ISTANBUL TECHNICAL UNIVERSITY ★ GRADUATE SCHOOL OF SCIENCE ENGINEERING AND TECHNOLOGY

DIGITAL VIDEO STABILIZATION WITH SIFT FLOW

M.Sc. THESIS

İnci Meliha BAYTAŞ

Electronics and Communication Engineering Department

Telecommunication Engineering Program

MAY 2014

ISTANBUL TECHNICAL UNIVERSITY ★ GRADUATE SCHOOL OF SCIENCE ENGINEERING AND TECHNOLOGY

DIGITAL VIDEO STABILIZATION WITH SIFT FLOW

M.Sc. THESIS

İnci Meliha BAYTAŞ
(504121319)

Electronics and Communication Engineering Department

Telecommunication Engineering Program

Thesis Advisor: Prof. Dr. Melih PAZARCI

MAY 2014

İSTANBUL TEKNİK ÜNİVERSİTESİ ★ FEN BİLİMLERİ ENSTİTÜSÜ

SIFT AKIŞI İLE SAYISAL VİDEO SABİTLEME

YÜKSEK LİSANS TEZİ

İnci Meliha BAYTAŞ
(504121319)

Elektronik ve Haberleşme Mühendisliği Anabilim Dalı

Telekomünikasyon Mühendisliği Programı

Tez Danışmanı: Prof. Dr. Melih PAZARCI

MAYIS 2014

İnci Meliha BAYTAŞ, a M.Sc. student of ITU Graduate School of Science Engineering and Technology 504121319 successfully defended the thesis entitled **"DIGITAL VIDEO STABILIZATION WITH SIFT FLOW"**, which he/she prepared after fulfilling the requirements specified in the associated legislations, before the jury whose signatures are below.

Thesis Advisor :	**Prof. Dr. Melih PAZARCI** Istanbul Technical University	.....
Jury Members :	**Prof. Dr. Bilge GÜNSEL** Istanbul Technical University	.....
	Asst. Prof. Dr. Koray KAYABOL Gebze Institute of Technology	.....

Date of Submission : **05 May 2014**
Date of Defense : **29 May 2014**

To my beloved family,

FOREWORD

I would like to thank my advisor Prof. Dr. Melih PAZARCI for giving me valuable support during this thesis. His advices always broaden my perspective to solve problems. I would like to thank the valuable academic staff who contributed to my graduate education in Electronics and Communication Engineering Department. I would like to give my special thanks to my precious parents for their eternal love, support and guidance. I appreciate the scholarship of the National Scholarship Programme for MSc Students of The Scientific and Technological Research Council of Turkey during my two year graduate education.

May 2014

İnci Meliha BAYTAŞ
(Telecommunication Engineer)

TABLE OF CONTENTS

Page

ABBREVIATIONS

BP	**:** Belief Propagation
ITF	**:** Inter-frame Transformation Fidelity
MRF	**:** Markov Random Field
MSE	**:** Mean Square Error
PSNR	**:** Peak Signal to Noise Ratio
RANSAC	**:** Random Sample Consensus
SIFT	**:** Scale Invariant Feature Transform

LIST OF TABLES

Page

LIST OF FIGURES

Page

DIGITAL VIDEO STABILIZATION WITH SIFT FLOW

SUMMARY

Videos which are recorded by hand held devices generally suffer from unintentional camera motion. The reasons of the unintentional motion may be the hand shake of users, or recording videos in a moving vehicle like car, bicycle, etc. Unwanted camera motion is not only encountered in amateur recordings made on mobile devices but also in video surveillance systems because of weather conditions like wind, or in videos which are recorded from an aerial vehicle. This unwanted motion decreases the quality of the video. In addition, the shaky movements may cause ambiguities in applications such as target detection or tracking. In other words, unwanted movements deteriorate the accuracy and the performance of the video processing applications. Therefore, the reduction of unintentional camera motion becomes a fundamental step for digital video processing.

Video stabilization can be defined as the correction of unstabilized video frames such that the new video which is constructed with the compensated frames has smoother frame to frame transitions. The possible methods for video stabilization can be divided into three categories such as mechanical, optical and digital video stabilization. In this study, digital video stabilization approach was taken into consideration. Digital video stabilization can also be divided into two categories such as offline and real-time video stabilization. Real-time video stabilization can be applied in mobile video recording devices. Real-time processing reduces the shakiness during the recording. However, this method is limited by the available processing time. Since the processing time is crucial, algorithms used in this method are generally chosen easy to implement and the motion models are also chosen simple to reduce the complexity. Stabilized videos with relatively sufficient visual quality are able to be produced by real time video stabilization methods. On the other hand, if the goal is improving accuracy and the performance of a video processing application, real time methods with simplified solutions may not be enough. In contrast, offline post-processing, which is the target of this study, allows us to use more robust and accurate methods. As a result, the quality and the accuracy of stabilized videos are consequently better than those for real-time.

Digital video processing has two main steps such as motion estimation and motion compensation. Motion estimation is the crucial part of a video stabilization scheme. There is a wide variety of approaches for motion estimation such as block matching algorithms, optical flow methods, pel-recursive methods, phase correlation methods, Bayesian methods, parametric motion estimation models, and 3D motion estimation. Correspondence matching or image alignment focuses on finding a feature which will be consistent across images. Raw pixels, corners, edges or some distinctive descriptors are used for this goal. As it is expected, using raw pixels is not a favored way because of its weakness for noise, illumination and orientation changes, etc. On the contrary, feature based motion estimation is proposed as a more robust method to these condition changes, since it uses some lighting, scale, orientation and geometric transformation invariant features for correspondence matching. In this thesis, a relatively new high

level image alignment technique called SIFT flow was used to extract the 2D flow field between consecutive video frames. SIFT flow can be briefly expressed as an algorithm whose computational framework is based on optical flow, but matches the SIFT descriptors instead of raw pixels. SIFT flow extracts pixelwise SIFT descriptors which are produced by local image structures and contextual information. These descriptors are then matched by a discrete, discontinuity preserving flow estimation algorithm. A discrete coarse to fine matching scheme based on the belief propagation is used to find flow vectors that minimize the cost function of the SIFT flow algorithm.

Although feature based methods try to match highly distinctive and robust features, there can still be undesired results during the motion estimation because of the feature points on moving objects and incorrect correspondence point matching. These kinds of points are expressed as outliers. Outliers are the points which do not fit the global motion model and alter the motion vectors locally. The success of a video stabilization scheme is affected by the outliers significantly. For this reason, outlier points must be eliminated. One of the widely used methods for the elimination of outliers is the Random Sample Consensus (RANSAC) algorithm. RANSAC tries to find inlier points in an iterative scheme. RANSAC process is repeated until reaching a predetermined number of trials. The error function is chosen as the Euclidean distance. Therefore, RANSAC tries to find points whose Euclidean distance between the actual points in the target frame and the transformed points from the reference frame are less than a distance threshold. This threshold can be determined heuristically according to the data. The maximum number of trials and the threshold for the consensus set size is calculated by considering the number of points and the inlier probabilities. Although RANSAC is a practical tool for removing outliers, there may be some points that belong to moving objects and cannot be easily eliminated by RANSAC. Therefore, a background point selection approach, which means choosing points compatible with the motion model, was utilized to overcome this problem.

After eliminating outliers, motion is estimated by using inlier feature points. Motion is generally expressed as a two dimensional vector whose elements are the horizontal and the vertical displacements. These two components are usually assumed to be independent. This assumption provides ease in computations. If a simple translation model is assumed as global camera motion, a global motion vector for a frame is looked for. On the other hand, there are also affine changes in real life videos and simple translation may not be enough for compensating the unstable frames. For example, the affine transformation constructs the camera motion model with scale, rotation, shear and translation together. If an affine parametric motion estimation is followed, global motion will be modeled as a global transformation between successive frames. In this study, global motion model was chosen as a 6 parameter affine transformation which is often preferred in literature. The last step of a video stabilization scheme is the motion compensation. Frames which have motion blur may yield wrong matching results. This may cause undesired affine transformation matrices. However, matching failures do not affect the translational motion as much as the affine part. Therefore, the frames with motion blur are compensated by using a translational motion model only. In conclusion, a feature based matching method was used to obtain flow vectors, outliers were eliminated by the RANSAC method, and shaky frames are compensated by taking the motion blurs into account in this thesis.

SIFT AKIŞI İLE SAYISAL VİDEO SABİTLEME

ÖZET

Bu çalışmada, bir videonun istenmeyen kamera hareketlerinin olabildiğince giderilmesi ele alınmaktadır. Videolardaki istenmeyen kamera hareketleri çekimin araba, helikopter gibi hareketli bir ortamda yapılması, kullanıcının elinin titremesi ya da güvenlik kameralarında rüzgar gibi hava koşulları sebebiyle meydana gelebilir. Bu hareketler, videonun görsel kalitesini bozarak izleyicileri rahatsız edebilir. Bununla birlikte, hedef takibi gibi sayısal video işleme uygulamalarında da belirsizliklere ve yanlışlıklara neden olmaktadır. Bu nedenle, sayısal video işleme uygulamalarına geçmeden önce istenmeyen kamera hareketlerinin giderilmesi (video sabitleme) gerekmektedir. İstenmeyen kamera hareketlerinin giderilmesi sonucunda görsel olarak daha yumuşak geçişleri olan bir video oluşturulması amaçlanmaktadır.

Literatürde, temel olarak üç çeşit video sabitleme yönteminden bahsedilmektedir. Bu yöntemler, mekanik, optik ve sayısal video sabitleme olarak adlandırılmaktadır. Mekanik video sabitlemenin amacı, kameranın üzerinde durduğu platformun hareketinin algılanarak kamerayı titreşimsiz bir çekim yapacak şekilde fiziksel olarak düzeltmektir. Oldukça iyi sonuçların alınabildiği mekanik video sabitlemede kamera dışında taşınması gereken aygıtlar bulunduğu için günlük kullanım ve amatör kullanıcılar için uygun olmayabilir. Bir diğer yöntem ise optik video sabitlemedir. Optik video sabitlemenin amacı ise kamera içindeki mercek grubunu görüntünün titreşimine uygun olarak değişikliğe uğratmak ve görüntü düzlemine ulaşan ışınların düzeltilmesini sağlamaktır. Son yıllarda, video kameraların pek çoğunda optik video sabitleme özelliği bulunmaktadır.

Buraya kadar bahsedilen iki video sabitleme yöntemi de istenmeyen hareketlerden arınmış videolar çekmeyi amaçlamaktadır. Üçüncü ve son yöntem olan sayısal video sabitleme, sayısal video işleme yöntemleri kullanılarak videolardaki titreşimi gidermeye çalışmaktadır. Uygulamaya göre gerçek zamanlı ya da çekim sonrası işleme şeklinde iki seçeneği mevcuttur. Gerçek zamanlı video sabitleme mobil cihazlara uygulanabilmektedir ve çekim sırasında titreşimli çerçeveleri düzeltmeyi amaçlamaktadır. Bu yöntemin sakıncalı yanı, zaman kısıtlaması olmasıdır. Kullanılan yöntemlerin hesaplama karmaşıklığının ve süresinin mümkün olduğunca az olması gerekmektedir. Bu nedenle, daha kolay uygulanabilir yöntemler tercih edilir. Örneğin, video sabitleme uygulamalarında en temel adım olan hareket kestriminde basitliği sebebiyle ötelemeye dayalı hareket modeli kullanınabilmektedir. Gerçek zamanlı video sabitleme ile mobil uygulamalar için yeterli olabilecek bir sabitleme gerçekleştirilebilir. Ancak doğruluğu daha yüksek bir uygulamaya ihtiyaç varsa zaman kısıtlması olmayan çekim sonrası işleme tercih edilmelidir. Bu yolla, gürültü, geometrik dönüşümler, ışık değişimleri gibi etkenlere daha dayanıklı ve hesaplama karmaşıklığı nispeten fazla olan daha kapsamlı yöntemler kullanılabilir. Bu nedenle, çekim sonrası video işleme yöntemlerinin doğruluğu ve görsel kalitesi gerçek zamanlı yöntemlere göre daha iyi olmaktadır. Bu yöntemin olumsuz yani ise artan hesap karmaşıklığı ile programların çalışma süresinin uzamasıdır. Her ne kadar gerçek

zamanlı yöntemler gibi hesaplama süresi üzerinde bir kısıtlama olmasa da titreşimsiz videoları elde etme süresinin kabul edilebilir ölçülerde olması gerekmektedir.

Sayısal video sabitleme, hareket kestirimi ve hareket karşılama olmak üzere iki temel adımdan oluşmaktadır. Özellikle hareket kestiriminin doğruluğu video sabitleme uygulamaları açısından büyük öneme sahiptir. Hareket kestirimi sırasında meydana gelebilecek herhangi bir hata, video sabitleme performansının bütününü etkilemektedir. Bu nedenle, dayanıklı ve doğruluğu yüksek bir hareket kestirimi yöntemi tercih edilmelidir. Hareket kestirimi, blok eşleme algoritması, optik akış yöntemleri, faz ilişkisi, vb. çeşitli yöntemler kullanılarak gerçekleştirilmektedir. Bahsedilen bu yöntemler kullanılarak komşu video çerçevelerinin pikselleri veya piksel blokları eşlenerek aralarındaki yer değiştirme kestirilmeye çalışılır. Bu eşleme esnasında, piksellerin gri seviye değerleri, kenar, köşe noktaları ya da çerçeveler boyunca istikrarlı kalabilecek bir takım öznitelikler kullanılabilir. Tahmin edileceği üzere gürültü, ışık değişimleri, ölçek ve geometrik dönüşüm değişimlerine karşı dayanıksızlığı sebebiyle doğrudan piksellerin gri düzeylerini kullanmak tercih edilen bir yol değildir. Bu nedenle video çerçevelerini eşlerken Scale Invarinat Feature Transform (SIFT), Speeded up Robust Features (SURF), Oriented FAST and Rotated BRIEF (ORB), vb. özniteliklerden yararlanılmaktadır. Öznitelik tabanlı görüntü işleme algoritmaları her ne kadar ayırt edici noktalar bulabilse de çerçeve içinde hareket eden nesneler ya da yanlış eşlemeler sebebiyle hareket kestiriminde istenmeyen sonuçlarla karşılaşılabilir. Yanlış eşlemeler nedeniyle seçilen hareket modeline uymayan ya da hareket eden cisimler nedeniyle hareket vektörlerinde yerel değişimlere sebep olan bu noktalar aykırı noktalar olarak adlandırılabilir. Aykırı noktalar, bir video sabitleme algoritması için büyük öneme sahip olan global hareketin kestirilme performansını olumsuz etkilemektedir. Bu nedenle, aykırı noktaların elenmesi gerekmektedir. 1981'de Fischler ve Bolles tarafından önerilen Random Sample Consensus (RANSAC) sık kullanılan bir aykırı nokta eleyici algoritmadır.

Ayrık noktalar elendikten sonra artık video sabitleme için gerekli olan global hareket, elemeden geçen noktalar ile kestirilmeye çalışılır. Ardışık video çerçeveleri arasında bulunan hareket, iki boyutlu bir akış alanı olarak ifade edilebilmektedir. İki boyutlu akış vektörlerinin elemanları, yatay ve düşey eksendeki yer değiştirmeyi göstermektedir. Hesaplamalarda kolaylık sağladığı için yatay ve düşey yer değiştirmeler genellikle birbirlerinden bağımsız olarak ele alınmaktadır. Bu akış vektörlerini elde edebilmek için ayrık, sürekli ya da kabadan inceye eşleme yöntemleri bulunmaktadır. Ayrıca komşu çerçeveler arasındaki hareketi geometrik bir dönüşüm olarak ifade etmek de mümkündür. Video sabitleme problemlerinde de sıkça kullanılan iki boyutlu parametrik hareket kestirimi, iki boyutlu yer değiştirmeleri kullanarak ardışık çerçeveler arasında geometrik bir dönüşüm bulmayı hedefler. Yaygın olarak kullanılan parametrik hareket modelleri iki boyutlu doğrusal koordinat dönüşümleridir.

Bu çalışmada SIFT özniteliklerinden yararlanan ve optik akış algoritmasından esinlenen bir eşleme yöntemi olan SIFT akışı kullanılmıştır. SIFT akışı, orijinal görüntüleri kullanarak her noktasında 128 boyutlu SIFT öznitelik vektörleri olan SIFT gürüntülerini elde eder. Böylece orijinal SIFT yöntemine göre daha sık bir SIFT gösterilimi elde edilmiş olur. Ancak SIFT akışı SIFT özniteliklerini hesaplarken orijinal SIFT öznitelik çıkarma adımlarının tamamını izlemez. Buna rağmen görüntü eşlemede piksellerin gri düzey değerlerini kullanmak yerine SIFT akışı yönteminde hesaplanan SIFT özniteliklerini kullanmak gürültü, geometrik

dönüşümler, ışık değişikleri vb. etkenlere karşı dayanıklılık sağlamaktadır. SIFT akışı, SIFT görüntülerini optik akışa benzer bir yaklaşım ile eşlemektedir. SIFT akışının enerji fonksiyonu, yer değiştirmenin akış vektörleri boyunca olacağı, bu akış vektörlerinin Taylor açılımını sağlayacak kadar küçük bulunacağı ve komşu akış vektörlerinin birbirine benzer olacağı yani süreksizliklerin kontrol altına alınabildiği bir yapıda seçilmiştir. Enerji fonksiyonu ayrıca parçalı Markov Rastgele Alanı (piecewise Markov Random Field) şeklinde modellenmiştir ve böylece bu maliyet fonksiyonunu enküçülten akış vektörlerinin bulunmasında Bayesçi bir yaklaşım olan inanç aktarımı (belief propagation) yöntemi kullanılabilmektedir.

SIFT akışı sonucunda ardışık iki çerçeve arasındaki yer değiştirmeleri ifade eden akış alanı elde edilmiş olmaktadır. Bu aşamada ortaya çıkabilecek aykırı noktalar RANSAC kullanılarak elenmektedir. RANSAC algoritmasında öncelikle, göz önüne alınan hareket modelinin çözümü için gereken en az sayıda nokta rasgele seçilir. Bu noktalar kullanılarak bir başlangıç hareket modeli hesaplanır. Daha sonra elimizdeki noktalardan bu modele uyan bir altküme seçilir. Noktaların modele uygunluğuna bakılırken kullanılan ölçüt ise Öklid uzaklığına dayanmaktadır. İlk iterasyon sonucu testten geçen noktaların sayısı eğer önceden belirlenen olası veri içindeki modele uyumlu nokta sayısından daha fazla ise program sonlandırılır, değilse başa dönülür ve işlemler tekrarlanır. RANSAC algoritması sonucunda elde edilen bütün uyumlu noktalar hareket modelinin bulunmasında kullanılmaktadır. Bu çalışmada model parametrelerini bulmak için en küçük kareler yöntemi kullanılmıştır. Ortalama almaya dayalı bir yöntem olduğu için en küçük karelerin sonucu aykırı noktaların varlığından oldukça etkilenmektedir. Bu nedenle, bu çalışmada en küçük kareler RANSAC ile aykırı noktalar elendikten sonra kullanılmıştır. Ancak bazı durumlarda sadece RANSAC algoritmasını kullanmak aykırı noktaların sonucu etkilemesini önlemeye yetmemektedir. Örneğin, videolarda hızla hareket eden ve oldukça çok yer kaplayan cisimlere ait noktalar RANSAC tarafından elenemeyecek aykırı noktalardır. Bu sorununun üstesinden gelmek için video çerçevesindeki noktalar kabul edilen hareket modeline uygunlukları açısından bir seçime tabi tutulmuştur. Hareketli cisimlerin genellikle çerçevenin ön planında yani çoğunlukla orta bölgelerde bulunduğu varsayılmaktadır. Bu nedenle, video sabitleme işleminin başında çerçevenin orta bölgesi dışında kalan noktalar RANSAC algoritmasında kullanılmıştır. Orta bölgenin büyüklüğü tahminen belirlenmektedir. İlk iki çerçeve için sadece arka plan noktaları kullanılarak hareket modeli hesaplanır. Sıradaki çerçeve çiftine geçmeden önce ön plan olarak kabul edilen bölgedeki noktalardan hesaplanan modele uyanlar da arka plan noktalarına katılır ve arka plan noktalarından modele uymayan noktalar elenir.Böylece, sıradaki RANSAC işlemi güncellenmiş arka plan noktaları kullanılarak yapılmaktadır. Bahsedilen işlemler bütün çerçeve çiftleri için tekrarlanarak devam eder. Dikkat edilmesi gereken nokta, bu tezde arka plan noktalarını seçmek ile ifade edilmek istenen her adımda hesaplanan hareket modeline uyan noktaların belirlenmesidir. Hareket modeli olarak olası kamera hareketlerinin çoğunluğunu içeren 6 parametreli ilgin dönüşüm tercih edilmiştir. Hareket modeli kestirildikten sonra video sabitleme yöntemlerinin son aşaması olan hareket karşılaması gerçekleştirilmektedir. Hareket karşılamada istenmeyen hareketlerin giderildiği yeni çerçeveler bir araya getirilerek sabitlenmiş videolar oluşturulmaktadır. İlgin dönüşüm bulunduktan sonra ikinci çerçeveye bulunan dönüşüm uygulanarak birinci çerçeve elde edilmektedir. Sıradaki çerçeve çiftine geçildiğinde ise bir önceki adımda düzeltilen çerçeve ile yeni çerçeve karşılaştırılır. Herhangi bir adımda ilgin dönüşüm hesaplamasında

meydana gelen hatalar eşleşmelerde düzeltilmiş çerçeveler kullanıldığı için katlanarak artabilmektedir. Bu durumun üstesinden gelebilmek için hata yapma olasılığının fazla olduğu çerçeveler belirlenerek bu çerçevelerde ilgin dönüşümün sadece ötelemeleri kullanılarak çerçeveler düzeltilmeye çalışılmıştır. Çünkü eşleme hataları ilgin dönüşümün öteleme kısmını nispeten daha az etkilemektedir. Eşleme hatasının olası olduğu çerçeveler ise çerçevelerin gradyenlerinden yararlanarak tespit edilmeye çalışılmıştır.

Çerçeveler düzeltilirken dikkate alınan bir diğer konu da bilinçli olarak kullanıcının ilgi alanının değişmesi sonucu yapılan yalpa ve yunus gibi kamera hareketlerini istenmeyen titreşim hareketlerinden ayırt edilmesidir. Bunun için bilinçli kamera hareketlerinin titreşim hareketlerine göre daha düzenli ve yumuşak hareketler olduğu sonucundan yararlanılmaktadır. Örneğin, yalpa hareketi için yatay öteleme parametreleri bir grup çerçeve için takip edilirse, parametrelerin tekdüze bir şekilde bir yönde arttığı görülmektedir. Bu çalışmadaki video sabitleme programınının yalpa hareketini bahsedilen şekilde fark edip yalpa hareketi süresince çok büyük bir düzeltme yapmaması sağlanmaya çalışılmıştır. Böylece istenen yalpa hareketinin takip edilebilmesi amaçlanmıştır. Bu tezde ayrıca yalpa hareketinin bilinçli bir kullanıcı tarafından yavaş bir şekilde yapıldığı varsayılamaktadır. Sonuç olarak, videolardaki istenmeyen hareketler literatürdeki çalışmalara benzer bir yaklaşımla giderilmeye çalışılmıştır. Öznitelik tabanlı bir eşleme yöntemi ile akış vektörlerine ulaşılmıştır. Seçilen hareket modeli ile uyumsuzluk yaratacak aykırı noktalar elenmiş, hareket modeli iki çerçeve arasındaki ilgin dönüşümü olarak belirlenmiş ve bu bilgiler ışığında titreşimli çerçeveler düzeltilerek sabitlenmiş videolar elde edilmeye çalışılmıştır.

1. INTRODUCTION

Hand held devices such as digital video cameras and cell phones are widely used all over the world. In addition, taking photos and recording videos became a pervasive habit. In amateur recordings, the unintentional camera motion caused by the hand shake of users or when recording from a moving car are inevitable. Unwanted camera motion is not only encountered in amateur recordings but also in video surveillance systems because of weather conditions like wind or in videos which are recorded from an aerial vehicle. This unwanted motion decreases the quality of the video. The aforementioned degradation in the quality of video has two aspects. First of all, the visual quality is decreased that is very disturbing for the viewers. Second and the most important aspect of the degradation is that the shaky movements may cause ambiguities in applications such as target detection or tracking. In other words, unwanted movements deteriorate the accuracy and the performance of video processing applications. Therefore, the reduction of unintentional camera motion becomes a fundamental step for digital video processing.

Video stabilization is the correction of unstabilized video frames such that the new video which is constructed with the compensated frames has smoother transitions. The possible methods for video stabilization can be divided into three categories such as mechanical, optical and digital video stabilization. Mechanical video stabilization provides a physical solution. It aims to adjust the entire camera to record stabilized videos. Some heavy devices with spinning wheels and a battery maintain the camera by using the information which comes from the motion sensors of the device. Mechanical video stabilization is able to give very good results. However, it is not a suitable solution for ordinary consumer use because of its power consumption and clumsiness. Second solution is optical video stabilization which manipulates the lens group with respect to the degree of image vibration. The light rays reaching the image plane can be steadied by using this method. In recent years, many video cameras have the optical video stabilization utility. The first two solutions stabilize videos while recording. The

last solution is the digital video stabilization [1]. Digital video stabilization can be also divided into two categories such as offline and real-time video stabilization. Real-time video stabilization is applied for mobile video recording devices. Real-time processing reduces the shakiness during the recording. However, this method is limited by the processing time. Since the processing time is crucial for real time use, algorithms used in this method are generally chosen easy to implement and simple motion models are chosen to reduce the complexity. For instance, the motion is assumed to be simple translation. Stabilized videos with relatively sufficient visual quality are able to be produced by the real time video stabilization methods. On the other hand, if the goal is improving accuracy and the performance of a video processing application, real time methods with simplified solutions may not be enough. In contrast, offline post-processing allows us to use more robust and accurate methods. Since there are no time limitations, more complex and comprehensive procedures may be applied. As a result, the quality and the accuracy of stabilized videos are obviously better than real-time but the drawback of the off-line post-processing is the computation time [2].

Digital video processing has two main steps such as motion estimation and motion compensation. Motion estimation is the crucial part of a video stabilization scheme. Any failure in motion estimation part affects the whole performance of the video stabilization procedure. Thus, an accurate and robust motion estimation is required for properly stabilized frames. There is a wide variety of approaches for motion estimation such as block matching algorithms, optical flow methods, pel-recursive methods, phase correlation methods, Bayesian methods, parametric motion estimation models and 3D motion estimation.

By using one of these methods the movement of pixels or blocks from one frame to another or briefly correspondence between consecutive frames is extracted. Correspondence matching or image alignment focuses on finding a feature which will be consistent across images. Raw pixels, corners, edges or some distinctive descriptors are used for this goal. As it is expected, using raw pixels is not a favored way because of its weakness for noise, illumination and orientation changes, etc. On the contrary, feature based motion estimation is proposed as a more robust method to these condition changes, since it uses some lighting, scale, orientation and geometric transformation invariant features for correspondence matching. Some prominent feature extraction

methods which are widely used in literature are Scale Invariant Feature Transform (SIFT), Speeded up Robust Features (SURF), Oriented FAST and Rotated BRIEF (ORB), etc. There are also modifications of SIFT in which computation complexity and cost are tried to be reduced.

There is a selection of the most representative and distinctive points in feature extraction methods. Therefore, a sparse representation of an image is generated. One drawback that may be encountered is an inadequate number of features extracted for some images. If there are not enough features, matching results will be poor. According to the application requirements, a dense representation may also be preferred.

Although feature extraction methods provide highly distinctive and robust features, there still can be undesired results during the motion estimation because of the feature points on moving objects and incorrect correspondence of matching points. These kind of points are expressed as outliers. Outliers are the points which do not fit the global motion model and alter the motion vectors locally. The success of the video stabilization depends on finding a global camera motion correctly. However, outlier points which especially occupy a large area in the frame affect the accuracy of global motion estimation process. For this reason, outlier points must be eliminated. One of the widely used methods to handle outliers is Random Sample Consensus (RANSAC) algorithm. RANSAC has the ability to remove points on moving objects in a frame [3].

After eliminating outliers, motion is estimated by using inlier feature points. Motion is generally expressed as a two dimensional vector. One of the components is for horizontal motion and the other one is for vertical motion. These two components are usually assumed to be independent. This assumption provides ease in computations. There are continuous, discrete and coarse-to-fine matching schemes to reach these motion vectors. At this point, there are different options to model the motion between consecutive frames. 2D parametric motion is generally preferred in video stabilization applications. The commonly used motion model is the affine motion model with six parameters. In a usual video stabilization scheme, after estimating the global motion parameters, motion smoothing is required to suppress high frequency jitters and obtain the intentional camera motion. Kalman filtering, Gaussian filtering, motion vector integration and particle filtering are some of the motion smoothing methods in

literature [2]. The last step of video stabilization is warping a frame by using smoothed motion parameters to obtain another frame. Different combinations of these methods explained above are utilized for video stabilization purposes in the literature.

1.1 Literature Review

In recent years, feature based motion estimation methods are commonly preferred in many studies reported in the literature and different combinations of the methods mentioned in the previous section are utilized for the video stabilization purpose. In this section, some selected work which have similar approaches as in this thesis are briefly summarized.

In [2], the ORB features were used in the motion estimation step. ORB features were extracted by using FAST keypoints detector and BRIEF descriptors. FAST or accelerated segment test is a corner detection method which uses machine learning. In spite of the high speed of the detector, FAST has no orientation operator. The lack of orientation operator destroys the robustness to noise. Therefore, rotated BRIEF descriptors, which utilize binary tests between pixels in a smoothed image patch, were used to provide robustness. Similar to SIFT, BRIEF has also robustness to lighting, blur and perspective distortion. The proposed feature extraction method in [2], combined FAST corner detector and BRIEF descriptor and it was called oriented FAST rotated BRIEF (ORB). It is stated that motion estimation with ORB was faster and had similar accuracy as motion estimation with SIFT. After ORB features were extracted and keypoints were matched, affine transformation model parameters were estimated to represent to global motion. RANSAC was also used to refine the affine parameters. The paper states that the ORB feature extraction produced a minimum number of features compared to SIFT and SURF, and this was why ORB feature based motion estimation needs less computational time. The reference [2] had an improved motion smoothing scheme that computed affine parameters by using unstable input frames and stabilized output frames. Gaussian filtering was preferred. They avoided accumulative errors with this smoothing approach. The proposed method in [2] was validated with real world videos. The reported results of experiments show that the approach in [2] was an efficient and robust video stabilization method.

In [3], particle filtering method was used for the video stabilization purpose. The main property of this approach was that, feature points should have different contributions to the estimation results, and good estimation should depend on feature points with similar degree of freedom (DOF). Features were extracted by the Speeded up Robust Features (SURF) method. Local motion vectors and incorrect correspondences were eliminated by RANSAC. Then, the weight of feature points was estimated by the particle filter approach, different depth of fields (DDOF) of different feature points were solved and weighted least square estimation was used to find global motion. They also used Kalman filter to estimate the intentional motion. 2D Affine motion model was assumed to represent the transformation between frames.

On the other hand, [4] classified the feature points as background and foreground in order to increase the accuracy of global motion estimation. The moving objects which produce outlier feature points are generally located at the foreground in a frame. [4] was aiming to use only the background feature points in the parameter estimation step to increase the performance of the RANSAC algorithm. Feature points were extracted and tracked by using the Kanade Lucas Tomasi (KLT) method and at the beginning of the feature point classification process, feature points were divided into two non-overlapping regions. Feature points that are located at the region near the middle of the frame were labeled as foreground. Conversely, the feature points that were extracted from the region near the boundaries of the frame were labeled as background feature points. Global motion model was extracted by using this initial set of background points in RANSAC. Before proceeding to the subsequent frames, foreground and background feature points were updated according to their compatibility with the global motion model. If a background feature point fitted the model, it remained as a background point in the following frames otherwise it switched to a foreground feature point. Contrarily, if a foreground feature point did not fit the model, it remained as a foreground point; otherwise it switched to a background point. It was stated that the proposed global motion estimation method colud be successful even with the presence of big foreground objects.

In [5], the similarity transformation model was preferred, and SIFT features were used in the global motion estimation. However, the feature extraction and selection of key points were different than conventional than SIFT algorithm. Their progress included

multi-block point extraction, in-block match selection and inter-block match selection. SIFT feature points were extracted from some blocks in the image and the feature points were also matched between these blocks. This was how they reduce the number of feature points and search range. In the first feature matching step, adjacent feature points in the same block were matched. In the second step, matched features in the first step were tried to be matched by using different blocks to increase matching accuracy. In this case, angle invariance property of similarity transform was considered. The paper states that their in-block and inter-block match selection scheme was faster than conventional methods and more accurate than RANSAC algorithm.

In [6], the affine motion model was preferred for the global camera motion and SURF was used for the feature extraction step. SURF features were matched by looking at the space distances and differences of descriptors between consecutive frames. After finding the features, RANSAC was utilized to deal with false matches and outliers and affine motion parameters were estimated. An iterative smoothing scheme was constructed by using a weak Gaussian kernel. Finally, the video was stabilized by frame warping with the compensation matrix. It is stated that the stabilized camera motion path was closer to the ideal path. As a result, there was less uncovered area in the stabilized video. According to the study, their method was more robust and adaptable to different video clips.

Finally, [7] extracted motion vectors by the Lucas and Kanade optical flow method. Four parameter affine motion model was adopted to estimate camera motion and parameters were estimated by the least squares solution. Motion parameters were used in a decision method which was called the collective motion estimate (CME) in the paper. CME was used to discriminate intentional and unintentional camera motions. The parameters of CME had small values in case of intentional motion. Therefore, CME values that were obtained from consecutive frames could be utilized to describe the change of camera motion in the entire video. According to the paper, after shaky frames were identified with the CME method, unintentional motion was rectified by using image morphing methods.

The purpose of this thesis is modeling the global camera motion and compensating for this unwanted motion in order to produce smoother and stabilized frames as much as possible. Motion estimation is carried out by using a new scene alignment

method which is called SIFT flow. After extracting the correspondence between adjacent frames with SIFT flow, an outlier rejection process is utilized to find motion parameters. Random Sample Consensus is preferred as the outlier rejection tool. Camera motion is modeled as a 6 parameter affine transformation. Affine motion parameters are calculated by least squares estimation with only inlier points. The final step is the motion compensation which is carried out by analyzing the amount of the unwanted motion in this thesis.

2. MOTION ESTIMATION

First step of a video stabilization scheme is motion estimation. The accuracy of the motion estimation is very crucial such that any error in the motion estimation step propagates through the following steps and affects the motion compensation performance. Motion is generally defined as a 2D motion vector field between successive video frames. Since there is a loss of information during the projection of images from a 3D scene to the 2D image plane, motion estimation is an ill-posed problem. Therefore, using a robust motion estimation method is very important. There are pixel based methods such as block matching algorithm, phase correlation method and optical flow, etc. These direct methods may be effective, but they are not robust enough to illumination changes or geometric transforms. Thus, indirect methods which are more robust are preferred in this thesis. Indirect methods use features and match these features to construct a correspondence between adjacent frames. In our case, motion between video frames relates to the entire image rather than being local or belonging to a moving object. Therefore, global motion estimation is required for the video stabilization. If a simple translation model is assumed as a global camera motion, a global motion vector for a frame is looked for. In this case, global motion vectors can be thought as the most frequent vector in the vector field that is extracted between adjacent frames. On the other hand, there are also affine changes in a real life videos and simple translation may not be enough for compensating the unstable frames. For example, the affine transformation constructs the camera motion model with scale, rotation, shear and translation together. If an affine parametric motion estimation is followed, global motion will be modeled as a global transformation between successive frames. In this study, global motion model is chosen as a 6-parameter affine transformation which is often preferred in literature. In Fig.s 2.1, 2.2 and 2.3, magnitude and angle histograms of flow vectors of different transformations are shown. The examples in the following figures are synthetic. The transformation matrices which belong to Fig. 2.1, 2.2 and 2.3 are shown in Eq.s (2.1), (2.2) and (2.3), respectively.

$$T = \begin{pmatrix} 0.9995 & 0.0314 & 0 \\ -0.0314 & 0.9995 & 0 \\ 0 & 0 & 1 \end{pmatrix} \tag{2.1}$$

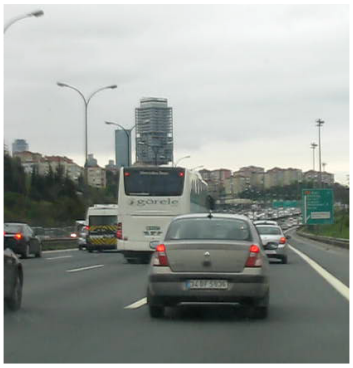
In Eq. (2.1), there is a clockwise rotation of $0.0314rad$.

$$T = \begin{pmatrix} 0.9995 & 0.1571 & -5 \\ -0.0314 & 0.9995 & 3 \\ 0 & 0 & 1 \end{pmatrix} \tag{2.2}$$

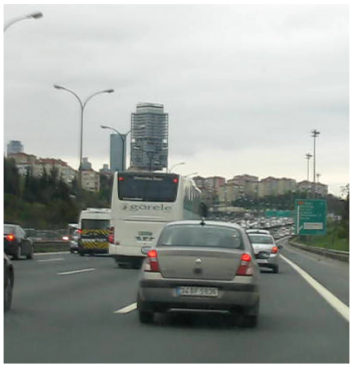
In Eq. (2.2), there are a clockwise rotation of $0.0314rad$, shear parallel to x axis with degree of 5, and 5 pixels of horizontal translation to the left and 3 pixels of downward vertical translation.

$$T = \begin{pmatrix} 1 & 0 & -25 \\ 0 & 1 & -15 \\ 0 & 0 & 1 \end{pmatrix} \tag{2.3}$$

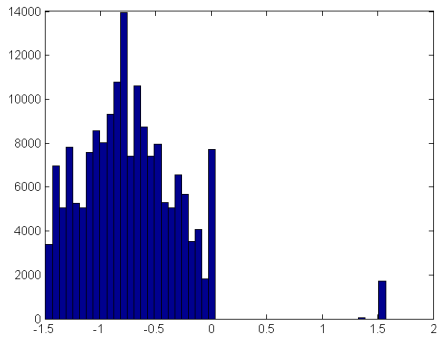
In Eq. (2.3), there are 25 pixels of horizontal translation to the left and 15 pixels of upward vertical translation.



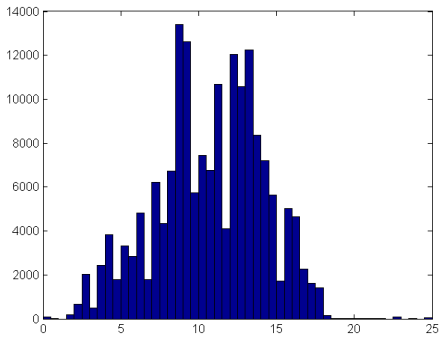
(a) Original image



(b) Image under only rotation

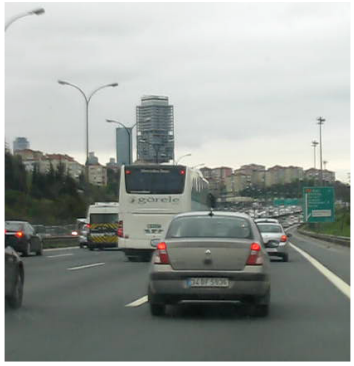


(c) Angle histogram (rad)

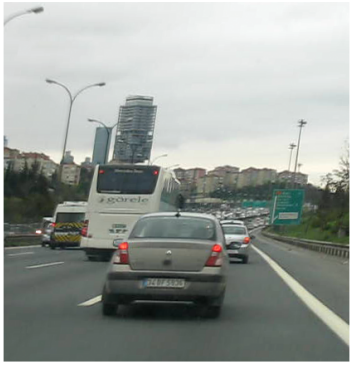


(d) Magnitude histogram (rad)

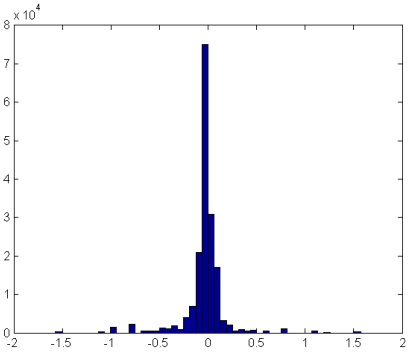
Figure 2.1: Histograms of flow vectors with only rotation.



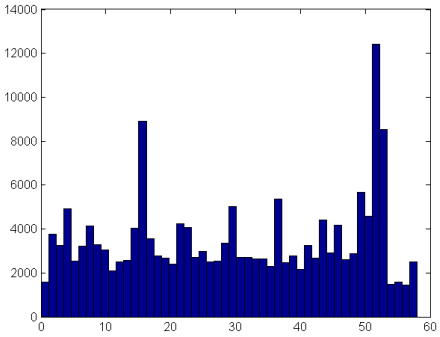
(a) Original image



(b) Image under affine

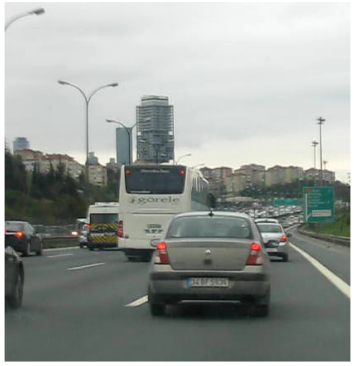


(c) Angle histogram (rad)

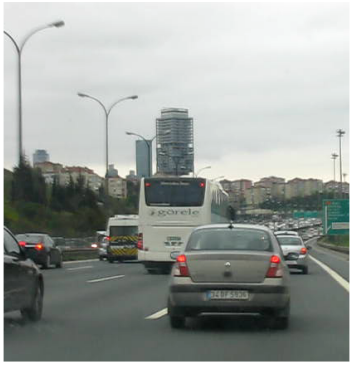


(d) Magnitude histogram (rad)

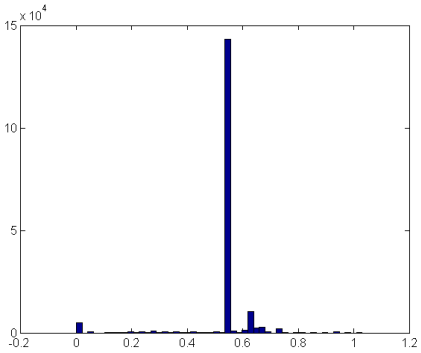
Figure 2.2: Histograms of flow vectors with affine transformation.



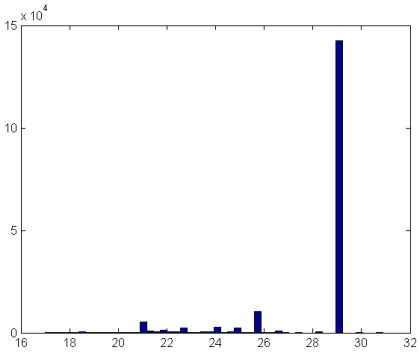
(a) Original image



(b) Image under simple translation



(c) Angle histogram (rad)



(d) Magnitude histogram (rad)

Figure 2.3: Histograms of flow vectors with simple translation.

As seen in Fig. 2.3, simple translation provides a global motion vector which is the most frequent among other flow vectors. However, a uniform magnitude histogram with a global angle is observed in affine transformation as it is seen in Fig. 2.2. These transformations and images were chosen arbitrarily. Therefore, histograms may be different for any other choice.

2.1 Correspondence Matching

In this thesis, a new high level image alignment method which is called SIFT flow was used to extract the 2D flow field between consecutive video frames. High level image alignment includes matching two images from different 3D scenes with similar scene characteristics. The problem considered in this thesis covered adjacent frames that belong to the same scene, but occlusion, clutter and multiple objects complicate the problem. [8] proposes a simple, effective and an object free image matching algorithm. SIFT flow can be briefly expressed as an algorithm whose computational framework is based on optical flow and this method matches SIFT descriptors instead of raw pixels. SIFT flow extracts pixel wise SIFT descriptors which are produced by local image structures and contextual information. These descriptors are then matched by a discrete, discontinuity preserving flow estimation algorithm. According to the reference [8], SIFT descriptors provide robust matching across different scene and object views, and discontinuity preserving spatial model provides matching of objects at different parts of the scene.

2.1.1 SIFT part of the SIFT flow algorithm

Scale Invariant Feature Transform (SIFT) was proposed by David G. Lowe in 2004 as a robust feature extraction method to perform dependable matching between images which have different views of the scene. SIFT produces distinctive features that are invariant to scale, rotation, view point, noise and illumination changes. The implementation steps of SIFT contain scale space extrema detection, keypoint localization, orientation assignment and keypoint descriptor. Briefly, scale space is constructed by filtering the image with Gaussian low pass filters in varying scales and the extrema points are found from the difference of Gaussian images in the scale space extrema detection step. Accurate subpixel locations of candidate keypoints are

obtained, and points with low contrast or on edges are eliminated in the keypoint localization step. After feature point extraction, one or more orientations which are calculated from local image gradients are assigned to each keypoint. Every keypoint has a scale, location and orientation as a result of the orientation assignment step. Finally, descriptors are obtained by dividing the pixel region around each keypoint into a $4x4$ array and calculated orientations are quantized into 8 bins. Thus, a $4x4x8 = 128$ dimensional feature vector is found for each keypoint [9].

As a result, SIFT can be defined as a sparse representation which has two basic steps such as feature extraction and feature detection. On the other hand, SIFT flow has only the feature extraction step of the original SIFT method. SIFT flow does not have any elimination of inaccurate or weak keypoints before obtaining SIFT descriptors. Moreover, SIFT flow does not build a scale space at the beginning of the feature extraction process. Feature extraction process of SIFT flow has the following steps. The neighborhood of each pixel in an image is divided into a $4x4$ cell array and the orientation is quantized into 8 bins in each cell. Vertical and horizontal edges are calculated by using gradient operations. Then the magnitude and the angle of gradients are calculated. After finding these parameters, orientations are calculated as it is shown in Eq. (2.4) and weighted with the magnitude of the gradients.

$$orientation_i = |\nabla I| \left(\cos\theta \cos\left(angle_i\right) + \sin\theta \sin\left(angle_i\right)\right)^{\alpha}, i = 1,2,3..,8 \qquad \textbf{(2.4)}$$

where α is a parameter for attenuation which must be an odd number and it is taken as 9 in [8]. θ is the angle and $|\nabla I|$ is the magnitude of the gradients. $angle_i$ are the 8 angles between 0 and 2π with an angle step of $\pi/4$.

As a result a $4x4x8 = 128$ dimensional feature vector is obtained for each pixel on a predetermined grid which is obtained according to the window size in which SIFT orientations are calculated. In [8], this per pixel SIFT descriptor is called as the SIFT image and the SIFT flow method is done with the SIFT algorithm after obtaining the SIFT image. All the steps of the SIFT algorithm are not carried out in the SIFT flow algorithm. The purpose of this feature extraction in SIFT flow is to find robust and dense features which have the local orientation information and matching them instead of intensity values of pixels. The aspect ratio of SIFT images are not the same as the

original image. Since the neighborhood of every pixel is divided into a 4x4 cell array, the calculations begin with the pixel that is four pixels away from the left and the top of the image. Therefore, the original width and height of the image do not remain the same and the matching is done between the grid points that depend on the window size for SIFT calculations. This resolution may be altered with a grid spacing parameter, if needed.

SIFT images are used in the correspondence matching step. In other words, 128 dimensional feature vectors of each pixel are matched. SIFT images will be $MxNx128$ dimensional matrix where N and M are the height and the width, respectively and the visualization of these images is not easy because of the third dimension. [8] proposes a visualization method to deal with this problem. They map the top three principle components of SIFT descriptors which are calculated from a set of images to the principle components of RGB color space. In this visualization, the pixels that have similar structures have similar colors. In addition, the visualization of SIFT images which is given in Fig. 2.4 shows that the sharp edges in original images are preserved. This is how we can see the discontinuity preserving property of the SIFT flow method.

(a) Original image (b) SIFT image

Figure 2.4: Visualization of per pixel SIFT image.

The SIFT image shown in Fig. 2.4 (b) is a $MxNx3$ image and it is not actually used in the matching step. The per pixel SIFT image that is used in the matching step is the $MxNx128$ matrix and the 128 dimensional feature vector of each pixel is matched to find the correspondence between adjacent video frames.

2.1.2 Optical flow part of the SIFT flow algorithm

In the previous section, how the SIFT features are extracted and a per pixel SIFT descriptor called the SIFT image is obtained are explained. Now, we have the entity

to match and what kind of a matching scheme is used in SIFT flow will be covered in this section. As it was mentioned before, matching scheme of SIFT flow is inspired by the optical flow method. In a video, image plane coordinates do not remain the same through the whole video. Pixels are moving because of the camera or object motion. This displacement of the pixels on the vertical and horizontal direction is modeled as a two dimensional vector which is called the correspondence vector. The correspondence vector can simply be thought as a difference of the corresponding point coordinates in adjacent frames. The temporal rate of change of the image plane coordinates because of the spatio−temporal variations of the intensity describes the optical flow vector. The correspondence field or optical flow field is determined as a vector field of pixel displacements which is also named as 'apparent 2D motion' [10]. Thus, 2D motion estimation problem becomes finding the correspondence vector. Optical flow method estimates the flow field by using the spatio−temporal image intensity gradients. The basic optical flow equation is given in Eq. (2.5) [11].

$$\frac{dE(x,y,t)}{dt} = 0 \tag{2.5}$$

where $E(x,y,t)$ is the image plane or the intensity value of pixel point at (x,y) at time t. Eq. (2.5) states that the intensity distribution remains the same through the time. If there is any intensity change for a point, this is only because of the displacement of pixels according to the main assumption of optical flow. Eq. (2.5) can be also expressed as shown in Eq. (2.6) by using the chain rule of the derivative.

$$\frac{\partial E}{\partial x}\frac{dx}{dt} + \frac{\partial E}{\partial y}\frac{dy}{dt} + \frac{\partial E}{\partial t} = 0 \tag{2.6}$$

where partial derivatives can be expressed as the gradient $\nabla E(x,y,t)$ and $u = \frac{dx}{dt}$ and $v = \frac{dy}{dt}$ are the unknown flow vectors. The flow vectors should be small enough that Taylor expansion can be valid. There are several approaches for the estimation of flow vectors. One of them is the Horn and Schunck method which looks for a flow field that minimizes the pixel to pixel variations along the flow vectors [10]. The cost function of Horn and Schunck method is shown in Eq. (2.7) [11].

$$E = \int\int \left((E_{of})^2 + \alpha^2 E_s^2 \right) dxdy \tag{2.7}$$

where E_{of} is the error in the optical flow function which is shown in Eq. (2.8)

$$E_{of} = \frac{\partial E}{\partial x}\frac{dx}{dt} + \frac{\partial E}{\partial y}\frac{dy}{dt} + \frac{\partial E}{\partial t} = E_x u + E_y v + E_t \tag{2.8}$$

E_s^2 is the smoothness constraint given in Eq. (2.9) and α is the smoothing parameter.

$$E_s^2 = \left(\frac{\partial u}{\partial x}\right)^2 + \left(\frac{\partial u}{\partial y}\right)^2 + \left(\frac{\partial v}{\partial x}\right)^2 + \left(\frac{\partial v}{\partial y}\right)^2 = u_x^2 + u_y^2 + v_x^2 + v_y^2 \tag{2.9}$$

If we let the points of the intensity pattern move independently, we may not find any reasonable velocity vectors. Therefore, the velocities of neighboring points are required to be similar and the flow field should be smooth. Flow discontinuities may occur when there are occlusions. As a result, flow discontinuities cause problems for an algorithm which has a smoothness constraint [11].

It can be seen from Eq. (2.9) that E_s^2 must be small enough to have a smoother flow field. Besides, we are able to control the smoothness constraint by α [10]. The cost function of Horn and Schunck can be optimized by a continuous estimation scheme which is reduced to solve partial differential equations with the Euler-Lagrange method. The solution is given step by step as below.

Let us $\left(E_{of}\right)^2 + \alpha^2 E_s^2$ be $L(x,y,u,v)$;

Euler-Lagrange equations are shown in Eq. (2.10)

$$\begin{aligned} \frac{\partial L}{\partial u} - \frac{d}{dx}\frac{\partial L}{\partial u_x} - \frac{d}{dy}\frac{\partial L}{\partial u_y} = 0 \\ \frac{\partial L}{\partial v} - \frac{d}{dx}\frac{\partial L}{\partial v_x} - \frac{d}{dy}\frac{\partial L}{\partial v_y} = 0 \end{aligned} \tag{2.10}$$

Let find the terms of Euler-Lagrange equations;

$$\begin{aligned} \frac{\partial L}{\partial u} =& \frac{\partial}{\partial u}\left(\left(E_x u + E_y v + E_t\right)^2 + \alpha^2\left(u_x^2 + u_y^2 + v_x^2 + v_y^2\right)\right) \\ =& 2\left(E_x u + E_y v + E_t\right)E_x \end{aligned} \tag{2.11}$$

$$\begin{aligned} \frac{\partial L}{\partial v} =& \frac{\partial}{\partial v}\left(\left(E_x u + E_y v + E_t\right)^2 + \alpha^2\left(u_x^2 + u_y^2 + v_x^2 + v_y^2\right)\right) \\ =& 2\left(E_x u + E_y v + E_t\right)E_y \end{aligned} \tag{2.12}$$

$$\begin{aligned}\frac{d}{dx}\frac{\partial L}{\partial u_x} &= \frac{d}{dx}\frac{\partial}{\partial u_x}\left((E_x u + E_y v + E_t)^2 + \alpha^2\left(u_x^2 + u_y^2 + v_x^2 + v_y^2\right)\right) \\ &= \frac{d}{dx}\left(2\alpha^2 u_x\right) = 2\alpha^2 u_{xx}\end{aligned} \tag{2.13}$$

$$\begin{aligned}\frac{d}{dy}\frac{\partial L}{\partial u_y} &= \frac{d}{dx}\frac{\partial}{\partial u_y}\left((E_x u + E_y v + E_t)^2 + \alpha^2\left(u_x^2 + u_y^2 + v_x^2 + v_y^2\right)\right) \\ &= \frac{d}{dy}\left(2\alpha^2 u_y\right) = 2\alpha^2 u_{yy}\end{aligned} \tag{2.14}$$

$$\begin{aligned}\frac{d}{dx}\frac{\partial L}{\partial v_x} &= \frac{d}{dx}\frac{\partial}{\partial v_x}\left((E_x u + E_y v + E_t)^2 + \alpha^2\left(u_x^2 + u_y^2 + v_x^2 + v_y^2\right)\right) \\ &= \frac{d}{dx}\left(2\alpha^2 v_x\right) = 2\alpha^2 v_{xx}\end{aligned} \tag{2.15}$$

$$\begin{aligned}\frac{d}{dy}\frac{\partial L}{\partial v_y} &= \frac{d}{dx}\frac{\partial}{\partial v_y}\left((E_x u + E_y v + E_t)^2 + \alpha^2\left(u_x^2 + u_y^2 + v_x^2 + v_y^2\right)\right) \\ &= \frac{d}{dy}\left(2\alpha^2 v_y\right) = 2\alpha^2 v_{yy}\end{aligned} \tag{2.16}$$

If the Euler-Lagrange equations are arranged again;

$$\begin{aligned}\left(E_x u + E_y v + E_t\right) E_x - \alpha^2\left(u_{xx} + u_{yy}\right) = 0 \\ \left(E_x u + E_y v + E_t\right) E_y - \alpha^2\left(v_{xx} + v_{yy}\right) = 0\end{aligned} \tag{2.17}$$

The terms $u_{xx} + u_{yy}$ and $v_{xx} + v_{yy}$ are the Laplacians of u and v. Therefore, Eq. (2.17) becomes;

$$\begin{aligned}E_x^2 u + E_x E_y v = \alpha^2 \nabla^2 u - E_x E_t \\ E_x E_y + E_y^2 v = \alpha^2 \nabla^2 v - E_y E_t\end{aligned} \tag{2.18}$$

The partial derivatives of brightness which are measured from a discrete set of image brightness can be estimated by using finite differences. For instance, Horn and Schunck proposed averaging four finite differences to estimate the derivatives as shown in Eq. (2.19) [10].

$$\begin{aligned}E_x \approx \frac{1}{4}\{&E_{i,j+1,k} - E_{i,j,k} + E_{i+1,j+1,k} - E_{i+1,j,k} + \\ &E_{i,j+1,k+1} - E_{i,j,k+1} + E_{i+1,j+1,k+1} - E_{i+1,j,k}\}\end{aligned}$$

$$
\begin{aligned}
E_y \approx & \frac{1}{4}\left\{E_{i+1,j,k} - E_{i,j,k} + E_{i+1,j+1,k} - E_{i,j+1,k} + \right. \\
& \left. E_{i+1,j,k+1} - E_{i,j,k+1} + E_{i+1,j+1,k+1} - E_{i,j+1,k+1}\right\}
\end{aligned}
\tag{2.19}
$$

$$
\begin{aligned}
E_t \approx & \frac{1}{4}\left\{E_{i,j,k+1} - E_{i,j,k} + E_{i+1,j,k+1} - E_{i+1,j,k} + \right. \\
& \left. E_{i,j+1,k+1} - E_{i,j+1,k} + E_{i+1,j+1,k+1} - E_{i+1,j+1,k}\right\}
\end{aligned}
$$

On the other side, Laplacians of u and v shown in Eq. (2.20) can be estimated by FIR highpass filtering.

$$
\begin{aligned}
\nabla^2 u &\approx \left(\bar{u}_{i,j,k} - u_{i,j,k}\right) \\
\nabla^2 v &\approx \left(\bar{v}_{i,j,k} - v_{i,j,k}\right)
\end{aligned}
\tag{2.20}
$$

where $\bar{u}$ and $\bar{v}$ are the local averages given in (2.21) [11].

$$
\begin{aligned}
\bar{u}_{i,j,k} = & \frac{1}{6}\left\{u_{i-1,j,k} + u_{i,j+1,k} + u_{i+1,j,k} + u_{i,j-1,k}\right\} \\
& + \frac{1}{12}\left\{u_{i-1,j-1,k} + u_{i-1,j+1,k} + u_{i+1,j+1,k} + u_{i+1,j-1,k}\right\} \\
\bar{v}_{i,j,k} = & \frac{1}{6}\left\{v_{i-1,j,k} + v_{i,j+1,k} + v_{i+1,j,k} + v_{i,j-1,k}\right\} \\
& + \frac{1}{12}\left\{v_{i-1,j-1,k} + v_{i-1,j+1,k} + v_{i+1,j+1,k} + v_{i+1,j-1,k}\right\}
\end{aligned}
\tag{2.21}
$$

If we apply the Laplacian approximation in Eq. (2.18):

$$
\begin{aligned}
E_x^2 u + E_x E_y v &= \alpha^2 (\bar{u} - u) - E_x E_t \\
E_x E_y + E_y^2 v &= \alpha^2 (\bar{v} - v) - E_y E_t \\
\left(\alpha^2 + E_x^2\right) u + E_x E_y v &= \alpha^2 \bar{u} - E_x Et \\
E_x E_y u + \left(\alpha^2 + E_y^2\right) v &= \alpha^2 \bar{v} - E_y E_t
\end{aligned}
\tag{2.22}
$$

If we use the Cramer rule to obtain the u and v, we will get the following solution in Eq. (2.23)

$$
\begin{aligned}
\left(\alpha^2 + E_x^2 + E_y^2\right) u &= \bar{u}\left(\alpha^2 + E_y^2\right) - E_x E_t - Ex E_y \bar{v} \\
\left(\alpha^2 + E_x^2 + E_y^2\right) v &= -E_y E_x u + \left(\alpha^2 + E_x^2\right) \bar{v} - E_y E_t
\end{aligned}
\tag{2.23}
$$

When an iterative solution such as Gauss Seidel is followed, the final solution will be as shown in Eq. (2.24) [10].

$$
\begin{aligned}
u^{(n+1)} &= \bar{u}^{(n)} - E_x \frac{E_x \bar{u}^{(n)} + E_y \bar{v}^{(n)} + E_t}{\alpha^2 + E_x^2 + E_y^2} \\
v^{(n+1)} &= \bar{v}^{(n)} - E_y \frac{E_x \bar{u}^{(n)} + E_y \bar{v}^{(n)} + E_t}{\alpha^2 + E_x^2 + E_y^2}
\end{aligned}
\tag{2.24}
$$

An instance of the solution for an optical flow method is given above. We can see from the solution that intensity distribution is directly used to find flow vectors that minimize the optical flow cost function. In SIFT flow, the desired matching of SIFT descriptors should also be along the flow vectors and flow field must be smooth in the same way as optical flow methods. The cost function of SIFT flow is shown in Eq.(2.25).

$$
\begin{aligned}
E(w) = &\sum_{p} min\left(\left\|s_1(p) - s_2(p + w(p))\right\|_1, t\right) + \sum_{p} \eta\left(|u(p)| + |v(p)|\right) + \\
&\sum_{(p,q)\in\varepsilon} min\left(\alpha\,|u(p) - u(q)|, d\right) + min\left(\alpha\,|v(p) - v(q)|, d\right)
\end{aligned}
\tag{2.25}
$$

where $p = (x, y)$ is the grid coordinate of the image, $w(p) = (u(p), v(p))$ is the flow vector at p, s_1, s_2 are SIFT images, ε contains all the spatial neighborhoods with respect to a four neighborhood system and t and d are the thresholds of truncated L1 norm which is preferred to deal with outliers and flow discontinuities. The first term in the sum in Eq.(2.25) is called the data term which provides matching of the SIFT descriptors along with the flow vector $w(p)$; the second term is called the small displacement term which constraints the flow vector to be as small as possible and it is controlled by the term η; the third and final term is called the smoothness term or the spatial regularization term which constrains the flow vectors of the adjacent pixels to be similar.

There are some analogies between the cost functions of Horn and Schunck optical flow and SIFT flow algorithms. For instance, the data term is similar to the error term of optical flow, but the grid points of SIFT images are considered rather than pixel points of original gray level images as in optical flow. Small displacement term and the smoothness term of SIFT flow also constrains the flow vectors and their neighbors to be small and similar as it is in optical flow. In contrast with the optical flow equations,

the smoothness term is decoupled in SIFT flow energy function. Thus, vertical and horizontal flow vectors become separated that makes it more useful in message passing in the belief propagation algorithm that will be used to optimize the SIFT flow cost function, and the complexity of the algorithm is said to be reduced from $O\left(L^4\right)$ to $O\left(L^2\right)$. SIFT flow prefers the truncated L1 norms instead of L2 norm in data and smoothness terms to deal with wrong matches and flow discontinuities which is a general problem of algorithms that have a smoothness constraint. L2 norm is able to deal with noise but it is not robust enough for outliers. One of the differences between SIFT flow and optical flow is the type of the values of flow vectors. As can be seen from the previous optimization example, optical flow provides subpixel precision. On the other hand, SIFT flow vectors are integers. Thus, SIFT flow provides pixel level accuracy whereas the optical flow can achieve subpixel accuracy. Moreover, flow vectors do not have to be integer valued in real life videos. Therefore, we apply a Gaussian low pass filtering to the flow vectors before using the vectors for further operations. The size of the Gaussian low pass kernel is chosen small, since we would like to avoid smoothing out the edge relationships of objects in the frames. The aim of this low pass filtering is to make the values of the flow vectors be similar to their $3x3$ neighbors and to deal with the possible quantization errors.

If we summarize the SIFT flow up to this point, SIFT flow is a discrete optical flow method which uses SIFT descriptors in matching and constructs a dense correspondence by looking for flow vectors for each grid point. In [8] a discrete coarse to fine matching scheme based on belief propagation is used to find flow vectors that minimize the cost function of SIFT flow algorithm.

2.1.3 Optimization of SIFT flow cost function

A coarse-to-fine matching scheme was preferred in [8] to avoid the computation time drawback. In coarse to fine matching, the flow at a coarse level is estimated first. Then the flow is propagated and refined step by step from the coarse levels to fine levels. It was shown that the coarse to fine matching can also yield lower energies most of the time compared to the standard matching. It was indicated that the relevance of the coarse-to-fine matching is generally encountered in optical flow methods, too. At each level, a dual layer loopy belief propagation algorithm is used to find optimal

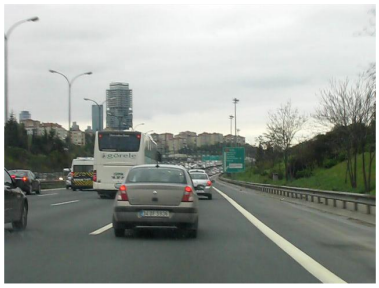
flow vectors in [8]. The energy function of the SIFT flow is formed as a Markov Random Field (MRF). MRFs are used in inference problems of computer vision like stereo matching, image segmenting, image reconstruction, etc. MRFs are explained as undirected graphical models that can encode spatial dependencies. As a graphical model, MRF has nodes, links and sometimes loops or cycles. There are observed and hidden variables in a MRF model, too. For instance, when a stereo matching problem is modeled as MRF, observed variables are the image intensities, and the hidden variables which are sometimes named as labels are the disparity that is aimed to be found [12]. In the SIFT flow case, the SIFT descriptors for grid points can be thought as observed variables and the correspondence between points can be thought as the labels. A general formulation based MRF is given below [13].

$$E(f) = \sum_{p \in P} D_p(f_p) + \sum_{(p,q) \in N} V\left(f_p - f_q\right) \tag{2.26}$$

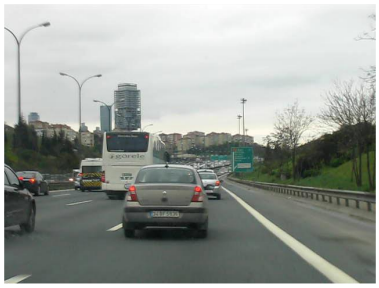
where f assigns a label f_p to each pixel $p \in P$, N are neighboring nodes, $D_p(f_p)$ is referred to as the data cost and $V\left(f_p - f_q\right)$ is the discontinuity or smoothness cost [13]. Data cost is supposed to be low for good matches. As a result, the aim is finding a labeling that minimizes this energy function corresponds to a maximum a posteriori estimation problem [13]. When we make an analogy between the Eq.(2.26) and (2.25), we may see that the label f_p is the flow vector, data cost is related to the data term in Eq.(2.25) and the term $V\left(f_p - f_q\right)$ is related to the third term in the Eq.(2.25). The second term in the Eq.(2.25) is added to apply optical flow's small displacement assumption. The difference of the smoothness term in the energy function of SIFT flow is that it is decoupled because of the benefits in optimization step in Eq. (2.25). There are different kind of approaches to solve such a problem based on Markov Random Fields in literature. The reference [8] prefers the belief propagation method.

Belief Propagation (BP) was proposed by Judea Pearl in 1982 and it is a message passing algorithm for performing inference on graphical models. For instance, belief propagation accomplishes to find an approximate solution for Bayesian networks and Markov random fields based formulations [13]. BP provides an efficient solution for inference problems by propagating local messages around neighboring nodes [14]. BP calculates marginal probabilities of hidden nodes and conditional probabilities of

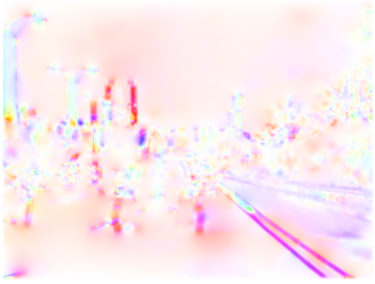
observed nodes by the iterative message passing approach. The messages are updated at each iteration until they converge to a consensus. The consensus obtains the marginal probabilities of all the variables. These estimated marginals are called as beliefs. We think that the most important part of the SIFT flow algorithm which is making a difference is its optimization scheme. Since it was assumed that the neighboring points should have similar displacements, forming the correspondence problem as a MRF was practical. Belief propagation is also a practical method to optimize an energy function based on MRF. In addition, the entity used in matching is chosen as a robust feature. As a result, we may say that SIFT flow is one of the powerful tools for image alignment and it benefits from both the sparse and dense representations. Reconstruction performances of SIFT and optical flow methods are shown in Fig. 2.5.



(a) Anchor frame



(b) Target frame



(c) Optical flow field

(d) SIFT flow field

(e) Optical flow reconstruction

(f) SIFT flow reconstruction

Figure 2.5: Reconstruction results of SIFT flow and optical flow algorithms.

The flow field representation in Fig. 2.5 uses the color codes to plot flow vectors. Magnitude of the flow vectors are represented by saturation and the orientation of the flow vectors are represented by the hue [15]. We can see from the colorful SIFT flow field representation that SIFT flow vectors which are extracted between two frames in Fig. 2.5 have generally similar magnitude and orientation. Therefore, a global camera motion can be obtained from the SIFT flow representation. On the other hand, there are more local flow changes in Horn and Schunck optical flow field. Thus, finding an accurate global motion may not be possible.

2.2 Outlier Rejection and Parameter Estimation

2.2.1 Random sample consensus

Random Sample Consensus (RANSAC) was first proposed by Fischler and Bolles in 1981 as a new paradigm for fitting a model to experimental data. In scene analysis, there are generally two conditions which may cause undesired results. One of them is finding the best match. Fischler and Bolles named the first case as a classification problem [16]. In other words, the feature points which must be consistent across frames are classified according to a model as a best match point or not. In many applications, it is possible to a have significant number of wrong matches after the correspondence matching process. The second problem is finding the best values of model parameters. Fischler and Bolles states that these two problems are not independent [16]. Therefore, the first problem must be handled to solve the second problem. Conventional parameter estimation methods try to optimize their cost functions which are kinds of functional descriptions of the model by using all of the available data. The methods like least squares are not interested in the rejection of erroneous data. The outlier data is assumed to be smoothed by the conventional parameter estimation methods. If feature detector obtains a feature point correctly but it can not find its location correctly, this error is called as measurement error and can be smoothed out. On the other hand, if the feature detector incorrectly finds a feature point, or correspondence matching step matches wrong couples because of condition changes, blur and moving objects in the scene, these kinds of errors have more destructive effects and cannot be smoothed out by an averaging approach [16].

Therefore, outlier points which are not compatible with the model must be eliminated before the parameter estimation process. Random sample consensus method was utilized in this thesis to handle the outliers.

2.2.1.1 RANSAC algorithm

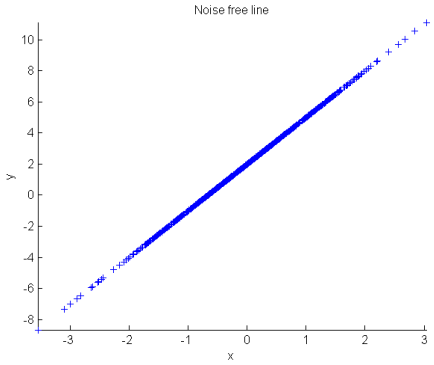
RANSAC randomly selects the minimum number of points that are required to estimate the model at the beginning of the algorithm. An initial model is calculated with the minimum number of points. Then this initial model is used for finding a subset of data points that are consistent with the model. This subset is called as the consensus set and its points are within an error tolerance which is based on Euclidean distance. If the size of the consensus set is larger than a predefined threshold, the final model is calculated by using the points in the consensus set. Otherwise, we return to the beginning of the algorithm and randomly select new points to obtain a new consensus set. The process is repeated until reaching a predetermined number of trials.

The error function was chosen as the Euclidean distance. Therefore, RANSAC tries to find points whose Euclidean distance between the actual points in the target frame and the transformed points from the reference frame is less than a distance threshold. This threshold can be determined heuristically according to the data. The maximum number of trials and the threshold for consensus set size is calculated by considering the number of points and the inlier probabilities. Let p be the desired probability that we have a good final consensus set. p can be calculated as shown is Eq.(2.27).

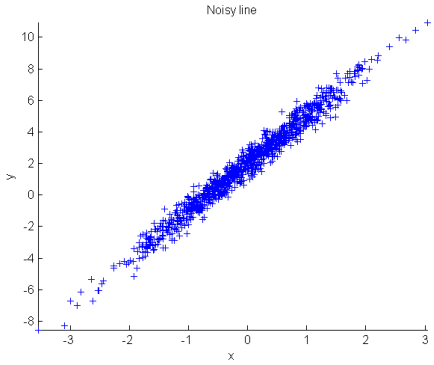
$$1-(1-(1-e)^s)^N = p \tag{2.27}$$

where N is the number of trials, s is the minimum number of points required for the model, and e is the probability that a point is an outlier. Here, $(1-e)^s$ is the probability of choosing inliers for each trial of s draws. $(1-(1-e)^s)^N$ is the probability that we had outliers for the N trials. As a result, $1-(1-(1-e)^s)^N$ is the desired probability that our samples contain inliers. If we derive N from Eq.(2.27), we obtain the expression in Eq. (2.28).

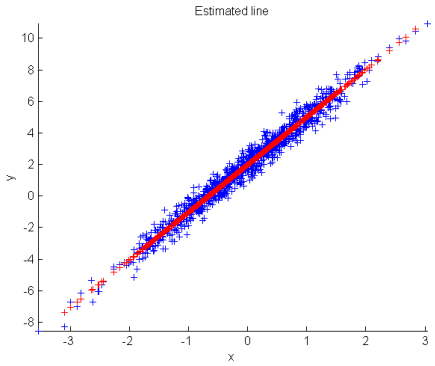
$$N = \frac{\log(1-p)}{\log(1-(1-e)^s)} \tag{2.28}$$



(a) Original line $y = 3x + 2$



(b) Line with zero mean Gaussian noise with 0.5 standard deviation



(c) Estimated line $y = 3.0113x + 1.9947$

Figure 2.6: RANSAC line fitting example.

The threshold for the consensus set size or for early termination of the algorithm can be found by the expression shown in Eq.(2.29).

$$T = (1-e)\,(TotalNumberofDataPoints) \tag{2.29}$$

We may express the term $(1-e)$ as a ratio of the number of inliers over the total number of data points. Therefore, the size of the consensus set should not be less than T. RANSAC is a practical method and it is easy to implement. A line fitting example of RANSAC is given in Fig. 2.6. 500 points are used to constitute a line. The original line, line with Gaussian noise and the estimated line are shown in Fig. 2.6.

RANSAC gives the inlier points that are compatible with the chosen model. Besides, a parameter estimation method is needed to find the model parameters by using the obtained inlier points. Least squares estimation is a common approach for parameter estimation. If data is contaminated by outliers, least squares parameter estimation will generally fail. Least squares estimation method can be expressed as a smoothing process which is not robust enough to outlier points. Therefore, least

squares estimation was utilized after the rejection of outliers with RANSAC. In the following section, least squares estimation of affine parameters is covered.

2.2.2 Least squares estimation of affine parameters

Affine transformation with homogenous coordinates is shown in Eq. (2.30), where x_i, y_i is the point in the reference frame and $x_i^{'}, y_i^{'}$ is the point in the target frame.

$$\begin{pmatrix} x_i^{'} \\ y_i^{'} \\ 1 \end{pmatrix} = \begin{pmatrix} a & b & c \\ d & e & f \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x_i \\ y_i \\ 1 \end{pmatrix} \tag{2.30}$$

Eq. (2.30) can be expressed as shown in Eq. (2.31)

$$\begin{aligned} x_i^{'} &= ax_i + by_i + c \\ y_i^{'} &= dx_i + cy_i + f \end{aligned} \tag{2.31}$$

Error function of least squares estimation is shown in Eq. (2.32).

$$\sum_{i=1}^{n} \left(x_i^{'} - ax_i - by_i - c \right)^2 + \left(y_i^{'} - dx_i - cy_i - f \right)^2 \tag{2.32}$$

The aim is finding the parameters a, b, c, d, e, f which minimizes the error function in Eq. (2.32). Therefore, the partial derivatives with respect to each parameter are taken and equated to zero as follows.

$$\begin{aligned} \frac{\partial E}{\partial a} &= 2\sum_{i=1}^{n} \left(x_i^{'} - ax_i - by_i - c \right)(-x_i) = 0 \\ \frac{\partial E}{\partial b} &= 2\sum_{i=1}^{n} \left(x_i^{'} - ax_i - by_i - c \right)(-y_i) = 0 \\ \frac{\partial E}{\partial c} &= 2\sum_{i=1}^{n} \left(x_i^{'} - ax_i - by_i - c \right) = 0 \\ \frac{\partial E}{\partial d} &= 2\sum_{i=1}^{n} \left(y_i^{'} - dx_i - ey_i - f \right)(-x_i) = 0 \\ \frac{\partial E}{\partial e} &= 2\sum_{i=1}^{n} \left(y_i^{'} - dx_i - ey_i - f \right)(-y_i) = 0 \\ \frac{\partial E}{\partial f} &= 2\sum_{i=1}^{n} \left(y_i^{'} - dx_i - ey_i - f \right) = 0 \end{aligned} \tag{2.33}$$

If Eq. (2.33) is arranged:

$$\begin{aligned}
\sum \left(ax_i^2 + bx_i y_i + cx_i\right) &= \sum x_i^{'} x_i \\
\sum \left(ax_i y_i + by_i^2 + cy_i\right) &= \sum x_i^{'} y_i \\
\sum \left(ax_i + by_i + c\right) &= \sum x_i^{'} \\
\sum \left(dx_i^2 + ex_i y_i + fx_i\right) &= \sum y_i^{'} x_i \\
\sum \left(dx_i y_i + ey_i^2 + fy_i\right) &= \sum y_i^{'} y_i \\
\sum \left(dx_i + ey_i + f\right) &= \sum y_i^{'}
\end{aligned} \tag{2.34}$$

The parameters can be solved by using the following expressions.

$$\begin{pmatrix}
\sum_{i=1}^n x_i^2 & \sum_{i=1}^n x_i y_i & \sum_{i=1}^n x_i & 0 & 0 & 0 \\
\sum_{i=1}^n x_i y_i & \sum_{i=1}^n y_i^2 & \sum_{i=1}^n y_i & 0 & 0 & 0 \\
\sum_{i=1}^n x_i & \sum_{i=1}^n y_i & \sum_{i=1}^n 1 & 0 & 0 & 0 \\
0 & 0 & 0 & \sum_{i=1}^n x_i^2 & \sum_{i=1}^n x_i y_i & \sum_{i=1}^n x_i \\
0 & 0 & 0 & \sum_{i=1}^n x_i y_i & \sum_{i=1}^n y_i^2 & \sum_{i=1}^n y_i \\
0 & 0 & 0 & \sum_{i=1}^n x_i & \sum_{i=1}^n y_i & \sum_{i=1}^n 1
\end{pmatrix}
\begin{pmatrix} a \\ b \\ c \\ d \\ e \\ f \end{pmatrix} =
\begin{pmatrix}
\sum_{i=1}^n x_i^{'} x_i \\
\sum_{i=1}^n x_i^{'} y_i \\
\sum_{i=1}^n x_i^{'} \\
\sum_{i=1}^n y_i^{'} x_i \\
\sum_{i=1}^n y_i^{'} y_i \\
\sum_{i=1}^n y_i^{'}
\end{pmatrix} \tag{2.35}$$

$$\begin{pmatrix} a \\ b \\ c \\ d \\ e \\ f \end{pmatrix} =
\begin{pmatrix}
\sum_{i=1}^n x_i^2 & \sum_{i=1}^n x_i y_i & \sum_{i=1}^n x_i & 0 & 0 & 0 \\
\sum_{i=1}^n x_i y_i & \sum_{i=1}^n y_i^2 & \sum_{i=1}^n y_i & 0 & 0 & 0 \\
\sum_{i=1}^n x_i & \sum_{i=1}^n y_i & \sum_{i=1}^n 1 & 0 & 0 & 0 \\
0 & 0 & 0 & \sum_{i=1}^n x_i^2 & \sum_{i=1}^n x_i y_i & \sum_{i=1}^n x_i \\
0 & 0 & 0 & \sum_{i=1}^n x_i y_i & \sum_{i=1}^n y_i^2 & \sum_{i=1}^n y_i \\
0 & 0 & 0 & \sum_{i=1}^n x_i & \sum_{i=1}^n y_i & \sum_{i=1}^n 1
\end{pmatrix}^{-1}
\begin{pmatrix}
\sum_{i=1}^n x_i^{'} x_i \\
\sum_{i=1}^n x_i^{'} y_i \\
\sum_{i=1}^n x_i^{'} \\
\sum_{i=1}^n y_i^{'} x_i \\
\sum_{i=1}^n y_i^{'} y_i \\
\sum_{i=1}^n y_i^{'}
\end{pmatrix} \tag{2.36}$$

The points which were used in least squares estimation are the inlier points which are obtained by RANSAC algorithm.

3. MOTION COMPENSATION

In the previous section, motion between adjacent frames was found as the SIFT flow vectors, and the geometric transformation between adjacent frames was represented as an affine transformation matrix. In the motion compensation step, frames will be warped by using the motion model to obtain a smoother video. As it was explained in the previous section, RANSAC and the least squares estimation were used to find the affine transformation parameters. However, RANSAC may have problems in eliminating some outlier points in real world videos. Therefore some additional steps are needed to enhance the robustness of RANSAC. Moreover, there are also some cases where matching failures are inevitable and using the whole affine transformation matrix will not be practical. The approach utilized to obtain and compensate problematic frame couples is explained in this section.

3.1 Background Point Selection

Although RANSAC is a practical tool for rejecting outlier points, some outlier points especially those on moving objects may not be eliminated by RANSAC and these points usually deteriorate the parameter estimation results. Therefore, such points were excluded in the RANSAC step in this thesis. Moving objects generally belong to the foreground objects of videos. On the other hand, background points are assumed to be the points which are stationary in a video. Moreover, background points are supposed to be affected only by the global camera motion. Therefore, using the background points in the parameter estimation step may yield more accurate results. By considering this assumption, the points that are in the central zone of the frames are assumed as foreground points and are not used in the RANSAC procedure [4]. The foreground area is determined heuristically. At this point, background points can be assumed to be the points that are compatible with the motion model, and the foreground points can be thought as outliers. After finding the affine model between the first two adjacent frames by using the initial background points, other possible points that fit

the model are searched among the set of foreground points. Since the foreground objects may disappear in time, foreground points are tested whether they fit the model at each step before proceeding to the next adjacent frame. In addition, the background points are also tested in case of an incompatibility with the current model, and the background points which do not fit the model are discarded. Then updated background points are used in the RANSAC process for the new frame couples. The test used in the foreground-background point determination is chosen the same as the Euclidean distance test of RANSAC. An example of the moving object case is shown in Fig. 3.1. In Fig. 3.1, there is no camera motion so the global affine transformation matrix should be an identity matrix. The transformation matrices obtained with and without background point selection are given in Eq. (3.1).

(a) Frame 1

(b) Frame 2

Figure 3.1: Moving object example.

$$M = \begin{pmatrix} 0.96 & 0.03 & 35.8238 \\ 0 & 0.98 & 0.2298 \\ 0 & 0 & 1 \end{pmatrix} M_{bg} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0.99 & 0.01 \\ 0 & 0 & 1 \end{pmatrix} \quad \textbf{(3.1)}$$

where M is the affine transformation matrix that was found without background point selection and M_{bg} is the affine transformation that was obtained by using background point selection. The moving object occupies a significant area in the frame in Fig.3.1. In spite of the stationary camera, there is a serious horizontal translation component calculated in the transformation matrix M. On the other hand, M_{bg} is a more desired result. In conclusion, the points on the moving object tend to be outliers that RANSAC is not able to deal with. Therefore, a background point selection is needed to deal with these kinds of situations. In the next step, frames are warped by using the affine transformation matrix found previously.

3.2 Warping the Frames

In the video stabilization scheme of this thesis, frames are processed as couples through the entire video. At the beginning of the video, the SIFT flow vectors are extracted between the first and the second frames. The affine transformation matrix is estimated by using flow vectors, background point selection, RANSAC and least squares parameter estimation respectively. The aim is warping the target (second) frame in this study. Therefore, the transformation matrix is applied to the target frame by using bilinear interpolation. In the first step, the reference frame which was the first frame of the original video and the warped target frame which was the second frame of the original video are the outputs as the first two frames of the stabilized video. In the next step, the reference frame is chosen as the compensated frame found in the previous step and the target frame becomes the third frame of the original video, and so on. Since we are matching stabilized frames with the original frames, any errors in the transformation matrix may accumulate through the following frames and undesired stabilized frames may eventually result. Problematic transformation matrices may be because of wrong matches. One of the situations which leads us to failures in the motion estimation is blur in the frames. If frames have severe blur due to the sudden camera motion, we may suffer from wrong matches. Since the effect of a failure in the affine transformation matrix may be quite destructive, using a simple translational

model can be reasonable in such blurry cases. For this reason, we need to identify the blurry frames before applying a suitable transformation.

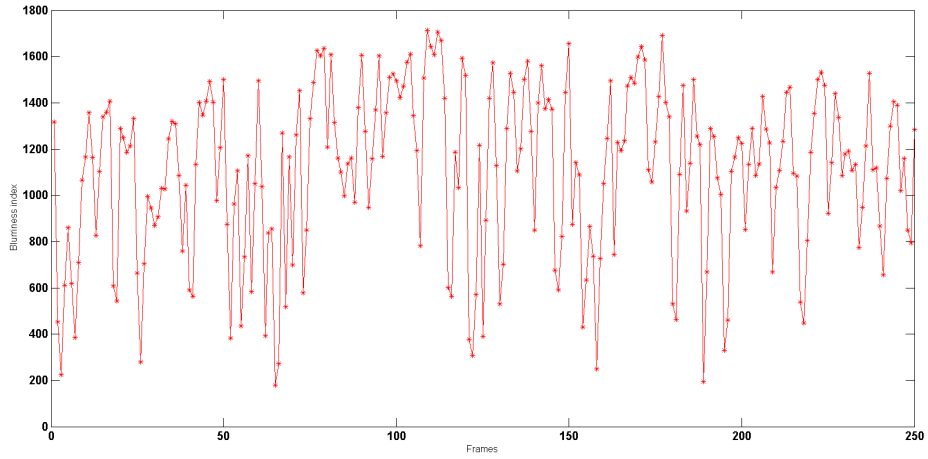
3.3 Detection of Motion Blur

The unintentional camera movements cause blur in video frames. This is a similar to the blur effect on moving objects in a video. The sudden motion of an object produces partial blurs in the frame. However, a sudden hand shake of the user or if the video is recorded in a moving car, a bounce of the car, produces a blur that affects the entire frame. We may figure out the severity of such unintentional movements from the amount of the motion blur in dealing with this situation. Therefore, a criteria is used to decide whether a frame has a severe blur. For this purpose, [17] and [18] proposed a method based on the frame gradients to obtain a measure of the blur. The criteria is called the blurriness index and is given in Eq. (3.2) [18].

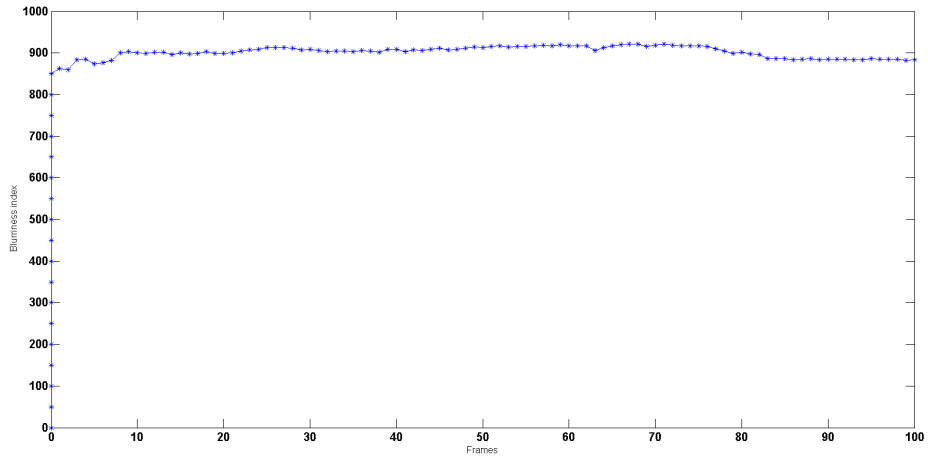
$$B_i = \sum_{p_i} \left(g_x^2 \left(p_i \right) + g_y^2 \left(p_i \right) \right) \tag{3.2}$$

where p_i are the pixels of ith frame, g_x and g_y are the horizontal and vertical gradients. The blurry frames have smaller blurriness indexes than less blurry or sharper frames [17]. Thus, we are able to use this fact to decide the blurriness of frames, but the value of the blurriness index varies depending on the amount of the blur. Therefore, a threshold is beneficial to come up with a final decision about the blurriness decision of frames. The aforementioned threshold is chosen as the mean of the blurriness index values of all frames in the video. Experiments showed that frames, whose blurriness index is less than the mean value, have a blur that can not be ignored. On the other hand, frames, whose blurriness index is greater than the mean value, are either sharp frames or have less blur. The noteworthy point about this step is that the purpose of the detection of blurry frames is not removing the blur from the frames in this thesis. The main reason of making a decision about the blur is to be able to choose a suitable transformation matrix to warp the frames. If both adjacent frames or one of them has a blurriness index less than the threshold, we use only the translations to compensate for the unwanted motion. On the other hand, if both frames have low blur or are sharp, we use the whole affine transformation matrix to warp the frames. Plots of blurriness

index of a shaky video and a video recorded with a stable camera through frames are shown in Fig. 3.2.



(a) Shaky video



(b) Stable video

Figure 3.2: Blurriness indexes of a shaky and a stable video.

The examples in Fig. 3.2 may differ for other videos with different conditions, but we can see from the plots that shaky videos may have very blurry frames because of unwanted camera movements. When there is an unwanted camera motion, the blur in a frame increases and the blurriness index decreases. If the amount of the jitter is slight, there will not be serious blur and the blurriness index increases again. Therefore, a blurriness index plot for a shaky video with respect to the frames has ups and downs. However, blurriness index values of a stable video are prone to be smoother through frames. Since there may be condition changes and moving objects

in videos, a constant blurriness index value for all frames of a stabilized video is not plausible but we observe a smoother blurriness index plot as given in 3.2 (b).

3.4 Discriminating Intentional Camera Motion

In real videos, there are also intentional camera movements. The intentional camera motion may contain pan, tilt and roll. Pan is caused by the horizontal movement of the camera and the vertical movement of the camera gives us the tilt. This intentional movement of the camera produces a motion that has different properties than the unwanted motion. The unintentional shaky movements cause random high frequency jitters, whereas the intentional camera motion generally has a smoother and continuous characteristics. For instance, panning the camera produces a smooth flow field with horizontal flow vectors. If we look at the affine parameters of the frames of a panning camera with our video stabilization scheme, we see that the horizontal translation parameter increases continuously in one direction during the pan. If our video stabilization program did not try to discriminate the camera pan, it would compensate this increasing horizontal motion up to a point and after that point the translation parameter would jump back for one frame and then begin to increase again. Therefore, we would not lose the frame totally by the increasing translational motion compensation but an undesirable temporal discontinuity would occur during the pan. This is an expected result, unless there is a control mechanism to recognize the intentional motion. Discriminating the camera motion as intentional or unintentional is not an easy task. There are no certain rules for the pan in real videos. A conscious user can be careful about panning the camera slowly. In this case, following the pan path by a video stabilization program will be straightforward. On the other hand, there may be a sudden change in the subject of interest of the user and the user may turn the camera very fast. In this case, there will be an abrupt scene change and recognizing this fast pan will be troublesome. In this thesis, we assume that the pan movement is smooth and consider that if the translational motion begins to increase in one direction monotonically, we may make a decision about the existence of the camera pan. Another complexity about pan is that some unwanted movements may behave like pan for several frames. Therefore, the algorithm may give a false alarm and the shaky frames are processed with the assumption of pan. We prefer to

process the frames in groups to realize the monotonically increasing pan translations. The size of the groups is chosen heuristically and large enough to avoid false alarms. If we find out the pan for a group of translation parameters, we do not use the original values of the translation parameters but we smooth them. Thus, the program does not compensate for the intentional pan and follows this intentional camera movement as closely as possible.

In conclusion, the video consisting of the stabilized frames is the final output of the video stabilization method of this thesis. Stabilized videos are supposed to have transitions as smooth as possible in this thesis.

4. EXPERIMENTAL RESULTS

In the experiments, videos are recorded by a 8 mega pixel Canon PowerShot SX100 IS hand held digital video camera. The SIFT flow algorithm implementation provided in [19] is used. The evaluation of the algorithms is performed on a PC with Intel Core i7, 2.8 GHz CPU with 16GB of RAM. The video stabilization algorithm developed in this thesis is run on MATLAB R2012b. The processing time to stabilize one frame is 20.5 seconds. The proposed approach is tested with several videos. Videos are divided into two groups as videos recorded with a stationary camera and those recorded with a mobile camera. It is clear that a video stabilization program should not attempt to stabilize the frames of a video recorded with a stationary camera. In the stationary camera case, there are also two subcases in which the objects move or are stationary. Although there is no global camera motion for the stationary camera videos, the moving objects may confuse the algorithm and cause problems with the video stabilization program. In other words, an affine transformation matrix different than identity may be found due to the outlier points on the moving objects. The effect of the moving objects in the motion parameter estimation stage depends on the size of the objects. If the moving objects occupy a large area in the frame, their deteriorating effect will be more significant than for small objects. In addition, even if no objects are moving, the video stabilization program should also be robust to lighting changes. In the mobile camera case, the video stabilization program needs to find an accurate global camera motion. Similarly, the stabilization program is supposed to be robust to the small or large moving objects.

4.1 Stationary Camera

As it was mentioned before, if our camera is not moving, we can not talk about any global transformation between frames. As a result, any movement of the pixels is only because of the local motion of objects. The video stabilization program is supposed to ignore the movements of the objects, and there should not be any global compensation

of the frames. Two examples are given in this section. The first video has moving objects with various sizes; the second video has stationary objects but the lighting changes. The mean of the global transformation matrices calculated per frame pairs in the videos with the moving objects and with the stationary objects are given in Eq.s (4.1),(4.2), respectively. Example frames from these videos are shown in Fig.s 4.1 and 4.2.

Figure 4.1: Three frames of the video with moving objects.

$$M_{moving} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \tag{4.1}$$

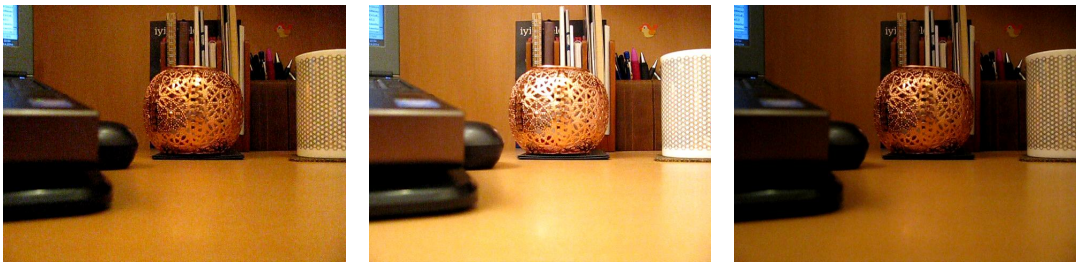


Figure 4.2: Stationary camera with lighting changes.

$$M_{stationary} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \tag{4.2}$$

RANSAC parameters used in these experiments are given in Table 4.1. The distance threshold plays a significant role. For instance, a greater distance threshold will not be enough to eliminate some outlier points because of the moving objects. The outlier probability may be chosen smaller. In this case, the maximum number of trials given in Eq.(2.28) will be greater. Thereby, the processing time per frame pair will be greater. The desired probability is generally chosen as 0.99 and the degree of freedom of the model is the minimum number of points needed to calculate the model parameters. In this case, our model is the 6 parameter affine transformation, so the degree of freedom is 3.

Table 4.1: RANSAC parameters.

Desired probability (p)	0.99
Outlier probability (e)	0.5
Degree of freedom of the model (s)	3
Euclidean distance threshold (T)	0.1

In conclusion, our video stabilization program was validated for being robust to illumination changes and moving objects in the stationary camera case. SIFT flow algorithm and RANSAC with background point selection provide us this robustness.

4.2 Moving Camera

In the moving camera case, the main purpose is estimating a global transformation between frames and compensating it such that the unwanted camera movements will be reduced. The video stabilization program of this thesis is tested by various videos with shaky movements. One of the main causes of unwanted motion is a moving vehicle like a car, bicycle or helicopter. Some videos recorded in moving cars, and videos recorded by users with shaking hands are used to test the video stabilization program. The explanations of the test videos are given below.

Car1 is a video recorded in a car. Since the car is going across the Bosphorus Bridge, there is a severe oscillatory movement in the video. Hence, *Car1* video is a very challenging case for a video stabilization implementation. The video has 250 frames with a resolution 640x480 and a frame rate of 30 fps. Example consecutive frames of this test video and the corresponding stabilized pairs are shown under the original frames in Fig 4.3.

Car2 is also a video recorded in a car. This case is not as challenging as the *Car1* case. The car keeps tracking the same lane and the motorway is quite smooth. There is unwanted motion because of the movement of the car and the hand shake of the user. There are also other cars that are changing lanes in the video. We may think of the other cars as moving objects. The video has 310 frames with a resolution 640x480 and a frame rate 30 fps. Example consecutive frames of this test video and the corresponding stabilized pairs are shown under the original frames in Fig 4.4.

Car3 video is also recorded in a car from the side window and has very fast pan like motion. In this video, we see the road side and the relative motion of the objects

(a) Original frame1 (b) Original frame2 (c) Original frame3

(d) Stabilized frame1 (e) Stabilized frame2 (f) Stabilized frame3

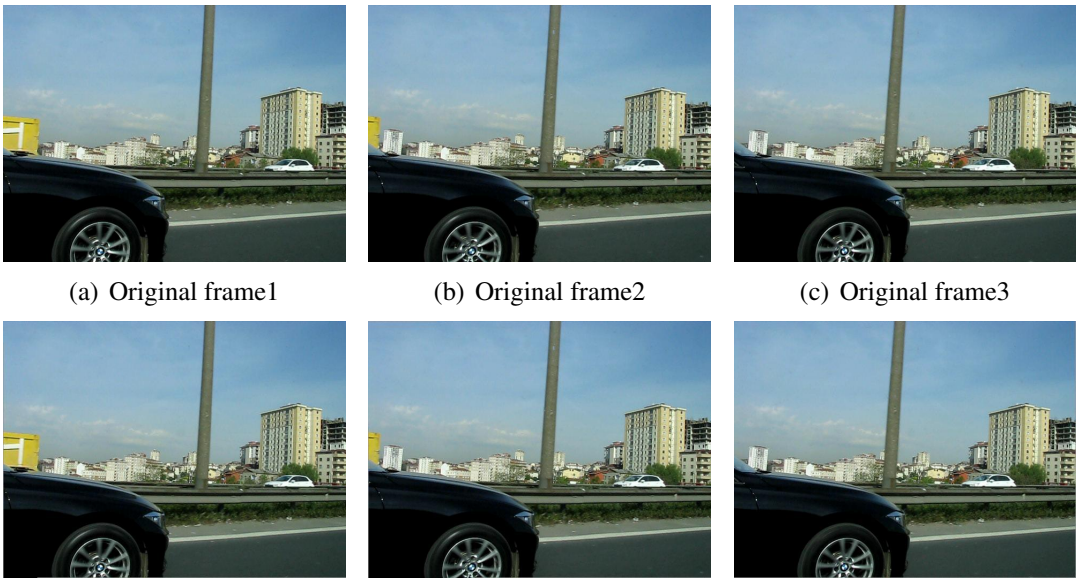
Figure 4.3: Stabilization result of *Car1*.

(a) Original frame1 (b) Original frame2 (c) Original frame3

(d) Stabilized frame1 (e) Stabilized frame2 (f) Stabilized frame3

Figure 4.4: Stabilization results of *Car2*.

like streetlights, buildings or other cars passing by. There is also the parallax effect which is a common situation for roadside videos. The frames of the *Car3* video has a continuous motion, some small shaky movements, and the objects are moving very fast during the recording. Therefore, this video is also a difficult case. Our algorithm does not attempt to make a serious compensation in this case, and the video stabilization program does not damage the original video with erroneous global transformations. Original frames and the output frames of the simulation can be seen in Fig. 4.5.



(a) Original frame1 (b) Original frame2 (c) Original frame3

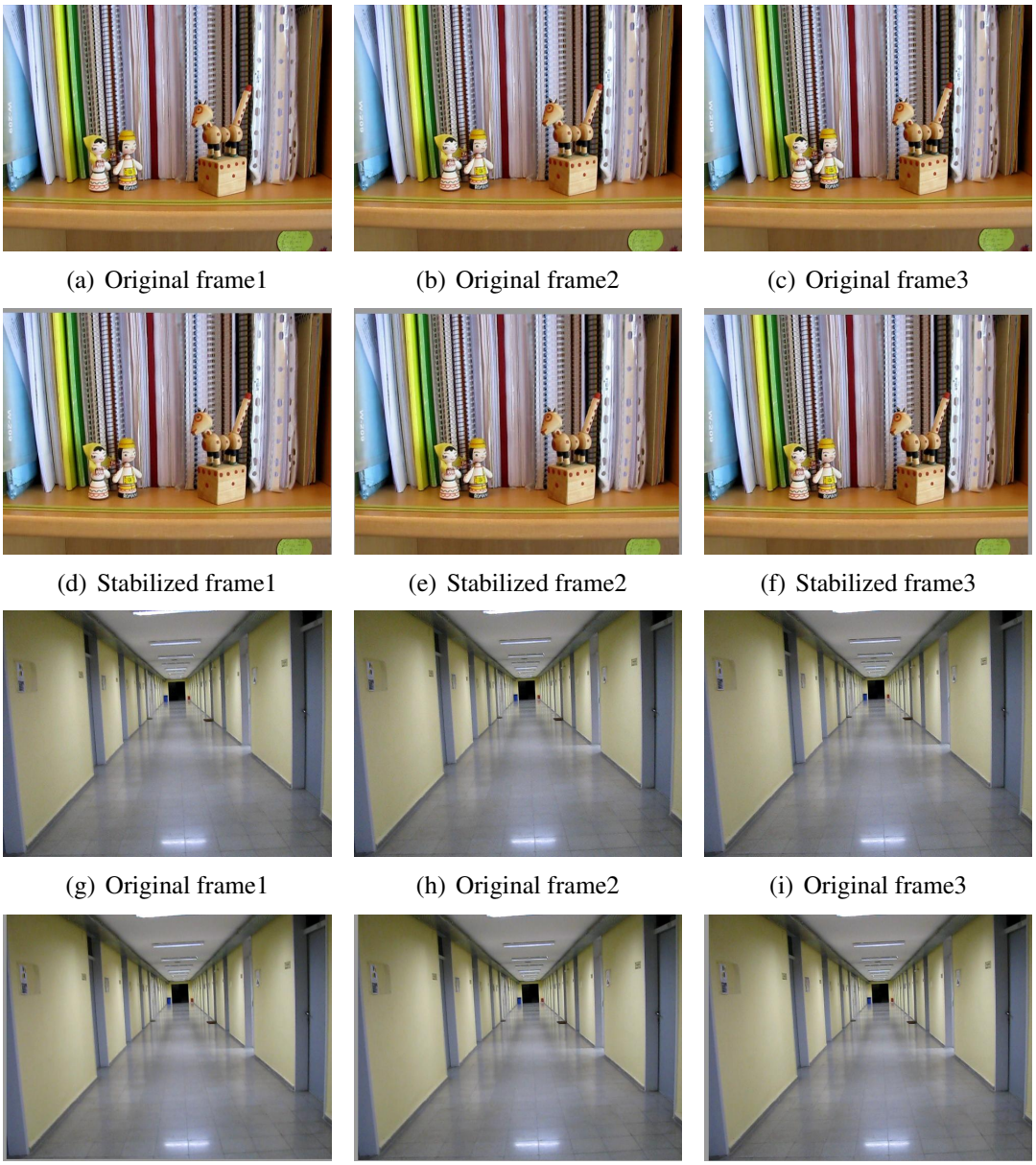
(d) Stabilized frame1 (e) Stabilized frame2 (f) Stabilized frame3

Figure 4.5: Stabilization result of *Car3*.

In the *Hand shake1*, the unwanted motion is due to the user's movement and hand shake. There are no objects in motion in the *Hand shake1* video. This video has also 191 frames with a resolution of 640x480 and a frame rate 30 fps. *Hand shake2* has similar properties as *Hand shake1*. This test video is recorded by a user walking down a corridor. *Hand shake2* video has 281 frames with a resolution of 640x480 and a frame rate 30 fps. Example consecutive frames of *Hand shake1* and *Hand shake2* videos and the corresponding stabilized pairs are shown under the original frames in Fig 4.6.

Finally, *Pan Camera* is recorded by a user walking down the street and there is a pan towards the end of the video. The video has 117 frames with a resolution 640x480 and a frame rate of 30 fps. Example consecutive frames of this test video and the corresponding stabilized pairs are shown under the original frames in Fig 4.7.

The horizontal translation variations with respect to frames are also shown in Fig 4.8.



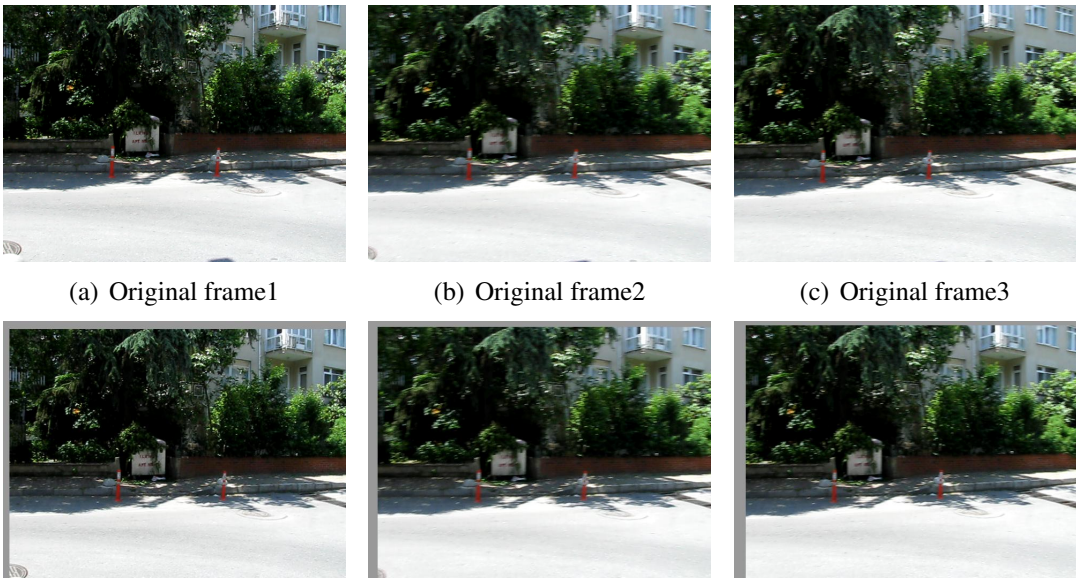
(a) Original frame1 (b) Original frame2 (c) Original frame3

(d) Stabilized frame1 (e) Stabilized frame2 (f) Stabilized frame3

(g) Original frame1 (h) Original frame2 (i) Original frame3

(j) Stabilized frame1 (k) Stabilized frame2 (l) Stabilized frame3

Figure 4.6: Stabilization results of *Hand shake1* and *Hand shake2*.



(a) Original frame1 (b) Original frame2 (c) Original frame3

(d) Stabilized frame1 (e) Stabilized frame2 (f) Stabilized frame3

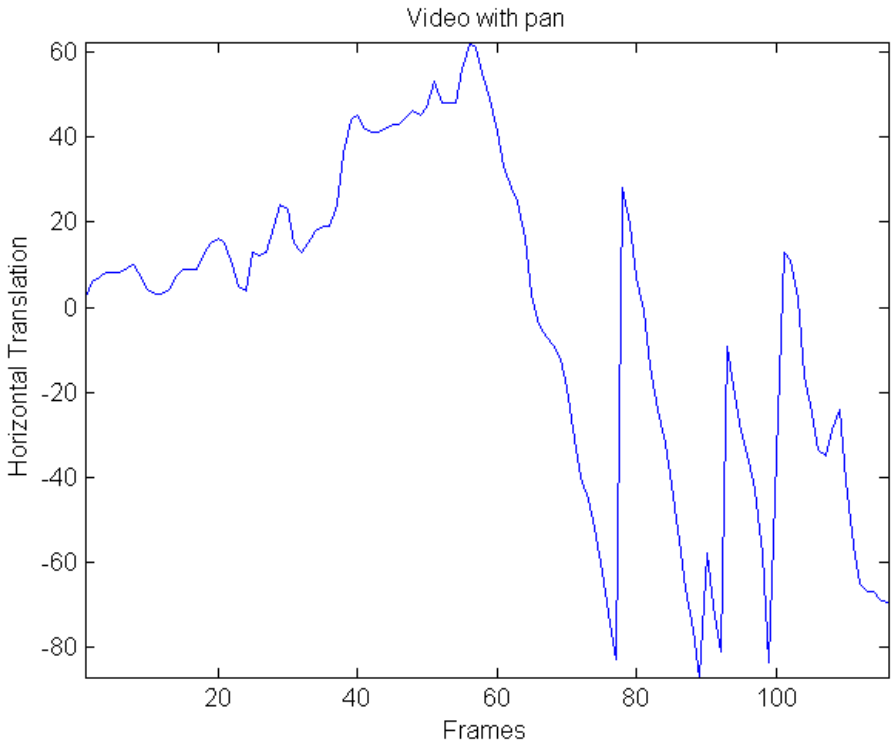
Figure 4.7: Stabilization result of *Pan Camera*.

We can observe from the Fig. 4.8 (a) that the camera pan begins after approximately the 60th frame. As it was explained in the previous section, the horizontal translation parameter increases continuously up to a point and the compensation takes back the frame at that point. Then, the translation begins to increase again. This sawtooth like pattern continues during the camera pan. On the other hand, the sawtooth pattern becomes smoother when we identify the camera pan and track it as shown in 4.8 (b).

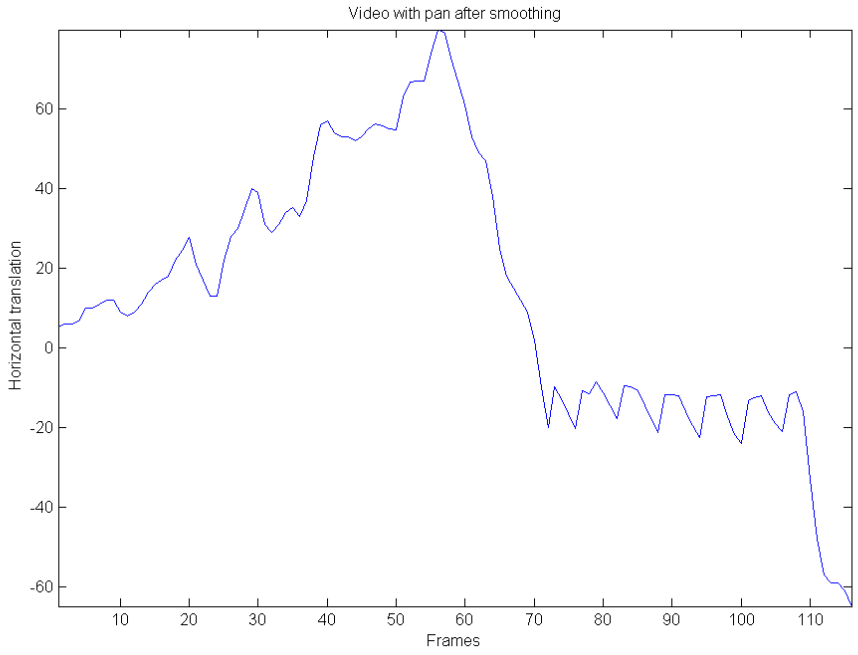
Herein, a criterion is needed to test the accuracy of the video stabilization program. The inter-frame transformation fidelity (ITF) measure is used to test the performance of the video stabilization. ITF measures the temporal smoothness [2]. The expression for ITF is given in Eq. (4.3).

$$
\begin{aligned}
ITF &= \frac{1}{N_{frame}-1}\sum_{k=1}^{N_{frame}-1} PSNR(k) \\
PSNR(k) &= 10log_{10}\left(\frac{I_{max}^2}{MSE(k)}\right) \\
MSE &= \frac{1}{MN}\sum_{i=1}^{M}\sum_{j=1}^{N}\left[I_1(i,j)-I_2(i,j)\right]^2
\end{aligned}
\tag{4.3}
$$

where I_{max} is the maximum pixel intensity, M and N are the height and the width of the images, MSE is the Mean Square Error, $PSNR$ is the Peak Signal to Noise Ratio and N_{frame} is the total number of frames in the video. PSNR measures the similarity between consecutive frames. Hence, the ITF value of a shaky video is expected to



(a) Horizontal translation variation of the original video



(b) Horizontal translation variation of the stabilized video

Figure 4.8: Horizontal translation variations of the original and the stabilized frames of *Pan Camera* video.

increase after applying the video stabilization program. This evaluation method is based on the fact that the difference between stabilized frame and the reference frame should get small after motion compensation. Obviously, the difference cannot be zero because of the pixels which belong to moving objects, and yet the difference between adjacent frames will decrease after reducing the unwanted global camera motion. The ITF results of the test videos are given in Table 4.2.

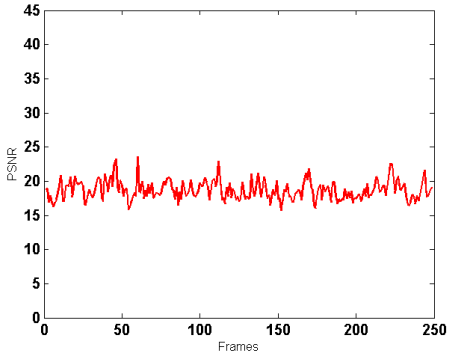
Table 4.2: ITF results of the proposed video stabilization scheme.

Video	ITF of Input Video	ITF of Stabilized Video
Car1	18.62 dB	22.18 dB
Car2	21.60 dB	25.68 dB
Car3	18.55 dB	18.55 dB
Hand shake1	19.03 dB	24.90 dB
Hand shake2	25.20 dB	32.10 dB
Pan Camera	17.18 dB	20.55 dB

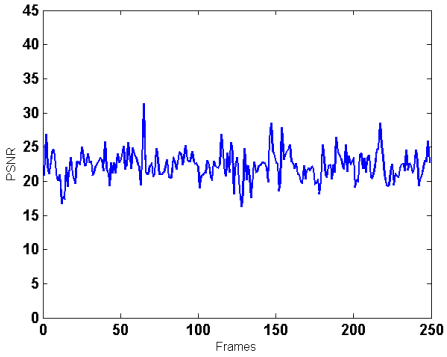
In Table 4.2, we can see that our video stabilization program is capable of increasing the mean PSNR of the test videos except for the *Car3* test video. The PSNR variations of frames for each test video except *Car3* are also plotted in Fig 4.9. PSNR values are generally greater in the stabilized videos.

One of the noteworthy point about the SIFT flow algorithm is that its flow vectors may be suitable for a simple k-means clustering. When we examine the histograms of the magnitude and the orientation of the SIFT flow vectors, we may deduce that flow vectors can be divided into clusters. If we can reach two clusters of flow vectors as outliers and inliers, we may use the cluster of inliers in the RANSAC stage. However, clustering does not look reasonable for the histograms of a real life video with affine changes. More than two clusters usually occur and the size of the clusters are not distinctive enough to choose one of them as the cluster of inliers. Even though we may assume that the cluster with the largest size will provide us with the global transformation, a proper affine transformation can not be established. As a result, k-means clustering of SIFT flow vectors is decided as inappropriate for our implementation.

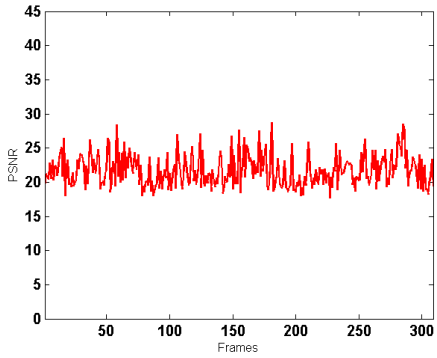
Another deduction from the experiments was about the iterative motion smoothing scheme that is especially preferred in [2] and [17]. Iterative smoothing scheme does not take a reference frame. For instance, if we consider a frame in the middle of the



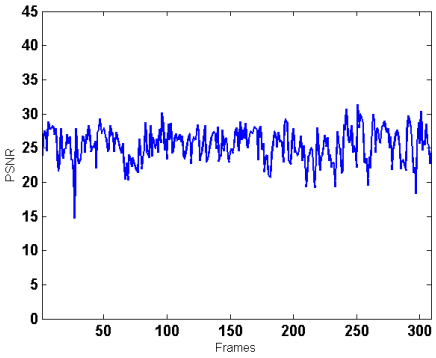
(a) Original Car1 Video



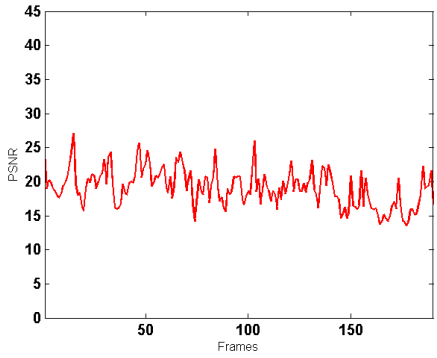
(b) Stabilized Car1 Video



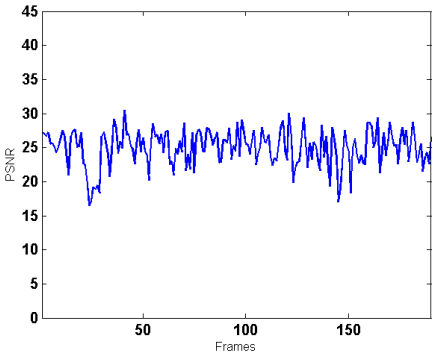
(c) Original Car2 Video



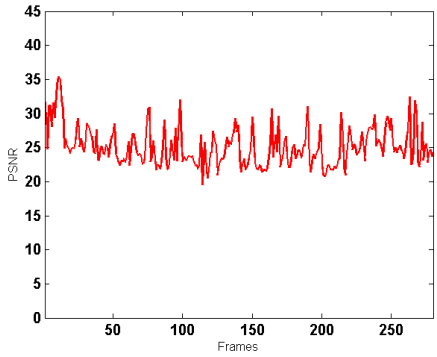
(d) Stabilized Car2 Video



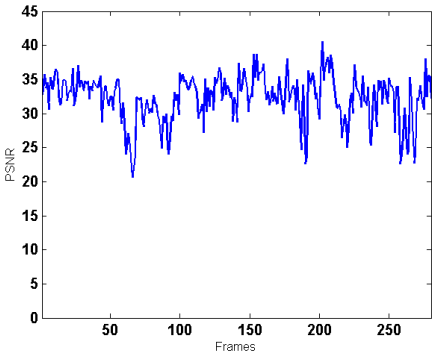
(e) Original Hand Shake1 Video



(f) Stabilized Hand Shake1 Video



(g) Original Hand Shake2 Video



(h) Stabilized Hand Shake2 Video

Figure 4.9: PSNR variations with respect to frames.

video, iterative smoothing calculates transformation matrices between the k frames before and after the corresponding frame. The relevant transformations are smoothed with a Gaussian kernel whose standard deviation is obtained according to the distance of frames to the center frame and the smoothed transformations are applied to the applicable frame. In the video stabilization implementation of this thesis, iterative smoothing was seen as a time consuming method with no worthwhile return. In addition, the first and the last frames of the video cause problems in this approach, since k is generally chosen greater then one frame in [2] and [17]. Therefore, the compensation of the first and the last frames is not sufficient for a stabilization purpose. Therefore, the iterative motion smoothing scheme is not used in this thesis.

5. CONCLUSIONS AND RECOMMENDATIONS

In this thesis, the flow field is estimated by a new image alignment method. SIFT flow has similarities as well as differences with SIFT and the optical flow methods. SIFT flow benefits from the advantages of both the dense and the sparse representation methods. Its algorithm establishes a pixel level correspondence, and SIFT flow is said to outperform the traditional sparse representation methods [8]. Sparse representations are famous for their robustness to condition changes. Since SIFT flow utilizes the SIFT descriptors in matching, the desired robustness gain is substantial. However, we should not forget that SIFT flow does not follow all the feature extraction steps of the traditional SIFT method. For instance, the scale space extrema detection is not performed in SIFT flow. Nevertheless, the pixel-wise SIFT descriptors are robust to lighting changes as indicated in the previous section.

As explained in previous sections, SIFT flow adopts the optical flow approach in its matching scheme. Both methods have pixel-wise correspondence, similar terms in their energy functions, and similar coarse to fine matching schemes. An important diversity in SIFT flow is that the SIFT flow vectors establish a pixel level accuracy, whereas the optical flow is able to provide subpixel precision. Since we get better results with subpixel precision, low pass filtering of flow vectors is applied to cope with this situation in this thesis.

[8] states that SIFT flow cannot take the place of optical flow methods. However, we believe that SIFT flow is a practical tool for offline video stabilization, but it may not be suitable for a real time video stabilization application because of the processing time. The computational load of SIFT flow field estimation process increases with larger matching window sizes. Therefore, the window size is chosen smaller than the default value of the software provided in [19].

After obtaining the flow vectors, outlier points are tried to be eliminated by RANSAC and background point selection. Motion compensation is applied to consecutive frames using the affine motion model parameters estimated by the least squares method. The

steps of applied motion compensation are designed to deal with error accumulation. As a result, undesired and disturbing motion in videos is eliminated as much as possible. Moreover, intentional pan camera motion is tried to be identified and the path of the pan motion is tried to be followed without compensation.

For future work, outlier elimination step may be developed to reach a higher robustness and accuracy. The blank border zones of the stabilized frames may be filled by using motion inpainting. The blur in the original frames due to motion of the camera remains in the stabilized video, since we are warping the original frames. In addition, the blur in stabilized videos may become more annoying than original videos. Hence, image deblurring may be applied to stabilized frames to produce stabilized and also enhanced frames. The control mechanism for discriminating the intentional and unintentional camera motion can be developed to handle the different kind of pan characteristics. The real time video stabilization problem is another challenging task which is still open to future improvements.

REFERENCES

[1] **Rawat, P. and Singhai, J.**, 2012. Hand Held Mobile Video Stabilization Using Differential Motion Estimation, Proceedings of the International Conference on Soft Computing for Problem Solving (Socpros 2011), Vol 2, pp.463–472.

[2] **Xu, J.**, **Chang, H.W.**, **Yang, S. and Wang, M.H.**, 2012. Fast Feature-Based Video Stabilization without Accumulative Global Motion Estimation, *Ieee Transactions on Consumer Electronics*, **58(3)**, 993–999.

[3] **Song, C.H.**, **Zhao, H.**, **Jing, W. and Zhu, H.B.**, 2012. Robust Video Stabilization Based on Particle Filtering with Weighted Feature Points, *Ieee Transactions on Consumer Electronics*, **58(2)**, 570–577.

[4] **Kim, S.K.**, **Kang, S.J.**, **Wang, T.S. and Ko, S.J.**, 2013. Feature Point Classification Based Global Motion Estimation for Video Stabilization, *Ieee Transactions on Consumer Electronics*, **59(1)**, 267–272.

[5] **Li, C.**, **Liu, Y.K. and Ieee**, 2012. Global Motion Estimation Based on SIFT Feature Match for Digital Image Stabilization, 2011 International Conference on Computer Science and Network Technology (Iccsnt), Vols 1-4, pp.2264–2267.

[6] **Xu, Y.C. and Qin, S.Y.**, 2010. A New Approach to Video Stabilization with Iterative Smoothing, 2010 Ieee 10th International Conference on Signal Processing Proceedings (Icsp2010), Vols I-Iii, pp.1224–1227.

[7] **Ejaz, N.**, **Kim, W.**, **Kwon, S.I. and Baik, S.W.**, 2012. Video Stabilization by Detecting Intentional and Unintentional Camera Motions, Third International Conference on Intelligent Systems Modeling and Simulation.

[8] **Liu, C.**, **Yuen, J. and A., T.**, 2011. SIFT Flow: Dense Correspondence across Scenes and its Applications, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **33(5)**, 978–994.

[9] **Lowe, D.G.**, 2004. Distinctive Image Features from Scale Invariant Keypoints, *International Journal of Computer Vision*, **60(2)**, 91–110.

[10] **Tekalp, A.M.**, 1995. Digital Video Processing, Prentice Hall Signal Processing Series.

[11] **Horn, B.K.P. and G., S.B.**, 1981. Determining Optical Flow, *Artificial Intelligence*, **17(1-3)**, 185–203.

[12] **Yuan, D.**, `http://http://nghiaho.com/?page_id=1366`.

[13] **Felzenszwalb, P.F. and Huttenlocher, D.P.**, 2006. Efficient Belief Propagation for Early Vision, *International Journal of Computer Vision*, **70(1)**, 41–54.

[14] **Yuan, D.**, `http://classes.soe.ucsc.edu/cmps290c/Spring04/proj/BPApp.pdf`.

[15] **Liu, C.**, `http://http://people.csail.mit.edu/torralba/courses/6.869/lectures/lecture19/lecture19.pdf`.

[16] **Fischler, M.A. and Bolles, R.C.**, 1981. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography, *Communications of the ACM*, **24(6)**, 381–395.

[17] **Matsushita, Y.**, **Ofek, E.**, **Ge, W.**, **Tang, X. and Shum, H.**, 2006. Full frame video stabilization with motion inpainting, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **28(7)**, 1150–1163.

[18] **Okade, M. and Biswas, P.K.**, 2011. Improving video stabilization in the presence of motion blur., 2011 Third National Conference on Computer Vision, Pattern Recognition, Image Processing and Graphics (NCVPRIPG), pp.78–81.

[19] **Liu, C.**, **Yuen, J. and A., T.**, `http://people.csail.mit.edu/celiu/SIFTflow/`.

CURRICULUM VITAE

Name Surname: İnci Meliha Baytaş

Place and Date of Birth: İstanbul January 01, 1990

Adress: Feneryolu Mah. Yıldıray Sok. Nur Apt. 9/11, Kadıköy/İstanbul

E-Mail: baytasi@itu.edu.tr

B.Sc.: Istanbul Technical University, Faculty of Electrical and Electronics Engineering, Electronics and Communication Engineering Department, Telecommunication Engineering Program

M.Sc.: Istanbul Technical University, Graduate School of Science Engineering and Technology, Electronics and Communication Engineering Department, Telecommunication Engineering Program

Professional Experience and Rewards: Research Asistant in Istanbul Technical University, Faculty of Electrical and Electronics Engineering, Electronics and Communication Engineering Department (since December 2013), Graduate Scholarship of the National Scholarship Programme for MSc Students of The Scientific and Technological Research Council of Turkey (2012-2014), Ranked 2nd in Faculty of Electrical and Electronic Engineering and Electronics and Communication Engineering Department.

List of Publications and Patents: **Baytaş İ.M.**, Günsel B., Head Motion Classification with 2D Motion Estimation, *IEEE 22nd Signal Processing and Communication Applications Conference (SIU)*, April 23-25, 2014 Trabzon, Turkey (Presented).