Summer 8-2-2017

PROJECTION-BASED ROTATION ESTIMATION FOR VIDEO STABILIZATION

Mahdi Zerara Mr

Mahdi Zerara

Candidate

Electrical and Computer Engineering

Department

This thesis is approved, and it is acceptable in quality and form for publication:

Approved by the Thesis Committee:

Prof. Majeed M. Hayat , Chairperson

Prof. Christos Christodoulou

Prof. Manel Martinez- Ramon

PROJECTION-BASED ROTATION ESTIMATION FOR VIDEO STABILIZATION

by

MAHDI ZERARA

B.E. INDUSTRIAL AUTOMATION AND PROCESS CONTROL
UNIVERSITY OF M'HAMED BOUGARA BOUMERDES
2014

THESIS

Submitted in Partial Fulfillment of the
Requirements for the Degree of

Master of Science
Computer Engineering

The University of New Mexico
Albuquerque, New Mexico

DECEMBER 2017

DEDICATION

To my beloved family, my best friends and all the beautiful souls out there!

PROJECTION-BASED ROTATION ESTIMATION FOR VIDEO STABILIZATION

By

Mahdi Zerara

B.E. INDUSTRIAL AUTOMATION AND PROCESS CONTROL, UNIVERSITY OF M'HAMED BOUGARA BOUMERDES, 2014

M.S. COMPUTER ENGINEERING, UNIVERSITY OF NEW MEXICO, 2017

ABSTRACT

Video stabilization has become a routine post-processing enhancement procedure, to remove unwanted shifts from video sequences. This thesis proposes a new computationally optimized approach for image rotational motion estimation, which has been studied and tested on shaky videos. The method is built upon a projection-based algorithm that reduces the 2-D video sequences into vectors of size 1×180, by averaging all the pixel values on the line-segments that cross the rotation center and form angles θ with the rows of the images, such that θ ranges from 0 to 179°. From the obtained projected vectors, we perform a 1D cross-correlation-based shift estimator, to measure the angle of the occurring rotation between each pair of successive frames. Most of the feature-based and block-based shift estimation algorithms require heavy computations, whereas the proposed method reduces the computation costs significantly, and shows high estimation performances when the center of rotation is successfully located. We rely on the mean-squared-error and the rotation estimation graphs, to provide a subjective assessment of the proposed method's efficiency on synthetic as well as real-case data.

Table of Contents

List of Figures

List of Videos

List of Tables

Chapter 1

Introduction

Stabilizing videos and scenes that suffer from jitters and moving frames has become a routine postprocessing approach in most applications where cameras are involved. The moving platforms and/or the extreme weather conditions under which video capturing is performed, makes the camera shake inevitable. Thus, correcting for the undesired vibrations that result in a degraded visual quality becomes essential. Applications of video stabilization vary, but they are usually encountered in aerial or terrestrial surveillance [1] as well as film post-production.

Video stabilization techniques can be grouped into three categories: 1) mechanical stabilization; 2) optical stabilization; and 3) digital stabilization [2]. Both 1 and 2 are hardware-based techniques and are beyond the scope of this presented work. Digital stabilization techniques are software-based and can themselves be divided into three steps: 1) motion estimation; 2) motion compensation; 3) image composition [3]. Optimizing the accuracy of the motion estimation step as well as its computation cost, is the subject of interest of most recently proposed algorithms in the literature of video stabilization.

In this thesis, we present a new approach for video frames shift estimation, which is digital stabilization based. The method uses a projection-based algorithm that was described in [11] and [12], to estimate the translational and rotational motions between each pair of successive frames. The algorithm then, compensates for the computed shifts and crop the corrected video frames to remove the black edges.

1.1. Motivation

The use of digital cameras became an important aspect of our daily lives. Millions of videos are being uploaded to the internet every year, as visual communication dominates today. Many of these videos that are taken with un-expensive camera devices whose CPU capacity is limited, suffer from shaky frames. This work aims to propose a video stabilization technique that estimates the undesired rotational motions, reduces the computation cost, and opens the possibility for real time implementation into unsophisticated imaging devices.

1.2. problem description

In most cases, imaging devices during exposure encounter relatively slight motion. These frame-to-frame shifts which sometimes cause unwanted blurring in the recorded videos, can be translation, rotation or both. Many algorithms have been proposed to correct for these unwanted shifts. However, most traditional techniques are sensitive to noise, don’t correct for rotational shifts, lack accuracy, computationally expensive or hard to implement in hand-held camera devices with a limited CPU capacity.

1.3. Goals

The main purpose of this research is to introduce a powerful technique that: 1) is faster than the proposed traditional shift estimation algorithms, 2) gives high estimation accuracy in the presence of noise, 3) performs frame-to-frame shift estimation when both translation and rotation occur, 4) reduces the computational complexity of the video stabilization process and 5) opens the possibility to implement a real-time stabilization system into small camera devices such as those associated with cellphones.

1.4. Related work

Although many algorithms have been proposed for digital video stabilization, there is still ongoing research because of the topic's importance and its massive need in many applications. Previous work in the literature of video motion estimation can be mostly grouped into block-based and feature-based algorithms [3]. In this section, we will shed light on some examples of the recently developed 2D stabilization methods that are still subject to future research.

An example of a block-based algorithm for video stabilization is described in Battiato *et al*'s work [4]. The idea here is to select 2D blocks from a pair of consecutive frames and generate the local motion estimation vectors. Depending on the estimation accuracy, an accept-reject based rule as well as a memory filtering process, retrieve the most significant values of the local motion vectors. Then, a correction and validation process for the shift estimation vectors is performed through some least-squares iteration and matrix calculation for error evaluation. Another block-based algorithm in Fei Yu *et al*' method [5], which consists of using 45° oblique vectors and a matching function between a chosen block from the referenced image and another from the current image. This helps in locating the information in the vertical and horizontal directions, and provide accurate shift estimations. Most of the block-based methods achieve precise results, but they often implicate excessive computations due to the high numbers of 2D blocks that the hardware needs to process [3].

Feature based algorithms rely on extracting and matching meaningful features between each pair of adjacent frames [3]. Kim *et al* [6] proposed a method that depends fundamentally on Kanade-Lucas-Tomasi (KLT) method which was described in [7], to separate the background and foreground in an image into two sets of feature points. Using the KLT method, we evaluate the transform function that maps the referenced image to the next adjacent image, which in turn gives estimates for the shifts between the pair of successive frames. Although feature based approaches have gained lots of popularity within the community of digital video processing for their high efficiency, they are still hard to implement in limited hardware devices. This is because of their high sensitivity to the image content [4]. Moreover, feature-based shift estimation algorithms are time consuming. They take 70% ~ 80% of the run-time of the video stabilization process [8].

Another state-of-the-art technique was proposed by Grundmann *et al* [9] which Youtube uses to stabilize low quality and casually taken videos. While this technique produces outstanding results, its computation cost makes its real-time implementation in hand-held devices challenging. The technique tracks the camera path (on x and y axes) at each frame and then apply the L1-norm minimization algorithm to generate the optimized camera path. Even though the mentioned algorithm is not feature-based in its core, in some cases it applies some feature detection within certain frames in the video to optimize the output. It also, removes the blurring from the output video using the Matsushita *et al* approach [10].

1.5. Contribution

This work is built upon a digital signal processing based technique that was proposed by Cain *et al* [11], to enable an accelerated and efficient translational shift estimation between a sequence of frames. Our contribution with this work lies on the expansion of the projection-based estimator's application to the rotational motion, to ensure a smooth and fast video stabilization system that combines simplicity and accuracy.

This in turn could be used in a myriad of purposes, such as surveillance cameras, unmanned aerial vehicles or a real-time system implementation into low resources hand-held devices.

1.6. Overview of Chapters

This thesis is organized as follows. In chapter-2, we state the theory behind the translational and rotational motion estimation, which is the pillar that this work stands on. In the first two sections, we provide a description of the projection-based method that was developed in [11] and the LIPSE algorithm that was described in [12], to measure the translational pixel and sub-pixel shifts. The second and third sections of chapter-2 are devoted to the theory of rotational motion estimation around known and unknown axes. Chapter-3 and chapter-4 discusses the experiment setups, as well as the results of applying the proposed projection-based method on synthetic and real data. We conclude this work by stating the future work that could be done to add improvements to the proposed rotation estimation system.

Chapter 2

Motion Estimation Theory

This chapter is devoted to the followed methodology for estimating the translational and rotational motions to achieve video stabilization. We first describe the projection-based estimator, proposed in [11], and the linear-interpolation-based sub-shift estimator (LIPSE), described in [12], to estimate the horizontal and vertical shifts. Then, we present our projection-based approach to address the rotation estimation problem in both situations where the center of the image rotation is known or unknown.

2.1. Projection-Based Method

Consider a video v with a sequence of frames $f_1(x_1, y_1)$, $f_2(x_2, y_2)$, ...etc. Each frame represents a measurement of a scene *s* at time *t*, such that the time lapse between successive frames results only in a slight variation of the captured scene. As a result, a translational shift that occurs in the horizontal and vertical directions between a pair of successive frames is represented by the following equations:

$$\begin{cases} f_i(x, y) = s(x, y) \\ f_{i+1}(x, y) = s(x - Hs, y - Vs) \end{cases}$$

Where *Hs* and *Vs* are respectively the horizontal and vertical shifts. Using the projection-based estimator to measure *Hs* and *Vs*, we first reduce each frame into row and column vectors. The row and column vectors are respectively the sum of all pixel values of an image along its vertical and horizontal directions. This is defined for an image of size $N \times N$

by:

$$f_i^H(y) = \sum_{x=1}^{N} f_i(x,y)$$

and

$$f_i^V(x) = \sum_{y=1}^{N} f_i(x,y)$$

We then perform a 1D cross-correlation operation between the row vectors and column vectors of the two subsequent frames to obtain:

$$P_y(z) = \sum_{x=1}^{N} f_i^y(x) f_{i+1}^y(x+z)$$

and

$$P_x(w) = \sum_{y=1}^{N} f_i^x(y) f_{i+1}^x(y+w)$$

Using the vectors of the cross-correlation functions, we compute:

$$\widehat{Hs} = argmax(P_y(z))$$

and

$$\widehat{Vs} = argmax(P_x(w))$$

Where $\widehat{Hs}$ and $\widehat{Vs}$ are respectively the horizontal and vertical shift estimations obtained with the projection-based method.

The projection-based method procedures for translational shift estimation are summarized in the following steps:

1) Compute the vertical and horizontal image projection vectors from each frame of a video sequence.
2) Obtain the vertical and horizontal 1-D cross-correlation vectors of each pair of successive frames from the computed projections in 1.
3) Find an estimation for the translational shift between a pair of successive frames along x and y axes using the argument of maximum value on the obtained 1-D cross-correlation vectors.
4) Repeat steps 1-3 for each pair of successive frames and then apply the corrections to compensate for the undesired shifts.

2.2. Linear-Interpolation-Based Sub-Pixel Shift Estimator

In order to estimate the translational sub-pixel shift using the linear-interpolation-based estimator [12], we consider a similar setting for a video v with a sequence of frames $f_1(x_1, y_1)$, $f_2(x_2, y_2)$,..., as described in the previous section. Briefly, the idea of the linear-interpolation-based sub-shift estimation algorithm is founded upon the concept of finding the mean-square-error (MSE) value, between the projected horizontal/vertical vectors P_i of a shifted image and the projected horizontal/vertical vectors $\hat{P}_i$ of the shifted frame estimate, using:

$$E_{(\Delta\alpha_i)}^{MSE} = \sum_{k=1}^{N} [P_i(k) - \hat{P}_i(k)]^2$$

At each iteration of the horizontal and vertical integer-based 1D correlation function between the projections of a frame and its estimate, we compute the MSE and obtain a vector of estimations for the sub-shift $\Delta\alpha_i$, using the formula:

$$\Delta\hat{a}_i = \frac{\sum_{k=1}^{N}(P_i(k)[P_{i-1}(k) - P_{i-1}(k+1)] + P_{i-1}(k)[P_{i-1}(k+1) - P_{i-1}(k)])}{\sum_{k=1}^{N}(P_{i-1}(k)^2 - 2P_{i-1}(k).P_{i-1}(k+1) + P_{i-1}(k+1)^2)}$$

The accurate estimation of the sub-shift $\Delta\alpha_i$ corresponds to the minimal mean-square-error (MMSE):

$$MMSE = \min\left(E_{(\Delta\alpha_i)}^{MSE}\right)$$

2.3. Rotation Estimation Around Known Coordinates

Although projection-based estimators (described in the first section) produce optimized translation estimation results, in some cases they fail to estimate the rotational shift between frames. This in turn, could reduce the reliability of the entire video stabilization process. Therefore, the use of a powerful and cost-effective algorithm to estimate the rotational motion is desired. A projection-based rotation estimation algorithm has been developed to achieve high stabilization performance in the presence of rotational shifts between video frames. Similarly to the projected-based translational shift estimator, our algorithm is built upon the idea of projecting each 2D frame into a row vector of size 1×180. Each column of the projected vector is computed by averaging all the pixel values along the direction that crosses the rotation center and forms an angle θ with the rows of the image, where $\theta \in [0,179]$. Using the 1D cross-correlation function, we find estimates for the rotational motion between a sequence of successive frames. In this section, we go with the assumption that the center of rotation between each pair of frames is given.

To begin our discussion of the projection-based rotation estimator, consider two image sequences $f_i(x_i, y_i)$, $f_{i+1}(x_{i+1}, y_{i+1})$ of a scene s, which were captured at time k and k+*1* respectively, and where a rotational motion around a specific point occurred between them. We assume that the angle of rotation is α and the rotation center coordinates (x_c, y_c) are known. Using the projection-based rotation estimation to compute $\hat{\alpha}$, we first crop each image into a square sub-image so that the rotation center coordinates become the center of the two cropped sub-images as shown in figure-1. The size of the two sub-images is similar and it is selected to encompass as much details on the images as possible. The crop operation is performed for the following purposes: 1) facilitate the extraction of the projections with square sub-images as it will be shown later in this section, 2) exclude the undesired information that might enter the scene because of the rotational shift, from the rotation estimation process, 3) reduce the computation cost by eliminating irrelevant processing.

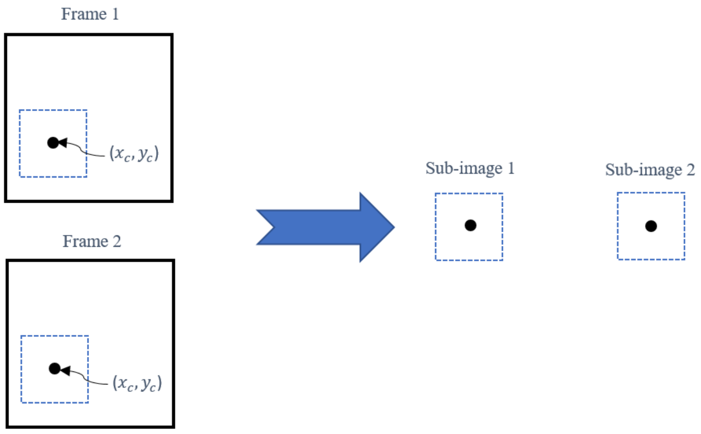


Figure 1: Square sub-images extraction from a pair of successive frames, such that the center of rotation is the center of the two sub-frames.

After we have obtained the square sub-images, we project each sub-frame into a row vector V_i of size 1×180, by averaging all the pixel values along the direction that crosses the rotation center and forms an angle θ with the rows of the sub-image, such that $\theta \in [0,179]$. This is done using the image rotation model [13]:

$$\begin{bmatrix} x_i' \\ y_j' \end{bmatrix} = \begin{bmatrix} \cos(\theta) & \sin(\theta) \\ -\sin(\theta) & \cos(\theta) \end{bmatrix} \cdot \begin{bmatrix} x_i \\ y_j \end{bmatrix}$$

To reduce the computation cost of the projection step from quadratic complexity O(n^2) to linear complexity O(n), we extract only the elements of the diagonal ($i = j$) at each iteration of $\theta \in [0,179]$.

From the computed projection vectors V_i and V_{i+1} of the pair of sub-frames, we compute the 1D cross-correlation vector by the rule

$$C(z) = \sum_{x=1}^{180} V_i(x) V_{i+1}(x+z)$$

The estimate $\hat{\alpha}$ of the rotation angle α is simply:

$$\hat{\alpha} = argmax(C(z))$$

Here, we summarize the projection-based rotation estimator procedures:

1) We crop each pair of frames into square sub-images so that the given rotation center coordinates become the center of the two cropped sub-frames.

2) we project each frame into a row vector V_i of size 1×180, by averaging all the pixel values along the direction that crosses the rotation center and forms an angle θ with the rows of the image, where θ is incremented in the interval $[0,179]$.

3) We compute the 1D cross-correlation vector between the computed projection vectors.

4) Estimate the rotation angle $\hat{\alpha}$ between a pair of frames using the argument of maximum value from the obtained 1-D cross-correlation vectors.

5) Repeat steps 1-4 for each pair of successive frames and then apply the corrections to fix the undesired rotations.

2.4. Rotation Estimation Around Unknown Coordinates

In a real-case video stabilization problem, it is hard to determine the center of the rotations that arise between a frame and its subsequent frames without a prior knowledge of the rotation angle or the coordinates of the displaced objects. Therefore, applying the projection-based rotation estimator algorithm with the assumption that the center of the image is the center of the occurring rotations, can sometimes lead to inaccurate estimations and cause the video stabilization system to fail. Figure-2 presents the effect of the above-mentioned problem on two distinct video sequences.

Table-1 shows some results of a study that we have done on a set of 1024×1024 images. On each image, we apply a rotation of an angle α of 4° and 8° respectively around a given point of coordinates (*x*,*y*) as shown in figure-3. Then, we assume that the center of the image is the center of the applied rotation and use the projection-based rotation algorithm to estimate the rotation angle $\hat{\alpha}$. Our study led to the following results:

	P$_c$ (513,513)		P$_1$ (684,342)		P$_2$ (650,575)		P$_3$ (200,600)	
	$\alpha = 4°$	$\alpha = 8°$	$\alpha = 4°$	$\alpha = 8°$	$\alpha = 4°$	$\alpha = 8°$	$\alpha = 4°$	$\alpha = 8°$
Frame 1	$\hat{\alpha} = 4°$	$\hat{\alpha} = 8°$	$\hat{\alpha} = 6°$	$\hat{\alpha} = 4°$	$\hat{\alpha} = 4°$	$\hat{\alpha} = 7°$	$\hat{\alpha} = 5°$	$\hat{\alpha} = 11°$
Frame 2	$\hat{\alpha} = 4°$	$\hat{\alpha} = 8°$	$\hat{\alpha} = 1°$	$\hat{\alpha} = 4°$	$\hat{\alpha} = 3°$	$\hat{\alpha} = 7°$	$\hat{\alpha} = 4°$	$\hat{\alpha} = 11°$
Frame 3	$\hat{\alpha} = 4°$	$\hat{\alpha} = 8°$	$\hat{\alpha} = 2°$	$\hat{\alpha} = 4°$	$\hat{\alpha} = 4°$	$\hat{\alpha} = 9°$	$\hat{\alpha} = 4°$	$\hat{\alpha} = 8°$
Frame 4	$\hat{\alpha} = 4°$	$\hat{\alpha} = 8°$	$\hat{\alpha} = 6°$	$\hat{\alpha} = 13°$	$\hat{\alpha} = 5°$	$\hat{\alpha} = 9°$	$\hat{\alpha} = 0°$	$\hat{\alpha} = -2°$

Table 1: Rotation estimation of the angle α when the rotation center coordinates are arbitrary and the projection-based rotation estimator is performed with the assumption that the center of the images is the center of rotation.

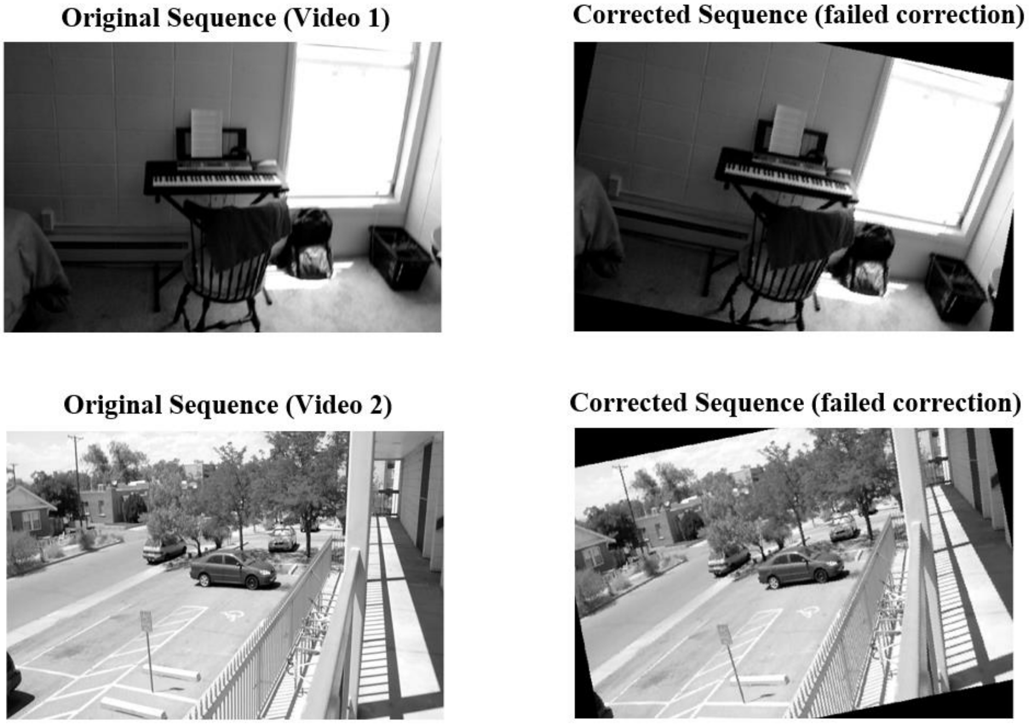

Original Sequence (Video 1)
Corrected Sequence (failed correction)
Original Sequence (Video 2)
Corrected Sequence (failed correction)

Figure 2: Failed rotation estimation using the projection-based rotation estimator under the assumption that the center of the images is the center of rotation.

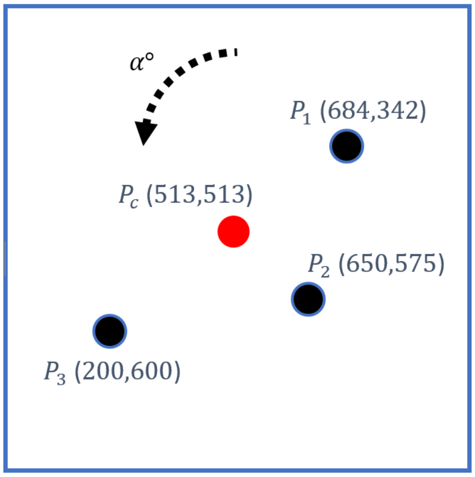


Figure 3: Simulating image rotations around four different rotation centers with an angle α of 4° and 8°.

From table-1, we notice that the most accurate estimations, are the ones that are performed in directions that cross the center of rotation (P_c in our case). In the example of *frame 4*, estimating the rotations of an angle $\alpha = 8°$ around P_1 and P_3 by performing the projections in the directions that cross P_c, led to undesired results. We conclude from the presented study that the estimation accuracy of the rotation angle using the projection-based rotation estimation algorithm, depends on determining the exact rotation center coordinates, which is a challenging task in a real-case rotation estimation problem.

To prevent inaccurate estimations from arising in a real-case video stabilization problem, we present another study that we have conducted on the previously-used set of 1024×1024 images. The study consists of extracting more than a sub-image from the original frame and its rotated form, compute the projections for each extracted pair of sub-images to estimate the rotation angle, and then obtain the global estimated rotation angle $\hat{\alpha}$ between a pair of frames by averaging all the measured estimations. In our study, we have considered rotations around three distinct rotation centers $P_i(x,y)$.

To increase the accuracy of the estimations, we choose a proper sub-image size to encompass sufficient image details. Figure-4 depicts some of the possible ways of extracting sub-images from an image. In our study, we extracted sub-images in three different ways that we refer to as the “X”, “+” and “*” distributions of sub-images. Table 2, 3 and 4 show the results of our study with rotation angles α equal to 5°, 10° and 15° respectively.

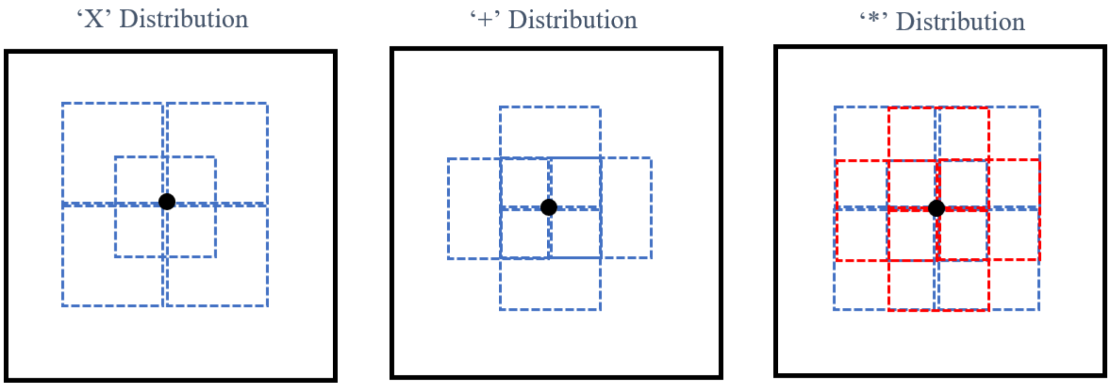


Figure 4: Three possible ways of extracting sub-images from a frame to estimate the rotation angle using the projection-based rotation estimation method.

	P1 (513,513)			P2 (400,450)			P3 (650,575)		
	$\alpha = 5^\circ$	$\alpha = 10^\circ$	$\alpha = 15^\circ$	$\alpha = 5^\circ$	$\alpha = 10^\circ$	$\alpha = 15^\circ$	$\alpha = 5^\circ$	$\alpha = 10^\circ$	$\alpha = 15^\circ$
Frame 1	$\hat{\alpha} = 1^\circ$	$\hat{\alpha} = 2^\circ$	$\hat{\alpha} = 28^\circ$	$\hat{\alpha} = 1^\circ$	$\hat{\alpha} = 1^\circ$	$\hat{\alpha} = 29^\circ$	$\hat{\alpha} = 0^\circ$	$\hat{\alpha} = 1^\circ$	$\hat{\alpha} = 3^\circ$
Frame 2	$\hat{\alpha} = 4^\circ$	$\hat{\alpha} = 12^\circ$	$\hat{\alpha} = 20^\circ$	$\hat{\alpha} = 3^\circ$	$\hat{\alpha} = 9^\circ$	$\hat{\alpha} = 5^\circ$	$\hat{\alpha} = 4^\circ$	$\hat{\alpha} = 14^\circ$	$\hat{\alpha} = 21^\circ$
Frame 3	$\hat{\alpha} = 3^\circ$	$\hat{\alpha} = 8^\circ$	$\hat{\alpha} = 8^\circ$	$\hat{\alpha} = 2^\circ$	$\hat{\alpha} = 5^\circ$	$\hat{\alpha} = 12^\circ$	$\hat{\alpha} = 5^\circ$	$\hat{\alpha} = 6^\circ$	$\hat{\alpha} = 9^\circ$
Frame 4	$\hat{\alpha} = 3^\circ$	$\hat{\alpha} = 8^\circ$	$\hat{\alpha} = 8^\circ$	$\hat{\alpha} = 4^\circ$	$\hat{\alpha} = 9^\circ$	$\hat{\alpha} = 10^\circ$	$\hat{\alpha} = 2^\circ$	$\hat{\alpha} = 1^\circ$	$\hat{\alpha} = -2^\circ$

Table 2: Rotation angle estimation following the ‘**X**’ Distribution of extracted sub-images.

	P_1 (513,513)			P_2 (400,450)			P_3 (650,575)		
	$\alpha = 5^\circ$	$\alpha = 10^\circ$	$\alpha = 15^\circ$	$\alpha = 5^\circ$	$\alpha = 10^\circ$	$\alpha = 15^\circ$	$\alpha = 5^\circ$	$\alpha = 10^\circ$	$\alpha = 15^\circ$
Frame 1	$\hat{\alpha} = 2^\circ$	$\hat{\alpha} = 6^\circ$	$\hat{\alpha} = -6^\circ$	$\hat{\alpha} = 4^\circ$	$\hat{\alpha} = 10^\circ$	$\hat{\alpha} = -3^\circ$	$\hat{\alpha} = 1^\circ$	$\hat{\alpha} = 3^\circ$	$\hat{\alpha} = 6^\circ$
Frame 2	$\hat{\alpha} = 3^\circ$	$\hat{\alpha} = 9^\circ$	$\hat{\alpha} = 11^\circ$	$\hat{\alpha} = 4^\circ$	$\hat{\alpha} = 8^\circ$	$\hat{\alpha} = 11^\circ$	$\hat{\alpha} = 2^\circ$	$\hat{\alpha} = 2^\circ$	$\hat{\alpha} = 16^\circ$
Frame 3	$\hat{\alpha} = 3^\circ$	$\hat{\alpha} = 5^\circ$	$\hat{\alpha} = 10^\circ$	$\hat{\alpha} = 1^\circ$	$\hat{\alpha} = 4^\circ$	$\hat{\alpha} = 2^\circ$	$\hat{\alpha} = 4^\circ$	$\hat{\alpha} = 9^\circ$	$\hat{\alpha} = 5^\circ$
Frame 4	$\hat{\alpha} = 3^\circ$	$\hat{\alpha} = 7^\circ$	$\hat{\alpha} = 7^\circ$	$\hat{\alpha} = 2^\circ$	$\hat{\alpha} = 7^\circ$	$\hat{\alpha} = 5^\circ$	$\hat{\alpha} = 3^\circ$	$\hat{\alpha} = 10^\circ$	$\hat{\alpha} = 14^\circ$

Table 3: Rotation angle estimation following the ‘+’ Distribution of extracted sub-images.

	P_1 (513,513)			P_2 (400,450)			P_3 (650,575)		
	$\alpha = 5^\circ$	$\alpha = 10^\circ$	$\alpha = 15^\circ$	$\alpha = 5^\circ$	$\alpha = 10^\circ$	$\alpha = 15^\circ$	$\alpha = 5^\circ$	$\alpha = 10^\circ$	$\alpha = 15^\circ$
Frame 1	$\hat{\alpha} = 1^\circ$	$\hat{\alpha} = 3^\circ$	$\hat{\alpha} = 11^\circ$	$\hat{\alpha} = 2^\circ$	$\hat{\alpha} = 5^\circ$	$\hat{\alpha} = 13^\circ$	$\hat{\alpha} = 1^\circ$	$\hat{\alpha} = 1^\circ$	$\hat{\alpha} = 3^\circ$
Frame 2	$\hat{\alpha} = 3^\circ$	$\hat{\alpha} = 7^\circ$	$\hat{\alpha} = 16^\circ$	$\hat{\alpha} = 3^\circ$	$\hat{\alpha} = 9^\circ$	$\hat{\alpha} = 8^\circ$	$\hat{\alpha} = 3^\circ$	$\hat{\alpha} = 8^\circ$	$\hat{\alpha} = 20^\circ$
Frame 3	$\hat{\alpha} = 3^\circ$	$\hat{\alpha} = 3^\circ$	$\hat{\alpha} = 10^\circ$	$\hat{\alpha} = 2^\circ$	$\hat{\alpha} = 5^\circ$	$\hat{\alpha} = 7^\circ$	$\hat{\alpha} = 4^\circ$	$\hat{\alpha} = 8^\circ$	$\hat{\alpha} = 8^\circ$
Frame 4	$\hat{\alpha} = 3^\circ$	$\hat{\alpha} = 3^\circ$	$\hat{\alpha} = 8^\circ$	$\hat{\alpha} = 3^\circ$	$\hat{\alpha} = 8^\circ$	$\hat{\alpha} = 7^\circ$	$\hat{\alpha} = 2^\circ$	$\hat{\alpha} = 4^\circ$	$\hat{\alpha} = 5^\circ$

Table 4: Rotation angle estimation following the ‘*’ Distribution of extracted sub-images.

From the presented tables, we notice that estimating the rotation angles based on sub-images extraction, reduces the accuracy of the estimation. However, in the presence of small rotations ($\leq 10°$), this approach eliminates the risk of inaccurate estimations that might lead the video stabilization system to fail due to the unknown rotation center.

Choosing the proper sizes, distributions and numbers of the extracted sub-images from an image is subject to optimization. In This thesis, we are not going beyond the three distributions that we have presented previously.

Chapter 3

Experimentation

To conduct a subjective assessment of the proposed projection-based method for rotation estimation and achieve the goal of this thesis, a Matlab code was written to estimate and correct for the translational as well as rotational motions between each pair of successive frames in a shaky video sequence. Our data was collected using a Nikon DSLR camera and contains videos with different vibration levels. In this chapter, we will be describing the conducted experiments that lead to the results that we will display in the following chapter.

3.1. Experiment Setup

3.1.1 Experiment 1

In the first experiment, we simulate 60 arbitrary image rotations around the center of a 424×640 image, by generating 60 random rotation angles in the interval of [0,10]°. The rotated image and the generated synthetic rotation angles are presented in figure-5. To estimate the rotation angles in our experiment, we use the two projected-based rotation estimation approaches that we discussed in the third and fourth parts of chapter-2. We will refer to them as method-1 and method-2 respectively, throughout all the current and the next chapter. In order to discuss the performance of each method, we first compare the estimation results with the generated synthetic rotation angles, and then, relying on the plotted mean-squared-error (MSE) graphs, we establish a comparison between both methods.

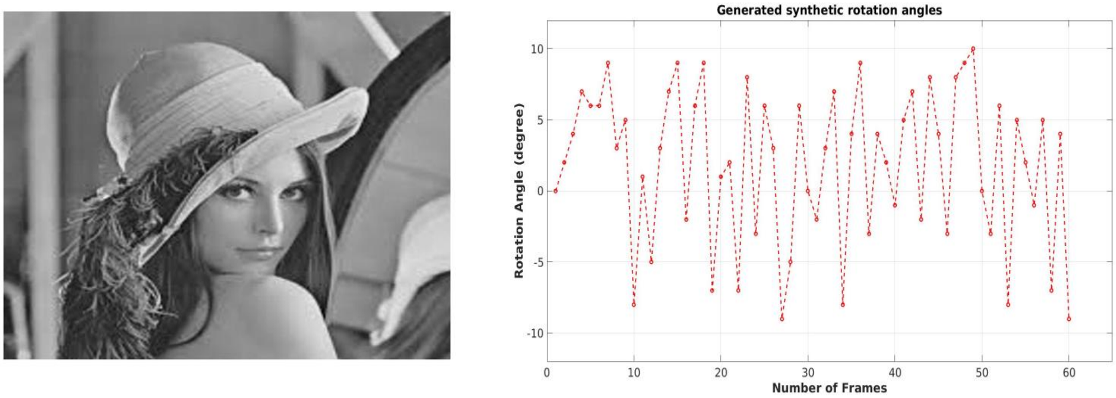


Figure 5: The original image has been rotated 60 times with the randomly generated angles that are displayed in the graph.

3.1.2. Experiment 2

The second experiment is similar to the previous one. Instead of using synthetic data, we apply the same proposed methods on a real-case 25 frames-per-second video of 424×640 resolution, which suffers from undesired rotations. The video contains 260 frames and was captured to replicate the past experience-i.e. the center of the occurring rotations is a short distance away from the center of the frames.

Video 1: A 25 frames-per-second shaky video of 424×640 resolution, which suffers from undesired rotations around coordinates that are a short distance away from center of the frames.

3.1.3. Experiment 3

This experiment features 60 simulated random rotations of a 424×640 gray scale image that contains less details than the one used previously. We have used the same generated rotation angles from the first experiment (figure-5) to perform image rotations around arbitrary *(x,y)* coordinates of the image. We then, apply the two above-mentioned approaches to plot the estimated angles as well as the mean-squared-error graphs and compare the efficiency of both methods in the presented scenario.

3.1.4. Experiment 4

This experiment considers two real-case 25 frames-per-second videos of 424×640 resolution, which suffer from random translational and rotational motions. Both videos capture sufficient details. For each pair of frames, we apply the translational shift and sub-shift estimation algorithms that we described in chapter-2, to reduce shaking. Then, using the two above-mentioned methods, we correct for rotational motions and establish comparisons between the performance of the two methods on the studied videos. In this experiment, we performed method-2 by extracting sub-images following the "+" distribution.

Video 2: A 25 frames-per-second shaky video of 424×640 resolution, which suffers from undesired translational and rotational motions.

Video 3: A 25 frames-per-second shaky video of 424×640 resolution, which suffers from severe translational and rotational motions.

Chapter 4

Results and Discussion

In this chapter, we conclude our work with presenting the results of the different experiments that we have realized in the previous chapter.

4.1. Experiment 1

In this section, we present the results that we obtained from the first experiment. applying the projection-based method between each pair of successive sequences, as described in the third part of the chapter-2, we obtained the following graphs:

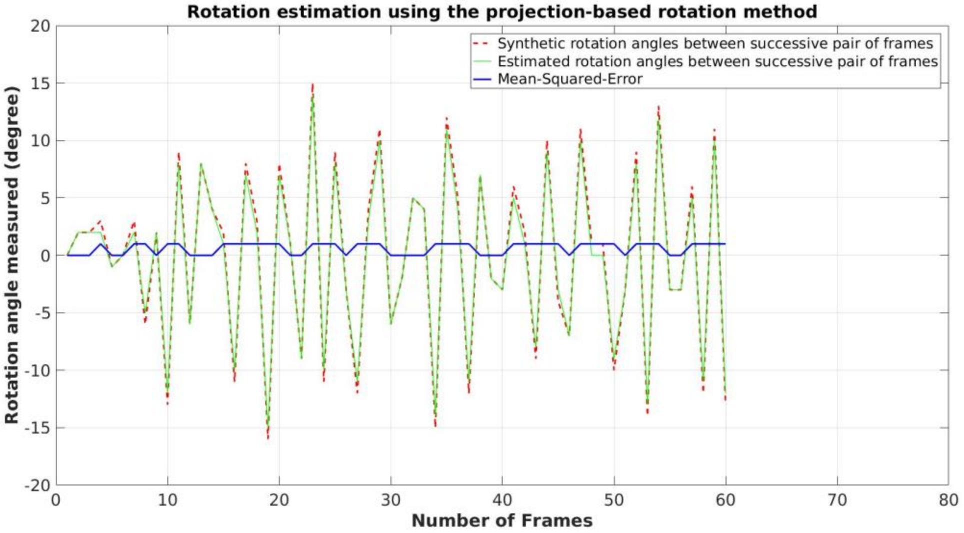


Figure 6: Comparison between the synthetic and estimated rotation angles using method-1.

The graphs that we obtained from applying method-2 between a pair of successive frames, are presented in figure-7. In our experiment, we followed the "+" distribution of sub-images extraction.

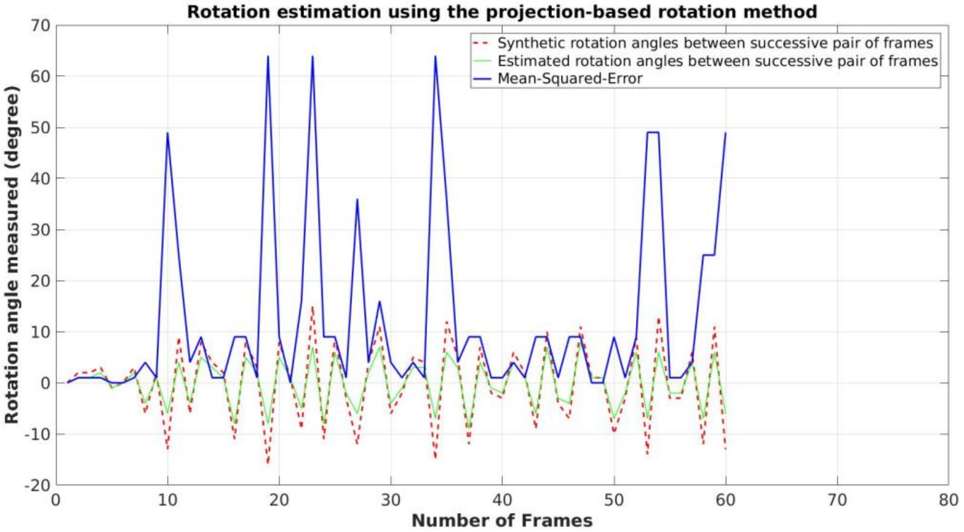


Figure 7: Comparison between the synthetic and estimated rotation angles using method-2.

In order to establish a fair comparison between the two followed approaches in our experiment, we considered the mean-squared-error graph as our criterion. From the obtained graphs, we can conclude that using projections of line-segments that cross the center of rotation, when the last-mentioned is known, leads to more accurate rotation estimations and outperforms the second method, which relies on sub-images extraction. This can be observed from the mean-squared-error graphs that are shown below.

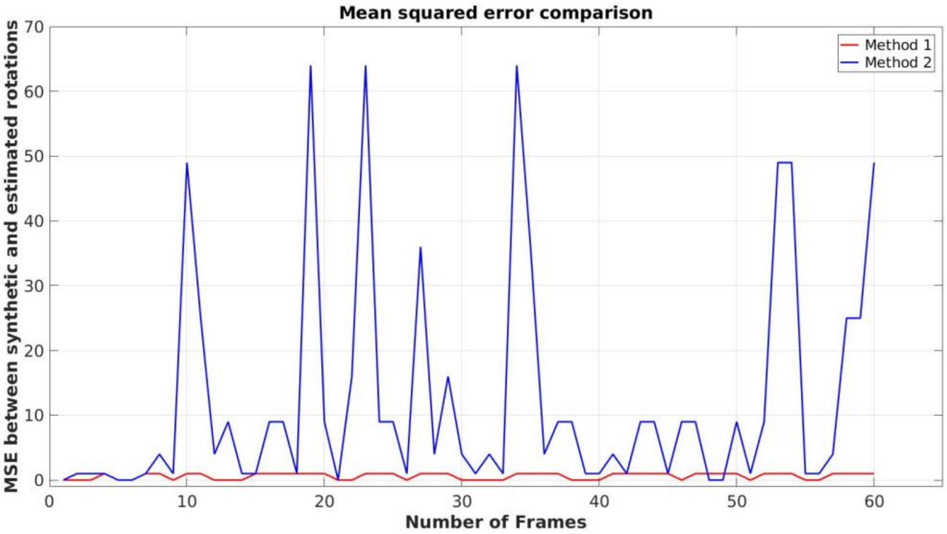


Figure 8: Comparison between method-1 and method-2, using the MSE graphs.

4.2. Experiment 2

After applying both methods on a real-case video that simulates undesired rotations around the center of the frames. We obtained the estimation graphs below. Although the first method performed better in this scenario, the second approach gave satisfying results as well, and reduced most of the undesired rotations from the video. Video 4 and 5 show the corrected versions of the input video using method 1 and 2 respectively.

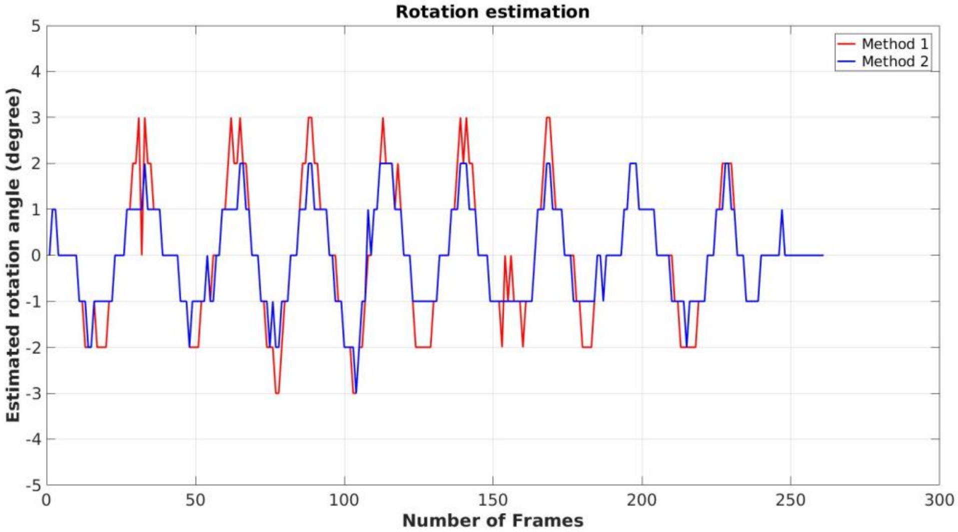


Figure 9: Comparison between method-1 and method-2 on a real-case video that suffers from undesired rotations around the center of the frames.

Video 4: correcting the input video from the undesired rotations using method-1.

Video 5: correcting the input video from the undesired rotations using method-2.

4.3. Experiment 3

Figure-10 and figure-11 show respectively, the rotation estimation and the mean squared error graphs that we have obtained from the conducted synthetic experiment. Although, the first method led to accurate estimations in the previous experiments, in some situations where the video sequences do not contain sufficient details, the method doesn’t achieve good performance. In the presented scenario, the first method failed to provide accurate estimations for some of the generated rotation angles, and caused unsuccessful stabilization process. The second method misestimated certain angles, but the overall stabilization process was uniform and satisfactory. This could be observed from the spikes in the generated mean-squared-error graphs below for the 10th, 18th ,34th and 36th frames.

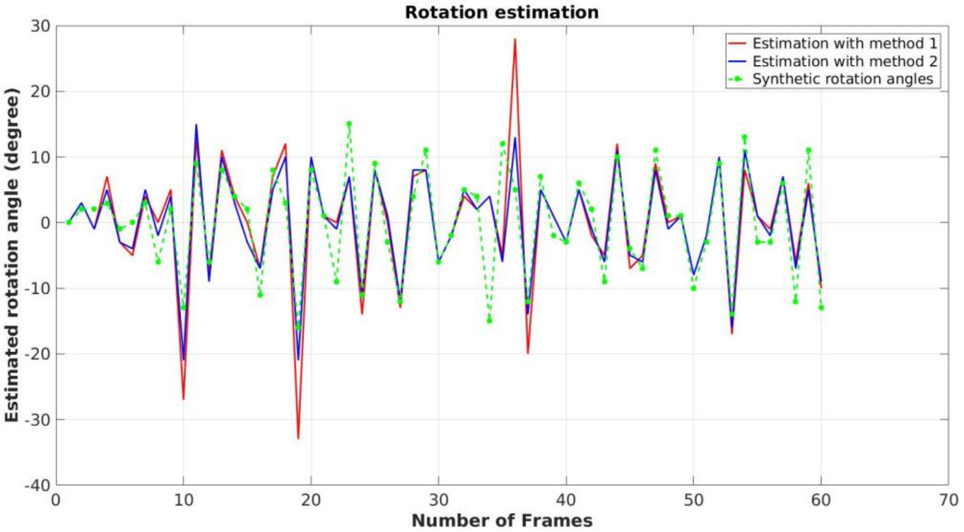


Figure 10: Comparison between the synthetic and estimated rotation angles, using method-1 and method-2 when the center of the occurring rotations is arbitrary.

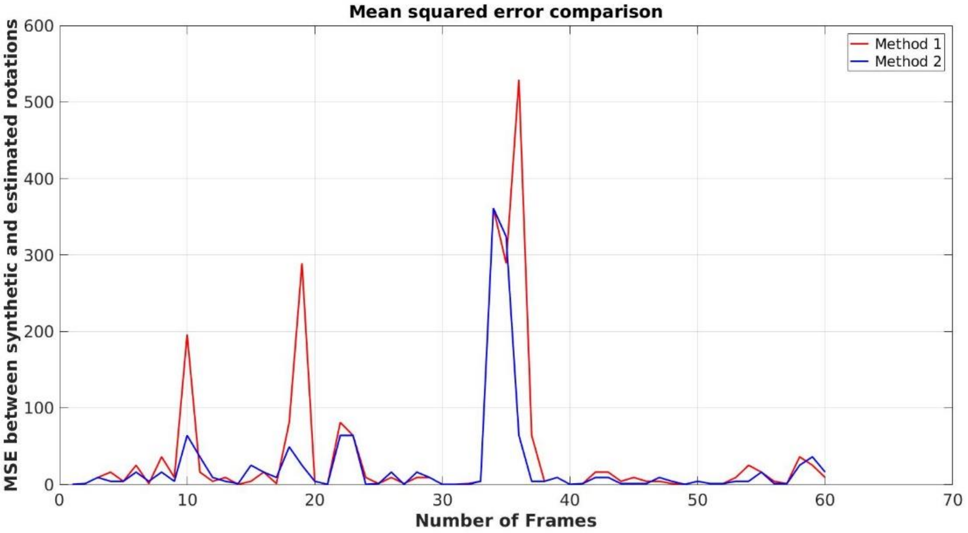


Figure 11: Comparison between method-1 and method-2 using the MSE graphs, when the center of the occurring rotations is arbitrary.

4.4. Experiment 4

- Video 1

The first video that we have used in our experiment contains 208 frames, and suffers from translational shifts as well as rotational motions around arbitrary coordinates. To tackle each problem independently, we first measured and compensated for translational pixel and sub-pixel shifts, and then, we estimate and correct for the undesired rotational motions. This approach achieves better rotation estimation results. The graphs below show respectively the estimated pixel and sub-pixel shifts. Video-6 shows the corrected video after compensating for the translational shifts.

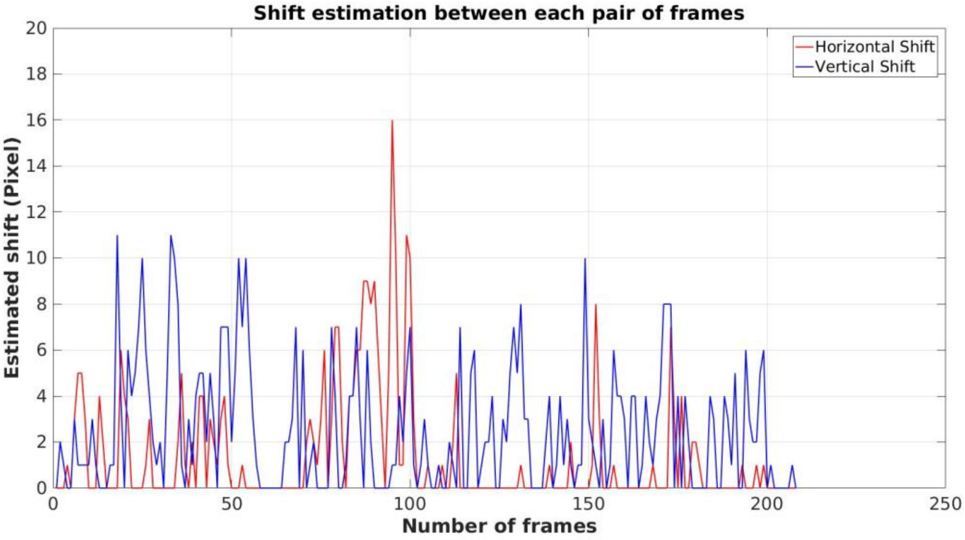


Figure 12: Horizontal and Vertical translational shift estimation in Video-1.

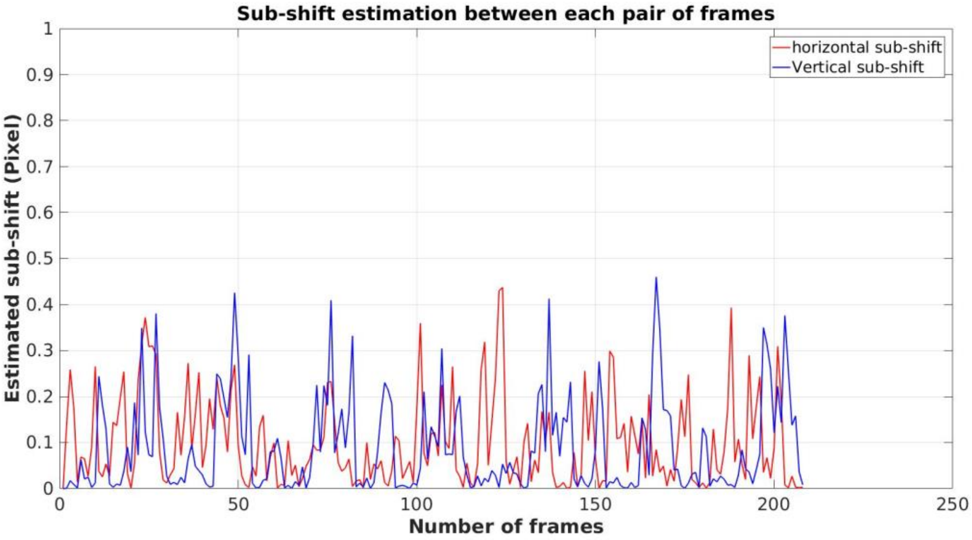


Figure 13: Horizontal and Vertical translational sub-shift estimation in Video-1.

Video 6: correcting the input video from the undesired translational shifts (half-correction).

After we correct for the translational shifts, we apply method-1 and method-2 to correct the rotational motion and compare between the outputted results of both methods. The estimated angles as well as the corrected videos are displayed in the graphs below.

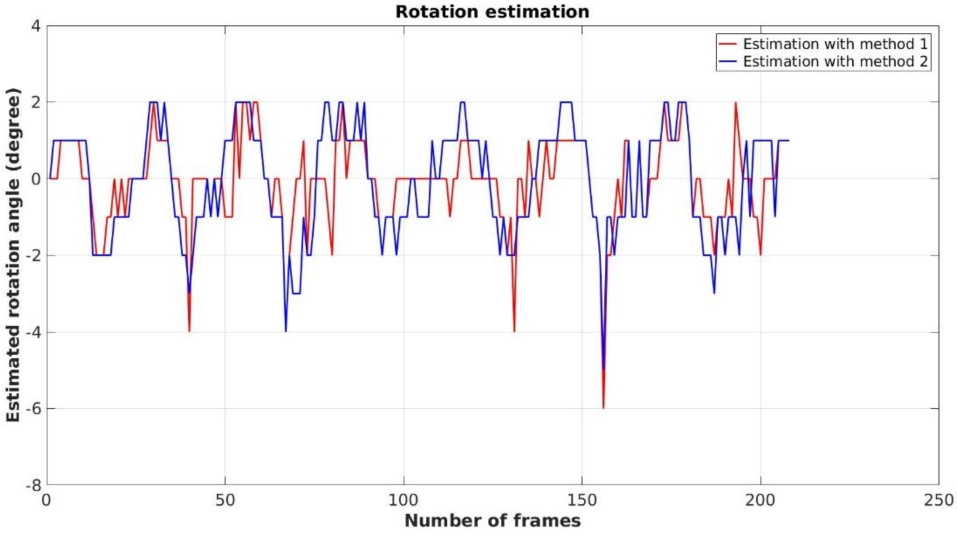
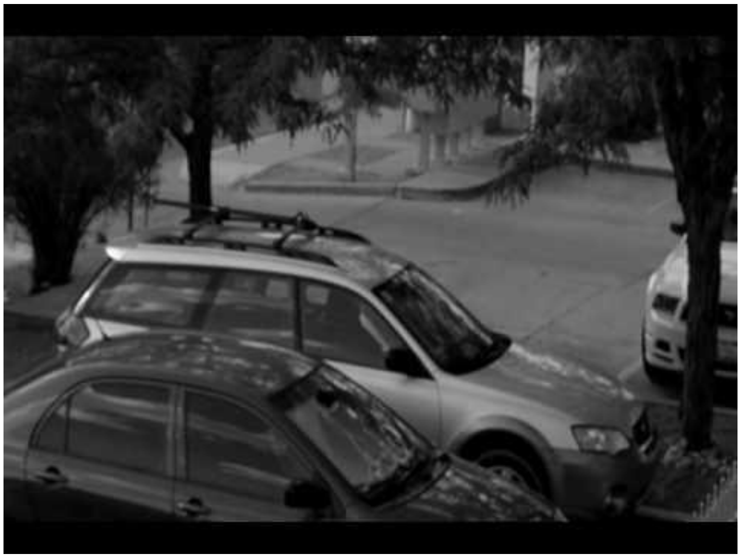


Figure 14: Rotational motion estimation using method-1 and method-2 on video-1.



Video 7: correcting the half-corrected input video from the undesired rotations using method-1.

Video 8: correcting the half-corrected input video from the undesired rotations using method-2.

Based on a visual observation of the outputted video, the first method failed again to provide a uniform stabilization process of the presented real-case unstable video. The rotation estimations between certain pair of frames using method-1, were inaccurate. This can be perceived from the graph above, between the pair of frames (155,156) and (184,185). Method-2 resulted in accurate estimation results and outputted an adequately corrected video.

- Video 2

Similarly to video-1, we apply the same procedures on a video that contains 367 frames and which suffers from severe vibrations. The estimation results that we obtained are presented in figure-15, 16 and 17. These are also joined by the output videos using method-1 and method-2.

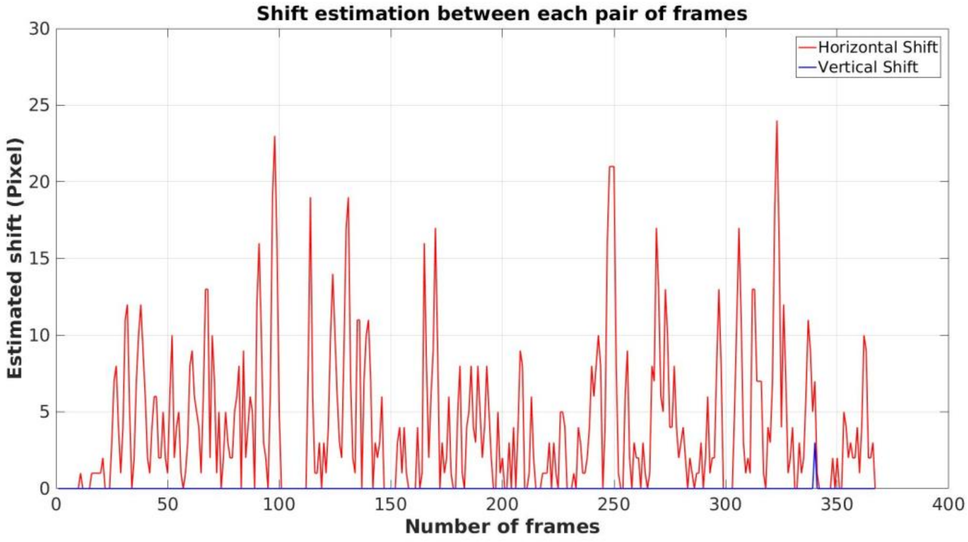


Figure 15: Horizontal and Vertical translational shift estimation in Video-2.

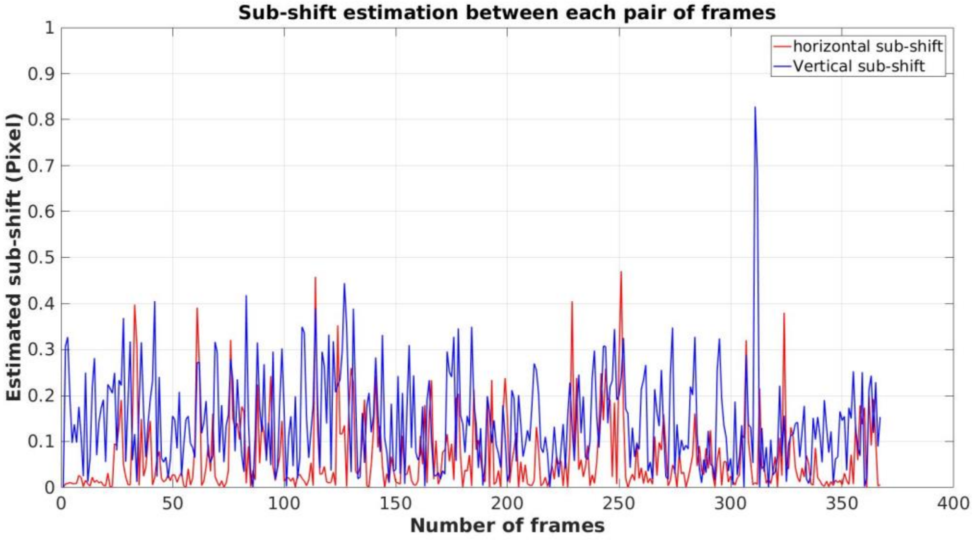


Figure 16: Horizontal and Vertical translational sub-shift estimation in Video-2.

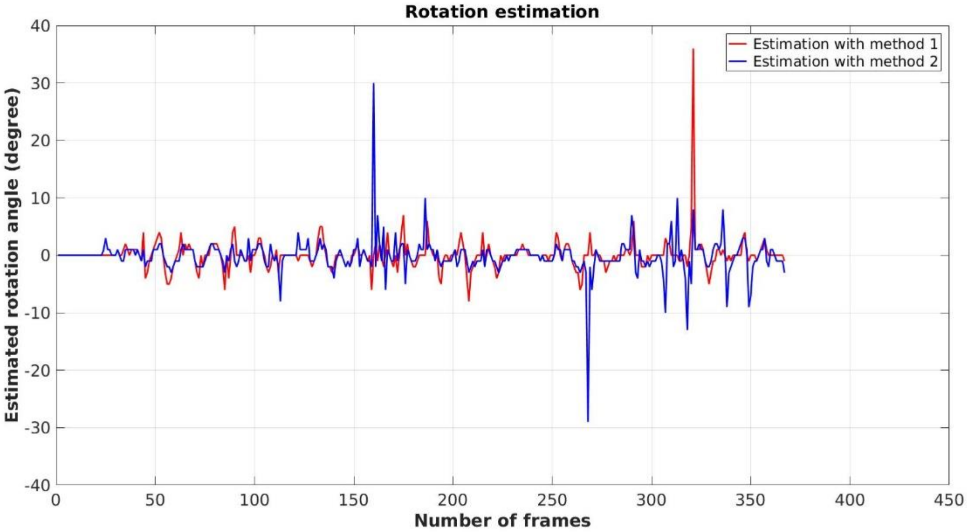
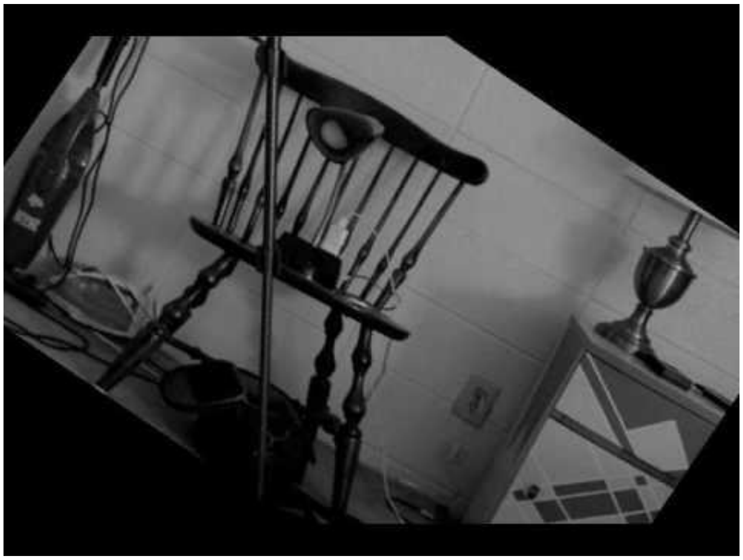


Figure 17: Rotational motion estimation using method-1 and method-2 on video-2.



Video 9: Failed rotation correction using method-1.

Video 10: Failed rotation correction using method-2.

In this scenario, we demonstrate the limitations of the projection-based estimators with an extremely unstable video. Although the methods that we have proposed in this thesis to estimate translational and rotational motions reduced the undesired shifts significantly, they failed to detect and correct for all the shaky scenes. Figure-16 shows the unsuccessful detection of the occurred horizontal shifts, using the projection-based estimator that we described in the first part of chaper-2, and form the peaks on the graphs in figure-18, we can observe the inaccurate angle estimations that we obtained by using the projection-based rotation estimator. Both method-1 and method-2 failed to provide accurate estimations for all the moving scenes. This video stabilization failure is caused mostly because of the severe displacements of the video frames while recording, and the insufficient details that enter the scene, which the projection-based estimators rely on to detect changes and evaluate the shifts correctly.

4.5. Conclusion and Future Work

In this thesis, we have proposed an effective approach that reduces the computation costs significantly, by projecting 2-D frames into 1-D vectors of size 1×180 each. Using the 1-D cross-correlation between each pair of successive projected frames, the projection-based rotation estimator achieves correct rotation estimations. The experiments on real-case unstable videos that we have conducted, demonstrated the robustness of the projection-based rotation estimator against rotations that occur around arbitrary coordinates along the video frames. Future work with the projection-based rotation estimator will involve testing the technique's efficiency in the presence of randomly distributed noise, developing a robust algorithm that determines the location of the center of rotation between each pair of successive frames, exploring deeply the sub-images extraction approach, to find a good tradeoff between accuracy and computational complexity.

References

[1] Thillainayagi, R., and Senthil Kumar. "Video stabilization technique for thermal infrared Aerial surveillance." Green Engineering and Technologies (IC-GET), 2016 Online International Conference on. IEEE, 2016.

[2] Ahmed, Abdelrahman, and Mohamed S. Shehata. "StableFlow: A novel real-time method for digital video stabilization." Pattern Recognition (ICPR), 2016 23rd International Conference on. IEEE, 2016.

[3] Li, Jianan, Tingfa Xu, and Kun Zhang. "Real-time feature-based video stabilization on FPGA." IEEE Transactions on Circuits and Systems for Video Technology 27.4 (2017): 907-919.

[4] Battiato, Sebastiano, Arcangelo Ranieri Bruna, and Giovanni Puglisi. "A robust Block-based image/video registration approach for mobile imaging devices." IEEE Transactions on Multimedia 12.7 (2010): 622-635.

[5] Yu, Fei, et al. "The application of improved block-matching method and block search method for the image motion estimation." Optics Communications 283.23 (2010): 4619-4625.

[6] Kim, Seung-Kyun, et al. "Feature point classification based global motion estimation for video stabilization." IEEE Transactions on Consumer Electronics 59.1 (2013): 267-272.

[7] J. Shi and C. Tomasi. "Good features to track." Computer Vision and Pattern Recognition, 1994. Proceedings CVPR'94., 1994 IEEE Computer Society Conference on. IEEE, 1994.

[8] Liu, Shuaicheng, et al. "CodingFlow: Enable Video Coding for Video Stabilization." IEEE Transactions on Image Processing 26.7 (2017): 3291-3302.

[9] Grundmann, Matthias, Vivek Kwatra, and Irfan Essa. "Auto-directed video stabilization with robust l1 optimal camera paths." Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on. IEEE, 2011.

[10] Y. Matsushita, E. Ofek, Weina Ge, Xiaoou Tang and Heung-Yeung Shum. "Full-frame video stabilization with motion inpainting." IEEE Transactions on Pattern Analysis and Machine Intelligence 28.7 (2006): 1150-1163.

[11] Cain, Stephen C., Majeed M. Hayat, and Ernest E. Armstrong. "Projection-based image registration in the presence of fixed-pattern noise." IEEE transactions on image processing 10.12 (2001): 1860-1872.

[12] Ratliff, Bradley Michael. "A generalized algebraic scene-based nonuniformity correction algorithm for infrared focal plane arrays." University of New Mexico, Albuquerque, NM (2004).

[13] Shuang, Zhang, Jin Gang, and Qin Yu-ping. "DSP-based parallel processing model of image rotation." Procedia Engineering 15 (2011): 2222-2228.

[14] De Castro, E., and C. Morandi. "Registration of translated and rotated images using finite Fourier transforms." IEEE Transactions on pattern analysis and machine intelligence 5 (1987): 700-703.

[15] Flusser, Jan, and Barbara Zitová. "Combined invariants to linear filtering and rotation." International Journal of Pattern Recognition and Artificial Intelligence13.08 (1999): 1123-1135.

[16] Govindu, V., Shekhar, C., & Chellappa, R. (1998, August). Using geometric properties for correspondence-less image alignment. In Pattern Recognition, 1998. Proceedings. Fourteenth International Conference on (Vol. 1, pp. 37-41). IEEE.

[17] Hsieh, Jun-Wei, et al. "Image registration using a new edge-based approach." Computer vision and image understanding 67.2 (1997): 112-130.

[18] Huttenlocher, Daniel P., Gregory A. Klanderman, and William J. Rucklidge. "Comparing images using the Hausdorff distance." IEEE Transactions on pattern analysis and machine intelligence 15.9 (1993): 850-863.

[19] Wang, Cheng-Ye, et al. "Some experiments in relaxation image matching using corner features." Pattern Recognition 16.2 (1983): 167-182.

[20] Zheng, Qinfen, and Rama Chellappa. "A computational vision approach to image registration." IEEE Transactions on Image Processing 2.3 (1993): 311-326.