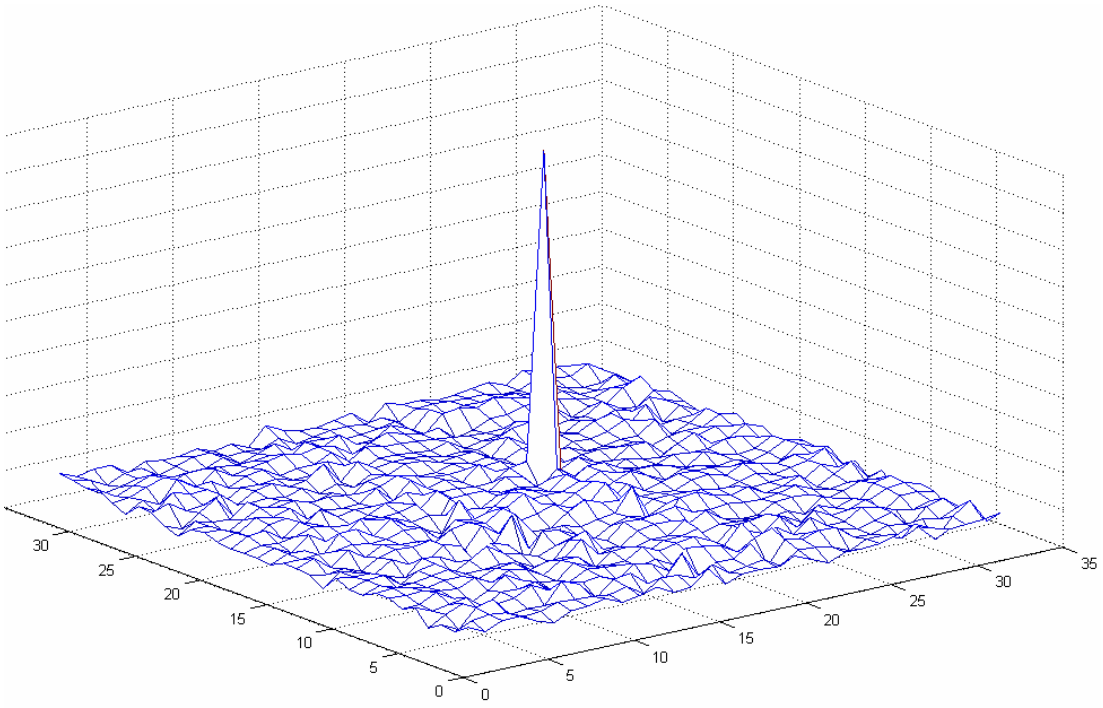

Figure 3.7 : Inverse fourier transform of cross spectrum of a block

Following pseudo code shows the implementation of block based phase correlation algorithm for one block selected on the reference image.

BEGIN----------

Let blocksize is MxM

Let (r_x, r_y) be center point of the block at the reference and current images

Let (d_x, d_y) be displacement of the block between reference and current images

Define a gaussian shape window whose size is MxM

- Take a block from the reference image whose center is (r_x, r_y)
- Take a block from the current image whose center is (r_x, r_y)

- Multiply window and reference block pixel by pixel, then take the resultant block as the reference block
- Multiply window and current block pixel by pixel, then take the resultant block as the current block
- Calculate fourier transform of the reference block
- Calculate fourier transform of the current block
- Calculate cross spectrum
- Calculate inverse fourier transform of cross spectrum
- Extract the location of peak point the difference from the origin of which gives (d_x, d_y) displacements for the block

END----------

Same process is performed for all sub blocks. At the end a translational motion which is composed of a vector in the *x* direction and a vector in the *y* direction is found for all blocks.

Global Motion Estimation

After obtaining a local motion for each block in the reference image, it needs to obtain a global motion between reference and current images. Global motion estimation in block based phase correlation algorithm is same as block matching and Lucas and Kanade algorithms. For the details, global motion estimation part of section 3.1.1.1 can be examined.

3.1.2. MECHANICAL APPROACH

As mentioned in Chapter 2, mechanical motion estimation is realized by a kind of inertial motion sensors such as accelerometers or gyros. Obtaining meaningful data from motion sensor (whatever used sensor is) is a very challenging issue. Because of the sensitivity and the characteristics of these sensors, it is difficult to obtain meaningful data by directly reading the output. Sensors generally tend to give a bias

and a drift term. Therefore some preprocessing operations are applied on the raw sensor data. The second challenge is to obtain meaningful data from the sensor synchronized with the frames of the video.

In this thesis, an IMU (Inertial Measurement Unit), Microstrain Inc. 3DM-GX1, is used as the motion sensor which serves as a complete solution for six axis motion analysis. Since it contains both accelerometers and gyros, all of angular velocities in roll, pitch, yaw directions in addition to linear accelerations in x, y, z directions can be measured. As mentioned in Section 2.1 and 2.2, distortions in roll, pitch and yaw directions are much more effective rather than distortions in x, y and z directions which are negligible. Therefore, stabilization of the camera generally covers the stabilization in roll, pitch and yaw directions.

3DM-GX1 IMU contains a good infrastructure to overcome all unwanted effects in obtaining of meaningful sensor data mentioned above. That is, it can be programmed to give compensated angular accelerations or to give compensated angular displacements etc. Since displacement is the main concern for global motion between the frames, IMU is programmed to output compensated angular displacements in yaw, pitch and roll directions. Programming details of 3DM-GX1 IMU are given in [41].

Estimating the motion mechanically and stabilizing the video digitally requires some conversions on the estimated motions. That is, although IMU produces angular motions in degree, stabilization evaluates the motions in pixel. Therefore, a relationship between angular measurements and pixels is needed to be found for translational corrections. That is, it must be found that how much angular rotation of camera corresponds to how many pixel shifts in the frame. This is a kind of calibration process in which calibration coefficients are determined and which is done one time at the beginning of stabilization. In the thesis, Microsoft LifeCam-VX-6000 camera is used with Microstrain 3DM-GX1 IMU. During the calibration, camera resolution is set to 288 x 352 pixels and IMU is programmed to give compensated angular displacement. Calibration is performed with the following steps;

Step 1: Camera and IMU are fixed on the same mass.

Step 2: By looking at the video on the camera, IMU is rotated in the YAW direction up to 50 pixel shift occurs on the video. When 50 pixel shift is reached, variation on the IMU in YAW direction gives the YAW calibration for 50 pixel shift.

Step 3: By looking at the video of the camera, IMU is rotated in the PITCH direction up to 50 pixel shift occurs on the video. When 50 pixel shift is reached, variation on the IMU in PITCH direction gives the PITCH calibration for 50 pixel shift.

Step 4: There is no need to make any calibration for ROLL direction. Because of the orientation of the camera and IMU, rotational displacements of the IMU directly correspond to rotational displacements of the video. Therefore, IMU output in ROLL direction which is in degree is used directly in the system.

After calibration process, following results are obtained. Calibration results tell us that 1 degree in YAW direction corresponds to 6.25 pixel shift in x direction, 1 degree in PITCH direction corresponds to 6.45 pixel shift in y direction and 1 degree in ROLL direction corresponds to 1 degree in rotational direction.

Table 3-1 : Camera Calibration Results.

Frame Pixels	IMU Angular Measure	Description
50.00 pixels shift	8.00 degree	In the YAW direction
50.00 pixels shift	7.75 degree	In the PITCH direction
8.00 degree	8.00 degree	In the ROLL direction

3.2. MOTION CORRECTION

The results of motion estimation part are the global motions between consecutive frames. Once motions are estimated, motion correction part distinguishes intentional

and unintentional motions between each other. Since intentional movements such as panning have to be kept within the video, frames are stabilized using only unintentional motions. Like motion estimation, there are various algorithms used for motion correction too. Kalman filtering [1 - 6], fuzzy filtering [7 - 12] and lowpass filtering [46] are the most popular and widely used algorithms. In this thesis, all of these algorithms are examined and implemented. In addition to these algorithms, because of its basic implementation and suitable structure, moving average filtering [17] is also implemented. Details of all implemented techniques are given in the following sections.

3.2.1. KALMAN FILTERING

Kalman filter is one of the most popular and widely used filter for different type of problems. The reputation of Kalman filter comes from its optimal solutions to various problems.

Kalman filter is mainly a set of mathematical equations that implement a predictor corrector type estimator to minimize the estimated error when some presumed conditions are met. The purpose is to estimate the exact states of a system from noisy measurements.

Kalman filter is essentially composed of two phases which are prediction phase and correction phase. Prediction phase contains time update equations and produces a priori estimate for the state of the system using a dynamic model which should be defined as close as possible to the ideal system. On the other hand, correction phase contains measurement update equations and estimates the exact states using a priori estimates and measurements. The following set of equations summarize the Kalman filter adaptation algorithm.

Let there is a discrete time system in the following form

$$x_t = A x_{t-1} + B u_t \tag{3.20}$$

where A and B are constant variables and t represents the time dependancy. Equation 3.20 is also known as state transition equation. Here it is seen that the present state x_t is dependent only to the x_{t-1} previous state and present input u_t.

If we consider that there is a process noise in the system, we can rewrite the state transition equation as follows;

$$x_t = A x_{t-1} + B u_t + w_t \tag{3.21}$$

where w is zero mean white process noise and uncorrelated with input x and u.

Lets assume that states x_t can not be measured directly. Instead, states z_t can be measured and there is a following relation between x_t and z_t which is known as observation equation.

$$z_t = H x_t + v_t \tag{3.22}$$

where v is zero mean white measurement noise uncorrelated with x, u and w, and H is a constant.

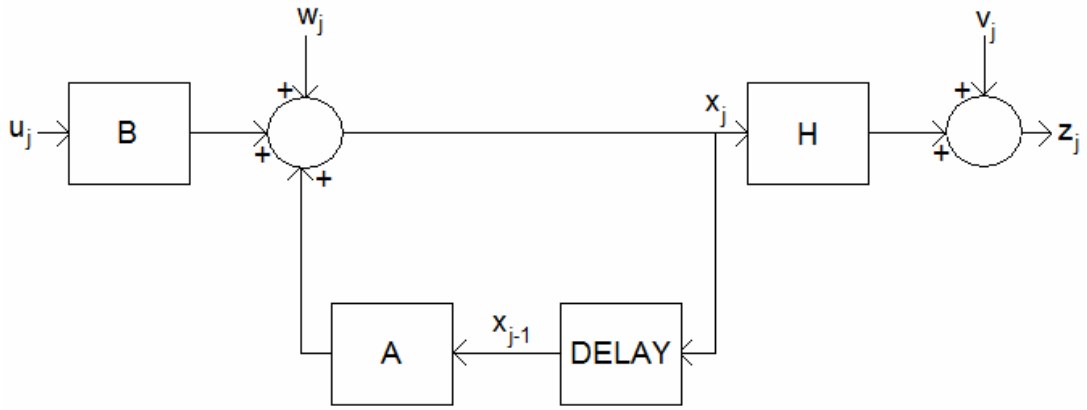

Figure 3.8: Discrete Time System

Figure 3.8 illustrates a discrete time system schematically. In such a system, since process noise w and measurement noise v are not known exactly, Kalman filter uses

the following equations instead of Equation 3.21 and 3.22 and examines the system without noise.

$$\hat{x}_t^- = A\hat{x}_{t-1}^- + Bu_t \tag{3.23}$$

$$\hat{z}_t = H\,\hat{x}_t^- \tag{3.24}$$

where $\hat{x}_t$ and $\hat{x}_t^-$ terms represent a posteriori and a priori estimates of x_t respectively. Since states and measurements are not the exact values, '^' is used to indicate that corresponding terms are just predictions.

Kalman filter defines the posteriori estimates which are the outputs of the filter by the following expression;

$$\hat{x}_t = \hat{x}_t^- + K_t(z_t - \hat{z}_t) \tag{3.25}$$

If we substitute $\hat{z}_t$ with $H\,\hat{x}_t^-$ in the Equation 3.25, we obtain the final equation;

$$\hat{x}_t = \hat{x}_t^- + K_t(z_t - H\,\hat{x}_t^-) \tag{3.26}$$

where K is the Kalman gain and calculated as follows;

$$K_t = P_t^- H^T (HP_t^- + R)^{-1} \tag{3.27}$$

In Equation 3.27, P_t represents the estimate error covarience and R represents the covarience of the measurement noise v. As it is seen from the equation that a priori estimate error covarience has to be recalculated at every time sample. This value is calculated as follows,

$$P_t^- = AP_{t-1}A^T + Q \tag{3.28}$$

where Q is the covarience of the process noise w and a posteriori estimate error covariance P_{t-1} is calculated using the following formula;

$$P_t = (I - HK_t)P_t^- \tag{3.29}$$

Consequently, Kalman filter uses Equation 3.23 and Equation 3.28 as time update equations and Equation 3.26, Equation 3.27 and Equation 3.29 as measurement update equations. Figure 3.9 below shows the final Kalman filter system.

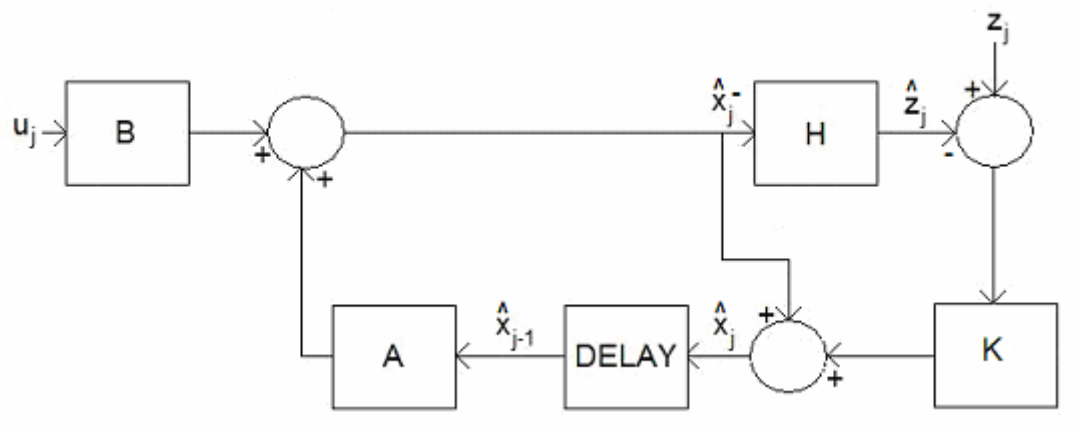

Figure 3.9: Kalman Estimator

Kalman Filtering in Video Stabilization

In video stabilization, Kalman filter is used to estimate global intentional movements of the camera from the estimated absolute frame positions.

Since Kalman filters produce intentional motions, jitter on the video is obtained by subtracting the output of Kalman filter from the estimated absolute frame positions. As a result, stabilization is performed with respect to these obtained jitter estimation.

Because of their structures, Kalman filters use a model and try to estimate the outputs with respect to the dynamics of that model. In video stabilization case, there are two models commonly used for motion correction in the literature. These models are constant acceleration and constant velocity models. But constant velocity model gives better performance with respect to constant acceleration model. Therefore, in this thesis, constant velocity model which assumes that the velocity of the camera is constant with respect to time is used for the Kalman filter. Since the accuracy of

Kalman filter depends on the model, to define the model as close as possible to the exact system improves the performance of the Kalman filter.

Constant velocity model accepts velocity and absolute frame position as inputs and produces an estimate for the exact absolute frame position refined from the jitter. Following 3.30 and 3.31 equations represent the state transition and observation equations of Kalman filter respectively for constant velocity model

$$\begin{bmatrix} x_t \\ m_t \end{bmatrix} = \begin{bmatrix} 1 & T \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x_{t-1} \\ m_{t-1} \end{bmatrix} + [w] \tag{3.30}$$

$$\begin{bmatrix} z_t \end{bmatrix} = \begin{bmatrix} 1 & 0 \end{bmatrix} \begin{bmatrix} x_t \\ m_t \end{bmatrix} + [v] \tag{3.31}$$

where x_t, m_t and z_t are estimated absolute frame position, velocity and measured absolute frame position of the frame respectively, $A = \begin{bmatrix} 1 & T \\ 0 & 1 \end{bmatrix}$, $H = \begin{bmatrix} 1 & 0 \end{bmatrix}$ and T is the time interval between successive frames in the video. Since video stabilization aims to remove the jitter in x and y directions, above equations are applied both of measured absolute frame positions in x and y directions individually.

The characteristics of Kalman filter can be adjusted by changing R and Q values. R value is the covariance of measurement noise. Higher R value means that standart deviation of the jitter is high. Or, if there is a jitter having high standart deviation, high R value is used in the Kalman filter to have better performance. On the other hand, Q value is the covariance of process noise. Since it is difficult to know the error on the process, determination of correct Q is more difficult than the determination of R. Consequently, higher R values result to have more smooth estimations, whereas, higher Q values results to have estimations which are closer to the noisy measurements. Before video stabilization is started, R and Q values have to be adjusted considering the model for the best estimation result.

Consequently, following sequence diagram shows the application of Kalman filter over the video stabilization.

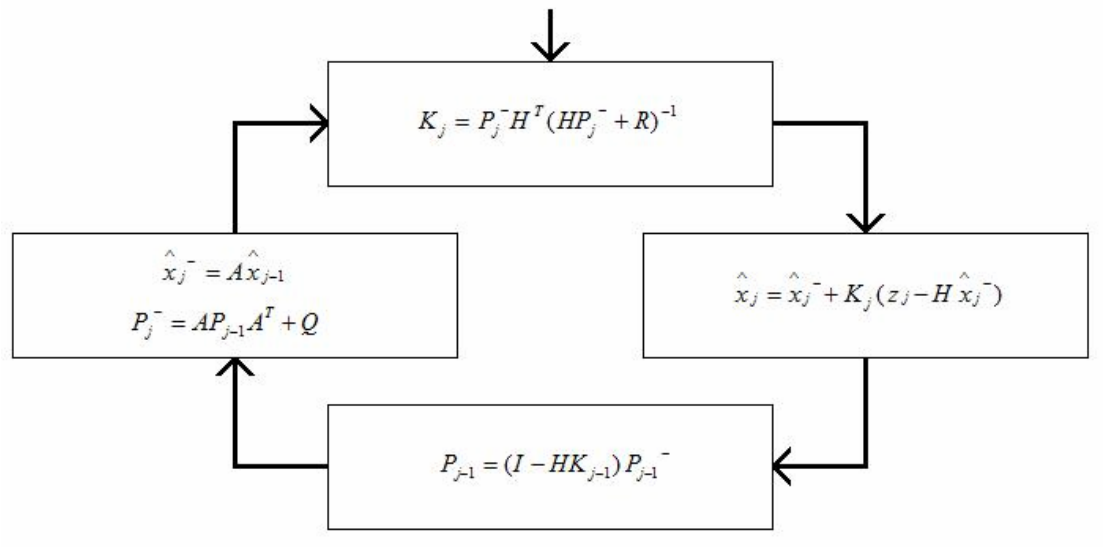

Figure 3.10: Kalman Filtering Sequence Diagram

3.2.2. FUZZY FILLTERING

Fuzzy filtering is a kind of filtering that uses fuzzy logic which is a problem solving control system methodology applicable to wide range of systems. Unlike classical logic which requires exact equations, precise numeric values or a deep understanding of a system, fuzzy logic incorporates an alternative way of thinking to model complex systems using a higher level of abstraction originating from our knowledge and experience. Fuzzy logic allows expressing this knowledge with subjective concepts such as very hot, bright red or a long time which are mapped into exact numeric ranges. Fuzzy logic concept contains the following three main phases; fuzzification, fuzzy engine and defuzzification. Following figure illustrates the fuzzy concept schematically.

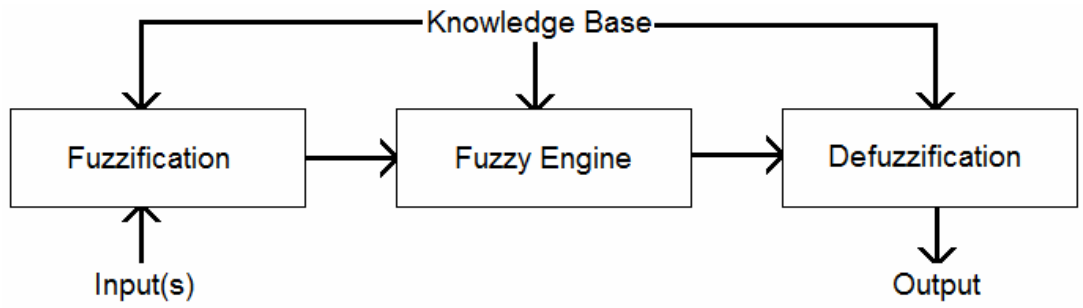

Figure 3.11: Fuzzy Correction System

Fuzzification

Fuzzification is a process where crispy input(s) are converted to fuzzy inputs. That is, a degree of membership is found for the input for all membership functions. In fuzzy logic, membership function is a function which maps a crispy input to a value between 0 and 1. Shape of membership function depends on the system. For example, triangular shape functions or gaussian shape functions can be used as membership functions. Figure 3.12 summarize the operation of fuzzification on a triangular shape membership function.

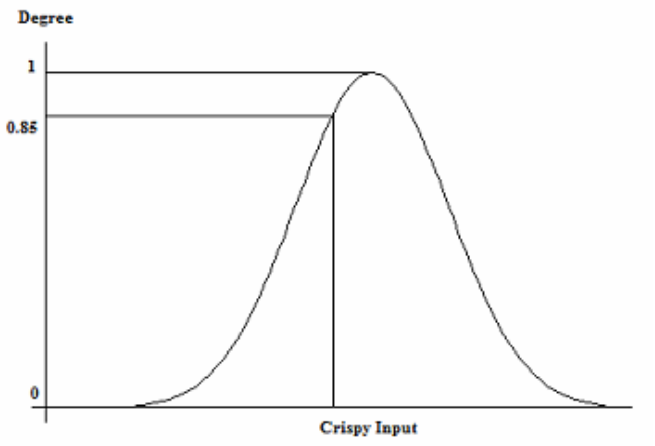

Figure 3.12: A Sample Membership Function

There can be more than one membership function in the system the number of which depends on the system complexity. If system is complex, it needs to have more than one membership function. In such a case, same crispy input is applied to all membership functions and a degree is obtained for each membership function. Figure 3.13 shows a system having multiple membership function.

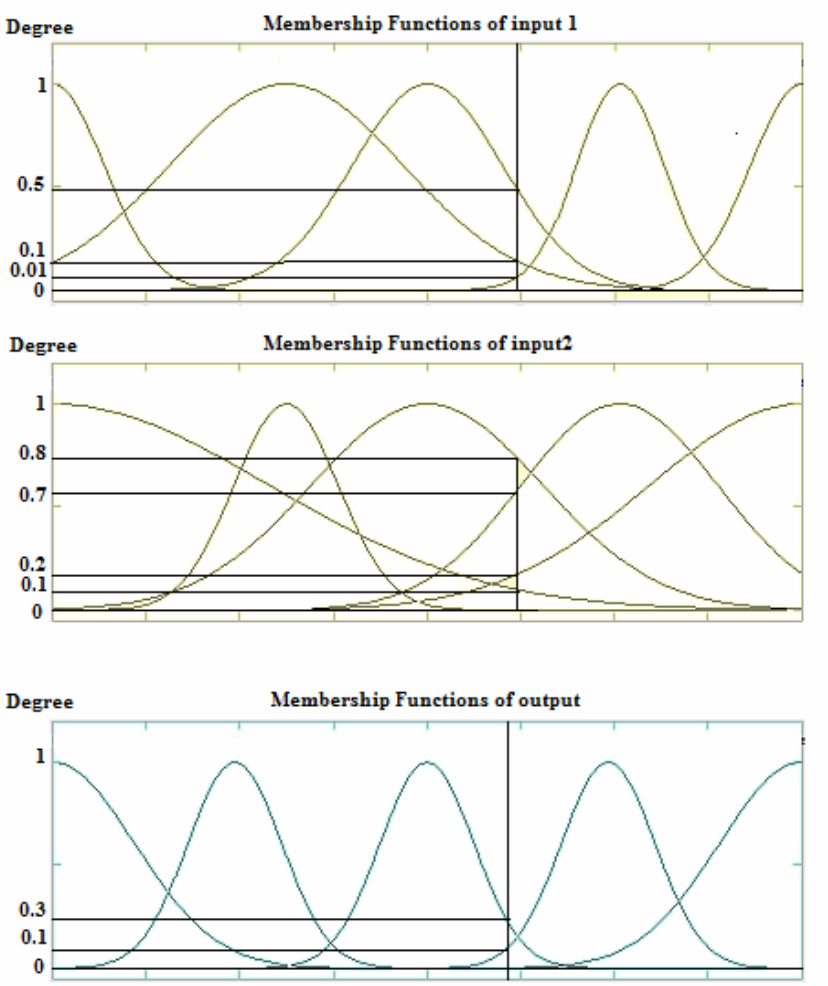

Figure 3.13: Membership Functions of inputs and output

Fuzzy Engine

Fuzzy engine takes all fuzzy inputs and obtains an output using some linguistic rules. Figure 3.14 illustrates the operation of fuzzy engine.

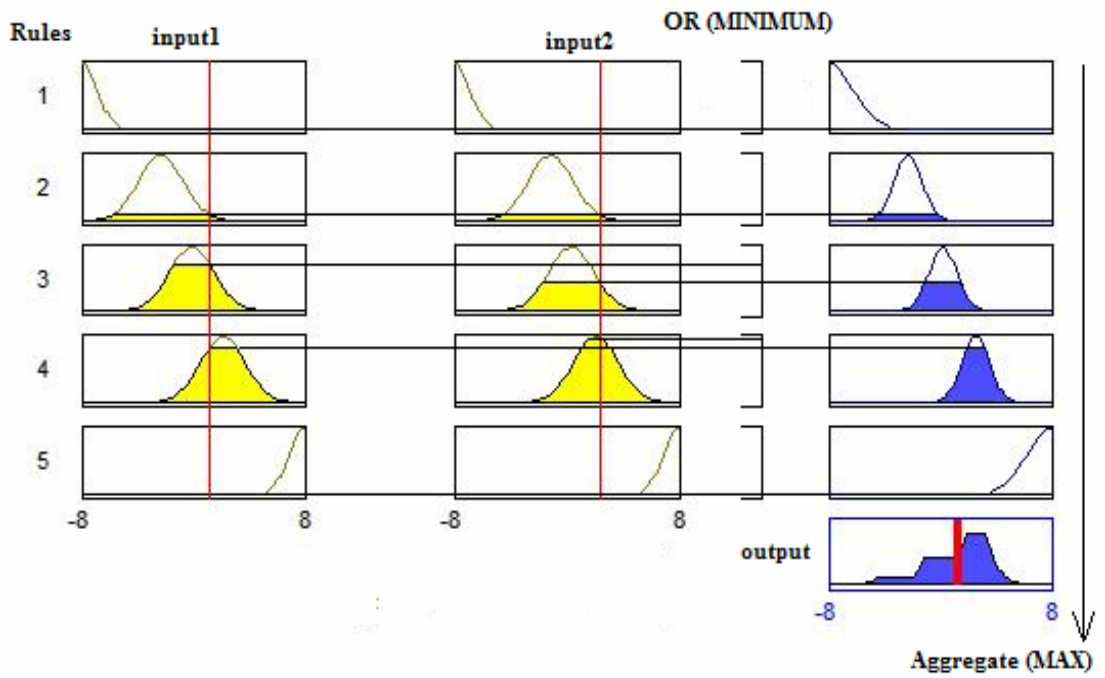

Figure 3.14: Fuzzy Engine

As it is seen from Figure 3.14, there are fuzzy operations and fuzzy rules in the fuzzy engine.

Fuzzy rules are a kind of "if-then" rules and completely dependent to the system model and directly affect the results. Therefore rules have to be determined very precisely. On the other hand, fuzzy operation is a kind of boolean logic which works a little bit different from classical boolean logic. Commanly used fuzzy operations are "AND", "OR" and "NOT". Fuzzy "AND" operation outputs the minimum of the inputs, "OR" operation outputs the maximum of the inputs and "NOT" operation outputs the ones complement of the inputs which is 1 - inputs.

Defuzzification

Defuzzification a kind of process which makes averaging and weighting the resulting outputs from all the individual rules into one single output decision or signal which

tells a controlled system what to do. The output is a precise appearing, defuzzified, "crisp" value. The defuzzification process is given in Figure 3.15.

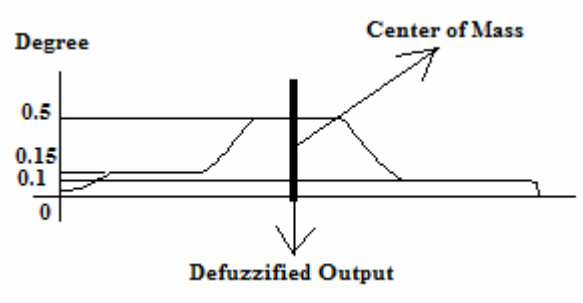

Figure 3.15: Defuzzification

Fuzzy Filtering in Video Stabilization

Lets define a discrete time system to understand fuzzy filtering approach;

$$x_t = f(x_{t-1}) + w_t \tag{3.32}$$

where f is a function that uses previous state to obtain the present state, w is zero mean white noise of the process and t represents the time dependency.

Assume that states x_t are not measured directly. Instead, we can measure states z_t and there is a following relation between x_t and z_t.

$$z_t = h(x_t) + v_t \tag{3.33}$$

where h is a function and v is zero mean white measurement noise.

To estimate states x_t with fuzzy filtering approach, another set of equation is defined using the recursive predictor-corrector architecture which is commanly used estimator structure.

$$\hat{x}_t = f(\hat{x}_{t-1}) + g(z_t, \hat{x}_{t-1}) \tag{3.34}$$

We can use above equation to estimate the jitter between video frames in a video sequence with fuzzy filtering approach.

If there is a video sequence taken from a platform moving with constant velocity, we can estimate the states (which correspond to the exact frame positions) using the following 3.35 and 3.36 equations;

$$\hat{x}_t^- = \hat{x}_{t-1} + T\hat{v}_{t-1} \tag{3.35}$$

$$\hat{x}_t = \hat{x}_t^- + g(z_t, \hat{x}_t^-) \tag{3.36}$$

where T is update period and v is rate of change estimation of frame motion speed;

$$\hat{v}_t = \frac{\hat{x}_t - \hat{x}_{t-1}}{T} \tag{3.37}$$

Function g in Equation 3.36 is correction function of the system and can be defined by fuzzy logic approach as mentioned in the fuzzy logic part of the text.

As mentioned in the fuzzy logic section, there are membership functions and fuzzy rules for fuzzy logic approach. The usage of fuzzy filtering on video stabilization depends on the determination of these membership functions and fuzzy rules.

Membership functions can be various types. A most suitable membership function has to be chosen for the system. Because of the simple adabtability and smooth variation, gaussian membership functions were used in the thesis. A gaussian membership function can be expressed as;

$$y = e^{-\frac{1}{2}\left(\frac{x-c}{\sigma}\right)^2} \tag{3.38}$$

where c is center point and σ is standart devation.

The selection of fuzzy rules is the other parameter. Rules have to be determined considering the system behaviour. System behaviour means the dynamic range of the

inputs. In our system, there are two parameters as inputs for fuzzy filter. These inputs are INPUT 1 and INPUT 2 which are formed using z_t and x_t as;

$$\begin{aligned}(INPUT\,1)_t &= z_t - \hat{x}_t^- \\ (INPUT\,2)_t &= (INPUT\,1)_t - (INPUT\,1)_{t-1}\end{aligned} \tag{3.39}$$

The following fuzzy rule table is used for the video stabilization.

Table 3-2: Fuzzy Rules

		INPUT 2				
		NB	N	Z	P	PB
INPUT 1	NB	NB	N	N	Z	Z
	N	N	N	Z	Z	P
	Z	N	Z	Z	Z	P
	P	N	Z	Z	P	P
	PB	Z	Z	P	P	PB

TERMS

NB : NEGATIVE BIG, N : NEGATIVE, Z : ZERO, P : POSITIVE, PB : POSITIVE BIG

In the table each term (NB, N, Z, P and PB) corresponds to a membership function. But terms in the INPUT 1 section correspond to the membership function of input 1, the terms in the INPUT 2 section correspond to the membership function of input 2 and the terms in the intersections of INPUT 1 and INPUT 2 correspond to the membership function of output.

3.2.3. LOWPASS FILTERING

Lowpass filter passes low frequencies while attenuating high frequencies with respect to a predetermined cut off frequency.

Lowpass Filtering in Video Stabilization

In video stabilization, intentional motions such as panning exhibit low frequency characteristics relative to unintentional motions. Therefore intentional motions can be extracted from the whole motion by lowpass filtering. Filter is applied to absolute frame position like Kalman and fuzzy filtering. The important point for lowpass filtering is to determine the cut off frequency. If cut off frequency is selected accurately considering the characteristics of the system, lowpass filter gives a reasonable performance on differentiation of the jitter. Ones intentional motion is extracted, it is subtracted from the estimated absolute frame positions to obtain jitter which is used for the image correction to obtain stabilized video.

For lowpass filters, filter length (filter order) is another parameter that affects the stabilization performance. Filter length determines the dependency of the input signal to the previous inputs. If filter length is defined too big, effect of current input on the current result decreses.

3.2.4. MOVING AVERAGE FILTERING

Moving average filter is the most commonly used filter in wide range of applications, mainly because it is the easiest digital filter to understand and use. In spite of its simplicity, moving average filter is an optimal filter to reduce random noise while retaining a sharp step response.

Moving average filter is a kind of filter that replaces each value in a series with the average of its neighbourhood. Following equation realizes moving average filter

$$x_t = \frac{x_{t-N} + \ldots + x_t + \ldots + x_{t+M}}{M + N + 1} \tag{3.40}$$

where x_t is the state at time t, N is the number of previous neigbouring states and M is the number of future neigbouring states. If moving average filter is wanted to be used for real time applications, since there is no future state information at current

state, M is set to zero. Moving average filter smooths the input and produces slowly varying outputs. The smoothness can be adjusted by the length of the filter. Increase in the length of the filter increases dependency to the previous states which increases the smoothness.

Moving Average Filtering in Video Stabilization

Moving average filter takes the estimated absolute frame positions as the inputs and smooths them to produce intentional motions. Then, jitter is obtained by subtracting the intentional motions from the estimated absolute frame positions. Following formula shows the calculation of intentional motions.

$$xx_t = \frac{x_{t-N} + x_{t-N+1} + ... + x_t}{N+1} \tag{3.41}$$

where x_t are the estimated absolute frame positions with respect to reference image, xx_t are the intentional motions of the camera and N is the filter length. Since video stabilization aims to remove the jitter in x and y directions, above equation is applied to both directions individually.

3.3. IMAGE CORRECTION

Image correction is the third and final step in video stabilization. Realization of image correction may change with respect to the video stabilization methods. That is, if mechanical or optical video stabilization is considered, image correction is realized by motors and a kind of mechanical structure. But, in digital video stabilization, image correction is realized only by software. Therefore, digital video stabilization is the most cost effective among all methods.

Digital image correction part takes the frames of the video and aligns them by shifting and rotating with respect to the output of motion correction part. Due to shifts and rotations, some unknown areas occur on the frames. This is an important

difference in image correction with respect to other video stabilization methods. In optical and mechanical video stabilizations, since there is no image processing operations, unknown regions do not occur on the frames.

In this thesis, only translational and rotational disturbances are considered as jitter. Therefore, only translational and rotational corrections are applied to over the images. The idea behind digital image correction is to give the inverse of estimated jitter to the frames.

Lets think two successive frames in a video sequence. If the output of motion correction is that there is +2º rotational misalignments in addition to translational misalignment having +4 pixels and -3 pixels misalignments in x and y directions respectively, -4 pixels and +3 pixels translational correction in x and y directions respectively and -2º rotational correction should be applied.

Rotational and translational corrections require interpolation over the frames. There are various interpolation techniques each of which has different accuracy and computational cost. Nearest neigbourhood, bilinear, and bicubic interpolation are most commonly used interpolation techniques in the literature. Even if bicubic interpolation has the best accuracy, it has considerable computational load. On the other hand, nearest neighnbourhood technique is the fastest algorithm. But it has not enough accuracy. Therefore, bilinear interpolation is used in this thesis since it has reasonable accuracy and enough computational time.

Consequently, after application of image correction to all frames in the video sequence, stabilized video is obtained.

CHAPTER 4

EXPERIMENTS AND RESULTS

The main concern of this study is to propose a solution for an optimum real time video stabilization for a camera on a mobile platform. As it has been mentioned in the previous chapter, video stabilization is composed of three main parts. Since image correction part is a kind of process in which images are aligned by shifts and rotations with respect to the output of stabilization process, it is much more related to the image quality analysis and it can be discarded for the performance evaluation of video stabilization. Depending on the computation time, image resolution or other requirements, different image correction algorithms can be used.

In this research, the performance of video stabilization has been evaluated with respect to the motion estimation and motion correction parts. Since these parts work sequentially, they can be considered as two different parts and thus they can be tested individually.

For the experiments, MATLAB environment has been utilized and all mentioned algorithms have been implemented in MATLAB (revision R2007a). In addition to MATLAB, Borland C++ Builder IDE (version 6.0) and OpenCV library (version 1.0) have been utilized to be able to capture the real test images and videos. Furthermore, Borland C++ Builder IDE (version 6.0) has been also utilized with a setup as it is

shown in Figure 4.1 below in order to obtain inertial motion information from the IMU for motion estimation in mechanical approach.

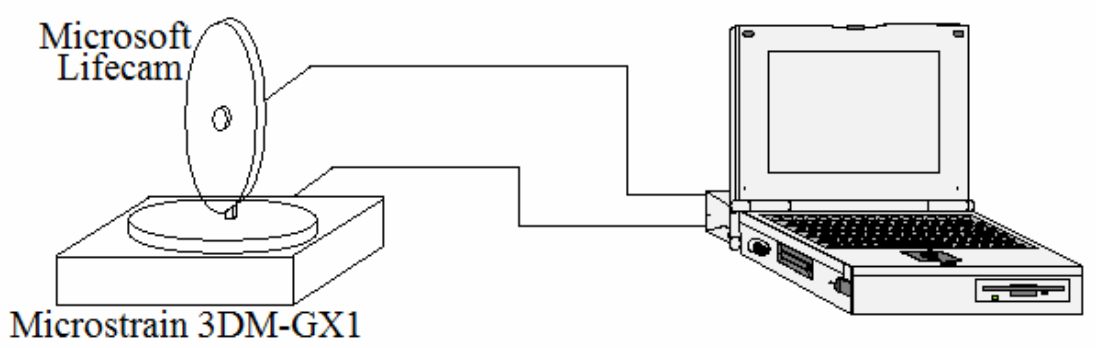

Figure 4.1 : Experiment Setup

4.1. MOTION ESTIMATION EXPERIMENTS

This section covers the experiments related to motion estimation both for digital and mechanical approaches. In the experiments, algorithms have been performed in order to estimate distortions in translational and rotational directions between each successive frames of the videos. Accuracy and complexity are taken as the comparison parameters for the motion estimation algorithms. 288 (rows) x 352 (columns) pixels resolution CIF formatted synthetic and real video sequences have been utilized in the experiments. For the real video sequences, new videos have been captured by the setup given in Figure 4.1 and examined. On the other hand, synthetic videos have been formed by putting the frames one after another whose frames are obtained by means of giving different amount and type of distortions to a high resolution image and taking the region of interest from high resolution image.

For the real video sequences, algorithm performances are evaluated with respect to mean square error (MSE) criteria and visual inspection. MSE of whole video can be found by calculating the mean square errors for each successive frame couple after stabilization is performed and then taking the average of all mean square errors.

$$MSE = \frac{1}{N_F - 1} \sum_{n=2}^{NF} \left[\frac{\sum_{x=1}^{M} \sum_{y=1}^{N} \left(E_{n,xy} - E_{n-1,xy} \right)^2}{MN} \right] \tag{4.1}$$

Here N_F represents the number of frames in the video, M represents the height of the frames in pixel, N represents the width of the frames in pixel and $E_{n,xy}$ represents the intensity value of n^{th} frame at (x, y) location.

In addition to MSE and visual inspection criterion, another indicator of quality measurement is used for synthetic videos. Since the distortions between any successive frames are known exactly in the synthetic videos, estimated values can be compared to the exact values by calculating the absolute estimation error in either direction. Absolute estimation error can be calculated for one direction, by the following formula;

$$X_{error} = \left\| X_{exact} - X_{estimated} \right\| \tag{4.2}$$

where X_{error} is the absolute estimation error, an X_{exact} is the exact distortion and $X_{estimated}$ is the estimated motion between each successive frames in x direction. Absolute estimation errors in other directions (y and rotational directions) can also be calculated using the same formula. However, in order to evaluate the performance over the whole video, average of the absolute estimation errors can be used;

$$X_{avg,error} = \frac{1}{NF - 1} \sum_{n=2}^{NF} X_{t,error} \tag{4.3}$$

Here $X_{t,error}$ is the absolute estimation error between t^{th} and $(t-1)^{\text{th}}$ frames, $X_{avg,error}$ is the average of absolute estimation errors in the x direction, and N_F is the number of frames in the video sequence. Average of the absolute estimation errors in other directions (y and rotational directions) can also be calculated using the same formula. In addition to accuracy measurement, complexity analyses in

motion estimation have been also given in this section to evaluate the computation time of each algorithm. At the end of the motion estimation experiments, a general evaluation of the experiments and commenting about the performances of digital and mechanical motion estimation algorithms have been given.

Each of the motion estimation algorithms has different parameters. Depending on these parameters, accuracy and computation time of the algorithms change drastically. Consequently, before conducting the experiments, parameters of each algorithm are set to predetermined values which were determined experimently considering computational load and accuracy. Thus, this preliminary operation makes the algorithms comparable.

4.1.1. SYNTHETIC VIDEO EXPERIMENT

As it has been stated above, frames of the synthetic video have been obtained by means of giving known translational and rotational distortions to a reference image. Since the video in this experiment has been synthetically formed, there is no interframe motion information obtained from the IMU. Therefore, synthetic video experiment is used to compare only the digital motion estimation algorithms among the each other. In this experiment, randomly generated motions having ±2 pixels maximum values have been applied to a high resolution image as translational distortion in x and y directions. In addition to translational distortion, randomly generated rotational motions having ± 0.5 degree maximum values have also been applied as rotational distortion. After the application of distortions in either direction, a region of interest which must be same for all synthetically generated images is extracted from the whole image. Then, all extracted regions are used as frames and put one after another to obtain the synthetic video used in this experiment. Following Figure 4.2 and Figure 4.3 show the high resolution image to which random disturbances were applied at time t and $t+1$ respectively and corresponding region of interests which are used as the frames of the synthetic video at time t and $t+1$ respectively. As a result, a synthetic video having 100 frames and 352x288 pixels CIF resolution is obtained.

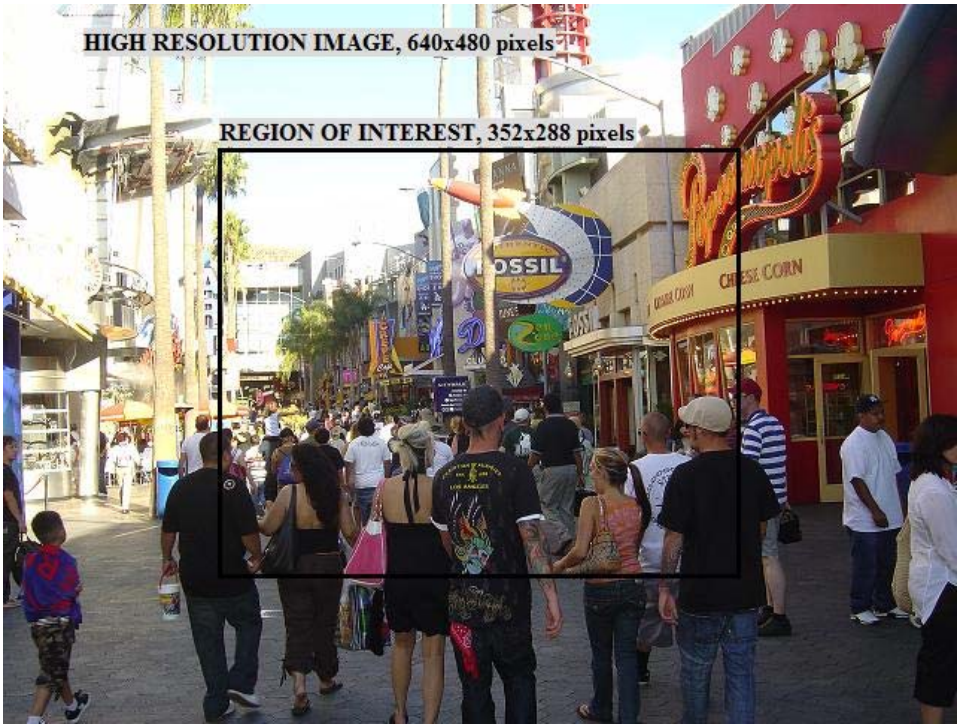

Figure 4.2 : High resolution image and region of interest at time t

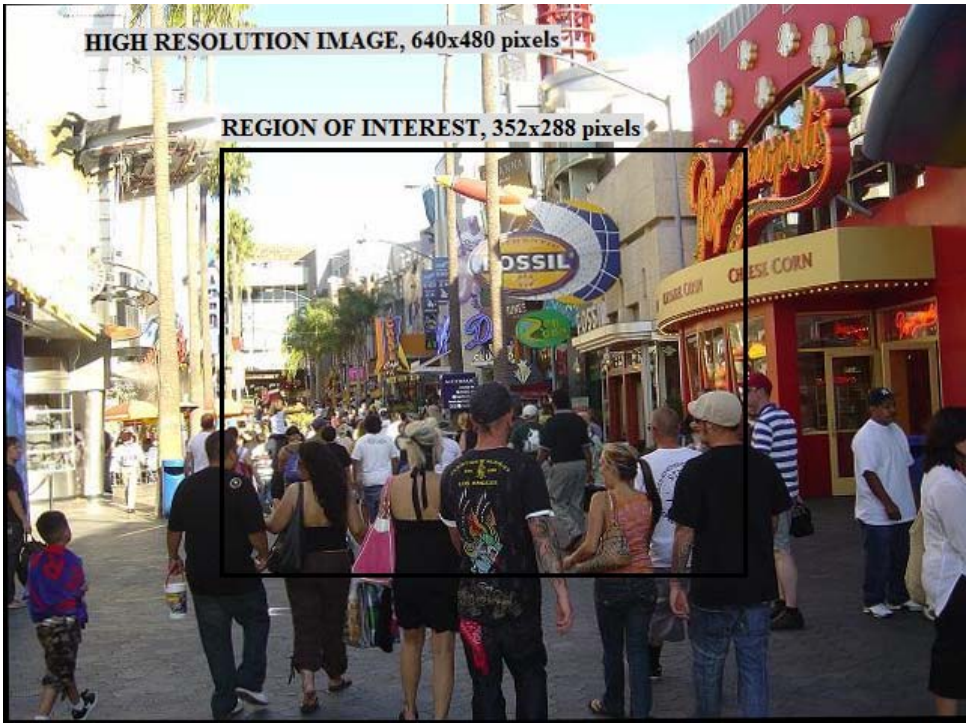

Figure 4.3 : High resolution image and region of interest at time $t+1$

Then, all digital motion estimation algorithms are performed over the synthetic video to find the global motions between the frames. Since exact distortions are known in this experiment, following figures are used to show estimation errors of each algorithm. Figure 4.4 shows cumulative exact interframe distortions and the cumulative interframe distortions estimated by area based correlation algorithm in either direction. Estimation errors of the algorithm for each direction are also illustrated in the figure. Figure 4.5 and Figure 4.6 show the same information with Figure 4.4 but for Lucas and Kanade and block based phase correlation algorithms respectively. Finally, both interframe mean square errors and average mean square error of the video which is stabilized by the algorithms separately are shown in the last figure of this experiment which is Figure 4.7.

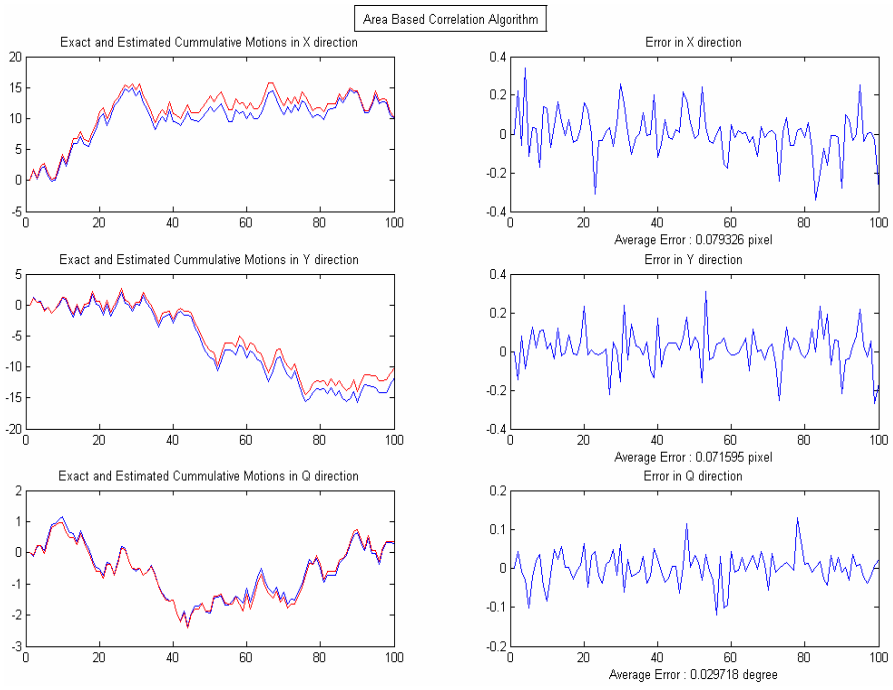

Figure 4.4 : Estimated motions of Area Based Correlation algorithm

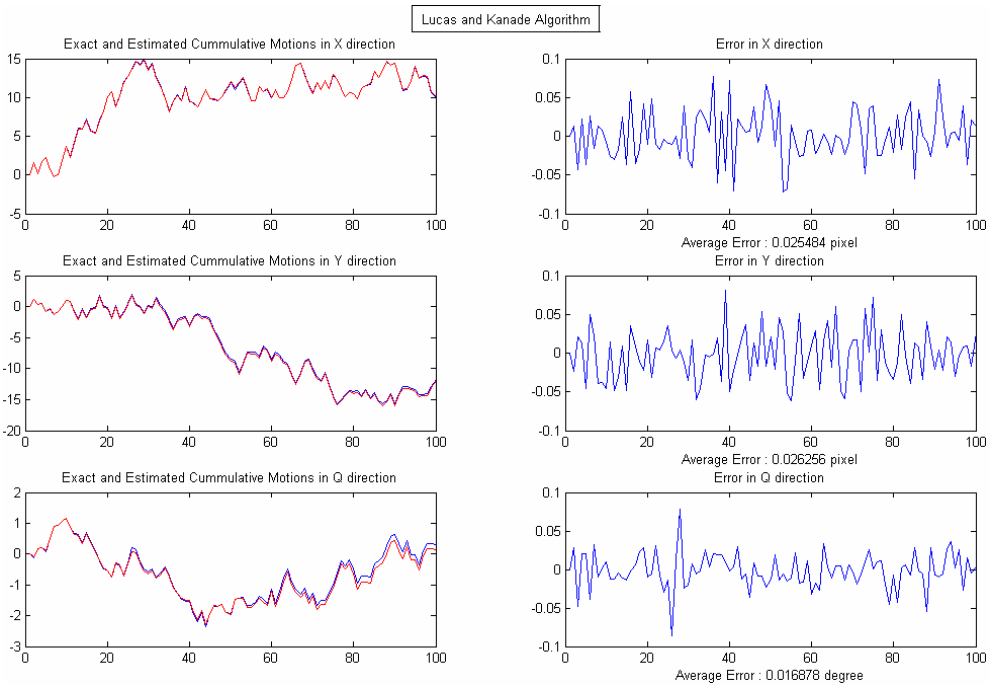

Figure 4.5 : Estimated motions of Lucas nad Kanade algorithm

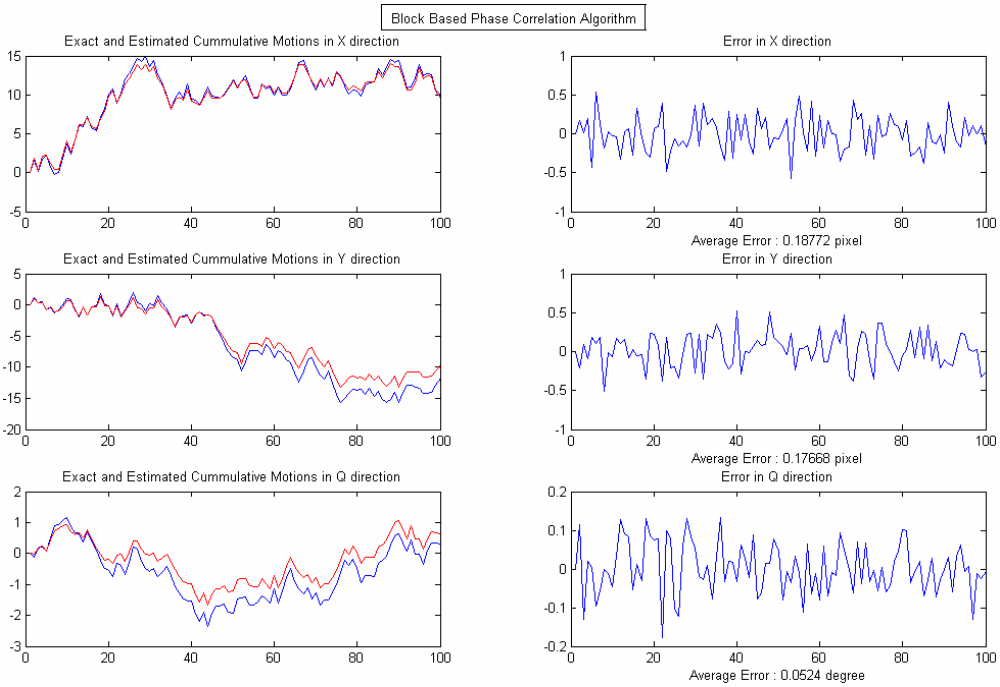

Figure 4.6 : Estimated motions of Block Based Phase Correlation algorithm

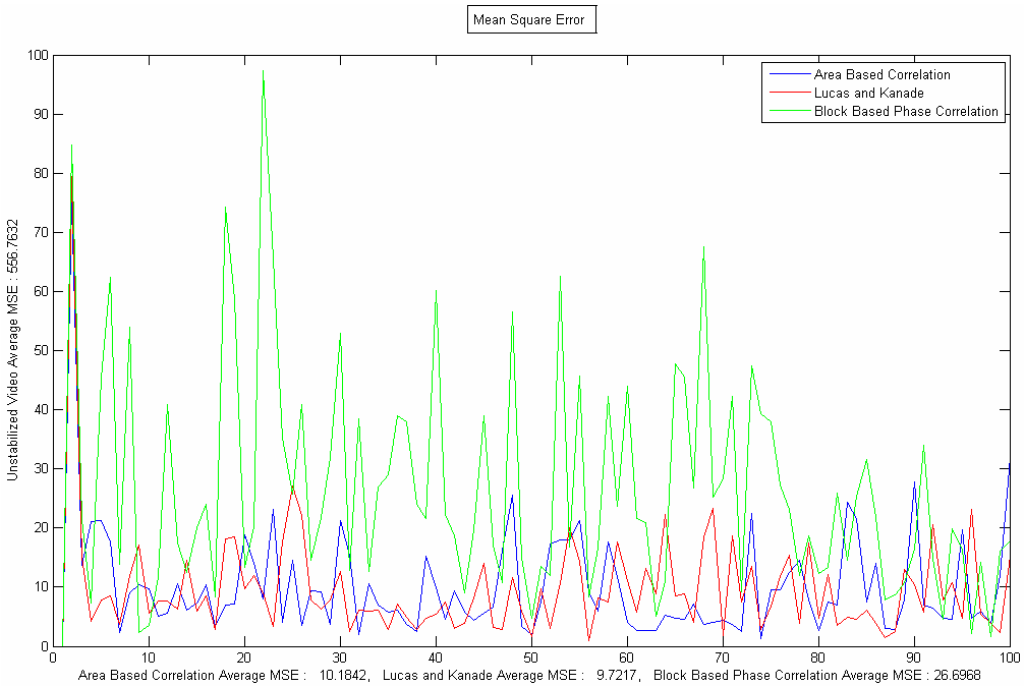

Figure 4.7 : MSEs of all digital motion estimation algorithms

4.1.2. REAL VIDEO EXPERIMENTS

Real videos have been captured by a camera with the setup given in Figure 4.1. In these experiments, camera setup is given some movements in any direction by hand as jitter while capturing the videos. Since the camera setup contains an IMU, motions of the camera can be taken from the IMU also and mechanically estimated motions can be compared to the digital motion estimation algorithms. But the exact distortions of the camera are not known. Therefore, it is not possible to compare the algorithms with respect to estimation error criteria for real videos. Mean square error and visual inspection are the only criterion for the comparison of all motion estimation algorithms including mechanical motion estimation.

4.1.2.1. Real Video with Low Amplitude Jitter

In this experiment, small translational and rotational global motions are desired on the video in order to compare the motion estimation algorithms under small distortions. Therefore, a real video having 100 frames with 352x 288 pixels CIF

resolution has been captured using the experiment setup while small movements are given to the setup as jitter in either direction. Motions of the camera are also taken from the IMU while capturing the video. A sample frame from the unstabilized real video has been given in the following figure.

Figure 4.8 : A sample frame from the real video used in experiment 4.1.2.1.

After obtaining the real video, it is stabilized using the estimated motions and a stabilized video is obtained for each algorithm including mechanical motion estimation. Figure 4.9 below shows the interframe mean square errors and average mean square error for each stabilized video.

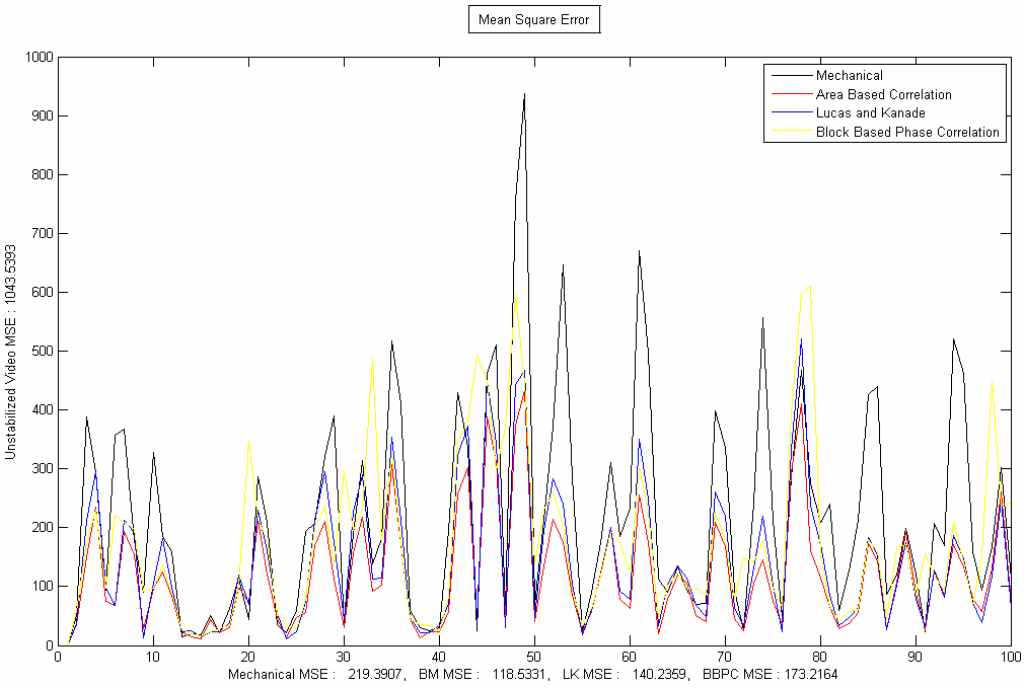

Figure 4.9 : MSEs of all motion estimation algorithms

4.1.2.2. Real Video with High Amplitude Jitter

This experiment is similar to the experiment handled in Section 4.1.2.1. Difference is the amount of distortion applied to the camera setup and the usage of mechanically estimated motions in the digital algorithms. Since big translational and rotational global motions are desired on the video, camera setup is given big movements as jitter in either direction. The reason of giving big amount of distortion is to compare motion estimation algorithms under big distortions. Same as the experiment handled under Section 4.1.2.1, a real video having 100 frames and 352x288 pixels CIF resolution has been captured and camera movements estimated by IMU have also been collected while capturing. Then, the video is stabilized using the estimated motions and a stabilized video is obtained for each algorithm which is similar to the previous experiment under Section 4.1.2.1. But, in this experiment, a different approach is used for the estimation of jitter. That is, mechanically estimated motions are used as the initial estimate for all digital algorithms and then the video is stabilized with respect to these motions also. A sample frame from the unstabilized

real video has been given in the following Figure 4.10. Like the previous experiment, only mean square error and visual inspection are used as the comparison criterion. Figure 4.11, below, shows the interframe mean square errors and average mean square error for the videos which are stabilized using the pure mechanically estimated motions, pure digitally estimated motions and the composite approach. The reason of not showing the results of all digital algorithms in the figure is that all digital algorithms exhibit approximately the same performance in the estimation of jitter having high amplitude.

Figure 4.10 : A sample frame from the real video used in experiment 4.1.2.2.

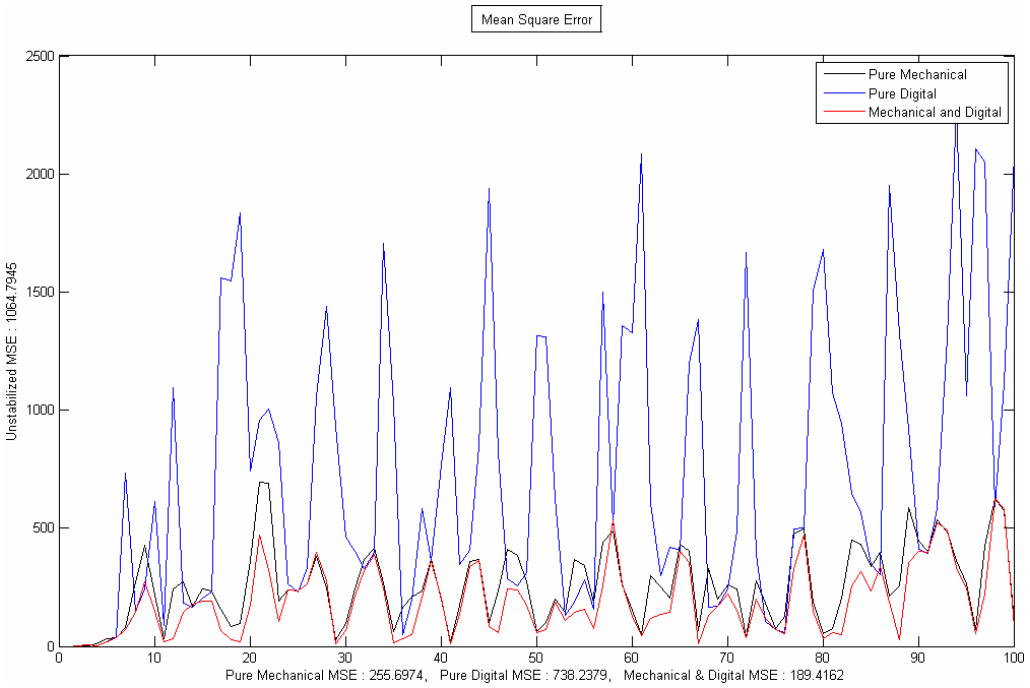

Figure 4.11 : MSEs of pure mechanical, pure digital and composite methods

4.1.3. COMPUTATION TIME COMPLEXITY ANALYSIS

Motion estimation is the most time consuming part of video stabilization especially in digital approach. Depending on the parameters and the size of the video, algorithms may run in different computation times. Since real time performance has been taken into consideration, computation time complexity analysis is needed for the motion estimation algorithms. Computation time complexities have been calculated from the code implementations of the algorithms. Furthermore, pseudo code of each algorithm has been given under the corresponding section in Chapter 3. Computation time complexity of each algorithm has been shown in the following table.

Table 4-1 : Computation Time Complexity of Motion Estimation Algorithms

Algorithms	Complexity
Area Based Correlation	$O\{ H W SS^2 \}$
Lucas and Kanade	$O\{ IT BS^2 (H W + BS^2 2^{2PYR} – (H + W) BS 2^{PYR}) / 2^{2PYR} \}$
Block Based Phase Correlation	$O\{ H W BS^2 \}$
Mechanical Motion Estimation	Negligible
H: Height, **W:** Width, **BS:** Block Size, **SS:** Search Block Size, **IT:** Iteration Number, **PYR:** Pyramid Level	

Since all digital motion estimation algorithms are performed over the images, they have very heavy computational loads. Furthermore, they also have different parameters which affect computation time. On the other hand, motion estimation in mechanical approach has negligible computational complexity. Reading of inertial positions of the camera from the IMU is the only operation in order to estimate motion. In fact, obtaining the inertial position has a sequence of operations such as signal gain adjustment, analog to digital conversion and filtering. Consequently, computation time of motion estimation in mechanical approach is negligible or can be assumed as zero with respect to digital approach.

4.1.4. EVALUATION OF MOTION ESTIMATION ALGORITHMS

Experiments conducted in Section 4.1 show that all digital motion estimation algorithms work well enough for small translational and rotational shifts. If the amount of given distortion is increased, they generally tend to fail. Figure 4.9 and Figure 4.11 show the performance of digital algorithms under the videos which have low and high distortions respectively. In order to overcome this failure, parameters of the digital algorithms are needed to be adjusted which causes to increase in computational time.

Block Matching is a classical and reliable algorithm. It separates each image into blocks and tries to find the matched blocks between two images. Best correlated blocks in the first image are searched within a region in the second image. Therefore, range of the search region must be defined big enough to cover all of the jitters on the video. On the other hand, if you define the range of the search region big enough, computation time of the algorithm may last seconds for an image. Except search region size, block size and threshold values are other parameters which are given in Table 4-1. Although they are not effective over the computation time, they are effective for the accuracy of found motion vector between two images. If the block size is defined too small, the possibility of finding more than one block within a search region increases. Since correlation operation is performed by small number of pixels, different matches may be found. Threshold values prevent from finding wrong motion vectors between the block by putting some criteria over the correlation operation. That is, even if a motion vector is found between two blocks, it is not allowed to contribute to the calculation of global motion vector. If there is a block whose standard deviation is less than a predetermined value or a block whose sum of absolute difference value is greater than a predetermined value, found motion vector is not taken as meaningful motion vector and therefore, it is not taken into consideration in the calculation of global motion vector.

Lucas and Kanade algorithm is a kind of optical flow algorithm. It calculates the optical flow between two blocks and determines a motion vector between them. According to the definition of optical flow motion estimation, it is assumed that small motion is allowed between two blocks. Therefore, big motions cannot be found in Lucas and Kanade method. In order to overcome this problem, Lucas and Kanade method is implemented with a pyramidal structure. In this structure, images are reduced in size by half for each pyramid level. In other words, if there is a 4 pixels global motion between images, it is reduced to 0.25 pixel motion in five level pyramidal structures. Block size is another parameter of Lucas and Kanade algorithm. Block size has to be defined big enough to find the maximum motion at the bottom level. Since block size increases with the number of pixels in the calculation of the

optical flow for a block, larger block size should be selected. Another parameter in the Lucas and Kanade algorithm is iteration number. Algorithm calculates the optical flow for each iteration and tries to reach the exact displacement value by recalculating the location of block in the second image. Exact displacement requires greater number of iterations. But algorithm may be used to find approximate exact displacement with predetermined iteration number. There is a trade-off between computation time load and accuracy. However, the algorithm starts to give closer results to the exact values after three iterations. Therefore, iteration number can be taken as three for general usage. All the parameters of Lucas and Kanade algorithm which are effective on the computation time and accuracy are given in Table 4-1.

Block Based Phase Correlation applies classical phase correlation method on each block and obtains a number of local motions such as area based correlation or Lucas and Kanade algorithms. Block size is the only criterion which determines the accuracy and the computation time. Greater block size can find greater displacements between the blocks. Therefore, block size has to be selected considering the maximum displacement on the video which is same with other algorithms.

If we evaluate the digital motion estimation algorithms, all of them find the displacement very accurately and close to each other. But because of its pyramidal structure, dynamic range of Lucas and Kanade algorithm is superior with respect to other algorithms and it also finds the motion between the images slightly more accurately. Following Lucas and Kanade algorithm, area based correlation has superior performance in the accuracy compared to block based phase correlation. On the other hand, block based phase correlation is the fastest algorithm among all algorithms. Figure 4.4, Figure 4.5, Figure 4.6 and Figure 4.7 show the accuracies and Table 4-1 shows the computational complexity of all implemented digital motion estimation algorithms.

All digital motion estimation algorithms work better over the synthetic videos rather than real videos. The reason is that synthetic videos have no corruptions produced by the imperfections of the camera, illumination changes, etc. Even if they use video

content and there are different undesired effects on the video, they can exhibit more accurate results compared to mechanical motion estimation. This statement is valid provided that the parameters of the algorithm are adjusted to suitable values to compensate the whole jitter. Otherwise, all of digital algorithms exhibit very poor accuracies. Figure 4.9 shows mean square errors of the videos which have been stabilized using the motions estimated by mechanically and estimated by digital algorithms having suitable parameters. However, mechanical motion estimation has always some error due to the characteristics of the motion sensors. But since motion sensors have greater dynamic ranges considering digital algorithms, mechanical motion estimation is superior in case there are big disturbances on the video. Figure 4.11 shows mean square errors of the videos which have been stabilized using pure mechanically estimated motions and pure digitally estimated motions.

Digital motion estimation is completely composed of software operations. This brings systems to have high portability, but in contrast, brings systems considerable CPU time, system memory, and implementation complexity. Due to these reasons, digital motion estimation algorithms are less preferable for real time mobile applications. The mechanical approach in motion estimation suggests a solution for computation time. Since, motions are read directly from sensors, no complex algorithms are implemented and no complex operations are performed. If a system has a good motion sensor, motion information can be obtained very accurately besides consuming no time. In this study, Microstrain 3DM-GX1 IMU has been used as the mechanical motion sensor. This sensor has intermediate quality, and therefore, exhibits intermediate accuracy. If more reliable sensor is used, more reliable and accurate values are obtained. That is, the quality of motion sensor is the most important parameter of mechanical motion estimation. Moreover, mechanical and digital estimation techniques have been also used together in this study. Motion sensor information has been used as the initial estimate for digital motion estimation algorithms. As it is seen from the Figure 4.11, since initial estimate has been satisfied by the motion sensors, the parameters of digital algorithms have been adjusted to find small motions which reduce the computation time while increasing the accuracy.

And therefore, this approach exhibits superior performance over the pure mechanical and pure digital motion estimation algorithms.

4.2. MOTION CORRECTION EXPERIMENTS

This section covers the experiments and evaluations about all implemented motion correction algorithms. During the motion correction experiments, various real video sequences having various global motion characteristics have been tested. Some of these videos have been captured by the setup given in Figure 4.1 and some of them which have been also used in different video stabilization researches in the literature have been obtained from the Internet. Details concerning the mentioned videos have been given in the related sections where the results of each experiment have been illustrated. Since motion correction algorithms are performed over the global motion vectors, not over the images, whether the video is real or synthetic does not affect the result. Therefore, only real videos have not been utilized here.

Visual inspection over the video is the only criterion for the comparison of the motion correction algorithms since it is not possible to know the exact intentional and unintentional motions. In fact, graphical representation of the estimated and the corrected motion vectors can also be used for the evaluation of the motion correction algorithms. Both cumulative sum and difference in global motion vectors have been utilized within the evaluation. Therefore, the results of the correction algorithms have been illustrated graphically.

Motion correction has been realized for only translational direction; that is, no motion correction has been applied to rotational motions. It means that all estimated rotational motions have been interpreted as rotational distortion. The reason is that it is not realistic for real life to give intentional rotational motions to the camera while

capturing. However, if there is such a case, motion correction can also be performed in rotational direction.

As motion estimation algorithms, each of motion estimation algorithms has different parameters. Depending on these parameters, performance of the algorithms changes drastically. Therefore, before carrying out the experiments, parameters of each algorithm are set to predetermined values which were determined experimently.

4.2.1. REAL VIDEO EXPERIMENTS

As it has been stated above, different real videos have been used in order to compare motion correction algorithms. Each test within this section has been performed with a different video. Details and the results of each experiment having different test videos have been indicated under corresponding sections.

4.2.1.1. Real Video Captured by the Experiment Setup

In this experiment, small translational and rotational global motions over the small translational intentional motions are desired on the video in order to compare the motion correction algorithms under small intentional motions. Therefore, a real video having 100 frames with 352x 288 pixels CIF resolution has been captured using the experiment setup while small unintentional movements in either direction are given to the setup while it is moving in the translational direction intentionally. In this experiment, interframe global motions are obtained from the motion sensor. That is, correction is applied to mechanically estimated motions. Result of the correction algorithms is given in the following figures. Figure 4.12 shows the estimated cumulative raw motion vectors in X direction and their forms after correction is applied. On the other hand, Figure 4.13 shows the estimated cumulative raw motion vectors in Y direction and their forms after correction is applied.

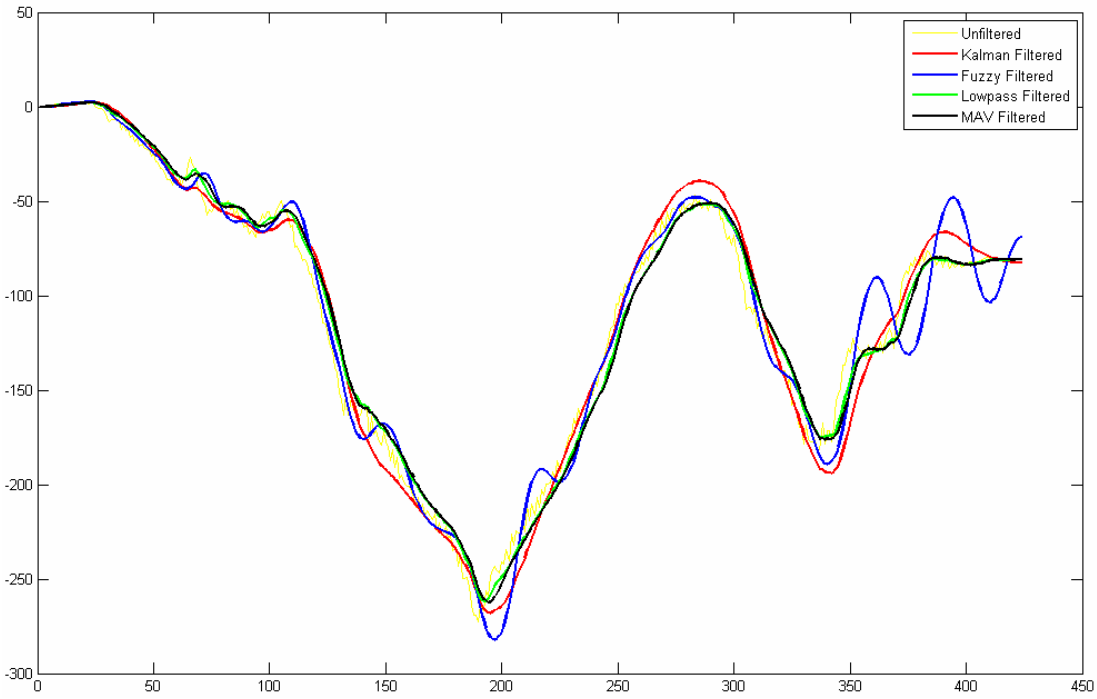

Figure 4.12 : Correction of real video captured by the setup in X direction

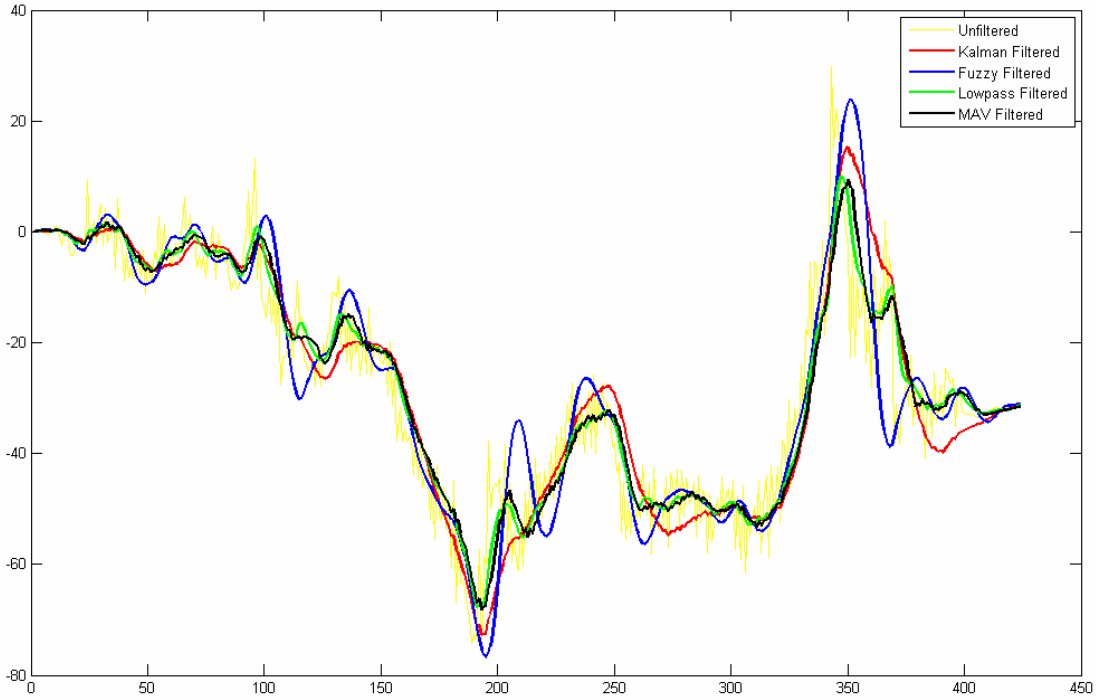

Figure 4.13 : Correction of real video captured by the setup in Y direction

4.2.1.2. Real Video Taken from the Internet

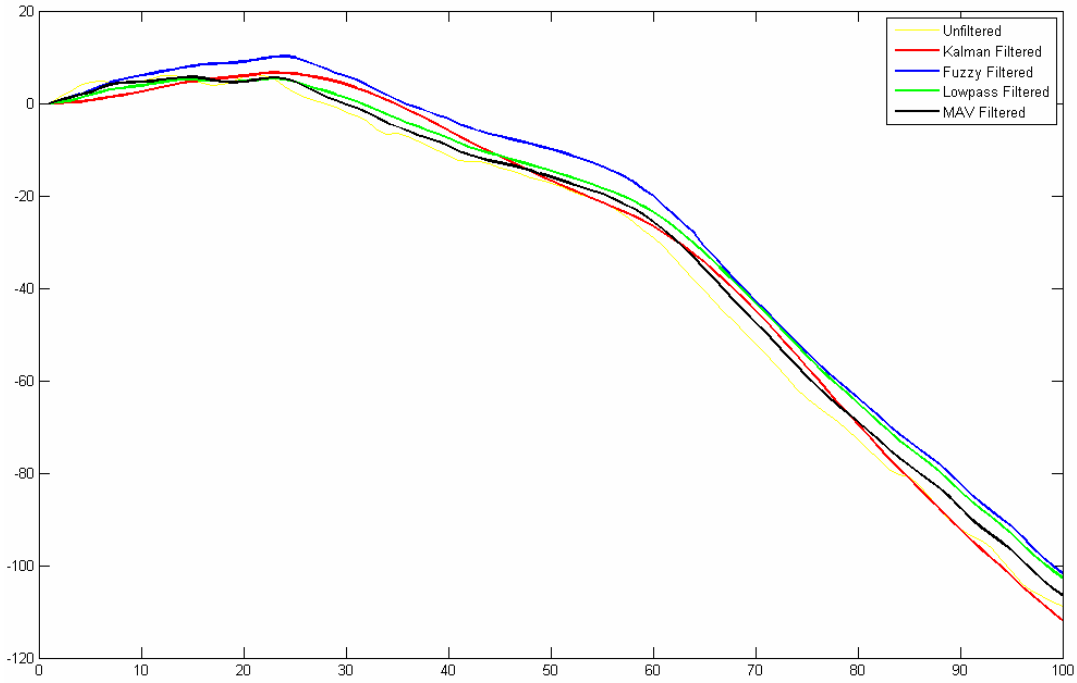

Figure 4.14 : Correction of real video obtained from the internet in X direction

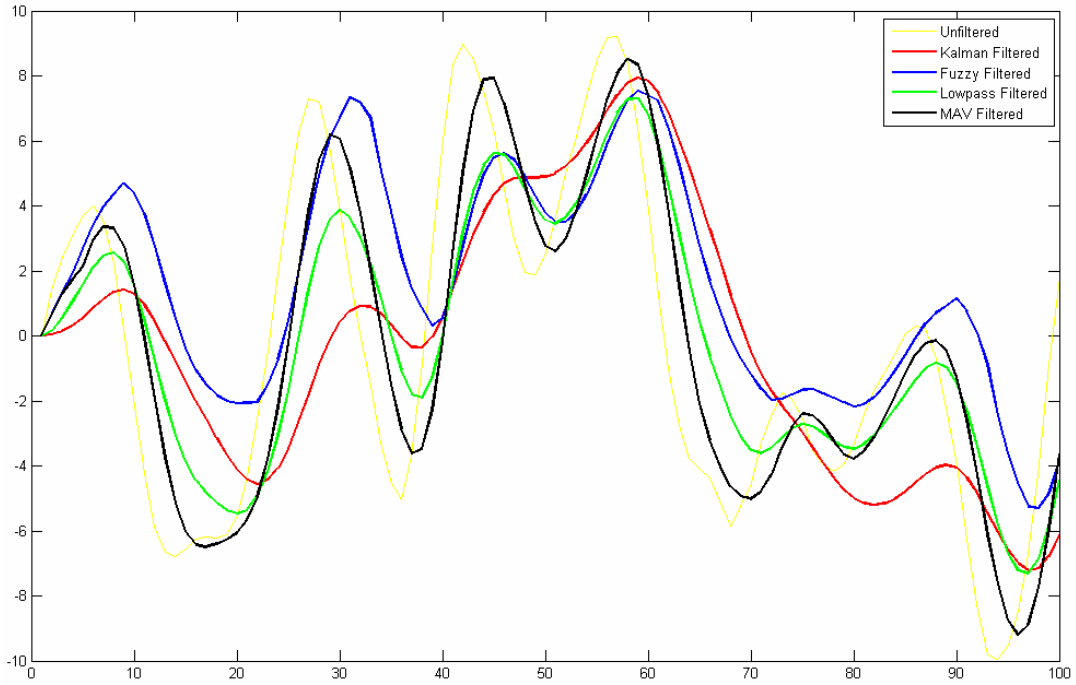

Figure 4.15 : Correction of real video obtained from the internet in Y direction

In this experiment, another real video is used which is obtained from the internet. This video is one of the used videos, called “off-road sequence” [1], for video stabilization researches in the literature. Result of the correction algorithms is given in the figures above. Figure 4.14 shows the estimated cumulative raw motion vectors in X direction and their forms after correction is applied. On the other hand, Figure 4.15 shows the estimated cumulative raw motion vectors in Y direction and their forms after correction is applied.

4.2.2. EVALUATION OF MOTION CORRECTION ALGORITHMS

Although interframe global motions are obtained very accurately on the video, good stabilization sensitivity cannot be reached unless intentional and unintentional motions cannot be differentiated between each other.

Kalman filtering in motion correction is the most commonly used correction method among all such kind of methods. In general, constant velocity model which describes better the characteristics of intentional camera movements is used for the system model in Kalman filtering. Kalman filter has also some parameters for its adaptation to the system model. Because of that it is more adaptive with respect to the low pass filtering and moving average filtering.

Fuzzy filtering is another method used for motion correction. Like Kalman filter, it uses constant velocity motion model. It has also some parameters for its adaptation to the system model. Fuzzy filter exhibits very different characteristics in terms of the values of the parameters which are more predictable within other correction algorithms. Thus, parameters of fuzzy filter have to be adjusted very accurately.

Lowpass filtering is a classical way to remove high frequency components from a signal. Since jitters have high frequency characteristics, Lowpass filters are used for the differentiation of the jitter. The important point for the Lowpass filtering is to determine the cut off frequency.

Moving average filtering is the simplest one among all algorithms. It has only the parameter of filter length which determines the number of past signals which is used in the calculation of the current signal.

If we evaluate the Figures from 4.12 to 4.15, Kalman and lowpass filters seem to make smoother the signals. On the other hand, moving average filter seems to be more successful for the signals which have relatively high frequencies. But if we evaluate the all algorithms theoretically, Kalman and fuzzy filters have very adaptable and superior structures with respect to the other algorithms. Since they use a model for the intentional motion, they can differentiate the intentional motions from the unintentional motions more accurately if the model is defined as close as possible to the real.

CHAPTER 5

CONCLUSION AND FUTURE WORK

5.1. CONCLUSIONS

The main concern of this study is to propose a solution related to video stabilization for a kind of mobile robot application in which a series of different image/video processing operations such as depth estimation, object detection, object recognition, etc. are performed in addition to video stabilization. If we put these operations in order, video stabilization comes first within the whole process and then all other operations take their actions using the output of video stabilization. Therefore, the results of video stabilization directly affect the results of other operations.

In this study, video stabilization has been divided into three main parts and examined under these parts which are motion estimation, motion correction and image correction. Motion estimation and motion correction have been given more importance among these three parts since image correction is much more related to the image quality rather than stabilization quality. Therefore, three motion estimation algorithms which are known by their considerable performances and four motion correction algorithms which are the most commonly used ones in the literature have been examined. Furthermore, a different approach in motion estimation has been also examined in addition to other algorithms. This additional method uses a mechanical component, IMU, to estimate the motion to which camera or mobile platform is exposed. Since motion estimation using any digital algorithm is very time-consuming

operation, estimation of motion mechanically brings considerable computation time efficiency which is the main reason of using a mechanical sensor in video stabilization. If we discard the mechanical motion estimation, digital video stabilization is the main frame of this study.

If we compare the motion estimation algorithms among each other, we can see that all digital motion estimation algorithms are superior in accuracy in terms of mechanical motion estimation. However, if the parameters are adjusted to get better time efficiency by means of reducing the search region range or reducing the iteration number, etc., digital motion estimation algorithms start to fail. For example, assume that there is a video having maximum ±10 pixels jitter. If you perform area based correlation algorithm with ±10 pixels range of search region, all the jitter can be detected correctly. But if the range of search region is decreased to ±7 pixels in order to reduce the computation time, some jitters which are greater than the range of search region cannot be detected correctly. On the other hand, mechanical sensors have wider dynamic ranges in motion estimation. Therefore, they can exhibit same estimation characteristics for different range of motions. However, mechanical sensors have some undesirable effects such as drift and bias due to their natures. Thus, the exact estimation results cannot be obtained anytime for mechanical estimation. In conclusion, digital algorithms exhibit superior accuracy performance with respect to mechanical motion estimation for the jitter whose amplitude is in the range of digital motion estimation algorithms. Yet, for the jitter whose amplitude is out of the range of digital motion estimation algorithms, digital algorithms exhibit poor accuracy performance in terms of mechanical motion estimation. In this study, in addition to usage of both digital and mechanical motion estimation methods separately, composition of them has been also used. The aim of this composite method is to decrease the computation time while keeping the accuracy same. In this method, mechanically estimated motions are used as the initial estimates for the digital motion estimation algorithms. Since there is an initial estimate, there is no need to keep the range of digital algorithms wide which leads to decrease in computation time. Figure 4.11 shows the performance of this composite method. For

example, ±10 pixel jitters can be detected accurately with area based algorithm having ±3 pixels search region. Because the best correlated match is not searched around the current region with ±3 pixels range, but it is searched a distance which is determined by mechanical motion estimation away from the current region. Furthermore, since it is applied over the mechanical estimation, composite method has superior accuracy performance in terms of mechanical estimation.

For the motion correction case, Kalman filtering and Fuzzy filtering are the algorithms which have superior characteristics in terms of Lowpass filtering and moving average filtering. The reason is that they are formed considering the system model. In this study, constant velocity model which accepts the velocity of the camera is constant for intentional motions has been used for the system model. Since motion correction operation is a kind of smoothing operation, Lowpass filtering and moving average filtering give reasonable performances.

5.2. FUTURE WORK

In this research, only the disturbances in roll, pitch and yaw directions have been estimated by mechanical sensors and they have been used to remove the jitter. Although they cause most of the disturbances over the video, the disturbances on x, y and z directions have also some effects on the video. As a result, removing all the disturbances cause to increases in the stabilization accuracy and visual quality of the video.

Since mechanical motion estimation is realized by mechanical sensors, there is always an error on the estimation of motions in either direction. The quantity of the error is directly related to the quality of the sensor. In this study, Microstrain 3DM-GX1 IMU is used which is an intermediate IMU having MEMS technology. However, if high quailty and reliable mechanical sensor is used, the accuracy of mechanical motion estimation can be increased considerably with respect to digial motion estimation in every range of data. Subsequently, if enough estimation sensitivity is reached with high quality mechanical sensor, there is no need to apply

extra digital motion estimation onto the mechanical motion estimation which reduces the computation time while increasing the accuracy.

REFERENCES

[1] S. Ertürk, “Real-time digital image stabilization using Kalman filters”, Real-Time Imaging, vol. 8, no. 4, pp. 317–328, 2002

[2] E. Yaman and S. Ertürk, “Image stabilization by Kalman filtering using a constant velocity camera model with adaptive process noise”, International Conference on Electrical and Electronics Engineering, ELECO2001, Bursa, vol., pp. 152-157, 2001

[3] G. Welch, G. Bishop, “An Introduction to the Kalman Filter”, Transactions of the ASME - Journal of Basic Engineering, 82 (Series D), pp. 35-45

[4] S. Ertürk, “Image sequence stabilisation based on Kalman filtering of frame positions”, Electronics Letters, 37, (20), pp. 1217-1219, 2001

[5] O. Kwon, J. Shin, J. K. Paik, “Video Stabilization Using Kalman Filter and Phase Correlation Matching”, ICIAR 2005, pp. 141-148

[6] R. E. Kalman, “A New Approach to Linear Filtering and Prediction Problems,” Transaction of the ASME - Journal of Basic Engineering, pp. 35-45, March 1960

[7] M. K. Güllü, S. Ertürk, "Image Sequence Stabilization using Membership Selective Fuzzy Filtering", Lecture Notes in Computer Science, Springer Verlag, 2869, pp. 497-504, 2003

[8] N. R. Hughes, G. N. Roberts, G. R. Wilson, “Application of Fuzzy Signal Processing to Three Dimensional Vision”, 5th International Conference on FACTORY 2000, conference publication no: 435, April 2-4, 1997

[9] M. K. Güllü, E. Yaman and S. Ertürk, “Image sequence stabilisation using fuzzy adaptive Kalman filtering”, Electronics Letters, 39, (5), pp. 429-431, 2003

[10] M. K. Güllü, S. Ertürk, "Fuzzy Image Sequence Stabilisation", Electronics Letters, 39, (16), pp. 1170-1172, 2003

[11] N. Kyriakoulis and A. Gasteratos, "A Recursive Fuzzy System for Efficient Digital Image Stabilization", Advances in Fuzzy Systems, vol. 2008, article id: 920615

[12] M. K. Güllü and S. Ertürk, “Membership function adaptive fuzzy filter for image sequence stabilization,” IEEE Transactions on Consumer Electronics, vol. 50, no. 1, pp. 1-7, 2004

[13] B. Zitova, J. Flusser, “Image registration methods: a survey”, Image and Vision Computing, 21, pp. 977–1000, 2003

[14] L.G. Brown, “A survey of image registration techniques”, ACM Computing Surveys 24 (1992) 326–376

[15] Richard Szeliski, “Image Alignment and Stitching: A Tutorial”, Foundations and Trends in Computer Graphics and Vision, vol. 2, iss. 1, pp. 1-104, January 2006

[16] S. David, N. Steven and G. Danel, “Image Stabilization Technology Overview,” http://www.invensense.com/shared/pdf/ImageStabilizationWhitepaper_051606.pdf, 2007, last access time: 25.11.2008

[17] J. A. Ramirez, E. Rodriguez, J. C. Echeverria, “Detrending fluctuation analysis based on moving average filtering”, Physica, A 354, pp. 199-219, 2005

[18] A.Engelsberg, G. Schmidt, “A comparative review of digital image stabilising algorithms for mobile video communications”, IEEE Transactions on Consumer Electronics, vol. 45, no. 3, August 1999

[19] M. Niskanen, O. Silven and M. Tico, “Video Stabilization Performance Assessment”, Proc. IEEE International Conference on Multimedia & Expo (ICME 2006), Toronto, 405-408

[20] T. E. Wett, “Measuring the effecetiveness of Image/Video Processing for Stabilizing a Video Image Using a Commertial Media Processor”, technology.asu.edu/files/documents/tradeshow/Dec03/WettThomas.pdf , last access date: 25.11.2008

[21] Y. Matsushita, E. Ofek, X. Tang, H.-Y. Shum, “Full-frame Video Stabilization”, CVPR (1) 2005, pp. 50-57, 2005

[22] S. Piva, M. Zara, G. Gera, C. S. Regazzoni, “Color-Based Video Stabilization for Real-Time On-Board Object Detection on High Speed Trains”, Proceedings of the IEEE Conference on Advanced Video and Signal Based Surveillance, pp. 299, July 21-22, 2003

[23] S.-H. Yang and F.-M. Jheng, “An adaptive image stabilization technique,” IEEE International Conference on Systems, Man, and Cybernetics (SMC2006), Taipei, Taiwan, October 8-11, 2006

[24] N. D. Cahill and A.C. Gallagher, “Automatic Global Alignment of Aerial Imagery”, Western New York Image Processing Workshop, September 2004

[25] D. Barreto, L. D. Alvarez, J. Abad, “Motion Estimation Techniques in Super-Resolution Image Reconstruction. A Performance Evaluation”, Virtual Observatory, Plate Content Digitalization, Archive Mining and Image Sequence Processing, in Sofia, Bulgary, April 2005

[26] F. Gensolen, G. Cathebras, L. Martin, and M. Robert, “An Image Sensor with Global Motion Estimation for Micro Camera Module”, Publisher Springer Berlin / Heidelberg, Lecture Notes in Computer Science, Advanced Concepts for Intelligent Vision Systems, vol. 3708/2005, pp. 713-721

[27] A. C. Brooks, “Real-Time Digital Image Stabilization”, EE420 Image Processing Computer Project Final Paper, March 2003

[28] http://www.aselsan.com.tr/urun.asp?urun_id=100&lang=tr, last access time: 25.11.2008

[29] http://www.aselsan.com.tr/urun.asp?urun_id=55&lang=tr, last access time: 25.11.2008

[30] G. Pang and H. Liu, “Evaluation of a Low-cost MEMS Accelerometer for Distance Measurement”, Journal of Intelligent and Robotic Systems archive, vol. 30 , iss. 3, pp. 249 – 265, March 2001

[31] E. Vermeulen, “Real-time Video Stabilization For Moving Platforms”, 21st Bristol UAV Systems Conference, April 2007

[32] J.-Y. Bouguet, “Pyramidal Implementation of the Lucas Kanade Feature Tracker: Description of the algorithm”, Technical Report, 1999. 2

[33] E. Estalayo, L. Salgado, F. Jaureguizar, and N. García: “Efficient Image Stabilization and Automatic Target Detection in Aerial FLIR Sequences”, Proceedings of the SPIE: Defense and Security Symposium, Orlando (FL), April 2006.

[34] S. Baker, and I. Matthews, “Lucas-Kanade 20 Years On: A Unifiying Framework”, International Journal of Computer Vision, vol. 56 3, pp. 221-255

[35] C. Fan, J. Gong, J. Zhu and L. Zhang, “An Improvement Approach Based On Keren Sub- Pixel Registration Method”, ICSP2006 Proceedings, 2006

[36] B. Lucas, and T. Kanade, “An Iterative Image Registration Technique, with an Application to Stereo Vision”, International Journal Conference in Artifcial Intelligence, pp. 121-130, 1981

[37] T. Amiaz, E. Lubetzky and N. Kiryati, “Coarse to Over-Fine Optical Flow Estimation”, vol. 40, iss. 9, pp. 2496-2503, September 2007

[38] Y. Keller, A. Averbuch, “FFT based image registration”, IEEE International Conference on Acoustics, Speech, and Signal Processing, Orlando,USA, May 2002

[39] B. K. P. Horn and B. G. Schunck, “Determining optical flow”, Artificial Intelligence, vol. 17, pp. 185-203, 1981

[40] C. Guestrin, F. Cozman and E. Krotkov, “Image Stabilization for Feature Tracking arid Generation of Stable Video OverIays”, IEEE International Conference on Intelligent Robots and Systems, vol. 1, pp. 19-24, Canada, October 1998

[41] Z. Prime, “Using the Microstrain 3DM-G(X1) IMUs”, October 4, 2007

[42] www.mathworks.com, last access time is 25.11.2008

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

[44] P. Vandewalle, S. Süsstrunk and M. Vetterli, “A Frequency Domain Approach to Registration of Aliased Images with Application to Super-Resolution”, EURASIP Journal on Applied Signal Processing (special issue on Super-resolution), vol. 2006, article id: 71459, 2006

[45] P. H. S. Torr and A. Zisserman, “Feature Based Methods for Structure and Motion Estimation”, Proceedings of the International Workshop on Vision Algorithms: Theory and Practice, pp. 278–294, 1999

[46] S. Ertürk and T. J. Dennis, “Image sequence stabilisation based on DFT filtering”, IEE-Proc., Vis. Image Signal Process., 147, (2), pp. 95-102, 2000

[47] http://www.ece.cmu.edu/~ee899/project/deepak_mid.htm, last access date: 25.11.2006

APPENDIX A

TRANSFORMATION MATRIX

Transformation matrix determines the global (entire) displacement using local motions obtained from locally matched points or regions between images.

Transformation matrix is formed considering some mathematical relations between the images. The issue here is to find this relation using some matrix operations.

Generally, images are assumed to be exposed to some known transformations such as translational, euclidean, similarity, affine or projective transformation, and all matrix operations are preceded under the consideration of this assumption.

Lets think two images and assume that there is a kind of transformation between the images. (x, y) be the points in the first image and (x', y') be the corresponding points in the second image. Different transformations use different rules or mathematical relations while transforming the points in one image to another. These relations are given below for some commanly used transformations whose graphical illustrations are also given in Figure A.1.

Translational transformation consists of only constant translational offset and expressed as the following matrix;

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} dx \\ dy \end{bmatrix} + \begin{bmatrix} x \\ y \end{bmatrix} \tag{A.1}$$

where dx and dy represent the translational displacements in x and y directions respectively.

Euclidean transformation consists of rotational offset in addition to translational offset and expressed as the following matrix;

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} dx \\ dy \end{bmatrix} + \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} \tag{A.2}$$

where θ represents the rotation between images.

Similarity transformation, sometimes called rigid transformation, includes a scale factor to rotational and translational offset between images.Similarity Transformation matrix is;

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} dx \\ dy \end{bmatrix} + \begin{bmatrix} S \\ S \end{bmatrix} \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} \tag{A.3}$$

where S is scale factor.

Affine transformation deforms the angle between parallel lines. Therefore it includes a shearing factor and expressed as;

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} a_{13} \\ a_{23} \end{bmatrix} + \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} \quad or \quad \begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \tag{A.4}$$

where a_{11}, a_{12}, a_{13}, a_{21}, a_{22} and a_{23} are the parameters of affine transformation.

Projective transformation, sometimes called as perspective transformation, is more complicated transformation rather than other transformations. It has eight parameters. The angles of the parallel lines are also deformed with this transformation. Projective transformation matrix can be expressed as follows;

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \tag{A.5}$$

where p_{11}, p_{12}, p_{13}, p_{21}, p_{22}, p_{23}, p_{31} and p_{32} are the parameters of projective transformation.

For example to find only the translational displacement between the images, it is enough to use translational transformation matrix or to find rotational displacement, it is enough to use euclidean transformation matrix.

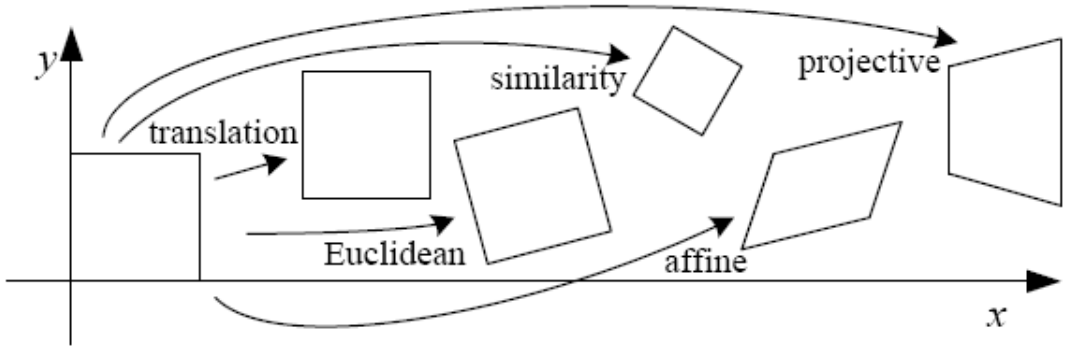

Figure A.1 : Transformations

There are also other trasformations different from the transformations mentioned above. But these are commanly used ones in the image processing.

In this thesis, it is assumed that there is an euclidean transformation between each successive frames of a video and, therefore, translational and rotatinal motions are dealt with. To find euclidean transformation, Equation A.2 is used. If we rearrange Equation A.2, we can obtain the following equation.

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} x & -y & 1 & 0 \\ y & x & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos\theta \\ \sin\theta \\ dx \\ dy \end{bmatrix} \tag{A.6}$$

In Equation A.6, there are two equations but three unknowns which are θ, dx and dy. Therefore, to solve the equation it needs to have at least one more equation. For local motion based global motion estimation algorithms, Equation A.6 represents only the values of one matched block between the images. Therefore, we can increase the number of equations using the values of other matched blocks. Lets call the points with respect to origin of the reference image as (x_j, y_j) and the points with respect to origin of the second image as (x'_j, y'_j), where j starts from one and goes to up to m which is the number of matched block. Since transformation parameters do not change for all matched points we can expand Equation A.6 to Equation A.7.

$$\begin{bmatrix} x_1' \\ y_1' \\ x_2' \\ y_2' \\ \dots \\ \dots \\ \dots \\ x_m' \\ y_m' \end{bmatrix} = \begin{bmatrix} x_1 & -y_1 & 1 & 0 \\ y_1 & x_1 & 0 & 1 \\ x_2 & -y_2 & 1 & 0 \\ y_2 & x_2 & 0 & 1 \\ \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots \\ x_m & -y_m & 1 & 0 \\ y_m & x_m & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos\theta \\ \sin\theta \\ dx \\ dy \end{bmatrix} \tag{A.7}$$

To make matrix operations easier, Equation A.7 can be written in the following form.

$$C = AB \tag{A.8}$$

If we perform the following matrix operations

$$(A^T)\,C = A^T AB \tag{A.9}$$

$$(A^T A)^{-1}(A^T)\,C = (A^T A)^{-1}(A^T A)B \tag{A.10}$$

$$(A^T A)^{-1}(A^T)\,C = (A^T A)^{-1}(A^T A)B \tag{A.11}$$

$$(A^T A)^{-1}(A^T)\,C = B \tag{A.12}$$

Consequently, using above operations matrix B is obtained whose third term, fourth term and tangent of second and first terms give the global displacement between the images in the x direction, in the y direction and in the rotational direction respectively.