MATCHING DAY AND NIGHT LOCATION IMAGES USING SIFT AND LOGISTIC REGRESSION

by

Nazlı TEKİN

A thesis/dissertation submitted to

The Graduate Institute of Sciences and Engineering

of

Meliksah University

in partial fulfilment of the requirements for the degree of
Master of Science of Engineering

in

Electrical and Computer Engineering

August, 2015
Kayseri, Turkey

APPROVAL

This is to certify that I have read the thesis titled "Matching Day and Night Location Images Using SIFT and Logistic Regression" by Nazlı TEKİN and that in my opinion it is fully adequate, in scope and quality, as a thesis for the degree of Master of Science in Electrical and Computer Engineering, the Graduate Institute of Science and Engineering, Melikşah University.

____

August 11, 2015 Asst. Prof. Dr. Kadir Aşkın Peker

Supervisor

I certify that this thesis satisfies all the requirements as a thesis for the degree of Master of Science.

____

August 11, 2015 Prof. Dr. M. Murat UZAM

Head of Department

<u>Examining Committee Members</u>

Title and Name		Approved
Asst. Prof.Dr. Kadir Aşkın PEKER	August 11, 2015	__________________
Assoc. Prof.Dr. Ahmet UYAR	August 11, 2015	__________________
Assoc.Prof.Dr.Alper BAŞTÜRK	August 11, 2015	__________________

It is approved that this thesis has been written in compliance with the formatting rules laid down by the Graduate Institute of Science and Engineering.

____

Asst. Prof. Dr. M. Evren SOYLU

Director

August 2015

MATCHING DAY AND NIGHT LOCATION IMAGES USING SIFT AND LOGISTIC REGRESSION

Nazlı TEKİN

M.Sc. Thesis Electrical and Computer Engineering
August 2015

Supervisor: Asst. Prof. Dr. Kadir Aşkın PEKER

ABSTRACT

The purpose of this thesis is to match day and night urban scene images. Matching images based on similarity of their content is a computer vision problem that we face in many applications such as object and location recognition, scene understanding, robotic navigation, stereo reconstruction, etc. Being able to match images of the same location across different time of day conditions is a desired capability for many applications.

SIFT is a method widely used for matching images of location or recognizing objects in an image. The method can handle light varying condition at some points. However, the degree of success decreases when we try to match images of locations taken at day to until night time period. Especially, the success of matching between day time image and night time image is very low. The reason of lower success is because it becomes very different image with nightfall and turning on lights. In this study, we examined how the performance of SIFT method diminishes with the change of daylight.

We propose a novel method for scene matching based on classification; we examined and compared the proposed method with the generally used distance and threshold based matching approach. Logistic regression classifier is used to match SIFT descriptors extracted from day and night images. The classifier is trained and tested with ground truth training set we collected and extracted using a fixed camera and image alignment. In our experiment with images of urban scene taken started from day till night, we make success comparison with Lowe's proposed threshold criterion method and logistic regression classification method. As a result, if we look at number of matching feature point pairs on images, the precision of two methods is high but it misses many possible matching feature point pairs, thus we identify the recall rate as the primary problem. According to the experimental results, logistic regression classification achieves better recall rates.

Keywords – image matching; image retrieval; local descriptors; SIFT; logistic regression; day and night image matching.

GECE VE GÜNDÜZ LOKASYON İMGELERİNİN SIFT ÖZNİTELİKLERİ VE LOGİSTİK REGRESYON İLE EŞLEŞTİRİLMESİ

Nazlı TEKİN

Yüksek Lisans Tezi – Elektronik ve Bilgisayar Mühendisliği
Ağustos 2015

Tez Yöneticisi: Asst. Prof. Dr. Kadir Aşkın PEKER

ÖZ

Bu tez çalışmanın amacı gece ve gündüz çekilen aynı lokasyon imgelerinin eşleştirilmesidir. Benzerliğe bağlı imge eşleştirme bilgisayar görmesinde nesne tanıma, lokasyon tanıma ve robotik gezintide karşılaştığımız gibi birçok uygulamada önemli bir problemdir. Günün farklı koşulları altında çekilen görüntülerin eşleştirilmesi birçok uygulamada istenen bir yetenektir.

SIFT (Ölçekten Bağımsız Öznitelik Dönüşümü- Scale Invariant Feature Transform) bir nesne veya mekân görüntülerinin tanınması ve eşleştirilmesinde sıkça kullanılmaktadır. SIFT yöntemi belirli oranda ışık normalizasyonu yapmaktadır. Ancak bir mekânın gündüzden geceye kadar olan zaman diliminde alınan görüntüleri eşleştirme yapıldığında aynı başarıyı gösteremediği görülmektedir. Özellikle gece gündüz resimlerinin eşleştirilmesine başarılı olamamaktadır. Bu başarının düşmesinin sebebi havanın kararmasıyla ve ışıkların devreye girmesiyle birlikte imge üzerinde ciddi değişiklikler meydana gelmektedir. Bu çalışmada SIFT özniteliklerinin eşleşme başarımının gün ışığı değişimi ile nasıl değiştiği nicel olarak ortaya konulmaktadır.

Tez çalışmada eşleşme kararı için Lowe’un önerdiği ölçüt veya doğrudan uzaklık eşiklemesine alternatif olarak sınıflandırıcı tabanlı bir eşleşme yöntemi önerilmektedir. Bu çalışmada logistik regresyon sınıflandırıcı kullanılmıştır. Logistik regresyon gece ve gündüz imgelerinden çıkarılan SIFT tanımlayıcı vektörlerini eşleştirmek için kullanılmıştır. Sınıflandırıcı hazırlanmış eğitim setiyle eğitilir ve yeni gelen örneklerle ile test edilir. Gündüzden geceye çekilen lokasyon imgelerini kullanarak yaptığımız deneyde bu iki eşleştirme yöntemleri karşılaştırılmıştır. İmgeler üzerindeki anahtar nokta eşleşme sayılarına bakılarak elde edilen hatırlama(precision) değerleri her iki yöntemde de yüksek çıkmıştır fakat eşleşmesi gereken birçok anahtar noktanın kaçırıldığı görülmektedir. Hassasiyet(recall) değerlerine baktığımız zaman logistik regresyonla sınıflandırma yönteminin daha iyi sonuçlar verdiği görülmektedir.

Anahtar Kelimeler — imge eşleştirme; imge tarama; yerel betimleyiciler; SIFT; lojistik regresyon; gece ve gündüz görüntü eşleştirme.

To my family

ACKNOWLEDGEMENT

My sincere appreciation is to my advisor, Kadir Aşkın Peker for continuous support of my MS thesis study and for his patience and motivation. I would like to thank him to share his knowledge and great encourage for my thesis.

I would like to thank all instructors in all of my life who help me to come this point.

My greatest deepest gratitude is to my parents, is to my father Hüseyin Tekin who always support my decisions, my mother Özay Tekin, my sisters Esra Tekin and Büşra Eda Tekin and my little brother Mehmet Emre Tekin. They were always supporting me throughout my life. I would like to thank them their unconditional love.

Table of Contents

LIST OF TABLES

TABLES

LIST OF FIGURES

FIGURE

LIST OF SYSMBOLS AND ABBREVIATIONS

SYMBOL/ABBREVIATION	
GPS	Global Positioning System
LIDAR	Laser Imaging Detection and Ranging
SIFT	Scale Invariant Feature Transform
SURF	Speeded up Robust Feature
RADAR	Radio Detection and Ranging

CHAPTER 1

INTRODUCTION

1.1 Motivation

Interpreting digital images of scenes is one of the goals of computer vision. Determining some specific objects or features or activities in given images is very crucial for many applications. Identifying and describing images for matching two images has become an important challenge in recent years.

Matching images is used in a wide array of applications such as location recognition or object recognition on any picture. For instance, image matching is used in determining whether given two images are of the same place or same object or if they have similar or common aspects of content. An example of object recognition is shown in Figure 1.

Figure 1.1 Images of objects are shown in first one. The result of object recognition is on the second image with using feature points.

Another area which uses matching images is content based image retrieval. For instance, with search by images you give an image to searching engine like Google, it helps to understand where the given location belongs to or what the object in an image is. Thus, it is used for finding the objects or location in real world from digital images (see Figure 2).

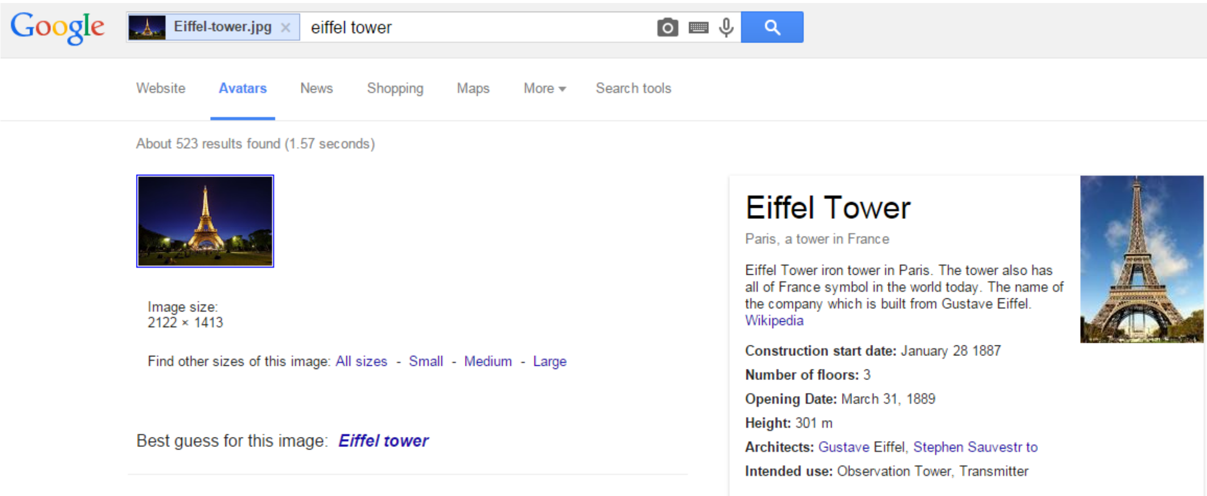


Figure 1.2 Example content based image retrieval using Google image search

Recognition of given location images is used to navigate robotic vehicles as well. For navigation the vehicle should know the places that it visited before. It is very crucial recognizing locations in varying lighting conditions for outdoor navigation. An autonomous car - also known as driverless car or robotic car - has transportation capability that a traditional car has. An autonomous vehicle senses its surroundings with the technique of LIDAR or RADAR or GPS or computer vision. Here also image matching for location and object recognition is employed.

Therefore, successfully identifying image and matching images solves many problems in real life. We can give automatic object recognition, location recognition, scene identification and classification applications and autonomous robots and cars applications and content based image search as examples.

For image matching local descriptors are widely used. There are several local descriptors such as SIFT, SURF and feature point detectors such as HARRIS CORNER detector. Among these detectors SIFT (Scale Invariant Feature Transform) is one of the most common and widely used methods. SIFT descriptors are found to perform best in many applications [14]. Since SIFT is more successful, in this research we use SIFT descriptors for extracting the features of images and describe this features. The methods developed and results obtained in this thesis using SIFT descriptors can be easily extended to other similar local descriptors as well.

Matching images operation is also requested in changing conditions such as day and night. For example, if a robot is required to recognize a location based on matching an acquired image to a reference database which has images collected during day time, we need to be able to match night images to the day images in the database. In many other applications invariance under day and night light changes is a required or desired feature. We specifically address this problem in this thesis.

1.2 Objective

Our motivation in this study is to first estimate the performance of existing SIFT matching method for image matching under day and night variations, and then develop a new classification based matching approach for image matching to increase performance under day and night variations.

Features extracted from images with SIFT descriptors are scale invariant, rotation invariant and partially illumination invariant. However, SIFT descriptor fails to match day and night appearances of location as the lighting conditions are very different. This is a more complicated problem compared to decreasing of lights problem because night time lighting such as street lights etc. change the scene drastically. Feature points in an image are decreasing with nightfall and different feature points appears with turned on lights in outdoor location images. Therefore, matching feature points is not very successful in this case. In this study, our main purpose is to match location images under varying illumination conditions. In order to increase performance of the SIFT descriptors, we propose a classification based technique. In the proposed method, we want to let a classifier make the decision for matching of two SIFT descriptors, instead of computing a vector distance and thresholding it.

1.3 Our Approach

Matching the descriptors from two images is a first step in matching images because it indicates if the two points in the two different images are the same point (object or location) or not. Lowe developed an algorithm [10] which is used for matching two images. In this algorithm, first it extracts the feature points of each image and then describes the feature points using the SIFT local descriptors. A SIFT local descriptor in this case is a 128 dimensional vector that describes the image region around a feature point. After describing local descriptors, it decides whether two images are matching or not by looking at the similarity of each descriptor in the first image with each descriptor in the second image. Matching of the descriptors is decided using a threshold value applied to the vector distance between the two descriptor vectors. This is the generally used matching method that we use as the reference prior approach in our work.

Our approach in this thesis for matching images is to use a classifier to decide whether two SIFT feature points are matching or not. We believe that in the case of a complicated and harder matching problem such as matching day and night images, fixed threshold method is not sufficient. A machine learning approach in this case can be able to learn the complicated relation between the day-time SIFT descriptor and the night-time SIFT descriptor of the same object or place in an image. We use logistic regression as the classifier in this thesis, however the basic idea applies to any classifier available in machine learning, including support vector machines or neural networks, etc.

1.4 Contribution

Trying to match two images is a critical problem especially if images are in different lighting condition such as day and night images. In this thesis, we study the use of logistic regression classification method for matching SIFT descriptors for the purpose of matching day and night images. Firstly, in our study, we examined the success of matching algorithm of Lowe under changing daylight conditions. We show that this matching algorithm is not effective for matching day and night images. Showing the limitations of the threshold based matching method of SIFT for day and night images in a systematic and quantitative way is one of the contributions of this thesis. The most important contribution of this study is to show that using a machine learning classification method for matching local descriptors extracted from day and night location images gives more successful results than the existing threshold based method. We propose a simplified 3 dimensional input vector to the classifier consisting of the distance, mean, and standard deviation of descriptors; but we also point towards more advanced alternatives for future work. We also develop a methodology of obtaining reference samples of positive matching SIFT descriptor pairs using aligned images of a static scene photographed from day to night using a fixed camera.

1.5 Structure of the Thesis

In continuation of this thesis, firstly, in second chapter, previous related work about image matching is presented. The methodology that is used in this study is also explained in the second chapter. In the third chapter, experimental work on establishing the level of performance of existing method and also the experiments that the test the proposed method are presented. In the fourth and concluding chapter, the results are discussed and future work is outlined.

CHAPTER 2

METHODOLOGY AND APPLICATIONS

2.1 Related Work

The location recognition problem which is under different lightning conditions and different times of day are encountered in many places. In robotic operation in outdoor environments, changing illumination is a crucial problem especially day and night light changes. Recently appearance based LIDAR is used for scanning environment [11]. However, power and weight penalties are critical for this system. This appearance based system suffers from appearance change in 24 hour period of day. In this thesis we want to develop a general purpose solution that will not require the use of LIDAR or other imaging techniques so that it can be used for instance for image based retrieval in a search engine.

Although, thermal infrared imagery is not directly affected by illumination changes, it suffers from light changing during day and night time period. The objects heated by the Sun along the day time have different thermal properties. Thermal camera or thermal sensors are not very useful for recognition of object or location in outdoor places. Therefore, even if thermal imaging is available, image matching still has an important role especially in outdoor robotic navigation.

Image matching is also used to navigate urban environments using mobile telephones with cameras. This system works as searching a query image from a large database which is collected before [1]. The limitation in this system is to identify day appearance and night appearance of the same place. To solve this problem, it has been taken photo in specific area at changing rotation and different period of time in a day [1].

There are needs to SIFT descriptor to recognize the location, historical places or buildings. In these applications, it scans image in the database regardless of image taken at day or night time. To recognize the places it should match the given image among the images in database. Even if there only exists day time in the storage, it should match images under varying lightening. As the database size grows, the invariant feature matching falls down. For the city scale location recognition, it is used vocabulary tree. The vocabulary tree is occurred with carefully selected among most informative features so that retrieval performance is significantly improved [2]. Vocabulary trees method determines the informative features about given location and builds a tree with these features. Thus, searching performance critically increases.

2.2 Background

For image matching, interest points in an image should be extracted and feature description of each interest point should be calculated. The description of each point provides information about that point and this helps to compare interest points. Comparing interest points of two images helps to compare the content of those two images. SIFT is arguably the most commonly used and the most successful among the local descriptors [14]. In this study, we chose SIFT method as the local descriptor to extract the feature points of an image. We describe local descriptors and the SIFT method in more detail in the following section.

In the proposed matching method used in this thesis, we trained a classifier with sample matching points and test new points if they are matched or not. We used the logistic regression classifier for this purpose. Logistic regression classifier is also described in more detail below.

2.2.1 Local Features

In image processing, a feature is a piece of information extracted from an image and captures and describes an important aspect from the content of the image. In an image, marked spots such as edges, points or corners can be examples of features. The feature points can also be local maxima and minima or line endings (see Figure 3). Features capture the significant and distinctive parts of an image. These feature points then can be used to understand, classify or compare and match images. Thus, feature extraction and detection is very significant for image matching. With the development of local features there has been substantial progress on the image matching problem.

Figure 2.1. Extracted feature points in an image are shown. SURF detector on the left, blob detector on center and corner detector on the right side.

For image matching or object recognition in an image, local descriptors are computed at each feature point of an image. The descriptor computer for a feature point describes the image region around that feature point. This description is used to compare two feature points to decide whether they belong to the same object or location. This local descriptors based approach has proven to be the most successful method in applications that involve object and location recognition or image matching [6]. There has been several studies on making this descriptors invariant to various image transformations such as affine transformations or illumination changes. For instance, there are two types of affine invariant regions build by Van Gool and Tuytelars [7] which are combination of feature points and edges and the image intensities. Lowe recommended scale invariant regions based on local extreme [8], which are invariant to rotation and scale changes, and also - to some degree - illumination changes.

In addition, local descriptors also called filters are used for texture classification. Texture is very frequently shown in images. In analysis of image applications like image segmentation or image retrieval, texture classification is very crucial. Texture classification plays important role in image analysis, object recognition and content based image retrieval applications [9]. The system of the texture design is to first extracting features and then classifying these features. Therefore, local descriptors are also significant in texture classification. For example Rabiu uses SIFT descriptors to classify fabric textures [22].

One of the methods used to determine feature points in an image is edge detection. With the edge detection methods it can be identified the points in an image where the brightness changes sharply or it has discontinuities. These points are considered to be important and distinctive points in an image. Thus, edge detection tool is sometimes used in computer vision and image processing in feature detection and feature extraction.

Canny edge detector was developed by John F. Canny is one of the most common used among the edge detector tools [23]. The process of the canny edge detector algorithm is as follows: It first applies a Gaussian filter to an image in order to remove noise. Then it finds the intensity gradients of an image. By applying a double threshold (hysteresis) it determines the potential edges. Finally it eliminates the weak edges and the edges which are not connected to others. These potential edges can also be potential feature points so that edge detector is used for finding feature points of an image [24].

On the other hand, feature points also called interest points of an image are related to the notion of corner detection. The main goal of corner detection is to obtain well-defined, stable image features which can be used for object tracking and object recognition in images [28]. However, most corner detectors do not just find the corner points in an image, but they also recognize the points which have high degree of variation in all directions [27]. Hence these points are acceptable for specific and distinctive points of an image. We explain corner detection and more specifically the Harris corner detector in more detail in the following section.

In computer vision, blob detection methods are also used for detecting regions with different properties such as regions in different brightness or color compared to the color of the surrounding region [29]. The purpose of the blob detector is to provide information about regions in an image which are not detected by edge detection and corner detection. These blobs also capture the important features of an image and enable matching parts of an image to other images for object or location region. To do this, it uses a differential method which takes the derivatives of position and local extreme methods which finds local maximum and local minimum of the function [30]. This method is used by SIFT for extracting the feature points of an image. Detail of the SIFT method is explain in Section 2.2.1.3.

In the following sections we describe some of the important feature point detection and description techniques.

2.2.1.1 Harris Corner Detector

Intersection of two edges generates a corner. To detect the feature points in an image Harris Corner Detection algorithm tests if there is a corner point by comparing each pixel to its neighborhood. A corner point can be recognized in a window, as shown in Figure 4. If there is a corner point in an area, shifting a window in any direction should give a large change in intensity. If the window is on a flat region or on an edge, there is no intensity change in all directions.

Changing in intensity for the shift $[u,v]$ with window function $w(x,y)$ is shown below:

$$E(u,v)=\sum_{x,y} w(x,y)\left[I(x+u,y+v)-I(x,y)\right]^2 \tag{2.1}$$

Window function $w(x,y)$ can be 1 inside the window and 0 outside; or it can be a Gaussian function. Harris detector uses an approximation of the above formula which is based on Taylor expansion and which includes image gradient terms. Average of this value over an image patch results in a matrix (Harris matrix) such that, if the eigenvalues of this matrix are both large positive values than the image patch contains a corner.

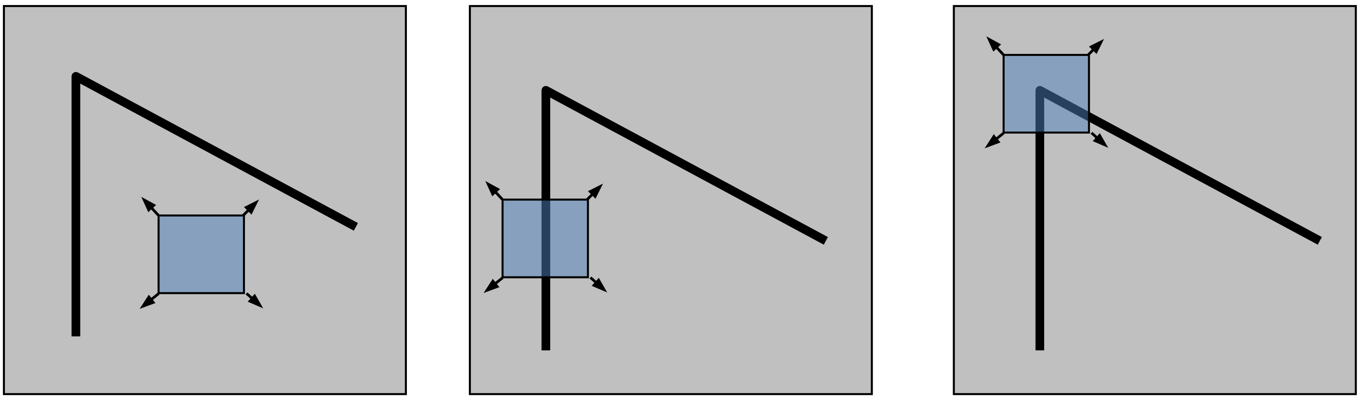


Figure 2.2. The representation of applying window function on an image. The first image shows flat region, second shows edge and the third one shows corner region in an image.

2.2.1.2 Speeded Up Robust Features (SURF)

Speeded up robust features is one of the popular feature detectors and descriptors. In object recognition, face recognition or image matching SURF descriptors can be used. The SURF algorithm is based on the three parts which are interest point detection, description and matching [19].

2.2.1.2.1 SURF Interest Point Detection

SURF uses square shaped filters for smoothing and filtering images with square filter more rapidly with using integral image. Integral image can be defined as

$$S(x,y)=\sum_{i=0}^{x}\sum_{j=0}^{y} I(i,j) \tag{2.2}$$

This integral image $S(x,y)$ represents sum of all pixels within rectangular region in original image $I(i,j)$ [19]. Calculation time is not crucial for integral images. Therefore using integral image is much faster.

For interest point detection, SURF algorithm uses Hessian matrix approximation. The determinant of Hessian matrix is using for measuring for the intensity changing around the point. The interest points are chosen according to maximal value of the determinant.

2.2.1.2.2 SURF Interest Point Description and Matching

Surf descriptor describes the intensity changes around the feature points. Every feature point has its own unique evaluated description. It is similar to gradient evaluation in SIFT descriptor but it build on distribution of Haar wavelet responses in x and y direction [19]. This is more quickly calculated.

The first step is to fix reproducible orientation based on information circular area from feature point. Then it constructs square region with orientations and extracts the SURF descriptors. Finally, features are matched by looking at the unique descriptors of features in two images.

2.2.1.3 Scale Invariant Feature Transform (SIFT)

Scale Invariant Feature Transform is an algorithm developed by David Lowe in 2004. Before this method there are algorithms which are rotation invariant for detecting feature on images but they were not invariant under varying scaling conditions. Therefore, this algorithm is considered for need of feature detection in scale invariant applications. With this method, despite scale of images changing, it can detect same features on images.

SIFT generates feature vectors on images which are rotation invariant, scale invariant and partially illumination invariant. SIFT algorithm use difference of Gaussians for keypoint localizations. The maxima and minima of difference of Gaussian are defined as keypoint locations on images. Keypoints are localized according to dominant orientations.

2.2.1.3.1 Keypoints Detection

In order to find keypoints, significant regions or blobs are detected using difference of Gaussian filters. After finding the extrema points in the filtered images, these points are localized. In the last step of keypoint detection, each keypoints orientation is estimated.

2.2.1.3.1.1 Finding Scale Space Extrema

This is the first stage of detecting keypoints. To decide the location of keypoints, Difference of Gaussian is used. It is critical to be scale invariant. By using Difference of Gaussian in different scale, stable keypoints can be found. DoG $D(x, y, \sigma)$ image can be occurred by the estimation written below. $L(x, y, k\sigma)$ is the convolution of original image $I(x, y)$ with Gaussian $G(x, y, k\sigma)$ at scale $k\sigma$.

$$D(x, y, \sigma) = L(x, y, k_i\sigma) - L(x, y, k_j\sigma) \tag{2.3}$$

$$L(x, y, k\sigma) = G(x, y, k\sigma) * I(x, y) \tag{2.4}$$

Therefore, DoG images are obtained with the different scale values. After DoG images have been obtained, keypoints local maxima and local minima recognized by comparing each pixel with its neighbors.

2.2.1.3.1.2 Keypoint Localization

After detecting extreme points, there are too many keypoints. Some of them are unstable because they are localized along the edge or have low contrast. To eliminate these candidate features, first determine the position of candidate keypoint and locate the keypoint on the candidate one [10]. However, these keypoints are not still accurate and stable. To determine accurate keypoints, it is calculated interpolated location of the extremum [10]. With discarding low contrast keypoint and eliminating edge responses SIFT descriptor finds its accurate keypoints.

2.2.1.3.1.3 Orientation Assignment

Image rotation variance affects the accurate keypoints location. If the image rotates, the key points locations can change and this cause losing the accurate key points locations. Because of preventing the fault caused rotation variance, it is assigned to every keypoints its orientation.

For each image sample $L(x, y, \sigma)$ (Gaussian-smoothed image) at scale σ the gradient magnitude $m(x, y)$ and orientation $\theta(x, y)$ are precomputed using pixel differences [10].

$$m(x,y) = \sqrt{\left(L(x+1,y) - L(x-1,y)\right)^2 + \left(L(x,y+1) - L(x,y-1)\right)^2} \tag{2.4}$$

$$\theta(x,y) = a\tan 2\left(L(x,y+1) - L(x,y-1), L(x+1,y) - L(x-1,y)\right) \tag{2.5}$$

A gradient histogram is shaped according to its neighbors of each keypoint. The highest gradient is chosen the orientation of the keypoint. In addition to highest peak it is chosen also local peak within %80. Thus, there can be keypoints which are same location and same scale but different orientation.

2.2.1.3.2 Keypoint Descriptor

In this stage, SIFT descriptor describes the description of each keypoint selected on image. This descriptor of each keypoints is rotation invariant, scale invariant and partially illumination invariant. The basic idea of estimating descriptor is first place 16x16 square windows into detected features. After placing the window, it computes the gradient orientation for each pixel. It creates histogram over edge orientations weighted by magnitude.

Divide 16x16 windows into 4x4 grids of cells shown in Figure 2.3 [10]. After that, compute orientation histogram for each cell. There are 8 orientations so that 16 cells 8 orientations 16x8=128 dimensional description vector. Therefore, there is unique descriptor for each feature on images that we can use it for matching images.

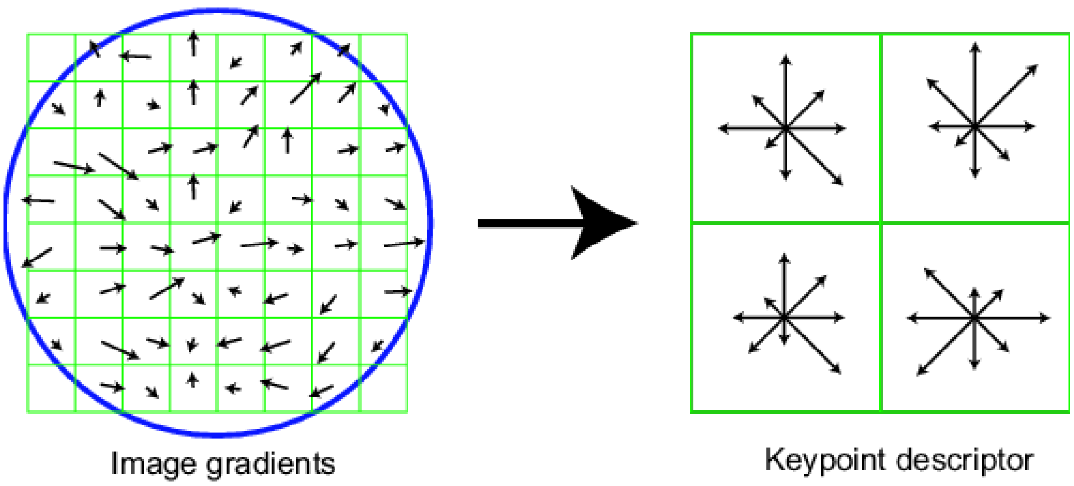


Figure 2.3. A sample keypoint descriptor is created by gradient magnitude. This figures shows 2x2 description array 8x8 set of samples where as in actual 4x4 descriptors computed from 16x16 sample array.

2.2.2 Logistic Regression

Classification problem means to identify new observation to which of set of categories belongs according to training data in statistic and machine learning. For instance, identifying given email is spam or not spam or deciding the tumor is malignant or benign is classification problem.

In machine learning, supervised learning method is to infer a function from label data to predict new data's categorization. Each training data consists of input generally vectors and desired output value. Supervised learning algorithm analyzes the training data values and produces inferred function to classify the new examples. On the other hand, unsupervised learning algorithm is clustering and grouping data in categories according to their similarity and distance.

Classifier is known as classification algorithm. Logistic regression is one of the classification algorithms that produce outcome considered possible values of dependent variable. Logistic regression classifier is used to assign a output value for given input value.

Logistic regression is probability model which was developed by D. R. Cox in 1958. Due to predictor features, binary logistic model predicts a binary response. By using logistic function, the possible outcomes of test set are modeled. There is binary logistic regression in which dependent variable is binary and the number of categories is two. There is also multinomial logistic regression which is for the number of categories is more than two.

There are two types of logistic regression model which are binomial and multinomial. In the binomial or binary logistic regression, there are two possible outcomes according to dependent variable such as a call with cancer or not. If there are three or more possible outcomes such as disease A, disease B and disease C etc., it is called multinomial logistic regression. In binary logistic regression, the possible outcome is selected as “1” or “0” according to being positive or negative result.

Logistic regression estimates the relationship among the independent variables. According to this estimation, it predicts the possible outcomes of test case. Logistic regression is widely used in many areas which is including medical and social sciences. For instance, logistic regression is used trauma care evaluation. Anatomic, physiologic and age characteristics are taken as independent features of injured patient to estimate the probability of survival [3]. Prevalence of depressive symptoms of cancer patient was observed by multinomial logistic regression [4]. Logistic regression cab be also used in engineering area especially for predicting of failure of given system or product. For example, it determines the failure probability of forming limit diagram [5].

The logistic function takes values from negative infinity to positive infinity as an input, returns values between “0” and “1”. The logistic function $\sigma(t)$ shown below:

$$\sigma(t) = \frac{e^t}{e^t + 1} = \frac{1}{1 + e^{-t}} \tag{2.6}$$

t is given as follows:

$$t = \beta_0 + \beta_1 x \tag{2.7}$$

The final logistic function can be defined as:

$$F(x) = \frac{1}{1 + e^{-(\beta_0 + \beta_1 x)}} \tag{2.8}$$

$F(x)$ is refers to probability of dependent variables. For instance, $F(x)$ is estimated probability $y = 1$ on input x. The sum of probality of being $y = 1$ and $y = 0$ is equal to 1.

$$F(x) = P(y = 1 \mid x; \theta) \tag{2.9}$$

$$P(y = 1 \mid x; \theta) + P(y = 0 \mid x; \theta) = 1 \tag{2.10}$$

CHAPTER 3

EXPERIMENTAL WORK

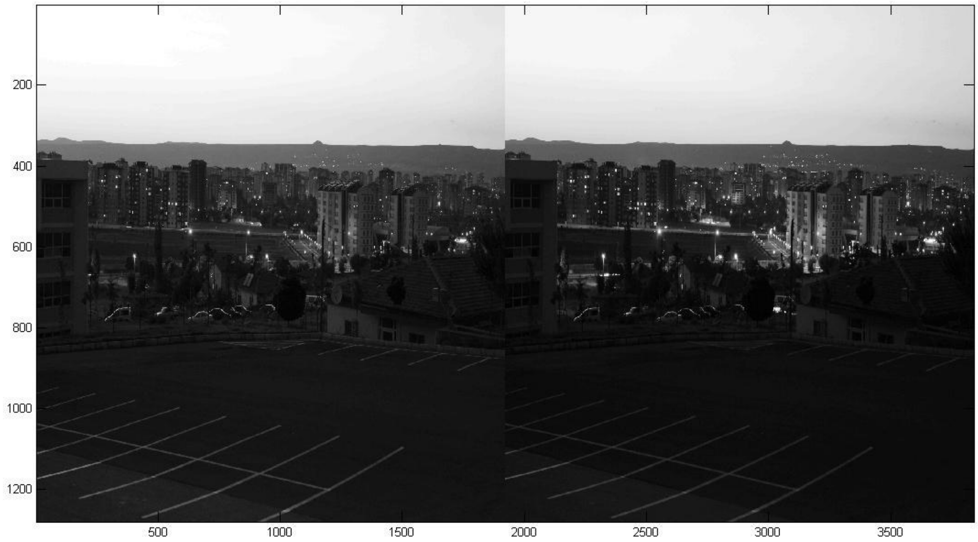
3.1 Image Matching with Vector Distance and Threshold Value

Matching SIFT features extracted from one image with SIFT features from another images is used in applications such as location determination and object recognition. Whether two images match is decided by comparing SIFT descriptor vectors extracted from the first image with the descriptor vectors from the second image. Hence, the decision of matching two descriptors is the fundamental component of the image matching problem.

Lowe [10] proposed following approach for this matching problem (see Appendix A for the Matlab code): The vector distances (Euclidean distance, or cosine distance which is quicker and gives the almost same result with Euclidean distance) are calculated for each descriptor vector from the first image with respect to all the descriptor vectors extracted from the second image. The first and second vector which is closest to the vector in the first image among the vectors in second image is taken. If the distance of the first vector to the vector in the first image is closer than the distance of the second vector, it is accepted that the first vector is matching with the vector in the first image. Otherwise, there is no vector matching with the vectors in second image. In other words, in order a descriptor in the first image to be considered to have a matching vector in the second image, the matching vector in the second image must be clearly ahead of the nearest candidate behind it. Lowe states that this match criterion gives a better result than applying a fixed threshold to vector distances [10]. Figure 3.1 shows the matching of SIFT features using Lowe’s matching method.

However, in many applications, feature vector matching decision is made by calculating the distance between the two vectors and then applying a fixed threshold. If the distance value is below the threshold, it is considered to be a valid match. In this thesis we call this approach direct matching with a threshold.

(a)



(b)

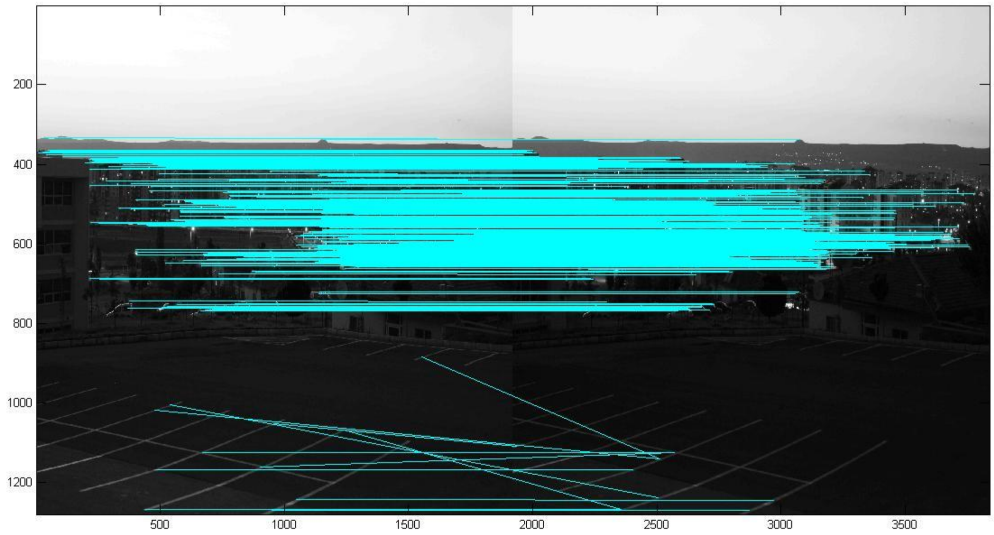


Figure 3.1 (a) Gray level of two same location but different lighting condition pictures desired to be matched. **(b)** Matching feature points using Lowe's matching method. Blue lines show matching feature points between two images with Lowe's method

3.2 Matching SIFT Features with Logistic Regression Classification

As explained above, the decision of matching two feature vectors is usually made by considering the information of the distance (or similarity) between two feature vectors. However, in some applications it can be difficult to define a fixed threshold value. The decision criteria of the compared vectors with some property and threshold value may need to be adapted. In this study, with the hypothesis of this adaptive approach can give better results while matching the day and night images we use supervised learning matching system. It can be shown that classifier based matching system is more compatible than matching with fixed threshold value system through the example of matching one dimensional two feature vectors. The condition of difference between two vector values must be smaller than t (threshold value) is the same as the band with the corresponding classification limit around y=x line with the width of t in two dimensional space. However, a classifier can learn more complicated classification limits. For instance, the band extended to positive direction around the y=x line corresponds to the decision system which is adaptive in direct proportion to the threshold value.

In our study, we used logistic regression classifier as a supervised learning matching system. The first step of this system is to train a logistic regression classifier with training examples and then test with new samples.

3.2.1 Preparation of the Training and Test Data Sets

To prepare the training set, we need to find matching pairs of feature points from two different images as positive samples and non-matching pairs as negative samples. Finding reliable correct (reference) matches to train the classifier is an important problem in our approach. For this purpose, we took pictures of an outdoor urban scene with a fixed camera at different times during day. We extracted SIFT features points and descriptors from each image in this data set. To decide positive samples we select the feature points from two images and look at if they are in the same location in the image. In other words, if we place first image over second image and on the same location if feature points are detected in both images, then these points are considered to be reference correct matches.

Since we assume that same location in all images correspond to same objects, we need to be very careful to make sure that the images are aligned. For this purpose, we used an image alignment method based on extracting SIFT features, matching SIFT features of consecutive images, and calculating the mean shift in x and y directions between images. Although the camera is fixed, there may be a few pixels of difference between some images (see Figure 3.2).

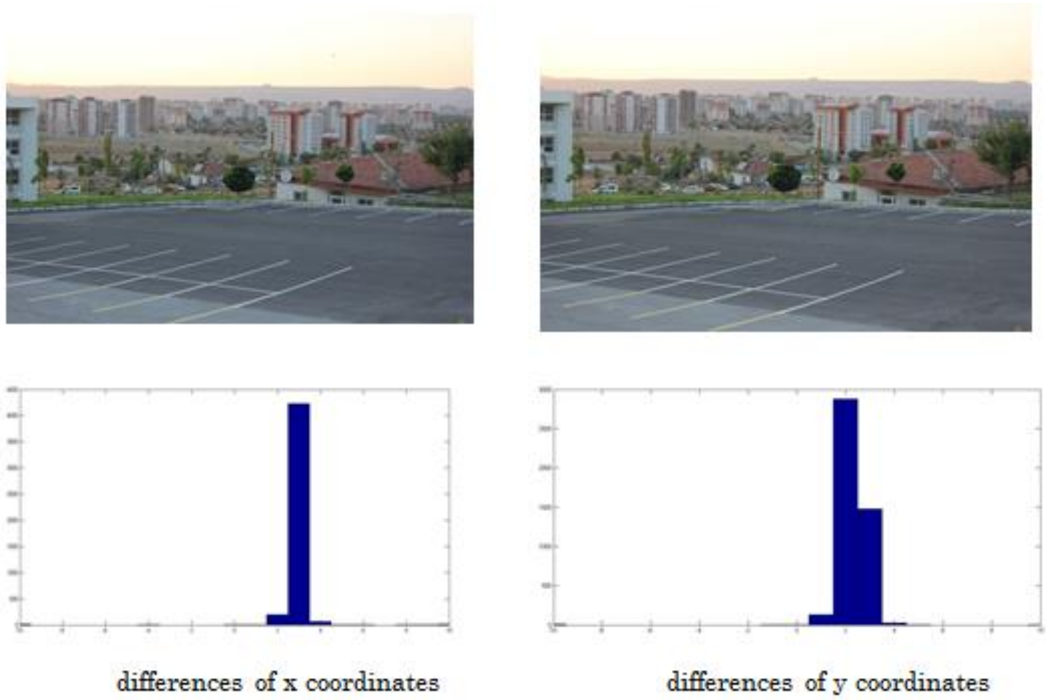


Figure 3.2 Image alignment example is shown. SIFT features extracted from the two images are matched. Then a histogram of the x- and y-coordinate differences of the matching SIFT features is computed. Difference histograms clearly show the amount of shift in x- and y-directions to align images. Differences are usually in the order of a few pixels.

After we took these points we calculated the vector similarity of these feature points. To verify the reliability of the reference correct matches, we selected feature points in pairs of images that are at the same location but the vector distance between them is high. We verified that these points are actually the same objects (e.g. same corner of a window on a building) even though the SIFT descriptors change significantly due to lighting variation.

For the negative training set, we simply selected random feature point pairs from two images. Random pairings of feature points are not supposed to match, so generating negative samples is a trivial step.

3.2.2 Input Vector to the Classifier

According to binary logistic regression, features vectors known as matching pairs in training set are set to positive samples whereas randomly selected feature vectors known as not matching pairs are set to negative class samples. The classifier needs to be trained using these sample pairs. For this purpose, we need to generate a combined feature vector from given two SIFT vectors as input to the classifier.

The first approach we tried was to concatenate the two SIFT vectors and input to the classifier. Thus, an extended feature vector is generated by adding two compared vectors whose size is 128 and the final feature vector becomes a 256 dimensional vector. The classifier is trained with this vector. The classifier trained with training samples prepared in this way is expected to learn which of the feature vectors pairs are matching and which are not. In the test classification and the implementation stage, the features vector pairs desired to be compared are given to classifier. The decision of match is made by the logistic regression classifier that we trained before. The use of all the available information of the vectors while making the matching decision should logically allow for a better decision than when only the distance between vectors is used to decide for a match. Hence, it is expected to give higher performance.

However, with the concatenated very high dimensional input, we faced an overfit problem were the performance is perfect on the training set but very poor on the test set. The high dimensional SIFT descriptor vector (two vectors combined to 256 dimensions) makes it difficult to efficiently train the classifier. In this study, we see that if we use 256 size vectors for training the classifier, we encounter the overfit problem. The definition of this overfitting problem is fitting a statistical model with too many degrees of freedom in the modelling process. One of the reasons that causes overfitting is to have too complex model which means to have too many predictors. In our cases the 256 size vector used as predictors caused this problem.

As a solution to the overfitting problem, in this study, we reduced the size of attributes for a more stable classifier with limited number of predictors. Instead of using 256 size vectors, we used the distance of these two vectors and the average and standard deviation value of these two vectors. The new input to the classifier in this case is a three dimensional vector instead of 256 dimensions. With this additional information, the classifier can learn appropriate measurement requirement for matching, such as bright or dark image areas with high-low contrast image regions matching. In the new solution, we lose a very large part of the original information in the sample pairs, but it is still more information than just using the distance between vectors. So, our experiments below show that even a small increase in the information used in matching decision can improve the matching performance. These values are presented to the classifier. We discuss in the Conclusion Section other possible solutions such as using a more powerful classifier with more training data to overcome the overfitting problem without losing information.

3.3 The Success of Matching Day and Night Images with Lowe‘s Matching Algorithm

In our study, we investigated the success of Lowe’s Matching algorithm for matching images. First of all, to demonstrate the performance of existing methods for matching SIFT features in a quantitative manner, we created the data set consists of the photographs taken with a fixed camera at regular intervals in the period starting from daytime until night when the lights are on. In Figure 3.3, there are two image dataset and one of them includes 18 image series which we used in this experiment and got result. The other set has 22 images.

Photos were taken at outdoor urban areas. One of the most critical things that make the difficult day and night matches is lighting urban areas with the night fall. There is a significant visual change in images. This change on the image is much more difficult than the scene becoming completely darkening and it is also more interesting for many applications. For this reason, in this study we particularly chose urban scenes with night lighting. In addition, we use a fixed camera for eliminating other factor effects in image matching such as different angle, scale and rotation, so that we have a controlled experiment focused on day and night changes. For this purpose we took pictures of the urban scene from a distance where moving objects such as cars etc. are not significant. Although we use fix camera for any shift towards images, to avoid any errors we align the entire image set, as described in Section 3.2.1.

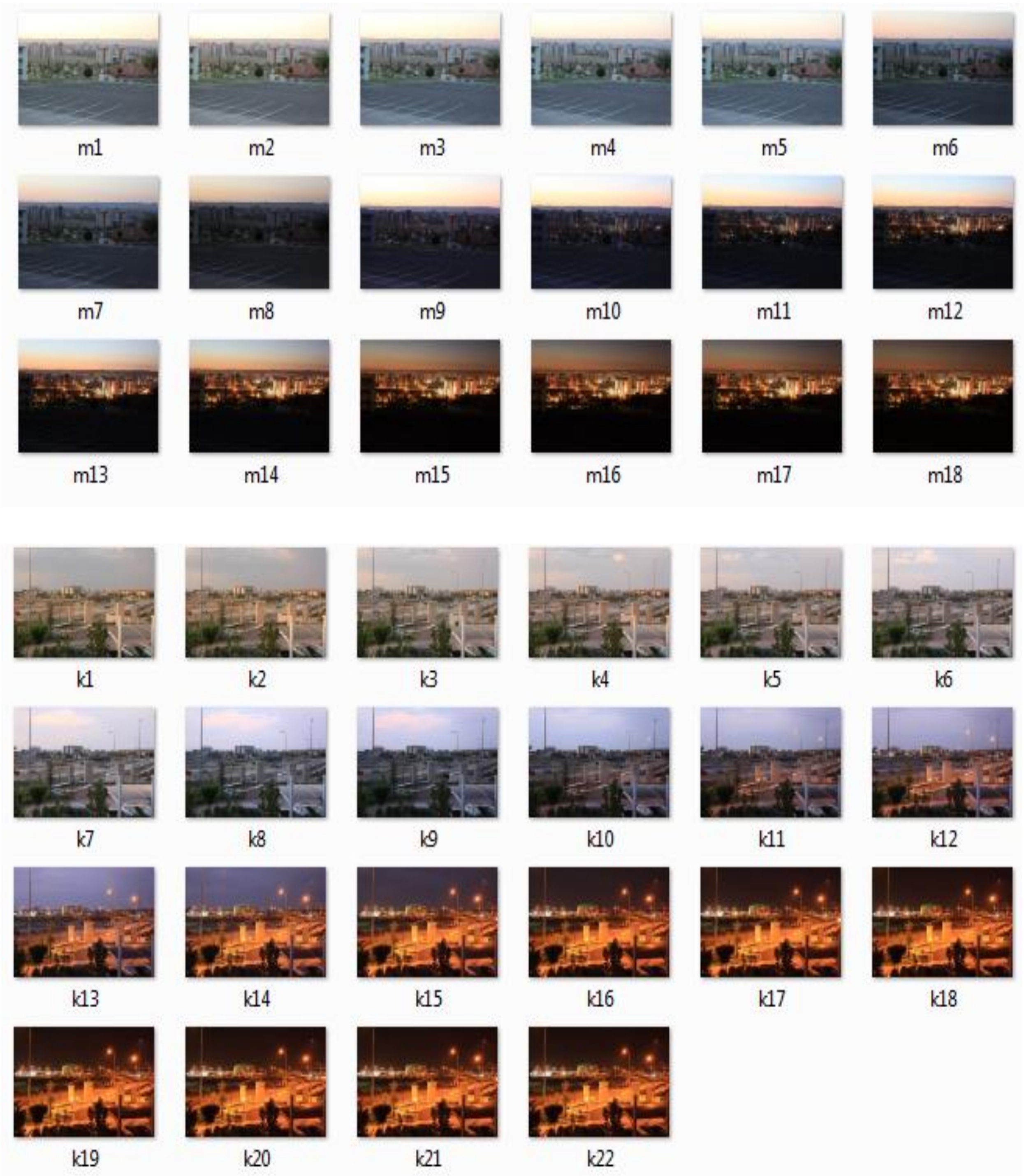
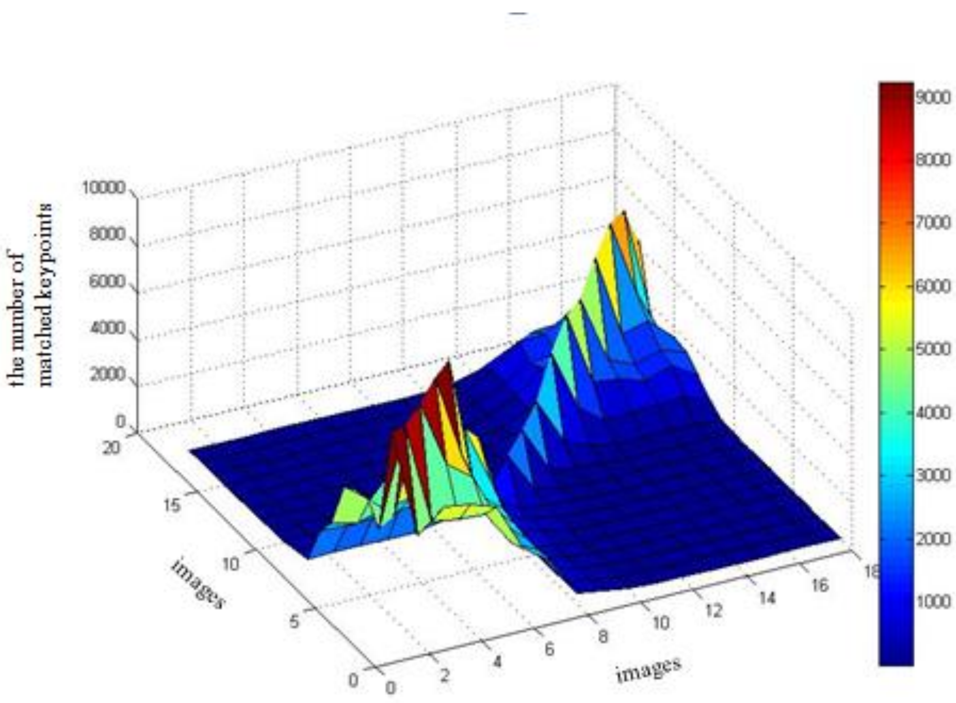


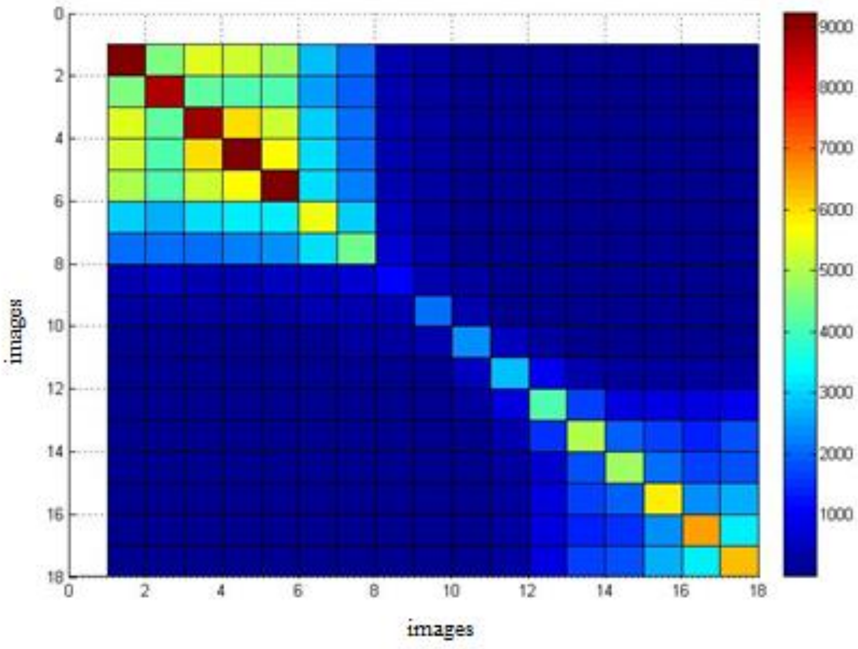
Figure 3.3 The two sample data sets (m and k series) are shown. The data sets prepared to measure the performance of day and night image matching. The images are taken in the same place from day to night with regular intervals to cover specifically the transition from day to night.

We perform an experiment with the 18 images and tried to match each image with each other (NxN=18x18 = 324 image comparisons) by using SIFT descriptor and Lowe's match criteria. For each pair of images, number of matched key points between two images is computed. Generally, number of matched key points are in the order of hundreds, but this number is highly variable and largely depends on the number of keypoints extracted from each image to begin with. If the images have too many keypoints than the number of matches proportionally increases. For this reason, for a fair comparison of the image match success, we normalize the number of matches between two images by dividing it with the number of keypoints in the first image. The reason we use the first image is that Lowe's method takes the first images keypoints as the starting point and looks for a match for each of them in the second image, so the maximum number of matches cannot be more than the number of keypoints in the first image. The normalized value of the average number of key points in two images, is used as "matching scores". The result of NxN matching result is shown in Figure 3.4 (a) and the normalized values shown in Figure 3.4 (b). The diagonal is the number of matches of an image with itself, which is actually a perfect match. In the non-normalized matrix, this diagonal values give us the total number of keypoints extracted from each image. Note that, the total number of extracted keypoints decreases during the period where the scene gets darker but the night lights are not turned on. The total number of keypoints increases when the night lights appear, but it is still less than the number for day time images. In the normalized matrix, the diagonal becomes 1, which means perfect match.

According to the graphs, the number of matched key points with Lowe's method among daytime images is very high, however matching of day and night images is very low, even close to zero. Therefore, among day light photo is sufficiently successful with this method, but day and night image matching is insufficient despite the light normalization that is done to a degree in SIFT descriptor computation.

(a)


the number of matched keypoints
images
images


images
images

(b)

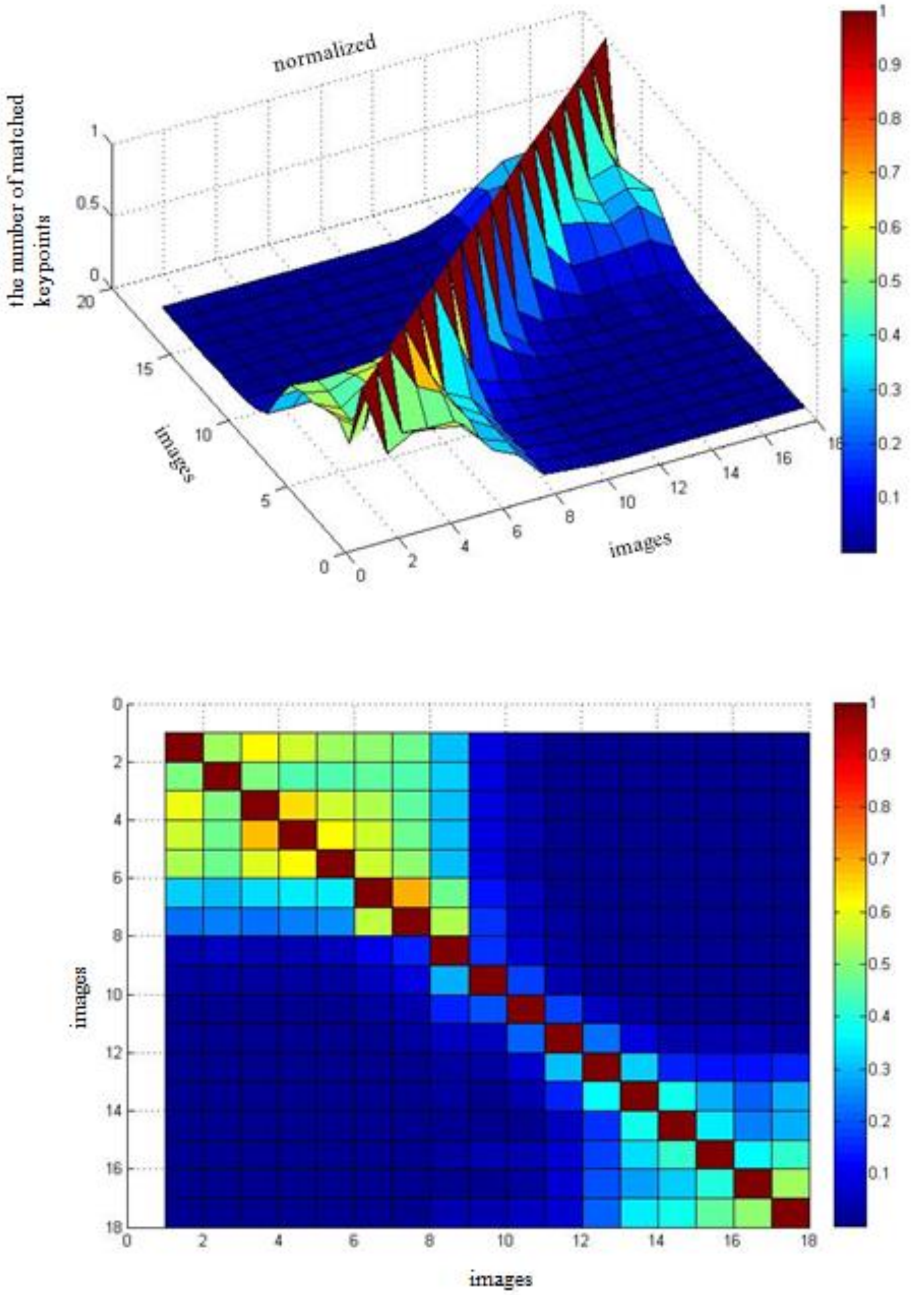


Figure 3.4 The matrix of matching 18 images which are taken from day to night is used. While daytime images matching among their own and the night images relatively matching successfully among them, the day and night pair of images matching is reduced to almost zero. **(a)** The matrix of non-normalized number of matches. **(b)** The matrix of normalized number of matches. One means perfect match.

Table 3.1: The number of matches between NxN pairs of images in the 18 image data set. The columns and rows correspond to images from 1 to 18 in order. The upper table shows the non-normalized matching numbers shown in Figure 3.4 (a). The lower table shows the normalized numbers shown in Figure 3.4 (b).

9190	4539	5477	5299	4925	2921	2114	416	257	97	43	19	14	21	10	24	18	10
4555	8669	4330	4173	4105	2628	2060	458	231	96	45	26	25	48	16	19	36	8
5441	4317	8890	6016	5208	3048	2083	414	226	97	42	17	14	17	15	19	13	6
5313	4139	6018	9238	5668	3206	2193	413	221	85	35	13	12	19	19	16	19	6
4880	4090	5246	5690	9192	3250	2324	439	210	83	44	14	20	34	19	26	19	13
2841	2588	2959	3129	3162	5541	3055	576	323	118	54	21	19	25	25	26	25	7
2049	1972	2023	2085	2244	3023	4368	646	387	164	76	25	17	38	27	18	33	11
333	345	328	334	329	523	590	1102	319	161	63	32	23	9	32	6	39	5
191	174	181	164	169	269	332	332	2035	405	118	57	32	21	67	11	86	7
76	67	76	78	73	120	143	174	411	2346	477	182	106	33	108	43	117	16
28	23	16	22	25	42	53	75	130	499	2827	848	401	171	183	207	182	99
16	24	16	9	8	13	25	39	52	211	921	4131	1548	699	762	777	860	412
11	11	16	5	7	18	14	22	17	89	410	1604	4932	1855	1659	1342	1732	989
6	9	8	11	5	7	10	12	17	48	208	727	1881	4881	2007	1552	1841	1548
5	9	10	6	2	2	7	3	8	40	199	811	1721	2026	5838	2311	2663	1631
9	11	5	4	9	13	12	12	9	75	256	865	1423	1596	2357	6522	3307	1233
4	13	5	6	7	6	8	7	9	44	195	931	1818	1869	2658	3300	6323	1541
1	1	4	2	1	1	3	4	1	23	115	416	974	1504	1576	1184	1483	4293

1,000	0,494	0,596	0,577	0,536	0,318	0,230	0,045	0,028	0,011	0,005	0,002	0,002	0,002	0,001	0,003	0,002	0,001
0,525	1,000	0,499	0,481	0,474	0,303	0,238	0,053	0,027	0,011	0,005	0,003	0,003	0,006	0,002	0,002	0,004	0,001
0,612	0,486	1,000	0,677	0,586	0,343	0,234	0,047	0,025	0,011	0,005	0,002	0,002	0,002	0,002	0,002	0,001	0,001
0,575	0,448	0,651	1,000	0,614	0,347	0,237	0,045	0,024	0,009	0,004	0,001	0,001	0,002	0,002	0,002	0,002	0,001
0,531	0,445	0,571	0,619	1,000	0,354	0,253	0,048	0,023	0,009	0,005	0,002	0,002	0,004	0,002	0,003	0,002	0,001
0,513	0,467	0,534	0,565	0,571	1,000	0,551	0,104	0,058	0,021	0,010	0,004	0,003	0,005	0,005	0,005	0,005	0,001
0,469	0,451	0,463	0,477	0,514	0,692	1,000	0,148	0,089	0,038	0,017	0,006	0,004	0,009	0,006	0,004	0,008	0,003
0,302	0,313	0,298	0,303	0,299	0,475	0,535	1,000	0,289	0,146	0,057	0,029	0,021	0,008	0,029	0,005	0,035	0,005
0,094	0,086	0,089	0,081	0,083	0,132	0,163	0,163	1,000	0,199	0,058	0,028	0,016	0,010	0,033	0,005	0,042	0,003
0,032	0,029	0,032	0,033	0,031	0,051	0,061	0,074	0,175	1,000	0,203	0,078	0,045	0,014	0,046	0,018	0,050	0,007
0,010	0,008	0,006	0,008	0,009	0,015	0,019	0,027	0,046	0,177	1,000	0,300	0,142	0,060	0,065	0,073	0,064	0,035
0,004	0,006	0,004	0,002	0,002	0,003	0,006	0,009	0,013	0,051	0,223	1,000	0,375	0,169	0,184	0,188	0,208	0,100
0,002	0,002	0,003	0,001	0,001	0,004	0,003	0,004	0,003	0,018	0,083	0,325	1,000	0,376	0,336	0,272	0,351	0,201
0,001	0,002	0,002	0,002	0,001	0,001	0,002	0,002	0,003	0,010	0,043	0,149	0,385	1,000	0,411	0,318	0,377	0,317
0,001	0,002	0,002	0,001	0,000	0,000	0,001	0,001	0,001	0,007	0,034	0,139	0,295	0,347	1,000	0,396	0,456	0,279
0,001	0,002	0,001	0,001	0,001	0,002	0,002	0,002	0,001	0,011	0,039	0,133	0,218	0,245	0,361	1,000	0,507	0,189
0,001	0,002	0,001	0,001	0,001	0,001	0,001	0,001	0,001	0,007	0,031	0,147	0,288	0,296	0,420	0,522	1,000	0,244
0,000	0,000	0,001	0,000	0,000	0,000	0,001	0,001	0,000	0,005	0,027	0,097	0,227	0,350	0,367	0,276	0,345	1,000

3.4 The Matching Results Using a Logistic Regression Classifier

To train the logistic regression classifier, firstly we must find reliable SIFT feature matches as positive samples. In particular, we want to train the classifier with pairs which the Lowe's method can't match but in reality they are matching key points. To find these pairs, the images in the dataset taken with a fixed camera so that image's (x,y) position could be used as a reference for correct match. To correct for a few pixels movement of the camera, alignment is made by looking at x and y coordinates of reference points in the images. We explain in more detail how we find such reference matches for training in Section 3.2.1.

For all the key points extracted from the first image, it is tested whether there is a key point in the same spot in the second image. If SIFT key points have been identified in the same spot in both of images, they were included as sample matches for the training set. To validate this method, some of the pair of keypoints (especially lower similarity of description) is selected, they were examined manually and were confirmed that these are indeed correct match of the same point.

With the described method we obtain usually a few hundred reference matching pairs out of a few thousand key points. Among the reliably correct matches obtained with this method, %70 of them used as a training set, %30 of them used as a test set. In addition, randomly selected mismatch pairs close to the number of correct matches (usually a bit more) from both images were included in the training and test sets as examples of the negative class. A logistic regression classifier is trained with this data and the performance is measured.

We also used a fixed threshold based matching in order to compare the performance of the classifier based method. The fixed threshold for each pair of images was computed from the training set so as to maximize matching performance in the training set. Then the performance was measured in the test set. In this manner the performance of the classical fixed threshold based method was calculated.

We compute the precision and recall values for all of the experiments described above. The results of these experiments are summarized in Table 1. The image pairs used in the experiments are shown in Figure 3.5. We observe that the precision values are high – sometimes equal to 1 – for both methods. This shows the distinctiveness of the SIFT descriptors. That is, if two SIFT descriptors match, then it is a correct match with a very high probability. But the problem that we want to solve is more about the recall rate. The main problem in matching day and night images is, because of the significant appearance change –

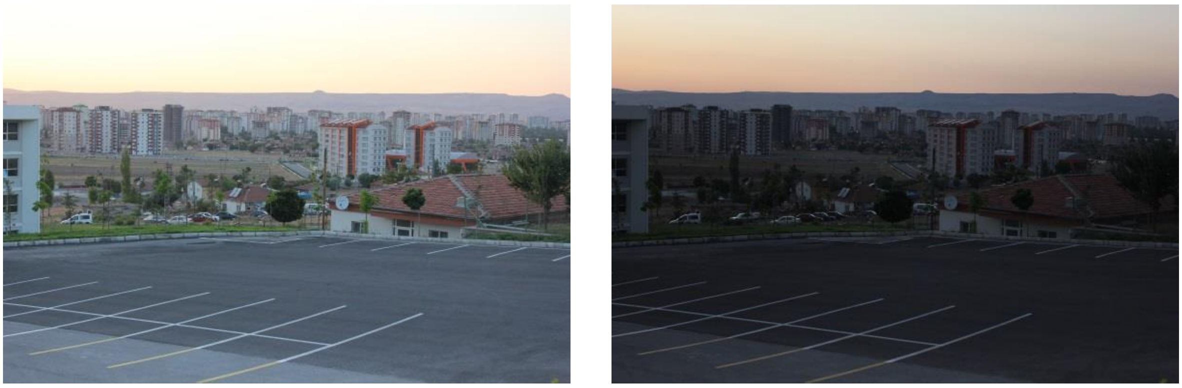
correct matches are missed. So, the main goal should be to increase recall rates while not reducing precision.

In this regard, we see that the recall rates for logistic regression are consistently better than threshold based method. In some cases the improvement is very high, as can be seen for m10-m13 pair, where recall rate is increased from 0.4 to 0.6. In most cases the increase is smaller, such as going from 0.83 to 0.90 or from 0.53 to 0.57. But what makes the results significant is that the recall rates are consistently better while the precision is preserved. As a result the suggested logistic regression matching method gives better results in a consistent manner compared to the widely used threshold method.

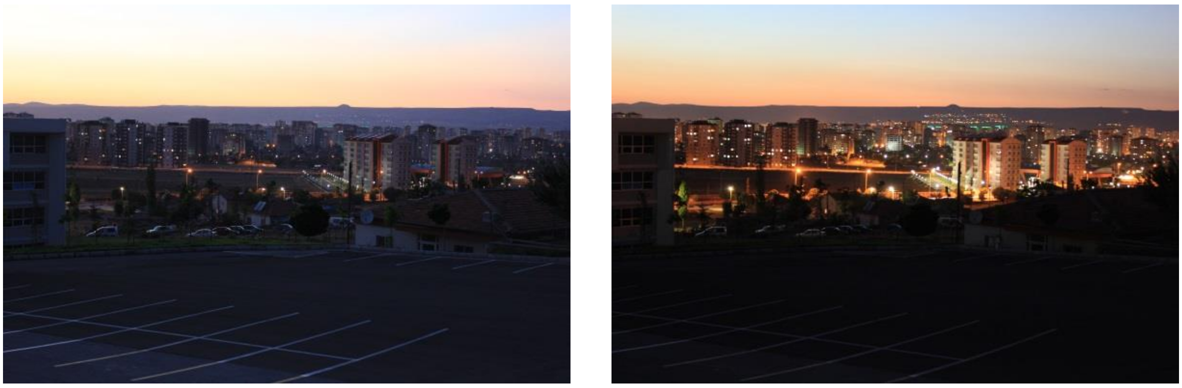
Note that the improvement in results come from a very small improvement in the matching method because we are using only the mean and standard deviation of the SIFT descriptors in addition to the distance. The threshold method uses only the distance. If a classifier is used only with the distance as the input, the theoretically best performance is given by a Bayesian classifier, which is the same as the best threshold training method we used in our experiment. So, we are comparing the performance of adding mean and standard deviation to the distance only input vector, and we see that even this small addition consistently improves the recall performance.

Table 3.2: The comparison of the performance of direct threshold value method (prior) and the performance of logistic regression method (proposed).

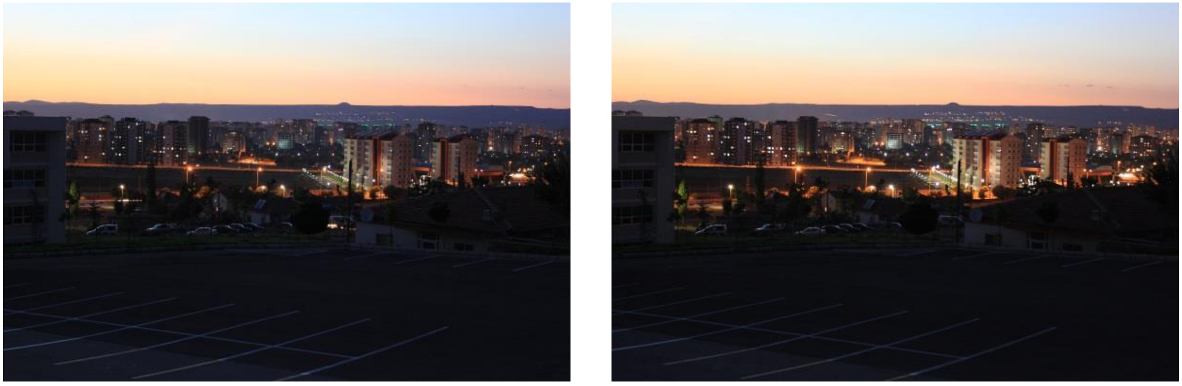
Image Pair Numbers	**Num. of keypoints in the 1st image**	**Num. of keypoints in the 2nd image**	**Matching reference pairs of keypoints**	**Matching with direct threshold**		**Matching with logistic regression**	
				Precision	**Recall**	**Precision**	**Recall**
m3-m8	8890	1102	308	1	0.9783	1	0.9891
m10-m13	2346	4932	135	1	0.4000	1	0.6000
m11-m12	2827	4131	100	1	0.8330	1	0.9000
m11-m15	5838	2823	236	0.9744	0.5352	0.9762	0.5775
k5-k9	13428	8423	642	0.9843	0.6443	0.9844	0.6495
k8-k15	10558	11296	235	0.8750	0.2000	0.8636	0.2714
k9-k10	8423	8581	853	0.9777	0.6836	0.9888	0.6914
k10-k15	8520	11926	376	0.9487	0.3274	0.9744	0.3363
k9-k19	8423	12837	532	1	0.3813	1	0.4000
k8-k17	10558	11650	463	0.9608	0.3525	0.9649	0.3957



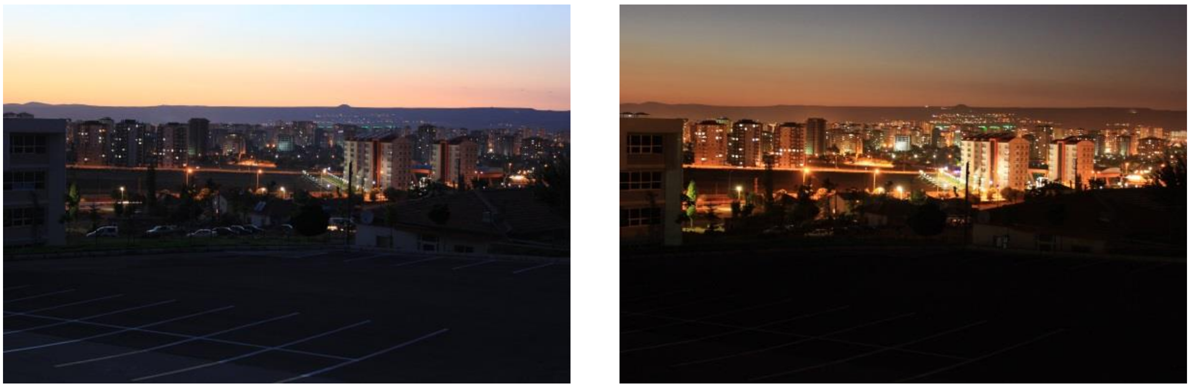
(a) m3-m8



(b) m10-m13

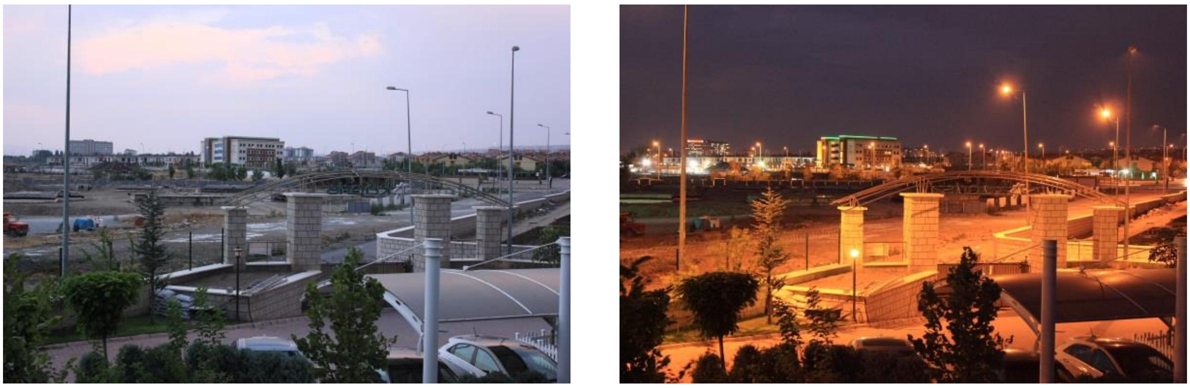


(c) m11-m12



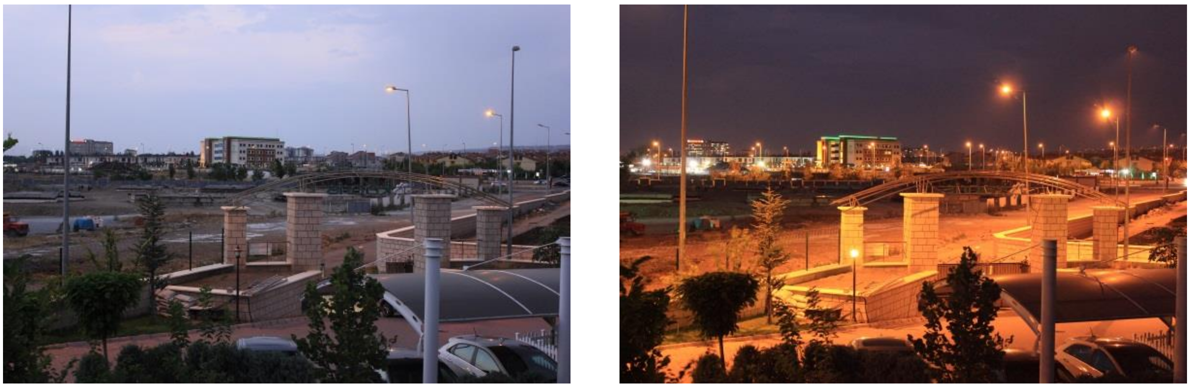
(d) m11-m15

(e) k5-k9



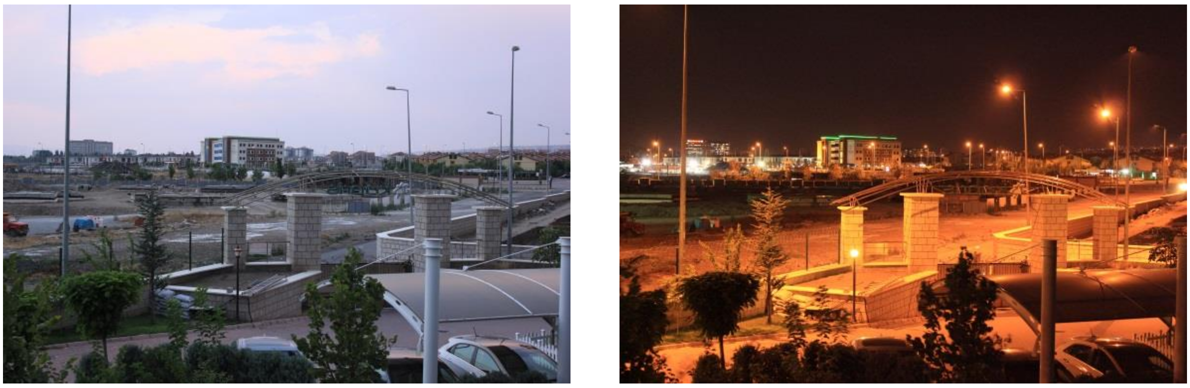
(f) k8-k15

(g) k9-k10



(h) k10-k15

(i) k9-k19



(j) k8-k17

Figure 3.5 Image pairs used in the matching experiments that compare logistic regression based matching and threshold based matching.

CHAPTER 4

CONCLUSION

Matching images is a widely used and needed operation in computer vision. Some of the areas that use image matching are object recognition, content based image retrieval, scene understanding, and location recognition.

Local descriptors have become the preferred technique used for image matching in recent years. SIFT and SURF are examples of the most commonly used among local descriptors. SIFT is chosen in this study as the local descriptor used for image matching, as it is the most widely used method in many applications. We first presented the dataset and the methodology we used for measuring the performance of location image matching in changing lighting conditions, especially from day to night. As the first step of image matching, we extracted feature points of images using the SIFT method. Then, we examined Lowe's algorithm to see how well it performs in matching images changing from day to night. The result is shown in Figure 3.4. According to the normalized graph in Figure 3.4 we see that the performance of a widely used threshold value method proposed by Lowe in matching images is completely inadequate especially for day and night location images. Therefore, the existing method is not sufficient in changing lighting conditions seen in day and night urban images.

In this thesis, we proposed a classification based model for image matching. We claimed that the increased information used in the matching decision, and use of a more powerful decision method such as a classifier from machine learning tools will allow making better decisions about matching local descriptors from day and night images.

A problem that needs to be solved for the classifier based approach is obtaining reference correct matches to train the classifier with. We obtained matching SIFT vectors from day and night images by taking images of a static urban scene with a fixed camera for the duration from day to night. We also aligned images to pixel level accuracy to obtain a reliable training set. We obtained correct matches from pairs of aligned images by finding SIFT descriptors that are at the same location in both images. This method of obtaining reference correct matches is also a novel contribution of this thesis.

In the experiments we conducted, we limited the input vector to the classifier to only three attributes: distance between descriptors, the mean of the descriptors, and the standard deviation of the descriptors. We also used a simple but effective classifier, which is the logistic regression method. It is successfully used in many applications and performs well. However, a more extended input vector – possibly concatenation of the two descriptors, which gives a 256 dimensional input vector – and a more powerful classifier such as multiple stage neural networks could further improve the results presented here. In that case, a much larger training data set would be required to avoid overfitting, which we faced when we used a 256 dimensional input with the logistic regression classifier.

The results we obtained in the experiments indicate a consistent improvement in the recall rates when logistic regression is used for matching SIFT descriptors. Note that this improvement is achieved with a very small improvement in the matching method, by just adding mean and standard deviation of the descriptors to the input. Hence, we conclude from this result that the idea of using a classifier and a richer input for the matching of descriptors is a promising direction that will help solve the problem of matching day and night images. The proposed method improves the descriptor to descriptor matching performance, which in turn improves the reliability of these numbers for image to image matching.

As for future work, we plan to work with more training samples. A large training data set will allow the use of a more powerful classifier such as a complex neural network. In addition, instead of using 3-dimensional feature vectors as input to the classifier, it can be improved with a more extended feature vector, possibly using full descriptors concatenated together. Also, this approach can be applied to matching of image under other types of variations such as seasonal changes in a scene.

APPENDIX

LOWE'S MATCHING ALGORITHM

```
function num = match(image1, image2)

% Find SIFT key points for each image
[im1, des1, loc1] = sift (image1);
[im2, des2, loc2] = sift (image2);

% for efficiency in Matlab, it is cheaper to compute dot products between
% unit vectors rather than Euclidean distances.  Note that the ratio of
% angles (acos of dot products of unit vectors) is a close approximation
% to the ratio of Euclidean distances for small angles.
%
% distRatio: Only keep matches in which the ratio of vector angles from the
%   nearest to second nearest neighbor is less than distRatio.
distRatio = 0.6;

% For each descriptor in the first image, select its match to second image.
des2t = des2';                          % Precompute matrix transpose
for i = 1 : size(des1,1)
   dotprods = des1(i,:) * des2t;        % Computes vector of dot products
   [vals,indx] = sort(acos(dotprods));  % Take inverse cosine and sort results

   % Check if nearest neighbor has angle less than distRatio times 2nd.
   if (vals(1) < distRatio * vals(2))
      match(i) = indx(1);
   else
      match(i) = 0;
   end
end

% Create a new image showing the two images side by side.
im3 = appendimages(im1,im2);

% Show a figure with lines joining the accepted matches.
figure('Position', [100 100 size(im3,2) size(im3,1)]);
colormap('gray');
imagesc(im3);
hold on;
cols1 = size(im1,2);
for i = 1: size(des1,1)
  if (match(i) > 0)
   line([loc1(i,2) loc2(match(i),2)+cols1], ...
        [loc1(i,1) loc2(match(i),1)], 'Color', 'c');
  end
end
hold off;
num = sum(match > 0);
fprintf('Found %d matches.\n', num);
```

REFERENCES

[1] Duncan Robertson and Roberto Cipolla. An image-based system for urban navigation. In BMVC, 2004.

[2] Schindler, Grant, Matthew Brown, and Richard Szeliski. "City-scale location recognition." *Computer Vision and Pattern Recognition, 2007. CVPR'07. IEEE Conference on*. IEEE, 2007

[3] Boyd, Carl R., Mary Ann Tolson, and Wayne S. Copes. "Evaluating trauma care: the TRISS method." *Journal of Trauma and Acute Care Surgery* 27.4 (1987): 370-378.

[4] Tang, Siew Tzuh, et al. "Prevalence of severe depressive symptoms increases as death approaches and is associated with disease burden, tangible social support, and high self-perceived burden to others." *Supportive Care in Cancer*(2015): 1-9.

[5] Strano, M., and B. M. Colosimo. "Logistic regression analysis for experimental determination of forming limit diagrams." *International Journal of Machine Tools and Manufacture* 46.6 (2006): 673-682.

[6] Mikolajczyk, Krystian, and Cordelia Schmid. "A performance evaluation of local descriptors." *Pattern Analysis and Machine Intelligence, IEEE Transactions on*27.10 (2005): 1615-1630.

[7] Tuytelaars, Tinne, and Luc J. Van Gool. "Wide Baseline Stereo Matching based on Local, Affinely Invariant Regions." *BMVC*. Vol. 412. 2000.

[8] Lowe, David G. "Object recognition from local scale-invariant features."*Computer vision, 1999. The proceedings of the seventh IEEE international conference on*. Vol. 2. Ieee, 1999.

[9] Liu, Li, and Paul Fieguth. "Texture classification using compressed sensing."*Computer and Robot Vision (CRV), 2010 Canadian Conference on*. IEEE, 2010.

[10] Lowe, David G. "Distinctive image features from scale-invariant keypoints."*International journal of computer vision* 60.2 (2004): 91-110.

[11] Maddern, William, and Stephen Vidas. "Towards robust night and day place recognition using visible and thermal imaging." *RSS 2012: Beyond laser and vision: Alternative sensing techniques for robotic perception* (2012).

[12] Joglekar, Jyoti, Shirish S. Gedam, and IIT CSRE. "Image matching with SIFT features–a probabilistic approach." *Proceedings of IAPRS* 38 (2010): 7-12.

[13] Zhou, Huiyu, Yuan Yuan, and Chunmei Shi. "Object tracking using SIFT features and mean shift." *Computer vision and image understanding* 113.3 (2009): 345-352.

[14] Juan, Luo, and Oubong Gwun. "A comparison of sift, pca-sift and surf."*International Journal of Image Processing (IJIP)* 3.4 (2009): 143-152.

[15] http://www.vlfeat.org/overview/sift.html

[16] Grauman, Kristen, and Bastian Leibe. *Visual object recognition*. No. 11. Morgan & Claypool Publishers, 2010.

[17] Harris, Chris, and Mike Stephens. "A combined corner and edge detector."*Alvey vision conference*. Vol. 15. 1988.

[18] Li-zhong, W. A. N. G., et al. "Application of feature point detection and matching using scale invariant feature transform method in the breeding cow conformation image processing." *Journal of Anhui Agricultural University* 3 (2010): 045.

[19] Bay, Herbert, et al. "Speeded-up robust features (SURF)." *Computer vision and image understanding* 110.3 (2008): 346-359.

[20] Brown, Matthew, and David G. Lowe. "Invariant Features from Interest Point Groups." *BMVC*. No. s 1. 2002.

[21] Mikolajczyk, Krystian, and Cordelia Schmid. "Indexing based on scale invariant interest points." *Computer Vision, 2001. ICCV 2001. Proceedings. Eighth IEEE International Conference on*. Vol. 1. IEEE, 2001.

[22] Abubakar Rabiu. “Automatic defect detection in fabrics using computer vision tecniques.” Master’s Thesis. Meliksah University. 2014

[23] Demarchi, Bruno Kewitz, and Felipe Pilon. "Canny Edge Detector."

[24] Ali, Mohamed, and David Clausi. "Using the Canny edge detector for feature extraction and enhancement of remote sensing images." Geoscience and Remote Sensing Symposium, 2001. IGARSS'01. IEEE 2001 International. Vol. 5. IEEE, 2001.

[25] Ding, Lijun, and Ardeshir Goshtasby. "On the Canny edge detector." Pattern Recognition 34.3 (2001): 721-725.

[26] Harris, Chris, and Mike Stephens. "A combined corner and edge detector." Alvey vision conference. Vol. 15. 1988.

[27] Chen, Bai-Fan, and Zi-Xing Cai. "Harris corner detection based on theory of scale-space." Zhongnan Daxue Xuebao (Ziran Kexue Ban)/Journal of Central South University (Science and Technology) 36.5 (2005): 751-754.

[28] Mao, Yan-ming, et al. "An Improved Corner Detection Method Based on Harris [J]." Computer Technology and Development 5 (2009): 036.

[29] Damerval, Christophe, and Sylvain Meignen. "Blob detection with wavelet maxima lines." *IEEE Signal Processing Letters* 14.1 (2007): 39-42.

[30] Fukui, Kazuhiro, and Osamu Yamaguchi. "Facial feature point extraction method based on combination of shape extraction and pattern matching." Systems and Computers in Japan 29.6 (1998): 49-58.

[31] Nazlı Tekin, Kadir Aşkın Peker, "Gece ve Gündüz Lokasyon İmgelerinin SIFT Öznitelikleri ve Lojistik Regresyonla Eşleştirilmesi", SİU2015, 23. Sinyal İşleme ve İletişim Uygulamaları Kurultayı, May. 2015, Malatya,Turkey.