# INTEREST POINT MATCHING ACROSS ARBITRARY VIEWS

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

BY

İLKER BAYRAM

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

JUNE 2004

Approval of the Graduate School of Natural and Applied Sciences

_____

Prof. Dr. Canan Özgen

Director

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

_____

Prof. Dr. Mübeccel Demirekler

Head of Department

This is to certify that we have read this thesis and that in our opinion it is fully adequate, in scope and quality, as a thesis for the degree of Master of Science.

_____

Assoc. Prof. Dr. A. Aydın Alatan

Supervisor

Examining Committee Members

Prof. Dr. Kemal Leblebicioğlu (METU, EE) _____

Assoc. Prof. Dr. A. Aydın Alatan (METU, EE) _____

Prof. Dr. Mete Severcan (METU, EE) _____

Assoc. Prof. Dr. Gözde Bozdağı Akar (METU, EE) _____

Uğur Topay, M.Sc. (TÜBİTAK-SAGE) _____

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

Name, Last name : İlker Bayram

Signature :

# ABSTRACT

INTEREST POINT MATCHING ACROSS ARBITRARY VIEWS

Bayram, İlker
M.Sc., Department of Electrical and Electronics Engineering
Supervisor: Assoc. Prof. Dr. A. Aydın Alatan

June 2004, 129 pages

Making a computer 'see' is certainly one of the greatest challanges for today. Apart from possible applications, the solution may also shed light or at least give some idea on how, actually, the biological vision works. Many problems faced en route to successful algorithms require finding corresponding tokens in different views, which is termed the correspondence problem. For instance, given two images of the same scene from different views, if the camera positions and their internal parameters are known, it is possible to obtain the 3-Dimensional coordinates of a point in space, relative to the cameras, if the same point may be located in both images. Interestingly, the camera positions and internal parameters may be extracted solely from the images if a sufficient number of corresponding tokens can be found. In this sense, two subproblems, as the choice of the tokens and how to match these tokens, are examined. Due to the arbitrariness of the image pairs, invariant schemes for extracting and matching interest points, which were taken as the tokens to be matched,

are utilised. In order to appreciate the ideas of the mentioned schemes, topics as scale-space, rotational and affine invariants are introduced. The geometry of the problem is briefly reviewed and the epipolar constraint is imposed using statistical outlier rejection methods. Despite the satisfactory matching performance of simple correlation-based matching schemes on small-baseline pairs, the simulation results show the improvements when the mentioned invariants are used on the cases for which they are strictly necessary.

Keywords: Interest point, matching, stereo correspondence, scale-space, invariant.

# ÖZ

RASTGELE GÖRÜNTÜLERDE İLGİ NOKTASI EŞLEME

Bayram, İlker
Yüksek Lisans, Elektrik ve Elektronik Mühendisliği Bölümü
Tez Yöneticisi: Doç. Dr. A. Aydın Alatan

Haziran 2004, 129 sayfa

Bir bilgisayarın 'görmesini' sağlamak, hiç kuşku yok ki günümüzün en güç problemlerinden birisidir. Başarılı olunduğu takdirde doğabilecek olağan uygulamaların yanı sıra, çözüm biyolojik görmenin de nasıl gerçekleştiğine ışık tutabilir veya en azından bir fikir verebilir. Başarılı algoritmalara giden yolda birçok problem, farklı, birbirine tekabül eden işaretlerin farklı bakış açılarından bulunabilmesine gereksinim duymaktadır. Bu probleme eşlik problemi denir. Örneğin, aynı sahnenin farklı açılardan çekilmiş iki fotoğrafı verildiğinde, eğer kameraların birbirine göre konumları ve kameraların iç parametreleri biliniyorsa, uzaydaki bir noktanın kameralara göre 3-boyutlu koordinatları, noktanın görüntüsünün nerede olduğu her iki fotoğrafta da tespit edildiği takdirde, saptanabilir. İlginçtir ki, kamera konumları ve iç parametreleri, yeterli sayıda birbirine karşılık gelen işaret bulunabiliyorsa, yalnızca bu bilgi kullanılarak elde edilebilir. Bu anlamda, iki alt problem, sözü geçen işaretlerin nasıl seçileceği ve bu yapıldıktan sonra bunların nasıl eşleneceği, incelenmiştir. Fotoğrafların rastgele olması

nedeniyle, eşlenecek işaretler olarak alınan ilgi noktalarının çıkarılmasında ve bunların eşlenmesinde birtakım değişmez yöntemler kullanılmıştır. Bu fikirleri ve yöntemleri takdir edebilmek amacıyla, ölçek-uzayı, dönel ve ilgin değişmezler gibi kavramlar tanıtılmıştır. Problemin geometrisi kısaca gözden geçirilmiş ve epipolar kısıtlaması istatistiksel aykırı değer reddetme yöntemleri kullanılarak uygulanmıştır. Basit ilintiye dayalı yöntemlerin kısa taban çizgisine sahip fotoğraf çiftlerindeki tatmin edici başarımına rağmen, benzetim sonuçları, bahsedilen değişmezlerin gerekli olduğu durumlarda kullanılmalarıyla başarımı iyileştirdiklerini göstermektedir.

Anahtar Sözcükler: İlgi noktası, eşleme, stereo eşlik, ölçek uzayı, değişmez.

To My Family and Gizem,

# ACKNOWLEDGEMENTS

I would like to express my gratitude to my supervisor Assoc. Prof. Dr. A. Aydın Alatan for his guidance throughout the preparation of this thesis.

I also would like to thank TÜBİTAK-SAGE for supporting the thesis.

# TABLE OF CONTENTS

# LIST OF FIGURES

# LIST OF TABLES

# CHAPTER I

# INTRODUCTION

Given images of the same scene from arbitrary views, finding tokens in each image which correspond to the same physical points is a crucial step for many higher order tasks in computer vision and related areas. The general problem is called the correspondence problem and depending on the relations between the images, the solutions take special forms. In the following, the problem will be introduced along with proposed solutions and main steps to reach these solutions.

## 1.1 Correspondence Problem

The correspondence process may be stated to be ([38]) : “the process that identifies elements in different views as representing the same object at different times, thereby maintaining the perceptual identity of objects in motion or change.” It should be noted that if the scene under observation does not undergo any change, the problem is equivalent to the stereo correspondence problem (see Figure 1.1), that is of matching two images of the same scene from different viewpoints ([13]).

The problem defined above is a fundamental problem for both biological and computer vision. In fact, the correspondence process may be claimed to be the lowest level operation to be performed by a vision system, since reliable solutions are required by higher order tasks like camera calibration ([41)], three-dimensional structure estimation, depth perception ([38], [15]), object recognition ([19]), etc.

If one would like to categorize the suggested solutions to the problem, she/he might distinguish between proposals based on affinity measures considering mainly, the relations of the elements to be matched

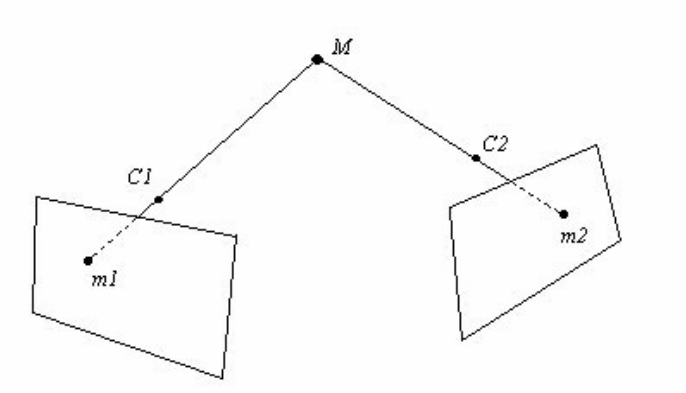



Figure 1.1 The stereo correspondence problem is to match the image of *M* in the first image *m1* with that in the second image *m2*. *C1* and *C2* are the centers of the lenses of the first and the second camera respectively.

with nearby elements ([13], [38]), and those, considering some sort of cross-correlation of the features representing the regions surrounding the elements with those of candidate elements' regions. The first approach is feasible for line segments, circular arcs or 'edges' in general, where simple local structures possesing somewhat unique characteristics in one image are sought in the other image ([13], [38]). This approach eliminates the difficulties arising from cross-correlating non-smooth surfaces with a great number of discontinuities, occlusions, etc. (cross-correlation is not a reliable measure in this case). What is dealt with is not gray-level structures but relationships of binary features. However, when it comes to point matching in a practical situation, if the number of points to be matched are rather high, considering the relationship of each candidate with its neighbors in one image and comparing these relationships with those of the other image makes the problem quite difficult to track. One is then practically, restricted to the small displacement case where the difference between the two images is quite low and the match for a particular point lies in the proximity of the point.

The other approach seems better suited for the case of point matching. If the points to be matched do not lie on discontinuities, it might be expected that cross-correlation of the raw gray-level intensities or outputs of a set of linear spatial filters ([15]) could be a good measure for detecting corresponding pairs. These alone can handle simple translations of any amount and yield satisfactory results. The outputs of these measures may also be used for reducing the number of candidate matches so that the affinity measure becomes applicable once again ([41], [21]). Unfortunately, even when there exists a slight rotation among the images where correspondences will be sought (which will be referred to as images to be matched), the outputs of the mentioned measures start to yield irrelevantly poor results. Changes in scale also affect the measures similarly. This shows itself when one decides on the sizes of the windows to be correlated or of the filters to be used. There simply does not exist an ideal window size for an arbitrary point of an arbitrary image. In order to overcome the problems mentioned, one is led to use invariant measures. These are realised by comparing invariant descriptors of the regions surrounding the points to be matched. The descriptors are invariant in the sense that they are not affected, if some particular transformation is applied to the image at hand. The transformations include upto a general affine transformation (including scale, rotations, stretchings, etc.), which approximates locally, a general perspective distortion for a planar patch ([12]).

In this thesis, point matching in two arbitrary images, have been set as the goal. The motivation for this goal is given the mentioned images and no other information to calibrate the cameras by which the images were obtained, i.e., estimate the relative positions and the internal parameters (like focal length, pixel characteristics, etc.) of the cameras. If a sufficient number of point matches with sufficient accuracy may be obtained, the mentioned attributes may be derived solely from the images ([12]).

## 1.2 Building Blocks for the Solution

The solution of the problem may be separated into different stages as in Figure 1.2.

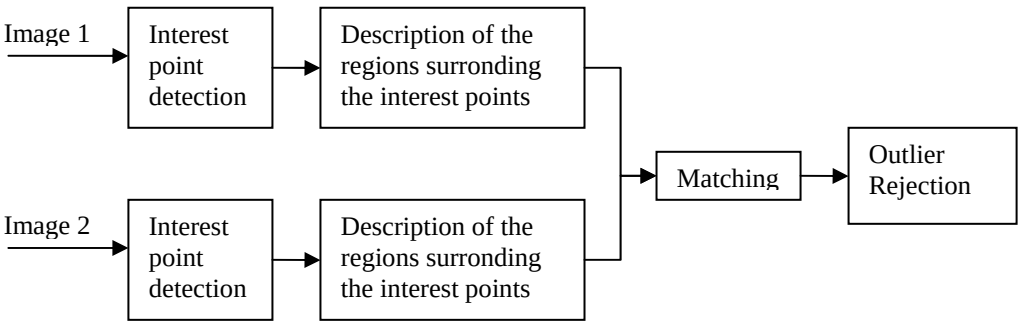



Figure 1.2 Building blocks for the solution to the correspondence problem

### 1.2.1 Interest Point Detection

The correspondence process may be visualized as a function mapping points in one image to those of the other ([38]). A natural question to ask is then, "What is the domain and the range of this function?". Should all the points in an image be mapped to the points in the other image or should a selection be made so as to narrow the domain and range? Considering two images from sufficiently different viewpoints, one immediately observes that not all points in one image exists in the other. Moreover, since points are no more expected to be matched within their close neighborhoods, there is no reliable way to match points where no significant change occurs. Even if the point lies on an 'edge' (whatever that might mean; the formal definition is given later in this thesis), it may be impossible to distinguish the match of the point from its neigbors, provided that the edge is a straight one in some neighborhood of the point. Following these observations, it may be postulated that 'corners' (another question mark to be clarified later) should be taken as the points to be

matched. A great deal of effort has been made to acccurately determine the corners and the matter will be discussed thoroughly. Another view is to define a measure and choose interest points as points for which the measure assumes a local maximum and is above some threshold. The approach is different from that of the corner detectors in the sense that the searched sructure is rather relaxed. A point that should be taken care of in this case is the invariance of the scheme under transformations of the image. For instance, in the given two images, the image of the same physical point should be chosen as the interest point for a correct match to occur. Note that a sharp corner may be transformed into a dull one but that the position of the corner in any case corresponds to the same physical point. Thus, there does not exist such a danger for corner detectors.

### 1.2.2 Description of Regions

After the interest point detection step, it comes to describe the patch surrounding these interest points for comparison with those of the other image. A straightforward solution would be to take the raw pixel brightness values in some neighborhood of the point. The first question to pose in an attempt to realise this scheme, would probably be about the sizes of the neighborhoods. Unfortunately, there exists no explicit answer. The solution should take into acount the particular scale of the interest point. Obviously, a scale invariant description is required. Similarly, it is seen that the description should be indifferent to other local transformations. The greatest degree of invariance achieved in this thesis will include affine transformations.

### 1.2.3 Matching

At this step, the interest points in each image are tried to be matched taking into consideration their invariant description, based on an appropriate comparison. Following the comparisons, it may be desired to apply also an affinity measure in connection with neighboring constraints

([41], [21], [9]). This might be the case, if the descriptions are not strong enough to uniquely characterise the image patch which is often the case, if a window of predetermined size, containing raw pixel brightness values is used as the descriptor.

### 1.2.4 Outlier Rejection

The results of the matching stage may contain a significant number of false matches. If however, one had the mentioned camera parameters at hand, she/he could eliminate most of these false matches, since they would not obey the constraints that the parameters would bring. Statistical outlier rejection methods ([41], [21]) are indeed very useful in such a setting. The idea is to select small subsets of putative matches, determine the related parameters that the model at hand posseses using this small subset and see if the constraints, brought by these parameters, are obeyed by a number of matches out of the chosen subset. This way, one can eliminate irrelevant (in terms of matching the model) matches.

## 1.3 Outline of the Thesis

Following this introduction, interest point detection methods are discussed in Chapter II. For this purpose, a corner model is given for insight on using different detection schemes. Major corner detection schemes are presented along with corresponding strengths, weaknesses and their performances are tried to be estimated through a theoretical reasoning.

Chapter III first presents the prerequisites for obtaining invariant region description and interest point detection methods. These are mainly concentrated around the concept of scale-space. Since the field itself is an influential and rich research area, a concise presentation of the key concepts are developed in a bottom-up manner. Rotational, scale and affine invariants that have proved to be useful are discussed in this chapter.

In Chapter IV, different matching schemes are explained which do or do not exploit the different invariants of the previous chapters. The chapter also includes a brief introduction to the scene geometry needed to develop the statistical outlier rejection methods. Matching schemes are presented starting from the weakest in terms of robustness against the arbitrariness of the views and ending with an algorithm which is intended to be able to handle local affine transformations.

Results and particular parameters of the methods for obtaining the results are shown in Chapter V. The results for interest point detectors and matching schemes are presented in order.

Finally, Chapter VI is the conclusion chapter where different methods presented in the thesis are evaluated.

# CHAPTER II

# CORNER DETECTION

Extraction of salient features in an image is an important task for many purposes like object recognition, image retrieval, stereo matching, etc. For the purpose of point matching, corners are natural candidates as the features to be used. Thus, accurate localization of these features ([8]), repeatability of the detectors ([31]), and invariance of the detectors under different geometric and photometric transformations ([26]) are important issues to be considered. Different detectors have been implemented, each representing a different reasoning. In this chapter, these methods are analyzed. The simulation results are presented briefly in Chapter V along with some details of their implementation.

(a) (b)

Figure 2.1 Corner model. (a) Ideal corner with $\Theta = 45^{\circ}$, (b) Ideal corner smoothed by a 2-D Gaussian of variance $\sigma^2 = 1$.

## 2.1 A Corner Model

In order to analyze the behaviors of the detectors, the corner model of [8] is being used. In this formulation, let *U* denote the step function:

$$U(x)=\begin{cases}1 & if \quad x>0\\ 0 & otherwise\end{cases} \tag{2.1}$$

An ideal corner with an edge along the x-axis and an angle Θ can be represented by the following 2-D function:

$$I_{\Theta}(x,y)=U(mx-y)\cdot U(y) \tag{2.2}$$

where $m=\tan\Theta$.

Convolving (2) with a 2-D Gaussian yields

$$S(x,y)=\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} g(x-a)\cdot g(y-b)\cdot U(ma-b)\cdot U(b)\;\; dadb \tag{2.3}$$

where

$$g(x)=\frac{1}{\sigma\sqrt{2\Pi}}e^{-\frac{x^2}{2\sigma^2}} \tag{2.4}$$

*S(x,y)* is presented in Figure 2.1.

## 2.2 A Note on Edge Detectors

Two well-known approaches on edge detection consist of extracting the local maxima of the gradient in the gradient direction and finding the zero crossings of the response to the Laplacian (or, alternatively Laplacian of Gaussian) operator. The former of these methods is equivalent to first suppressing the non-maxima of the gradient and then finding points where the second directional derivative in the direction of the gradient is zero. Explicitly, this is equal to (after non-maxima suppression), finding zeros of,

$$\frac{\partial^2 \overline{S}(x,y)}{\partial n^2}=\frac{\partial^2}{\partial t^2}\overline{S}\left(x+t\frac{S_x}{\sqrt{S_x^2+S_y^2}},y+t\frac{S_y}{\sqrt{S_x^2+S_y^2}}\right)\Big|_{t=0}$$

$$= \frac{\partial}{\partial t}\left[ \begin{array}{l} \dfrac{S_x}{\sqrt{S_x^2+S_y^2}}\overline{S_x}\left( x+t\dfrac{S_x}{\sqrt{S_x^2+S_y^2}}, y+t\dfrac{S_y}{\sqrt{S_x^2+S_y^2}} \right) \\ +\dfrac{S_y}{\sqrt{S_x^2+S_y^2}}\overline{S_y}\left( x+t\dfrac{S_x}{\sqrt{S_x^2+S_y^2}}, y+t\dfrac{S_y}{\sqrt{S_x^2+S_y^2}} \right) \end{array} \right]_{t=0} \qquad (2.5)$$

$$= \frac{S_{xx}S_x^2 + 2S_{xy}S_xS_y + S_{yy}S_y^2}{S_x^2+S_y^2}$$

where $\overline{S}(\ldots)$ denotes the function and *S*, the value that the function assumes at the given point, with subscripts denoting partial differentiation.

The zero crossings of the expression in the case of an angle of $\Theta = 45^\circ$ smoothed by a Gaussian, is shown in Figure 2.2.a. The rounding effect is well-observed in the vicinity of the corner. Particularly, it is noted that in the origin, the expression is non-zero, i.e. the corner is not an edge point according to this criterion. Deriche and Giraudon ([8]) determined the location of the particular edge point on the angle bisector and found that the displacement is a function of both the bandwidth of the Gaussian and the angle of the corner. The displacement increases with decreasing Gaussian bandwidth and/or decreasing corner angle.

In the latter edge detection scheme, the zero crossings of the Laplacian image is searched. The Laplacian image is given by:

$$\nabla^2 S(x,y) = S_{xx}(x,y) + S_{yy}(x,y) \qquad (2.6)$$

It is well known that near the corner, the zero crossings of the Laplacian deviates from the true edges. However, at the origin, the function takes the value of zero, i.e. the exact corner point is a part of the edge map (see Figure 2.2.b).

The Laplacian may be interpereted as follows:

Let [a,b] be an arbitrary direction, i.e.,

$$a^2 + b^2 = 1 \qquad (2.7)$$

Then, one can write

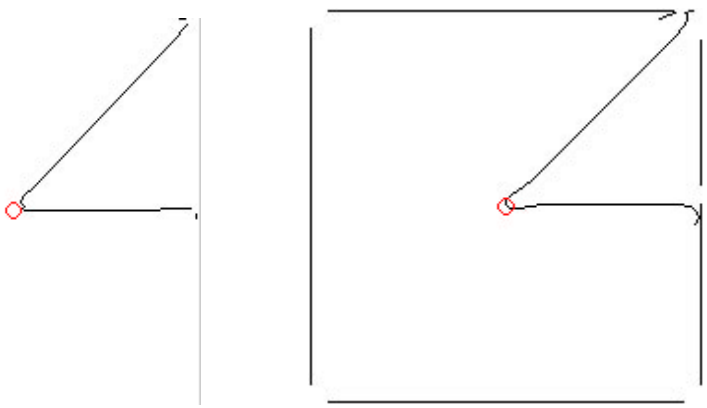
$$\nabla^2 S(x,y) = \begin{bmatrix} a & b \end{bmatrix}\begin{bmatrix} S_{xx} & S_{xy} \\ S_{yx} & S_{yy} \end{bmatrix}\begin{bmatrix} a \\ b \end{bmatrix} + \begin{bmatrix} -b & a \end{bmatrix}\begin{bmatrix} S_{xx} & S_{xy} \\ S_{yx} & S_{yy} \end{bmatrix}\begin{bmatrix} -b \\ a \end{bmatrix} \qquad (2.8)$$

In words, the Laplacian is equal to the sum of second directional derivatives in any two orthogonal directions. If one of these directions are taken to be the gradient direction, it can be written:

$$\nabla^2 S(x,y) = \frac{\partial^2 S}{\partial n^2} + \frac{\partial^2 S}{\partial n_\perp^2} \tag{2.9}$$

where the terms represent the second directional derivatives in the direction of the gradient and the direction orthogonal to the gradient direction.

The difference between the two approaches is observed more clearly, as the first term in (2.9) is equal to (2.5). The significance of the remaining term in (2.9) will be explained in the following sections.



(a) (b)

Figure 2.2 Edges of a smoothed ideal corner, detected using (a) maxima of the gradient, (b) zeros of LoG. The exact location of the corner is the center of the circle.

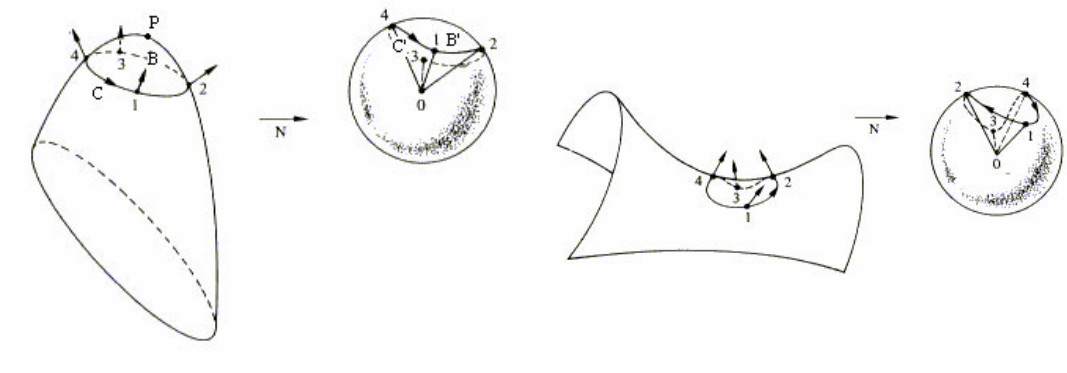
## 2.3 Gaussian Curvature

On a curve, the *curvature* of a point is defined as the magnitude of the rate of change of the tangent vector, when the curve is parameterized so that the curve has unit speed, i.e. the tangent vector has unit magnitude ([7]) (It should be noted that unit speed parameterization is not a restriction and that any curve may be parameterized this way). The

curvature may be said to measure how much the curve deviates from being straight or actually, 'curves'.

Correspondingly, for a surface, it is of concern to measure how much the surface at a specific point deviates from being a plane. This amounts to finding how the normal vector of the surface at or in the vicinity of the specified point changes. Observing that the normal vector changes rapidly for a sharp peak (the sharper, the more rapid), Gauss' approach to defining the curvature might be better enjoyed. The idea is as follows:

Consider for a point P on a surface (seeFigure 2.3(a)), a closed curve C enclosing a region B containing P. The image of the normal vectors of C on the sphere of unit radius is also a closed curve, C´, enclosing a region on the sphere B´. A measure is obtained if the areas of the regions are compared as area(B´)/area(B), where, in the limit, area(B) tends to zero. A distinction between saddle-like and peak-like points can also be made if the orientations of C and C´ are taken into account. For saddle-like points, the orientations are reversed (see Figure 2.3(b)) ([7]).




(a) (b)

Figure 2.3 Gauss map. (a)at an elliptic point, (b) at a hyperbolic point. Note that the orientation of the curve is reversed for the Gauss map around the hyperbolic point [7].

Interestingly, the measure outlined ('very' informally) above turns out to be equal to the following (when the measure is non-zero):

Consider a plane containing the normal N and parallel to the tangent direction $v$ to the surface at $p$ (Figure 2.4). The intersection of this plane with the surface is a planar curve C including $p$. The curvature of this planar curve at $p$ may be calculated. As this plane is rotated about the

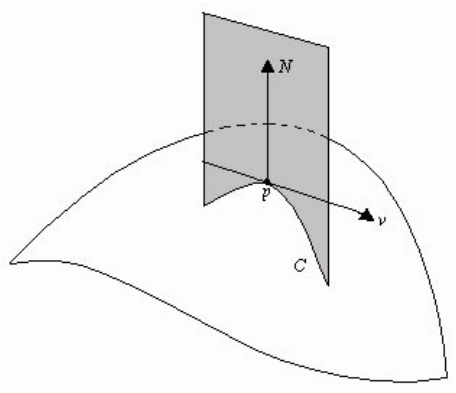



Figure 2.4 A surface, and its normal. The Gaussian curvature is the product of the maximum and minimum curvature yielding curves at $p$ generated by intersecting the normal with the surface.

normal to the surface, different curvature values at $p$ resulting from the different curves, obtained by intersecting the plane with the surface are obtained. The minimum and maximum (negative values are defined considering the direction of the normal) curvatures $K_{min}$, $K_{max}$ are called the *principal curvatures*. The *Gaussian curvature K* is defined as the product of the principal curvatures ([7]).

$$K = K_{\min} \cdot K_{\max} \tag{2.10}$$

The Gaussian curvature for the graph of a function $z = f(x, y)$, parameterized by $(x, y) \rightarrow (x, y, f(x, y))$ may be explicitly written as ([7])

$$K = \frac{I_{xx} I_{yy} - I_{xy}^2}{\left(1 + I_x^2 + I_y^2\right)^2} \tag{2.11}$$

A point on the surface is said to be, an *elliptic point* if $K>0$, a *hyperbolic point* if $K<0$, *a parabolic point* if $K=0$ and if one of the principal curvatures is non-zero. Figure 2.5 illustrates these points on the previous corner model.

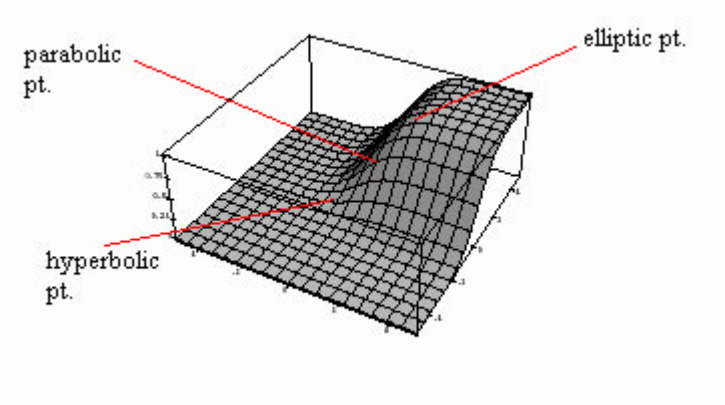



Figure 2.5 Examples of elliptic, parabolic, hyperbolic points on a Gaussian smoothed $90^{\circ}$ corner

Observing the Gaussian curvature values of the corner model given in Figure 2.6, the algorithm for finding corners proposed by Dreschler and Nagel may be appreciated. The algorithm is as follows ([8]):

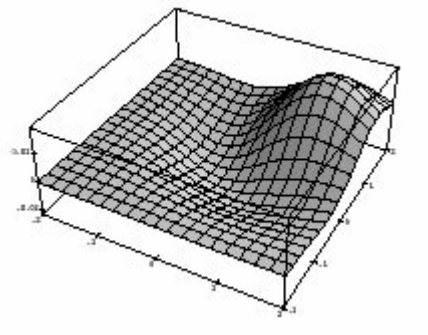


Figure 2.6 Gaussian Curvature on a Gaussian smoothed $90^{\circ}$ corner. Points above ground are elliptic, points below ground are hyperbolic and the points on the ground are parabolic or planar points.

Algorithm 2.1 :

1. Compute the Gaussian curvature.
2. Select locations of Gaussian curvature extrema.
3. Match each elliptic maxima with a hyperbolic minima. The principal curvatures of the matched extrema should approximately be alligned.

4. For a particular match, consider the segment joining the elliptic maximum with the hyperbolic minimum. The point at which the Gaussian curvature is equal to zero, on this segment is taken to be the corner for that particular match.

The major flaw in this approach ([8]) is noted to be related to the position of the hyperbolic extrema. As the elliptic extrema is always inside the corner, the hyperbolic extrema is expected to be outside the corner for accurately locating the corner. However, this is observed ([8]) not to be the case for corners with small angles and thus, the detection is not accurate.

## 2.4 Corners as Local Maxima of Curvature Along Edges

A different approach for detecting corners is based on the idea to first extract the edges as a chain code and then search for local maxima of the curvature on this chain code ([14]). Apparently, the first stage, consisting of extracting the edges requires an edge detector. Experiments were done using two different edge detectors, namely Canny edge detector and LoG edge detector. The main difference between the two is their scheme for obtaining the edge map. The first one searches for the maxima of the gradient magnitude in the direction of the gradient, while the other looks for the zero crossings of the Laplacian (particularly LoG) image. The difference of the two schemes is already discussed.

After the extraction of the edges, the edges are represented as a chain code, i.e., neighboring edge pixels are ordered in some direction and assembled in a list. During this process, *T-corners* ([27]) are marked as corners and taken out of the list. The next step is the calculation of the curvatures. The calculation of the curvature is handled via averaging *k-curvatures* ([14]), as follows:

Algorithm 2.2

1. Let the point for which the curvature will be calculated be the $t^{th}$ element in the chain code, denoted by $t$.
2. Representing a vector by the start and end points, subtract the difference of the directions of the vectors

defined by [$t$,$t$+$k$] and [$t$-$k$,$t$]. This difference is defined to be k-curvature at the point $t$.

3. Average k-curvatures with possibly different weights (emphasizing small $k$'s) to obtain the curvature.

The point yielding a local maximum for curvature is taken as a corner, if its curvature is above some threshold.

## 2.5 Kitchen-Rosenfeld Cornerness Measure

Exploiting a similar idea as the previous detector, a *cornerness measure* may be proposed, as the change of gradient direction along an edge contour multiplied by the local gradient magnitude. This is also intuitively reasonable, since along the edges where no corner is locally in sight, one expects nearly constant gradient direction, whereas near the corners, there is a significant change in the mentioned direction. Multiplying the change of gradient direction further by the gradient magnitude emphasizes the strong edges, corners, etc. For this purpose, after a non-maximum suppression (along the gradient direction) applied on the gradient magnitude of the image, the measure is computed by calculating the derivative of the gradient direction along the direction orthogonal to the gradient[1]. Explicitly, this is equal to calculating the derivative of $\tan^{-1}\left(\frac{I_y}{I_x}\right)$, along $\left(-I_y, I_x\right)$ where $I(x,y)$ is the image and $I_x$ and $I_y$ denote the partial derivatives of $I(x,y)$. As is well known in calculus,

$$\frac{d\tan^{-1}(x)}{dx} = \frac{1}{1+x^2} \tag{2.12}$$

[1] The direction orthogonal to the gradient is taken as the direction of the tangent to the edge, however if this were the case, all the edges would lie on an equibrightness curve. Thus, this is merely an intuitive approximation to the edge direction. The exact direction on a continuous curve defined by the zero crossings of (2.5), is obtained by considering (2.5) as a function of $x$,$y$, namely $F(x,y)=0$ and calculating $\frac{dy}{dx}$ implicitly in terms of this function, which gives $\frac{dy}{dx} = -\frac{F_x(x,y)}{F_y(x,y)}$ whenever the denominator is non-zero(otherwise the edge direction is parallel to the y-axis). The edge is in the direction of $\left[1, \frac{dy}{dx}\right]$.

Thus,

$$\frac{\partial \tan^{-1}\left(\frac{I_y}{I_x}\right)}{\partial x} = \frac{1}{1+\left(\frac{I_y}{I_x}\right)^2}\cdot\frac{\partial}{\partial x}\left(\frac{I_y}{I_x}\right) = \frac{I_x^2}{I_x^2+I_y^2}\cdot\left(\frac{I_{yx}}{I_x}-\frac{I_y I_{xx}}{I_x^2}\right)$$
$$= \frac{I_x I_{yx} - I_y I_{xx}}{I_x^2 + I_y^2} \tag{2.13}$$

Similarly,

$$\frac{\partial \tan^{-1}\left(\frac{I_y}{I_x}\right)}{\partial y} = \frac{I_x I_{yy} - I_y I_{xy}}{I_x^2 + I_y^2} \tag{2.14}$$

Using these equations, the proposed cornerness measure, denoted by *K*, is calculated as,

$$K = \begin{pmatrix} \frac{\partial \tan^{-1}\left(\frac{I_y}{I_x}\right)}{\partial x} & \frac{\partial \tan^{-1}\left(\frac{I_y}{I_x}\right)}{\partial y} \end{pmatrix} \bullet \frac{\begin{pmatrix} -I_y & I_x \end{pmatrix}}{\sqrt{I_x^2+I_y^2}} \cdot \sqrt{I_x^2+I_y^2}$$

$$= \frac{I_{xx}I_x^2 - 2I_{xy}I_xI_y + I_{yy}I_y^2}{I_x^2+I_y^2} \tag{2.15}$$

where • denotes the inner product. Expressing (2.15) another way, *K* is equal to,

$$K = \frac{1}{\sqrt{I_x^2+I_y^2}}\begin{bmatrix} -I_y & I_x \end{bmatrix}\cdot\begin{bmatrix} I_{xx} & I_{xy} \\ I_{yx} & I_{yy} \end{bmatrix}\cdot\begin{bmatrix} -I_y \\ I_x \end{bmatrix}\frac{1}{\sqrt{I_x^2+I_y^2}} \tag{2.16}$$

Recognizing $\frac{1}{\sqrt{I_x^2+I_y^2}}\begin{bmatrix} -I_y & I_x \end{bmatrix}$ as the direction orthogonal to the gradient, (2.16) thus *K* may also be interpreted as the second derivative along the direction orthogonal to the gradient. It is noted at this point that this is exactly equal to the second term in (2.9).

Despite its compact form, this intuitive definition of the corner suffers from the same localization problem as the maxima gradient searching edge scheme, since the local maximum of the cornerness measure is sought after a non-maximum suppression (of the gradient magnitude in the direction of the gradient) procedure is applied.

## 2.6 Zuniga-Haralick Method (Facet Model Approach)

For each neighborhood of any desired size (practically 7x7 or 5x5), the image patch $f(x,y)$ can be modeled by a bi-cubic polynomial ([10]):

$$f(x,y) \approx k_1 + k_2 x + k_3 y + k_4 x^2 + k_5 xy + k_6 y^2 + k_7 x^3 + k_8 x^2 y + k_9 xy^2 + k_{10} y^3 \tag{2.17}$$

The coefficients $k_n$ may be calculated by finding the least squares approximation or utilizing pre-determined masks obtained using a procedure exploiting Gram-Schmidt orthogonalization method (see Appendix-1).

Cornerness is then calculated as the rate of change of the direction of the gradient along the edge contour. This measure is noted to be equal to the previous one (2.15) except for the absence of the multiplication with the gradient magnitude. This suggests the use of,

$$K = \frac{I_{xx} I_x^2 - 2 I_{xy} I_x I_y + I_{yy} I_y^2}{\left(I_x^2 + I_y^2\right)^{\frac{3}{2}}} \tag{2.18}$$

Calculating explicitly the partial derivatives of the bi-cubic polynomial at the origin,

$$\begin{aligned} I_{xx}(0,0) &= 2k_4; \\ I_{xy}(0,0) &= k_5; \\ I_{yy}(0,0) &= 2k_6; \\ I_x(0,0) &= k_2; \\ I_y(0,0) &= k_3 \end{aligned} \tag{2.19}$$

Inserting equations (2.19) in (2.18), the cornerness measure for a particular pixel, the neighborhood of which is approximated by the polynomial (2.17) is obtained as,

$$K = 2\frac{k_4 k_2^2 - k_5 k_2 k_3 + k_6 k_3^2}{\left(k_2^2 + k_3^2\right)^{\frac{3}{2}}} \tag{2.20}$$

An edge point, detected after the non-maxima gradient magnitude suppression, is claimed to be a corner, if this measure has a local maximum and is above some threshold.

A particular pixel is claimed to be an edge point if the second directional derivative taken in the direction of the gradient has a negatively sloped zero-crossing within the boundaries of the pixel. At a particular point, consider the curve obtained by intersecting the surface with a plane tangent to the direction of the gradient and orthogonal to the row-column plane. For an independent variable $h$ in the domain of this curve, one replaces $x$ with $h \cdot cos\alpha$ and $y$ with $h \cdot sin\alpha$, where α is the direction of the gradient, obtaining

$$\begin{aligned} f_\alpha(h) = k_1 &+ \left(k_2 \cos\alpha + k_3 \sin\alpha\right)h + \\ &\left(k_4 \cos^2\alpha + k_5 \sin\alpha\cos\alpha + k_6 \sin^2\alpha\right)h^2 + \\ &\left(k_7 \cos^3\alpha + k_8 \cos^2\alpha \sin\alpha + k_9 \cos\alpha \sin^2\alpha + k_{10} \sin^3\alpha\right)h^3 \end{aligned} \tag{2.21}$$

One can also replace $sin\alpha$ with $\frac{k_3}{\sqrt{k_2^2 + k_3^2}}$ and $cos\alpha$ with $\frac{k_2}{\sqrt{k_2^2 + k_3^2}}$. The desired higher order derivatives may now be calculated easily by differentiating the polynomial function $f_\alpha(h)$ with respect to $h$, yielding new polynomial functions that are completely characterised by the coefficients $k_i$ .

It is noted that since the corners detected by this measure also lie on maxima of the gradient, the detected corner is not at the exact corner location, similar to the previous measure. Moreover, the approximation by the bi-cubic polynomial may also introduce some localization error.

## 2.7 Harris-Stephens Corner Detector

As an important detection scheme, Harris-Stephens Corner Detector makes the use of the following matrix

$$C(x,y)=\begin{bmatrix} \sum_R w_R I_x^2 & \sum_R w_R I_x I_y \\ \sum_R w_R I_x I_y & \sum_R w_R I_y^2 \end{bmatrix} \tag{2.22}$$

where R is a region centered around the point $(x,y)$, $w_R$ represents a Gaussian window in this region and $I_x$ and $I_y$ are the partial derivatives of the image. The points for which this matrix has full rank, or practically, has two significant eigenvalues, are considered as interest points. The matrix in (2.22) may also be recognized to be the one used for determining points for which optical flow may safely be computed.

A clear interpretation of this scheme is given in [32], as follows:

Correlating a patch with its neighboring patches, i.e. patches shifted in small amounts $(\Delta x, \Delta y)$, reveals a measure about the local changes in a function. For a given point $(x, y)$ and a direction $(\Delta x, \Delta y)$, the measure is given by:

$$m(x,y)_{(\Delta x,\Delta y)} = \left( \int_R \left( I(p,q) - I(p+\Delta x, q+\Delta y) \right)^2 dR \right) \tag{2.23}$$

where $R$ represents the region or patch centered around $(x, y)$. In the case of an image, where a discrete grid is used, the integral may be replaced with a summation.

If the function, at some point $(x, y)$, assumes high values for any direction $(\Delta x, \Delta y)$, the point is considered to be significantly distinct.

Considering further,

$$I(x+\Delta x, y+\Delta y) \approx I(x,y) + \Delta x I_x(x,y) + \Delta y I_y(x,y) \tag{2.24}$$

the expression to be summed may be reduced to

$$m(x,y)_{(\Delta x,\Delta y)} \approx \sum_R \left( \begin{bmatrix} I_x & I_y \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix} \right)^2$$
$$= \begin{bmatrix} \Delta x & \Delta y \end{bmatrix} \underbrace{\begin{bmatrix} \sum_R I_x I_x & \sum_R I_x I_y \\ \sum_R I_x I_y & \sum_R I_y I_y \end{bmatrix}}_{C'} \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix} \tag{2.25}$$

Thus, if $C'$ has two significant eigenvalues, the measure should yield high values signaling an interesting point.

In order to further concentrate the measure around the center of the patch, the summations taking part in (2.25) are done using a Gaussian window, centered around the point $(x, y)$, i.e., $C'$ is replaced with $C$ in (2.22).

Instead of calculating the eigenvalues, the product of the eigenvalues is compared to their sum squared via computing $Det(C) - kTrace^2(C)$, where k is usually taken as 0.04 or 0.06. The points, at which this measure is above some threshold and assumes a local maximum, are taken to be interest points.

## 2.8 SUSAN & Fast Corner Detection

In this section, two radically different interest point detectors will be presented, the latter of which is also tested via simulations. Despite their resemblance in the first glance, the only similar point between these detectors is the figures in which observations are achieved, forming grounds that the schemes are based on.

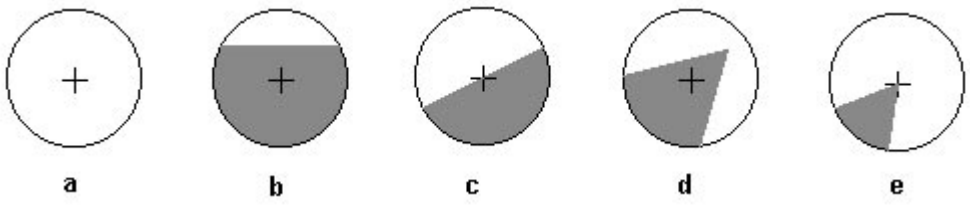



Figure 2.7 Circular masks for considering SUSAN. (a) Uniform region, (b) Region with nucleus near an edge, (c) Region with nucleus on the edge, (d) Region with nucleus near a corner, (e) Region with nucleus on a corner

Considering Figure 2.7, for each circular mask with the center, depicted as the *nucleus*, comparing the brightness of each pixel in the mask with that of the nucleus, an area is defined by the pixels having brightness similar to that of the nucleus. This area is called "Univalue Segment Assimilating Nucleus" (USAN) . Based on this definition, it is observed that for a uniform region (Figure 2.7.a), USAN is equal to the area inside the circle. When the nucleus is near to an edge (Figure 2.7.b), USAN decreases and when the nucleus is actually an edge pixel (Figure 2.7.c), USAN attains a certain (ideally) value (not equal to the half of the area of the mask due to the dicrete grid). However, near a corner (Figure 2.7.d), USAN significantly decreases and at the corner point (Figure 2.7.e) USAN attains a local minimum. Thus, a corner detection scheme might be proposed in the light of these observations [33]. The implementation should actually contain (mainly) two thresholds. The first threshold is utilized for deciding whether a pixel in the circular mask belongs to USAN or not, i.e. the so-called *brightness difference threshold*. The other threshold works as a geometrical threshold deciding whether a local minimum is a corner point. This geometrical threshold allows only sharper corners to be detected, if it were reduced. It is noted that the geometrical threshold may safely be chosen as a fixed value, whereas the brightness difference threshold seems to need (theoretically at least) a tuning regarding the noise amplitude and the contrast level. The authors however claim that this does not turn out to be a problem in practice and propose a fixed value.

Another method is due to Trajkovic and Hedley [36]. For this method, one should consider an arbitrary line $k$ containing the nucleus and intersecting the boundary of the circular window at two opposite points $P(l)$ and $P'(l)$, and the following corner response function (CRF):

$$R_C = \min_k \left[ \left( I_P - I_N \right)^2 + \left( I_{P'} - I_N \right)^2 \right] \tag{2.26}$$

where $I_P$, $I_{P'}$ and $I_N$ are the image intensities at $P(l)$, $P'(l)$ and the nucleus, respectively and $\min_k$ denotes the minimum value of the expression evaluated for different choices of the line $k$ (see Figure 2.8).

As observed in Figure 2.8, the CRF proposed in (2.26) isolates corners and regions near the corners.

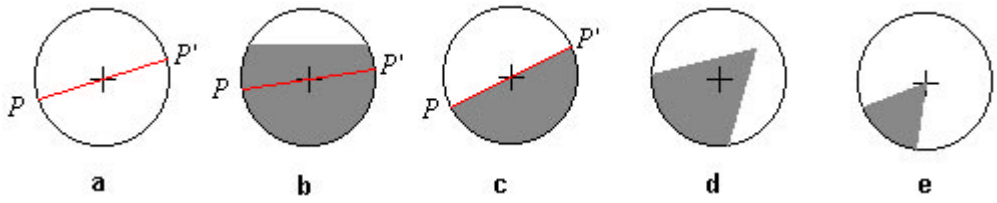



Figure 2.8 Arcs minimizing (2.26) imposed on (a) Uniform region, (b) Region with nucleus near an edge, (c) Region with nucleus on the edge, (d) Region with nucleus near a corner, (e) Region with nucleus on a corner

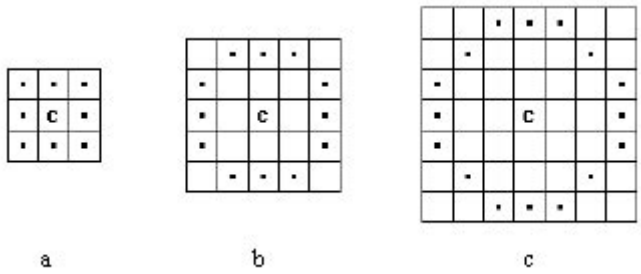



Figure 2.9 Digital circular masks of radius (a) 1, (b) 2, (c) 3

In practice, a discrete approximation of the circular window is used (Figure 2.9). If the size of the utilized window is large, the localization of the corner is poor, since the CRF does not help discriminate against the situations in Figure 2.8.d and Figure 2.8.e. On the other hand, when the window is taken to be of a small diameter, there are too few directions to evaluate (2.26) and false corners may be detected in result. This problem is overcome by the help of linear interpixel approximation [36].

Considering a window of radius 1, with center C and A,A',B,B' the pixel locations(Figure 2.10), the horizontal($r_A$) and vertical($r_B$) intensity variations can be computed as,

$$r_A = (I_A - I_C)^2 + (I_{A'} - I_C)^2 \qquad (2.27)$$

$$r_B = (I_B - I_C)^2 + (I_{B'} - I_C)^2 \tag{2.28}$$

Thus the CRF at the center $R_C$ is,

$$R_C = \min(r_A, r_B) \tag{2.29}$$

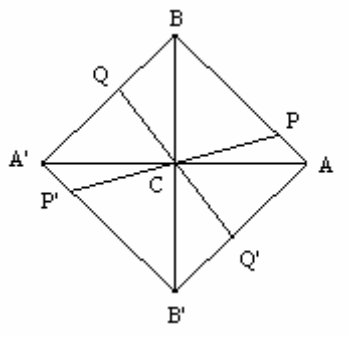



Figure 2.10 Interpixel location definitions

If $R_C$ is less than a given threshold, then no corner exists in the window, and the point is discarded. However, if this is not the case, diagonal edges(i.e. P,P',Q,Q') must be checked using linear interpizel approximation.

The intensity at interpixel locations are calculated as:

$$\begin{aligned} I_P &= (1-x)I_A + xI_B \\ I_{P'} &= (1-x)I_{A'} + xI_{B'} \\ I_Q &= (1-x)I_{A'} + xI_B \\ I_{Q'} &= (1-x)I_A + xI_{B'} \end{aligned} \tag{2.30}$$

Now, a response function for the diagonals containing P and P' may be written:

$$\begin{aligned} r_1(x) &= (I_P - I_C)^2 + (I_{P'} - I_C)^2 = A_1x^2 + 2B_1x + C \\ r_2(x) &= (I_Q - I_C)^2 + (I_{Q'} - I_C)^2 = A_2x^2 + 2B_2x + C \end{aligned} \tag{2.31}$$

where

$$
\begin{aligned}
C &= r_A \\
B_1 &= (I_B - I_A)(I_A - I_C) + (I_{B'} - I_{A'})(I_{A'} - I_{C'}) \\
B_2 &= (I_B - I_{A'})(I_{A'} - I_C) + (I_{B'} - I_A)(I_A - I_C) \\
A_1 &= r_B - r_A - 2B_1 \\
A_2 &= r_B - r_A - 2B_2
\end{aligned}
\tag{2.32}
$$

After some manipulations, it can be shown that the necessary and sufficient condition for a minimum of the function $R = \min_{x \in (0,1)} (r_1(x), r_2(x))$ to exist is in the interval (0,1):

$$B<0 \text{ and } A+B>0 \tag{2.33}$$

where

$$B = \min(B_1, B_2) \text{ and } A = r_B - r_A - 2B \tag{2.34}$$

Moreover, the value of the minimum is given by :

$$R_{\min} = C - \frac{B^2}{A} \tag{2.33}$$

## 2.9 Discussion

Although it is rather tempting to introduce a model for the feature being sought and try to propose operators, detectors that can succeed for the given model, it is noted that ([33]) an image may contain many structures (e.g. 'T','X','Y' junctions, crunks ([2]), etc.) that one would not like to miss and which do not fit the model. Obviously, different models and detectors can be introduced for each such 'anomaly' and these might be used in parallel. Even for this case, as noted in the previous sections, the performance of the detectors might still be questioned in some aspects (localization in this case). However, many applications do not require the exact position of a corner, or a 'T' junction, etc. A more severe requirement seems to be related to the repeatability (i.e. the ability to detect the same feature in different views) ([32]) and geometric invariance (i.e. the ability to recover the same position for a given feature in different views) ([26]). Thus, if the desired application allows, it is more natural to deviate from a strict model of a corner, junction, blob, etc. and look for features (whatever they

might be) that our robust, invariant operator (whatever it is) is able to detect.

In this regard, a view invariant detector should be sought. It is known that a perspective transformation may locally be approximated by an affine transformation ([12]). Thus, an affine invariant detector (including scale) is sufficient in this sense.

It was shown that, under a transformation of the image coordinates (about the featuıre point) as $\begin{bmatrix} x' \\ y' \end{bmatrix} = B \begin{bmatrix} x \\ y \end{bmatrix}$, a modified version (utilising, instead of simple derivatives, derivatives of Gaussians with varying bandwidths) of the Harris matrix ($C'$ in (2.25)) transforms as $B^T C' B$. Using Cholesky decomposition then, it is possible to back-transform this deformed matrix and obtain an identity matrix. This 'straightens' the local patch up to a rotation, and since the eigenvalues are not affected by a rotation [4], the modified Harris matrix is rendered affine-invariant. This method was also shown to have good repeatability, which makes it a good candidate for feature detection. It can also be shown that the Hessian (i.e. the 2x2 matrix containing the second derivatives) may also be modified in the same manner to gain affine-invariance. These issues will be discussed in the next chapter.

# CHAPTER III

# INVARIANT INTEREST POINT EXTRACTION & DESCRIPTION

All of the interest point detectors, which are presented in the previous chapter, rely on an assumption of, some fixed scale attached to the features present in the images, where they are applied. For instance, in order to estimate the curvature at a point, k-curvatures are averaged, but it is not known a-priori what the largest value of 'k' value to be used in the summation is. If it is taken too low, somewhat smooth corners might beare missed. On the other hand, if 'k' is too high, some sharp, but rather local corners, might be lost during averaging. Similar problems may arise regarding the derivative approximation for the Kitchen-Rosenfeld measure, or choice of the size of the region for estimation by using bi-cubic polynomials for the Facet-Model approach, or selection of the width of the Gaussian for weighted summation in the Harris detector and lastly, the radius of the digital circles utilized by the SUSAN and 'fast corner detection' schemes. Given an image, these selections may be regarded as parameters to be adjusted manually. However, even for manual selection, one may not be able to obtain satisfactory results. This is due to the fact that in an arbitrary image, there may exist both sharp and diffuse features ([17]), i.e., the mentioned characteristics are not properties of the image but belong to the features present in the image. Moreover, it is noted that a diffuse feature may be transformed into a sharp one by adjusting the scale of observation (Figure 3.1). From this point of view, the problem may be considered as one of scale. This necessitates a multi-scale description of the image. The idea of scale-space, which properly suits (which has been,

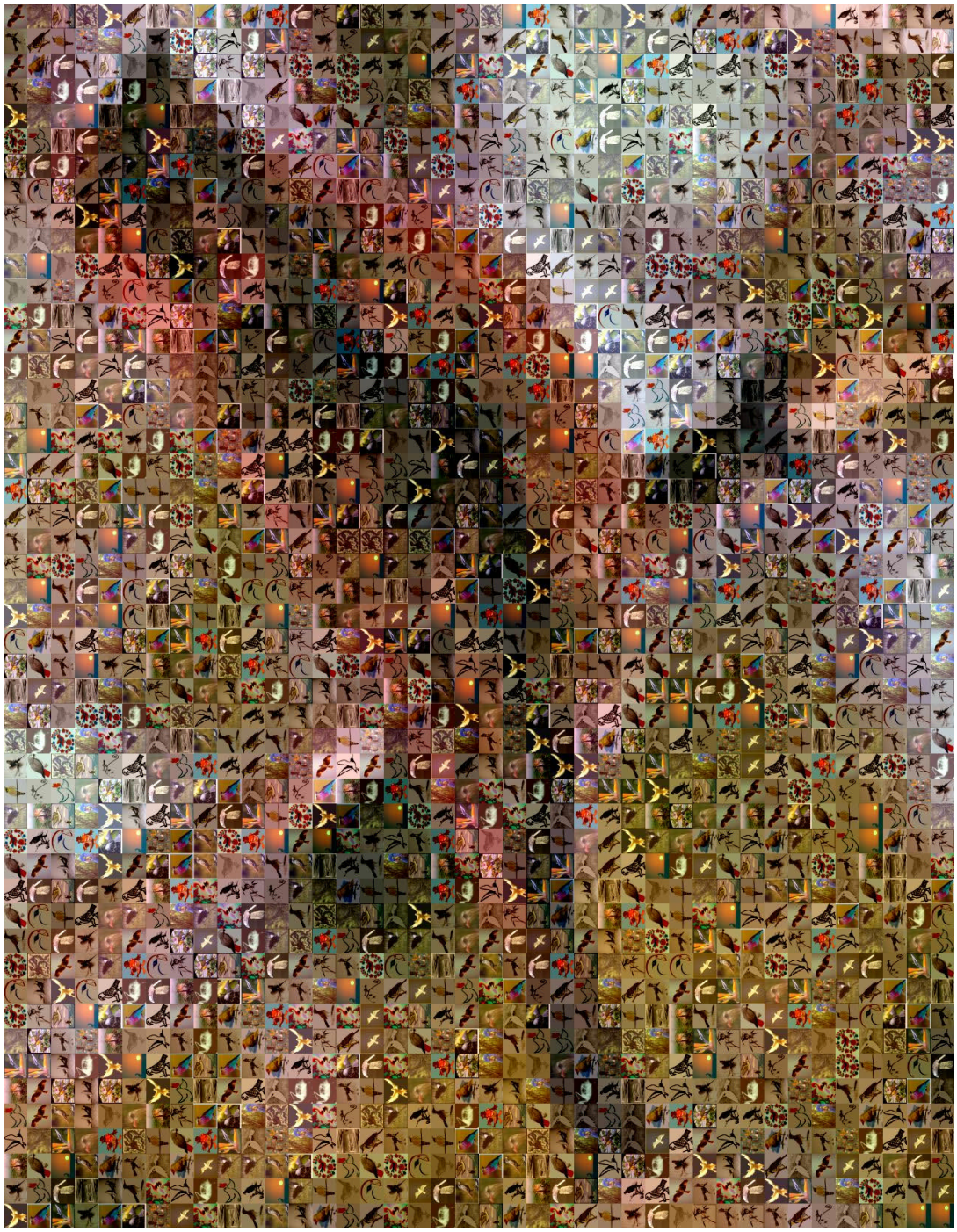


Figure 3.1 ‘Bird assembled from smaller birds’. The coarse features present in the image are observed more clearly at a distance (a few meters), whereas the details are more ‘dominant’ in the reading distance. A similar technique exploiting the same idea is called ‘dithering’, mentioned in [16]. It can be speculated from these observations that our vision system somehow disregards too-fine-scale information, paying greater attention to reliable data, at the expense of losing information.

indeed, proposed for this purpose ([40])) such a need, will be presented shortly.

Following the detection of the features, which usually correspond to a first step in a given task, one may need to locally characterise the region containing the feature. The most straight-forward approach would be to take a window of pre-determined size. However, in many applications, it is required that this characterisation be invariant to deformations (such as, rotation, stretching, scaling, etc.), arising from possible changes in the viewing positon. Thus, it is necessary to be able to find a characterisation invariant to affine transformations. For this purpose, different rotation invariants ([4], [29], [30], [31]) will be introduced in this chapter and by using these invariants, a method to obtain affine invariant description ([4], [18]) of any interesting region will be explained. This analysis will also lead to an affine invariant detector ([26]), with an ability to locate the same positon in a region regardless of the affine transformation that the region might be exposed to.

## 3.1 Scale-Space

The idea of multi-scale representation of a signal can be useful in various situations. Pyramid representation of the images were initially introduced ([6]) mainly for the purpose of data compression. This representation exploits the high-correlation of pixel values in a neighborhood. Any given image is low-pass filtered and subtracted from its original, resulting in a “decorrelated” image. This decorrelated image can be coded using fewer bits, since there is less redundant information in each pixel value. Then, the low-pass filtered image is sampled and processed in the same manner,resulting in another low-pass and “decorrelated” image pair, of reduced size. The process is repeated and thus, a pyramid representation of the original image is produced (Figure 3.2). Adding the “decorrelated” images and upsampling the sums, the original image is recovered.

(a) (b)

Figure 3.2 (a) 'House' image, (b) its pyramid representation.

The use of successive smoothing and subtraction may also be considered as bandpass filtering. Thus, filter banks, which may be utilised for the analysis of formants which enable the recognition of vowels in the context of speech recognition, etc., may be accepted as related approaches. Scale-space filtering, as introduced by Witkin ([40]), is intended to form a framework to separate events at different scales arising from distinct physical processes. In the following section, a brief, informal presentation will be made so as to give an intuitive feeling about the subject.

### 3.1.1 Continuous Scale-Space Representation

Given a signal $f : R \to R$, the scale-space representation $L : R \times R_+ \to R$ is defined ([16]) such that the representation at zero scale is equal to the original signal

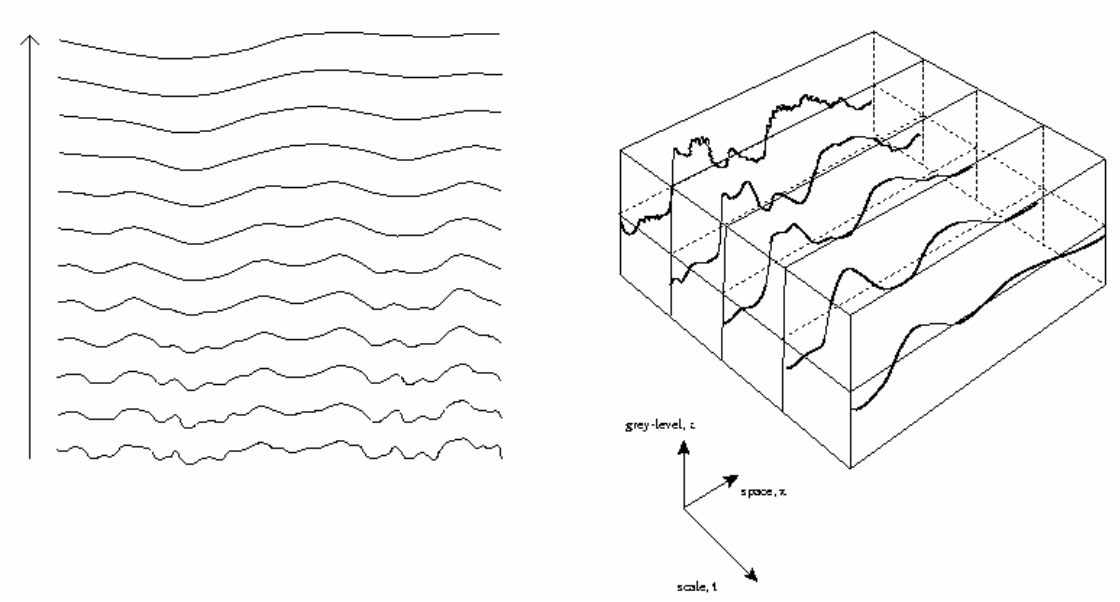
$$L(x;0) = f(x) \tag{3.1}$$

and the representations at coarser scales are obtained by convolution of the signal with Gaussian kernels of decreasing bandwidth (or increasing standard deviation) (Figure 3.3),

$$L(x;t) = \int g(a;t) f(x-a) da = g(a;t) * f \tag{3.2}$$

where

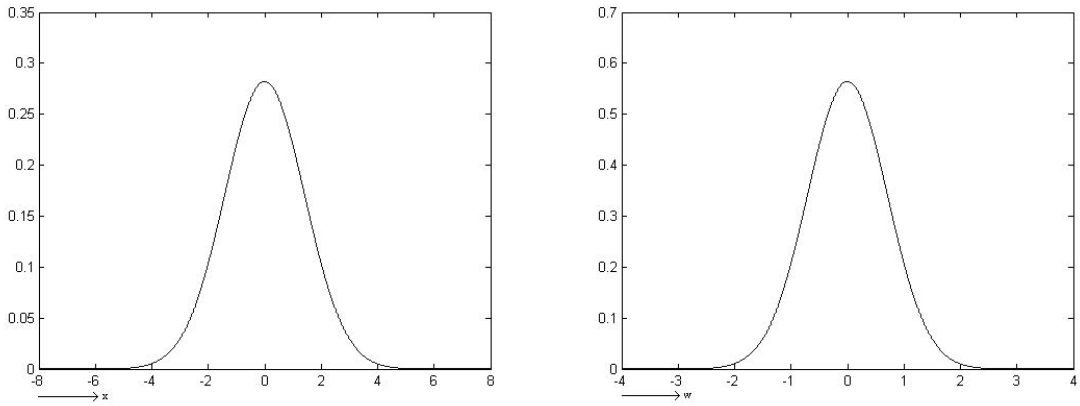
$$g(x;t) = \frac{1}{\sqrt{2\pi t}} e^{-\frac{x^2}{2t}} \tag{3.3}$$




(a) (b)

Figure 3.3 Scale-space representation of a signal. (a)The signal successively smoothed by Gaussians of decreasing bandwidth, (b)3-D illustration of the scale-space representation of the signal, explicitly shown as a function of both scale and position ([16]).

The Gaussian function posseses some interesting properties, such as, unimodality in both spatial and the frequency domain, symmetricity (Figure 3.4). As $t$ approaches zero, the convolution with the Gaussian approaches the unsmoothed signal and as $t$ increases, the convolution




(a) (b)

Figure 3.4 The Gaussian with $t$=2, (a) in the spatial domain, (b) in the frequency domain. The cut-off frequency is inversely proportional to $\sqrt{t}$ .

approaches to the mean value of the signal. The Gaussian is also normalized and infinitely differentiable. Despite these attractive properties, what is more important is related to the number of local extrema. As the scale parameter $t$ is increased, the number of local extrema in the representation $L(x;t)$ decreases monotonically. This property may also be formulated as all the first order minima of the convolution to increase and all first order maxima to decrease with increasing *t*. Smoothing of the signal in this manner ensures that artificial features or structres are not created and the salient features, which exist over a range of scales, may be safely detected. Since the representation is dependent on a continuous scale parameter and the same spatial sampling is preserved over all scales, the mentioned features may easily be tracked and the position of the feature at any desired scale may readily be obtained as a result (Figure 3.5).

Another important question to be posed is about the uniqueness of such a representation. Does there exist another family of functions, whose convolution with images lead to a scale-space representation possessing similar properties? A lemma by Babaud et.al. ([3]) states the uniqueness of the family of Gaussians and the representation thus obtained, if the family is restricted to possess some advantageous properties. The constraints, for the desired family, denoted by $g(x,t)$, are as follows :

1) $\frac{1}{\sqrt{t}}$ is a bandwidth parameter, i.e. $g(x,t)=\frac{1}{\sqrt{t}}h\left(\frac{x}{\sqrt{t}}\right)$.

2) g is symmetrical in $x$ : for all $x \in R$

$$g(x,y)=g(-x,y).$$

3) For all $t>0$, $g$ is normalized:

$$\int_{-\infty}^{\infty} g(u,t)du=1.$$

4) There exists an integer $p$ such that

$$h^{(2p)}(0)\neq 0.$$

where the last constraint is noted to be necessary due to the technical reasons in the proof for uniqueness.

In fact, the uniqueness of this representation may be obtained [16] with different constraints.

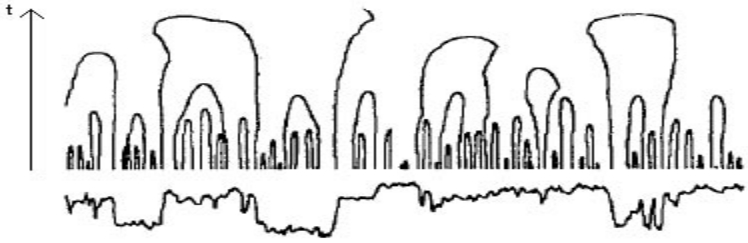



Figure 3.5 Zero crossings in scale. The signal is given below. On the above, the trace of the zero crossings of the second derivative in scale-space is shown. It is noted that, since new zero crossings are not created as the scale parameter *t* increases, the curves are always expected to be closed from above ([16]).

Another formulation of the scale-space utilizes partial differential equations (the heat equation). The scale-space family is described by the diffusion equation ([16]) :

$$\partial_t L = \frac{1}{2}\nabla^2 L = \frac{1}{2}\partial_{xx} L \tag{3.4}$$

The equation above may be interpreted as follows:

*Since at the local maxima, the second derivative* $\partial_{xx} L$ *assumes negative values,* $\partial_t L$ *will also be negative and as a consequence it is ensured that proceeding in the direction of increasing t (at least for a small amount), without changing the spatial position would decrease the value that the representation assumes at the given position and scale. Similarly, going in the direction of increasing t at a local minimum would have the effect of increasing the value of the representation.*

Thus, (3.4) is seen to monotonically smooth the given signal with a non-enhancement property (of the local extrema) that is required from the representation. It is noted that the Gaussian family, and thus, convolution of a signal with the Gaussian satisfies (3.4) .

The representation may be extended to higher dimensions ([16]) via the diffusion equation which, for an n-dimensional case takes the form,

$$\partial_t L = \frac{1}{2}\nabla^2 L = \frac{1}{2}\sum_{i=1}^{n} \partial_{x_i x_i} L \tag{3.5}$$

where the solution is obtained by convolving the n-dimensional signal $f(X)$ with a family of n-dimensional Gaussians,

$$L(X;t) = f(X) * g(X;t) \tag{3.6}$$

where

$$g(X;t) = \frac{1}{(2\pi t)^{\frac{n}{2}}} e^{-\frac{X^T X}{2t}} \text{ with } X = [x_1, x_2, \ldots, x_n]^T \tag{3.7}$$

with the initial condition $L(X;0) = f(X)$.

In particular, for the case n=2, it is expected that no new level curves are created throughout the smoothing process.

Another useful property of this representation is related to the convolution of a Gaussian kernel with another Gaussian kernel,

$$g(X;t) * g(X;s) = g(X;t+s) \tag{3.8}$$

which is named the semi-group property. Using this property,

$$\begin{aligned} L(X;t_2) &= g(X;t_2) * f(X) = g(X;t_2 - t_1) * g(X;t_1) * f(X) \\ &= g(X;t_2 - t_1) * L(X;t_1) \end{aligned} \tag{3.9}$$

In words, the representation at a coarse scale $t_2$ may be computed by convolving the representation at a finer scale with a Gaussian of standard deviation equal to the difference of the two scales.

### 3.1.2 Discrete Scale-Space Representation

All of the results and discussions cited above are concerned with continuous signals. However, the smoothing is usually performed using a computer which necessarily follows the spatial sampling of the signal. Thus, it is also important and interesting at the same time, to be able to form a discrete scale-space representation for practical situations. Lindeberg [16] presents an axiomatic approach to the problem, starting with the definition of a discrete scale space kernel.

*Definition 3.1 : A one-dimensional discrete kernel* $K : Z \to R$ *is said to be a scale space kernel if for all signals* $f_{in} : Z \to R$ *the number of local extrema in the convolved signal* $f_{out} = h * f_{in}$ *does not exceed the number of local extrema in the original signal.*

Considering the convolution of a scale space kernel with another scale space kernel, and using the associativity property of the convolution, it can be seen that,

$$(K_1 * K_2) * g = K_1 * (K_2 * g) \tag{3.10}$$

Since $K_2 * g$ contains fewer local extrema compared to *g*, and further convolution of the resulting signal with another scale space kernel, namely $K_1$, will further reduce the number of local extrema, it is concluded that $(K_1 * K_2)$ is also a scale space kernel.

Another interesting property of such kernels is their unimodality in both the spatial and frequency domains. For the proof of this fact in the spatial domain, it is sufficient to consider the discrete impulse as the input function. If the kernel is not unimodal, the result of the convolution will be the kernel itself which would contain more local extrema than the input (the impulse function), hence the necessity in time-domain follows. For the frequency domain, if the frequency domain representation is not unimodal, then there will exist frequencies $f_1$ and $f_2$, with $f_1 < f_2$, and $|H(f_1)| < |H(f_2)|$ where $H(f)$ represents the Fourier transform of the kernel. Considering a signal consisting of two sinusoids of frequencies $f_1$ and $f_2$ (possibly of finite duration), successive application of the kernel to the signal will result in the attenuation of the low frequency sinusoid $f_1$, resulting in an increase in the number of local extrema, therefore it can be concluded that unimodality in the frequency domain is also necessary.

The family of kernels are finite. These are of the form $\delta[n-a]$, where $a \in Z$, $a\delta[n] + b\delta[n-1]$ where $a,b>0$, $a^n u[n]$ where $u[n]$ denotes the step function and $|a|<1$, $a^{-n}u[-n]$ where $|a|>1$, and $e^{-\alpha t} I_n(\alpha t)$ for some $\alpha > 0$ where $I_n$ are the modified Bessel functions of integer order.

Among these, $e^{-\alpha t} I_n(\alpha t)$ is the only kernel, possesing the semi-group property, which makes it unique for the purpose of forming a discrete scale space representation. Lindeberg [16] calls it the discrete analogue of the Gaussian (or simply discrete Gaussian) and further shows that convolution of a discrete signal $f(n)$ with this kernel family satisfies a discretized version of the diffusion equation (3.5),

$$\partial_t L(n;t) = \frac{1}{2}\left(L(n+1;t) - 2L(n;t) + L(n-1;t)\right) \tag{3.11}$$

with initial condition $L(n;0) = f(n)$, and $t \in R_+$, $n \in Z$.

The discrete scale space representation for k-dimensional signals is constructed by extending (3.11) to higher dimensions. In the end, it turns out that the unique family of kernels to be used is

$$T_k(N;t) = \prod_{i=1}^{k} T(n_i;t) \tag{3.12}$$

where $T$ is the discrete analogue of the Gaussian kernel.

Another approach for the construction of a discrete scale-space might be through the sampling of the Gaussian family. It is in fact ensured by the following lemma ([16]) that these sampled functions are also discrete scale space kernels.

*Lemma 3.2 : Uniform sampling of a continuous scale-space kernel gives a discrete scale-space kernel.*

On the other hand, it is shown that using the sampled Gaussian may violate the semi-group property. That is, if $g$ is a sampled Gaussian of standard deviation $t_1$, the following relation holds only under special circumstances:

$$L(X;t_1 + t_2) = g(X;t_1) * L(X;t_2) \tag{3.13}$$

Despite this annoying fact, the sampled Gaussians are widely used (actually, they have also been used for the implementations that will be explained), due to the unavailability of the discrete analogue of the Gaussian in standard libraries.

### 3.1.3 Rescalings in Scale-Space:

Finally the behavior of the representation under rescalings of the original signal is considered. For an input signal $f : R \to R$, define $f' : R \to R$ by

$$f(x) = f'(sx) \tag{3.14}$$

Then,

$$L_f(x;t) = \int_{-\infty}^{\infty} f(a)g(a-x;t)da, \tag{3.15}$$

where $g(x;t)$ denotes the Gaussian.

Then, letting $a' = sa$, $t' = s^2 t$, it is seen that,

$$f(a) = f'(a') \tag{3.16}$$

$$g(a-x;t) = sg'(a'-x';t') \tag{3.17}$$

$$\frac{da'}{da} = s \tag{3.18}$$

where (3.16) follows from (3.14) and (3.17) from the properties of the Gaussian (consider the mentioned constraints).

Thus, (3.15) is equal to

$$\int_{-\infty}^{\infty} f'(a')g'(a'-x';t')da' = L_{f'}(x';t') = L_f(x;t) \tag{3.19}$$

This suggests that at a point, the problem of recovering scale may be reduced to finding the correct scale parameter *t*, for the Gaussian to be used ([23]). What is actually being done is to "zoom" in or out of the function using Gaussians with different variances, instead of dealing with the unknown function values. The result can be generalized to an affine transform as well ([22]), and this matter will be examined in detail in the next sections.

### 3.1.4 Scale-Space Derivatives

The calculation of the derivatives of an image (or any given signal) in a computer is usually considered an ill-posed problem ([31], [16]) (i.e. the

solution does not continuously depend on the input). For an illustration, consider the functions ([31]) $f_1(x)$ and $f_2(x) = f_1(x) + \varepsilon \sin(\omega x)$, for a small $\varepsilon$, the two functions are approximately the same. However, if $\omega$ is sufficiently high, the amplitudes of the first derivatives at the origin will be significantly different. This is a frequent situation in images due to the inescapable presence of noise from the sensors. The remedy for this case is to attenuate this high-frequency noise via some low-pass filter.

Choosing the response of the filter as a Gaussian, the properties of scale-space may be exploited. With such a choice of the smoothing function, the differentiation operation (at an arbitrary order *n*) amounts to calculating $\frac{\partial^n}{\partial x^n} L(x;t)$,

$$\begin{aligned} \frac{\partial^n}{\partial x^n} L_f(x;t) &= \frac{\partial^n}{\partial x^n} \int g(a;t) f(x-a) da = \int g(a;t) \left( \frac{\partial^n}{\partial x^n} f(x-a) \right) da \\ &= L_{f^n}(x;t) \end{aligned} \tag{3.20}$$

if $f(x)$ is a differentiable function. Moreover,

$$\begin{aligned} \frac{\partial^n}{\partial x^n} L_f(x;t) &= \frac{\partial^n}{\partial x^n} \int g(x-a;t) f(a) da = \int \left( \frac{\partial^n}{\partial x^n} g(x-a;t) \right) f(a) da \\ &= \left( \frac{\partial^n g}{\partial x^n} \right) * f \end{aligned} \tag{3.21}$$

Equation (3.20) implies that, for a differentiable function, smoothing after differentiating is equivalent to differentiating after smoothing. Apart from this fact, it is seen that $\frac{\partial^n}{\partial x^n} L(x;t)$ corresponds to $L_{f^n}(x;t)$ which actually is the scale-space representation of $f^n$. Thus, one expects to find fewer local extrema in the derivatives of the smoothed function, no matter when it was differentiated (either before or after smoothing). In addition, it is seen that the spatial derivatives of a scale-space representation should obey the scale-space axioms, as well.

On the other hand, (3.21) suggests a method to calculate the derivatives, somewhat regularizing the ill-conditioned problem. Instead of convolving the differentiated function with a Gaussian, the function is

convolved with a differentiated Gaussian (which is infinitely differentiable), enlarging the class of differentiable functions ([16]).

The reason of interest in these derivatives is actually related to the local description of the image. Considering the Taylor expansion around the origin, an infinitely differentiable function may be written as,

$$f(x) = f(0) + xf'(0) + \frac{x^2}{2} f''(0) + \frac{x^3}{3!} f'''(0) + \ldots + \frac{x^n}{n!} f^n(0) + \ldots \qquad (3.22)$$

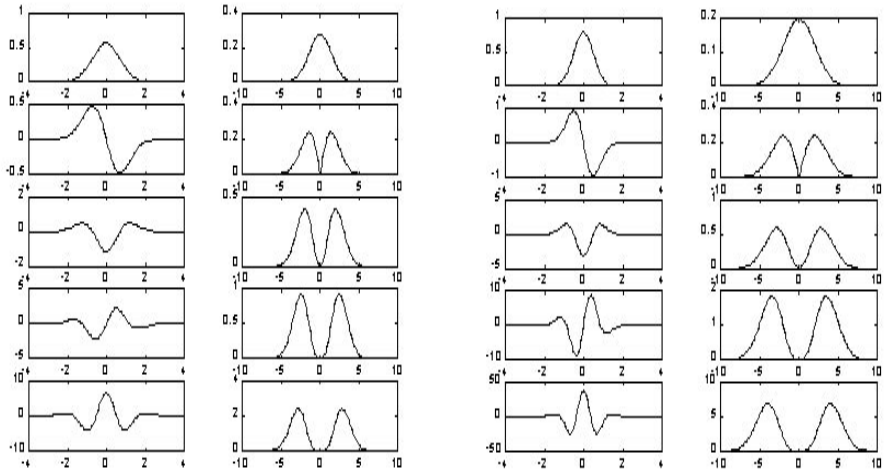
Thus, it is clear that derivatives can readily be used for local description at some scale *t*. An interesting property is about the Laplacian of the representation. If Laplacian derivatives at all scales are available, the representation, thus all the other derivatives may be obtained from these values. For a proof, consider the representation at some scale. If the representation tends to zero at infinite scale, it follows from the diffusion equation (3.4) that,

$$L(x;t) = -\left(L(x;\infty) - L(x;t)\right) = -\int_{-\infty}^{\infty} \frac{\partial}{\partial t'} L(x;t')dt' = -\frac{1}{2}\int_{t'=t}^{\infty} \nabla^2 L(x;t')dt' \qquad (3.23)$$

In fact, using scale-space derivatives with several different scales, one can obtain high-quality reconstructions of a patch ([15]). In this work ([15]), Singular Value Decomposition (SVD) is utilised to test the independence of the filter set (Gaussian derivatives in this case). Expressing the responses of the filters as a column vector and assembling the column vectors in some matrix, using SVD, one can obtain an orthonormal set spanning the same space as the original column vectors.

The use of Gaussian derivatives at different scales may be appreciated, if the Fourier Transforms (FT) are investigated. It can be shown that FT of a Gaussian with variance *s* is a Gaussian with variance *1/s* (see Appendix-2). Moreover, noting that differentiating a function in the spatial domain corresponds to multiplication of its FT with *-jω* in the frequency domain, the FT's of the derivatives of the Gaussian may also be computed easily. Considering a Gaussian derivative of arbitrary order as a filter kernel and noting that successive differentiation in the spatial domain will lead to successive multiplication with *-jω* in the frequency domain, one can expect that higher the derivative, higher will be the pass-

frequencies (see Figure 3.6). Thus, for some fixed variance, or scale, the Gaussian and its derivatives behave like a filter bank, where the Gaussian



(a) (b) (c) (d)

Figure 3.6 Gaussian, its derivatives and their Fourier Transforms. Gaussian and derivatives with increasing order, (a) *t=1/2*,(c) *t=1/4*, FT magnitudes for the functions (b) in (a), (d) in (c)

is the low-pass and the derivatives constitute the band-pass filters (actually this is the case for an arbitrary low-pass differentiable kernel). Moreover, adjusting the scale, one obtains a filter bank, which covers the high-frequencies with low-scale Gaussians and low-frequencies with high-frequency Gaussians. In this respect, the behaviour may be resembled to those of wavelets (if the low-pass Gaussians are left out) ([16]).

Lastly, the behaviour of these derivatives under rescalings of the original signal is considered. For this, let $f_1(x)$ be the input signal and let $f_2(x') = f_1(x)$ where $x' = sx$. Applying the differentiation operator (of arbitrary order *n*) with respect to *x* to both functions, one obtains

$$\frac{\partial^n}{\partial x^n} f_1(x) = \frac{\partial^n}{\partial x^n} f_2(x') = \frac{\partial^{n-1}}{\partial x^{n-1}}\left[\frac{\partial x'}{\partial x}\left(\frac{\partial}{\partial x'} f_2(x')\right)\right] = \ldots = s^n \frac{\partial^n}{\partial x'^n} f_2(x') \quad (3.24)$$

Using the equivalence of (3.15) and (3.19),

$$L_{f_1^n}(x;t) = s^n L_{f_2^n}(x',t'). \quad (3.25)$$

where $t' = s^2 t$ as defined previously.

Multiplying both sides of (3.25) by $t^{\frac{n}{2}}$,

$$t^{\frac{n}{2}} L_{f_1^n}(x;t) = s^n t^{\frac{n}{2}} L_{f_2^n}(x',t') = t'^{\frac{n}{2}} L_{f_2^n}(x',t') \tag{3.26}$$

Using further, the equivalence in (3.20), equation (3.26) may be written,

$$t^{\frac{n}{2}} \frac{\partial^n}{\partial x^n} L_{f_1}(x;t) = t'^{\frac{n}{2}} \frac{\partial^n}{\partial x'^n} L_{f_2}(x',t') \tag{3.27}$$

From these results, utilization of normalized derivatives, which are defined by ([17]),

$$\partial_{t,n} = t^{\frac{n}{2}} \frac{\partial^n}{\partial x^n} \tag{3.28}$$

may be well appreciated. Although the results are presented for the 1-dimensonal case, they may be extended to higher dimensions using similar arguments. These operators will be highly useful when trying to compare regions of arbitrary scales.

## 3.2 Invariant Image Description

As already pointed out, the characterization of a region, in a way immune to affine transformations, is necesssary for the recognition of the region, when viewed from different perspectives. This stems from the fact that a general perspective transformation may locally be approximated by an affine transform ([12]). Using simply the brightness value of each pixel, gives good results, when the transformation between the views is restricted to a translation. However, in case of a more “interesting” transformation, a rotation for instance, the success of recognition of the same region falls significantly ([31]). In such a case, a description invariant to rotation is required, bringing the use of rotational invariants into the picture.

Utilizing rotational invariants, more general (in terms of invariance to different types of transformations) invariants may be developed. These are scale and affine invariants. In the next chapter, the necessity of such invariants are also shown visually to appreciate their requirement. The

discussion in this section is only restricted to the construction of the mentioned invariants, given a feature.

### 3.2.1 Rotational Invariants

Three different types of rotational invariants (which have drawn attention, been used and been known widely ([31], [25], [30], [29], [4])) will be explained in the following subsections.

#### 3.2.1.1 Steering the Gaussian Derivatives in the Direction of the Gradient

Let $f(X)$ be a 2-D signal with $X = [x \quad y]^T$.

Writing the Taylor's expansion around the origin up to fourth order,

$$\begin{aligned} f(X) \cong\ & f + f_x x + f_y y + f_{xx} x^2 + 2 f_{xy} xy + 2 f_{xy} xy + \\ & f_{yy} y^2 + f_{xxx} x^3 + 3 f_{xxy} x^2 y + 3 f_{xyy} xy^2 + f_{yyy} y^3 + \\ & f_{xxxx} x^4 + 4 f_{xxxy} x^3 y + 6 f_{xxyy} x^2 y^2 + 4 f_{xyyy} xy^3 + f_{yyyy} y^4 \end{aligned} \tag{3.29}$$

where the terms containing $f_{i^n j^k}$ denote the value of the partial derivatives evaluated at the origin.

Thus, a reasonable vector characterising $f(X)$ around the origin is:

$$\lfloor f \quad f_x \quad f_y \quad f_{xx} \quad f_{xy} \quad f_{yy} \quad f_{xxx} \quad f_{xxy} \quad f_{xyy} \quad f_{yyy} \quad f_{xxxx} \quad f_{xxxy} \quad f_{xxyy} \quad f_{xyyy} \quad f_{yyyy} \rfloor \tag{3.30}$$

Considering $f(X) = g(X')$ where $X' = \underbrace{\begin{bmatrix} \cos\Theta & \sin\Theta \\ -\sin\Theta & \cos\Theta \end{bmatrix}}_{C} X$ (i.e., the two axes are related by a rotation of Θ), if it were possible to obtain (3.30) for $g(X')$ with respect to the coordinate axes $X'$, there would be no difficulty matching rotated versions of the same feature for a given window size, after properly rotating the image. However, the rotation angle between the two coordinate systems is unknown. At this point, it is noted that, the gradient direction (at the origin) in the two images are also related by the same amount of rotation. For this, differentiate both *f* and *g* with respect to *X* (using the fact that the inverse of a rotation matrix is its transpose):

$$\frac{\partial}{\partial X} g(X)\Big|_{X=\bar{0}} = \frac{\partial}{\partial X} f\left(C^T X\right)\Big|_{X=\bar{0}} = C f'(\bar{0}) \tag{3.31}$$

Then, one obtains :

$$g'(\bar{0}) = \left[ \frac{\partial g}{\partial x}(0,0) \quad \frac{\partial g}{\partial y}(0,0) \right]^T = Cf'(\bar{0}) = C\left[ \frac{\partial f}{\partial x}(0,0) \quad \frac{\partial f}{\partial y}(0,0) \right]^T . \tag{3.32}$$

Thus, if the images are rotated to align the gradient direction with the image's x-axis (see Figure 3.7), it can be claimed that the resulting images would be the same. In order to show this, let $G$ and $F$ be the rotation matrices which transform a unit vector in the x-axis to a unit vector in the gradient direction in the first and second images respectively (note that $G=CF$). Consider the rotated function $f_R$ for the first image:

$$f_R(F^{-1}X) = f(X) \tag{3.33}$$

so,

$$f_R(X) = f(FX). \tag{3.34}$$

Similarly, for the rotated function $g_R$ for the second image,

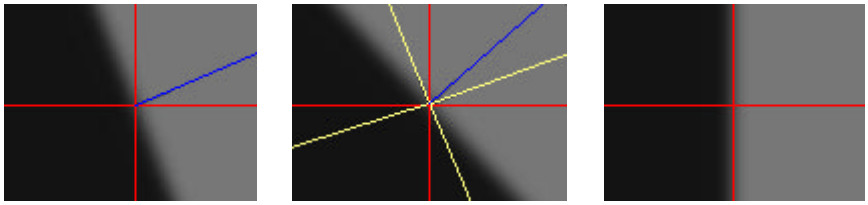
$$g_R(G^{-1}X) = g(X) = f(C^{-1}X) \tag{3.35}$$

hence,

$$g_R(X) = f(C^{-1}GX) = f(FX) \tag{3.36}$$

thus,

$$g_R(X) = f_R(X). \tag{3.37}$$



(a) (b) (c)

Figure 3.7 Graphs of rotated edges. (a) $f(X)$, $X$ plotted in red, gradient direction in blue, (b) $g(X)$, $X$ plotted in red, $X'$ plotted in yellow, gradient plotted in blue, (c) rotating both (a) and (b) so as to align the gradient direction with the $x$-axis yields the same image.

In fact, investigating the result of the equations, it can be suggested that each region to be described be rotated and the brightness values of

the rotated region be taken as a characterization of the region. Alternatively, instead of explicitly rotating the image, the vector (3.30), compactly representing the region may be obtained as if the gradient direction and the direction orthogonal to the gradient were the axes. This is equivalent to first rotating the image and then obtaining (3.30). In order to illustrate the calculation procedure, $f_{x_g y_g}$ will be computed (where $\vec{x}_g$ denotes the gradient direction and $\vec{y}_g$ the direction orthogonal to it, so that when $\vec{x}_g$ is rotated to match the x-axis, i.e. $\vec{x}$, $\vec{y}_g$ matches the y-axis, i.e. $\vec{y}$, and a general vector is written as $[x \quad y] = x\vec{x} + y\vec{y} = x_g \vec{x}_g + y_g \vec{y}_g$ ) :

Let $I_x = \dfrac{f_x}{\sqrt{f_x^2 + f_y^2}}$ and $I_y = \dfrac{f_y}{\sqrt{f_x^2 + f_y^2}}$ (evaluated at the point), then,

$$\begin{bmatrix} x_g \\ y_g \end{bmatrix} = \begin{bmatrix} I_x & I_y \\ -I_y & I_x \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} \Leftrightarrow \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} I_x & -I_y \\ I_y & I_x \end{bmatrix} \begin{bmatrix} x_g \\ y_g \end{bmatrix}, \tag{3.38}$$

the derivative along $\vec{x}_g$ is obtained as:

$$f_{x_g} = \frac{\partial f}{\partial x}\frac{\partial x}{\partial x_g} + \frac{\partial f}{\partial y}\frac{\partial y}{\partial x_g} = f_x I_x + f_y I_y \tag{3.39}$$

and,

$$\begin{aligned} f_{x_g y_g} = \frac{\partial f_{x_g}}{\partial y_g} &= \left( \frac{\partial f_x}{\partial x}\frac{\partial x}{\partial y_g} + \frac{\partial f_x}{\partial y}\frac{\partial y}{\partial y_g} \right) I_x + \left( \frac{\partial f_y}{\partial x}\frac{\partial x}{\partial y_g} + \frac{\partial f_y}{\partial y}\frac{\partial y}{\partial y_g} \right) I_y \\ &= -f_{xx} I_x I_y + f_{xy}\left(I_x^2 - I_y^2\right) + f_{yy} I_x I_y \end{aligned} \tag{3.40}$$

This procedure may be extended to any desired length of the vector. However, an important point to note is, as the vector proceeds, the terms get more dependent on a precise estimation of the gradient direction. This precision might be hard to achieve, which seems to be the major difficulty for applying the approach. For improving the estimation, some authors use a local histogram for the calculation of the gradient direction ([19]). Moreover, some of the terms depending on the second derivative may also be computed without referencing explicitly to the direction of the gradient as a property of the Hessian. This can be shown as follows:

The second order derivative of $f$ in some direction $[x\ y]$ is :

$$\frac{\partial^2}{\partial X^2}(X) = \begin{bmatrix} x & y \end{bmatrix} \underbrace{\begin{bmatrix} f_{xx} & f_{xy} \\ f_{yx} & f_{yy} \end{bmatrix}}_{A} \begin{bmatrix} x \\ y \end{bmatrix} = X^T AX \tag{3.41}$$

under a rotation as $X' = RX$ , where $R^T R = I$ ,

$$\frac{\partial^2 f}{\partial X^2}(X') = X^T R^T ARX = X^T BX \tag{3.42}$$

using the facts that (for two square matrices Q, W of the same size) $Tr(QW) = Tr(WQ)$, and $DET(QW) = DET(Q)DET(W)$, it is concluded that the trace of $A$ is equal to the trace of $B$ (i.e. the Laplacian) and the determinants of the matrices are equal. These (or any combination of these) may be replaced with two of the second order terms of (3.30). In fact, since the two eigenvalues are determined by these two numbers, a third coefficient is not necessary, since these are the only entities that remain constant under a rotation (i.e. invariance to rotation introduces a redundancy to the representation using the full set of partial derivatives).

Indeed, the term '*steerable filter*' stems from the work of Adelson and Freeman ([1]). In their research, what the authors have primarily proposed, is to obtain the responses of the same filter rotated in an arbitrary amount, without applying the actually rotated filter. Instead, they form a basis of filters so as to enable calculation of the response for any amount of rotation. This basis consists of the rotated versions of the filter. The idea may explicitly be expressed as (for a filter $f$, $f_\Theta$ denoting the '$\Theta$' rotated filter) ;

$$f_\Theta = \sum_{i=1}^{n} a_{\Theta_i} f_{\beta_i} , \tag{3.43}$$

$a_{\Theta_i}$ denoting the coefficient for each element of the basis, dependent on the rotation angle $\Theta$. Such a representation is possible if the original function may be written as:

$$f(X) = \sum_{n=-t}^{t} c_n(r) e^{in\Theta} , \tag{3.44}$$

i.e., if the function may be written as a finite sum of complex exponentials in polar angles multiplied by rotationally invariant functions. In other

words, the Fourier terms, (which are of a finite number) expressed in polar angles, form a basis for the function and any translation in the polar angle, may be handled by appropriately adjusting the coefficient of each element in the basis.

That the Gaussian derivatives are steerable follows from writing these as a sum of the form (3.44). Actually, considering a derivative of any arbitrary order, one can observe that :

$$\frac{\partial^{n+m}}{\partial x^n y^m} g(x,y,t) = \sum_{k=1}^{n} \sum_{l=1}^{m} x^k y^m a_{k,l}(r) \tag{3.45}$$

$a_{k,l}(r)$ being a rotationally symmetric function (r representing the radial variable). Inserting $x = r\cos\Theta$, $y = r\sin\Theta$ in (3.45), and moreover noting that $\cos\Theta = \frac{e^{i\Theta} + e^{-i\Theta}}{2}$, $\sin\Theta = \frac{-e^{i\Theta} + e^{-i\Theta}}{2j}$, it's seen that (3.45) may be written in the form (3.44). Thus, the Gaussian derivatives are steerable in the sense of (3.43).

However, it is noted that the procedure explained in the beginning exploits the properties of directional derivatives of a partially differentiable function and actually the basis filters are not rotated versions of each other (consider $f_{xxx}$, $f_{xxy}$, etc.; on the contrary, the first order derivatives exactly fit the framework outlined in (3.43)) (see Figure 3.8).

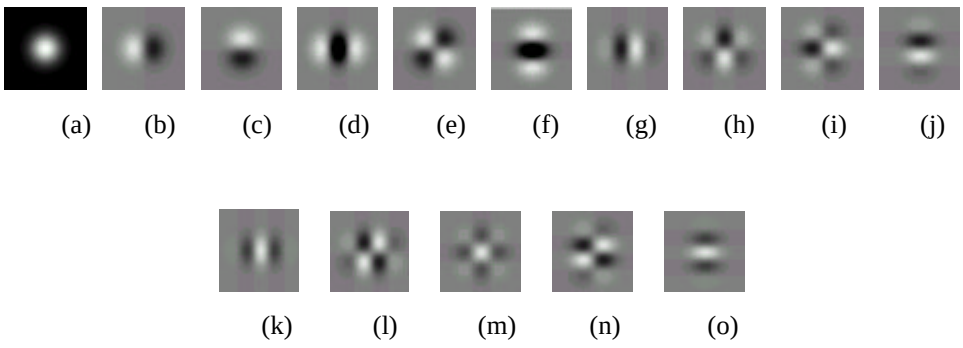



Figure 3.8 Images of a Gaussian and its partial derivatives (a) g, (b) $g_x$, (c) $g_y$, (d) $g_{xx}$, (e) $g_{xy}$, (f) $g_{yy}$, (g) $g_{xxx}$, (h) $g_{xxy}$, (i) $g_{xyy}$, (j) $g_{yyy}$, (k) $g_{xxxx}$, (l) $g_{xxxy}$, (m) $g_{xxyy}$, (n) $g_{xyyy}$, (o) $g_{yyyy}$. It is noted that (b) and (c) are simply rotated versions of each other and form a basis for the first order derivatives of the Gaussian in any direction. However, this is not the case for higher order derivatives and the elements of the basis (that is utilised) are not rotated verisons of each other.

The last point to note is about comparing two vectors. Considering each entry of the vector as a random variable, it is noted from the previous discussion about the non-orthonormality of the filter bank produced by the derivatives of Gaussians, that each entry will inevitably "contain some" from another entry. In other words, the entries will not be independent. Moreover, the expected value and covariance of each entry may be different. Thus, a fair comparison of the vectors are possible after normalizing the covariance matrix for the ensemble of these vectors. This is indeed what *Mahalanobis distance* does ([35]). According to this scheme, the distance between two (row) vectors $x$, $y$ are given by:

$$d(x,y)=(x-y)C^{-1}(x-y)^T \tag{3.46}$$

where $C$ is the covariance matrix estimated by:

$$C=\frac{1}{N}\sum_{k=1}^{N}(x_k-\mu)^T(x_k-\mu), \tag{3.47}$$

where each $x_k$ denotes a vector obtained through a distinct observation and $N$ is the total number of such vectors in the ensemble. The mean vector, $\mu$, is given by:

$$\mu=\frac{1}{N}\sum_{k=1}^{N}x_k\ . \tag{3.48}$$

### 3.2.1.2 Invariants Based on Transforms

It is known that for a 2-D function $f(x,y)$ (with Fourier Transform $F(\omega_1,\omega_2)$), if the function is translated in some arbitrary amount (in both directions), the magnitude of the resulting function's Fourier Transform(FT) will be the same as the original functions'. This follows simply from the properties of FT. What happens though under a rotation of the function? Let $f(r,\Theta)$ be the same function expressed in terms of polar coordinates, and similarly $F(R,\Phi)$ be the FT expressed in corresponding polar coordinates. Then, since,

$$F(\omega_1,\omega_2)=\iint f(x,y)e^{-j\omega_1 x}e^{-j\omega_1 y}dxdy \tag{3.49}$$

Expressing all the variables in the corresponding polar variables,

$$x = r\cos\Theta\,,\ \ y = r\sin\Theta\,,\ \ \omega_1 = R\cos\Phi\,,\ \ \omega_2 = R\sin\Phi\,, \tag{3.50}$$

and inserting into (3.49),

$$F(R,\Phi) = \iint f(r,\Theta)e^{-jR(\cos\Phi)r(\cos\Theta)}e^{-jR(\sin\Phi)r(\sin\Theta)}rdrd\Theta \tag{3.51}$$

$$= \iint f(r,\Theta)e^{-jRr[(\cos\Phi)(\cos\Theta)+(\sin\Phi)(\sin\Theta)]}rdrd\Theta \tag{3.52}$$

$$= \iint f(r,\Theta)e^{-jRr\cos(\Theta+\Phi)}rdrd\Theta\,. \tag{3.53}$$

From (3.53), it is seen that a rotation in the original function, which corresponds to a shift in Θ, causes the FT to be rotated (i.e. a shift in Φ) as well (in contrast to the translation propert of FT.) What can be mimicked to obtain an invariant, is the translation property as mentioned in the beginning. Thus, in order to obtain a transform whose magnitude is invariant to rotations, the following set of complex coefficients are proposed by Baumberg ([4]):

$$u_{n,m} = \int f(r,\Theta)\left(\frac{d^n}{dr^n}g(r,t)\right)e^{jm\Theta}rdrd\Theta \tag{3.54}$$

Under a rotation of the image $f'(r,\Theta) = f(r,\Theta+\alpha)$, these coefficients are transformed as follows:

$$u'_{n,m} = e^{-jm\alpha}u_{n,m}\,. \tag{3.55}$$

Thus, for the group ( *m = constant* ), the action under a rotation is the same. To render these coefficients invariant to rotation, one can simply take the magnitudes since $\left|e^{-jm\alpha}\right| = 1$. Alternatively, if the coefficients $u_{n,m}$ are divided by unit-length complex number proportional to $u_{0,m}$, the multiplicative factors $e^{-jm\alpha}$ can also be eliminated. Thus, the set of coefficients obtained this way are invariant to rotations.

The comparison of these invariants is also based on the Mahalanobis distance, since the kernels are not orthonormal.

### 3.2.1.3 Orthonormal Filters

Another major approach to obtain rotation invariance, is to use orthonormal filters. Consider the family of kernels ([29]),

$$K_{m,n}(x,y)=(x+jy)^m(x-jy)^n g(x,y) \tag{3.56}$$

where $g(x,y)$ denotes the Gaussian. Expressed in polar coordinates,

$$K_{m,n}(r,\Theta)=r^m e^{j\Theta m} r^n e^{-j\Theta n} g(r)=r^{m+n} e^{j\Theta(m-n)} \tag{3.57}$$

Applying the kernel on a given function $f(r,\Theta)$, the response is:

$$z_{m,n}=\iint f(r,\Theta) r^{m+n} e^{j\Theta(m-n)} g(r) r dr d\Theta\,. \tag{3.58}$$

Similarly, the response after rotating the function so that $f'(r,\Theta)=f(r,\Theta+\alpha)$ is:

$$z'_{m,n}=\iint f(r,\Theta+\alpha) r^{m+n} e^{j\Theta(m-n)} g(r) r dr d\Theta \tag{3.59}$$

$$=\iint f(r,\beta) r^{m+n} e^{j\beta(m-n)} e^{-j\alpha(m-n)} g(r) r dr d\beta \tag{3.60}$$

$$=z_{m,n} e^{-j\alpha(m-n)} \tag{3.61}$$

Thus, for the group ($m$-$n$) = *constant*, the action is the same under a rotation. Since swapping $m$ and $n$ simply results in complex conjugate filters, it is enough to calculate the response of the filters with $m \geq n$. Moreover, it is noted that $K_{m,n}$ and $K_{k,l}$ are orthogonal if ($m$-$n$) ≠ ($k$-$l$):

$$\int_{\Theta=0}^{2\pi} \int_{r=0}^{\infty} r^{m+n} e^{j\Theta(m-n)} r^{k+l} e^{j\Theta(l-k)} g^2(r) r dr d\Theta \tag{3.62}$$

$$=\int_{\Theta=0}^{2\pi} M_{m+n+k+l} e^{j\Theta[(m-n)-(k-l)]} d\Theta\,, \tag{3.63}$$

($M_{m+n+k+l}$ exists, since the Gaussian decays more rapidly than any polynomial).

Equation (3.63) equals 0, if ($m$-$n$) ≠ ($k$-$l$), and $2\pi M_{m+n+k+l}$ otherwise. The group of filters for which ($m$-$n$) = *constant*, may be orthonormalized using Gram-Schmidt orthonormalization procedure, and once this is achieved (since the group action for ($m$-$n$) = *constant* is the same under a rotation, the resultant group after orthonormalization using the mentioned method will also exhibit the same behaviour), the whole set of filters will be orthonormal. Orthonormality will allow the utilization of the simple Euclidian distance.

In order to achieve rotational invariance, simply taking the magnitude of the response (or using the scheme described in the preceding subsection) is sufficient. Since $\left|\left|z\right|-\left|w\right|\right| \le \left|z-w\right|$, the Euclidian distance for these invariants is noted (since the filter bank is orthonormal) to be a lower bound on the region's sum of squared differences (SSD).

The use of the Euclidian distance is favored (compared to the Mahalanobis distance), since for this distance, there is no need to estimate the covariance matrix for the characterization vectors, which is a rather tedious process.

### 3.2.2 Scale Invariants

The idea of scale-space offers operators and a representation to analyze the behavior of a signal throughout different scales, but it does not explicitly state the exact scale under which, a structure present in some signal should be observed. However, combining the outputs of the mentioned operators, schemes to detect a salient scale for a feature, which would be affected in the same manner as would some unit length in the given signal under rescalings of the signal. As an example, consider a one dimensional sinusoidal input signal

$$f(x) = \sin(\omega_o x) \tag{3.64}$$

The solution for the 1-D diffusion equation (3.4) with initial condition $L(x;0) = f(x)$ is given by :

$$L(x;t) = e^{-\frac{\omega_0^2 t}{2}} \sin(\omega_0 x) \tag{3.65}$$

This can be verified by simply observing that the equality (3.4) holds for *L* given by (3.65). Applying the $m^{th}$ order normalized derivative $\partial_{t,m} = t^{\frac{m}{2}} \frac{\partial^m}{\partial x^m}$, the amplitude at the origin, as a function of scale is given by:

$$\partial_{t,m} L = t^{\frac{m}{2}} \omega_0^m e^{-\frac{\omega_0^2 t}{2}} . \tag{3.66}$$

It is observed that this function first increases and then decreases (for non-negative $t$), assuming a unique maximum at the zero crossing of the derivative wrt $t$, given by:

$$t_{\max,L} = \frac{m}{\omega_0^2} \tag{3.67}$$

Defining a scale parameter $s$ by

$$s = \sqrt{t}\,, \tag{3.68}$$

it is observed that the scale at which the normalized derivative assumes its maximum is proportional to $\frac{2\pi}{\omega_0}$, i.e., the wavelength of the sinusoid.

Moreover, placing (3.67) in (3.66), the maximum value over scales is:

$$\sup \partial_{t,m} L = e^{-m} m^{\frac{m}{2}} \tag{3.69}$$

which is noted to be independent of the wavelength of the signal. In other words, using this scheme, one can treat sinusoidal signals of arbitrarily different frequencies, independent of their frequency (see Figure 3.9).

The operation outlined above constitutes a way of estimating length (or wavelength) based on local measurements performed at a single point. This sounds like what Short Time Fourier Transform (STFT) intends to achieve, trying to estimate the frequency content of a signal at a particular time (or, reversely, the time content at a particular frequency). However, in contrast to STFT, there is no need to explicitly set a window size for realizing the computations, which actually exists in the definition of STFT. In this view, the selection of a windowing function may be regarded as 'buried' in the diffusion equation. Thus, in the context of time-frequency analysis, this operation may be interpreted as similar to what wavelets do ([16]).

The measurement procedure based on derivatives may also be viewed as a pattern matching process. The signal is matched with Gaussian derivative kernels of different order, scale, with the corresponding normalization.

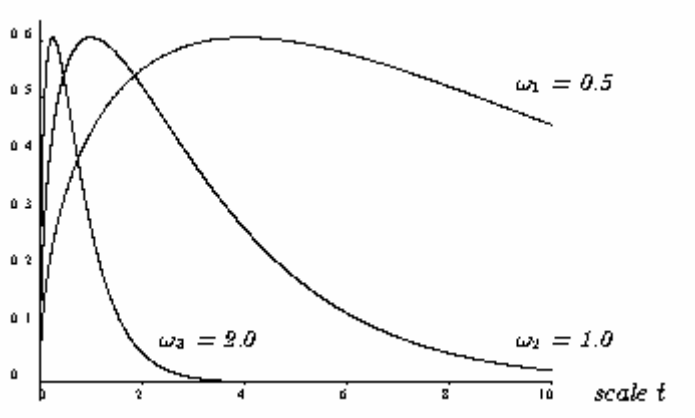



Figure 3.9 Amplitudes of normalized scale-space derivatives of first order for sinusiods with angular frequencies $\omega_1 = 0.5$, $\omega_2 = 1.0$, $\omega_3 = 2.0$ ([17]).

Following these ideas, Lindeberg ([17]) states the following principle:

> *"Principle for scale selection:*
>
> *In the absence of other evidence, assume that a scale level, at which some (possibly non-linear) combination of normalized derivatives assumes a local maximum over scales, can be treated as reflecting a characteristic length of a corresponding structure in the data."*

The remaining question awaiting an answer is : "Which particular combination of derivatives should be used?". In fact, different authors have used different combinations and have obtained good results in their own account. For example, Lindeberg ([17]) uses the normalized Laplacian of Gaussian and Lowe ([19]) utilizes difference of Gaussians, leading to an efficient computation scheme at the expense of sampling the signal at hand. Mikolajczyk and Schmid ([25]) have actually shown experimentally that among the following candidates given as:

| | | |
|---|---|---|
| *Square Gradient* | $s^2\left(L_x^2(x;s)+L_y^2(x;s)\right)$ | (3.70) |
| *Laplacian* | $\left\|s^2\left(L_{xx}(x;s)+L_{yy}(x;s)\right)\right\|$ | (3.71) |
| *Difference of Gaussian* | $\left\|I(x)*g(x;s_{n-1})-I(x)*g(x;s_n)\right\|$ | (3.72) |
| *Harris function* | $\det(C)-\alpha trace^2(C)$ | (3.73) |

with

$$C(x, s_i, s_d) = s_d^2 g(x; s_i) * \begin{bmatrix} L_x^2(x; s_d) & L_x(x; s_d) L_y(x; s_d) \\ L_x(x; s_d) L_y(x; s_d) & L_y^2(x; s_d) \end{bmatrix} \quad (3.74)$$

where $I(x)$ is the signal and $L(x)$ its scale-space representation with the scale parameter $s$ as previously defined in (3.68) (it is also noted that the scale parameter is discretized due to practical reasons, as $s_n$, where $s_0$ denotes zero scale), the normalized Laplacian gives the best results in terms of correctly[2] selecting the scale for a feature.

The Harris function is noted to be a scale-adapted version of the one described in Section 2.7 . In this respect, the derivatives are computed using derivatives of Gaussians with scale parameter $s_d$ (differentiation scale) and averaging is performed using a Gaussian of scale parameter $s_i$ (integration scale). Usually, the differentiation and integration scales are taken to be proportional, i.e., $s_i = \alpha s_d$ .

Based on the experimental facts mentioned, Mikolajczyk and Schmid propose the following scale invariant detector ([25]):

Algorithm 3.1:
1. Build a scale-space representation for the Harris function (68). Detect the candidate interest points (noting the scale at which they were detected) at each scale by thresholding the response of the function and eliminating the non-maxima.
2. A candidate point $x$ at some scale $s_n$ is taken to be an interest point if its response to the Laplacian (3.71), denoted by $F(x; s_n)$, forms a local maximum in scale, i.e., if

$$F(x; s_n) > F(x; s_{n-1}) \wedge F(x; s_n) > F(x; s_{n+1}) \quad (3.75)$$

where $\wedge$ denotes 'logical and'.

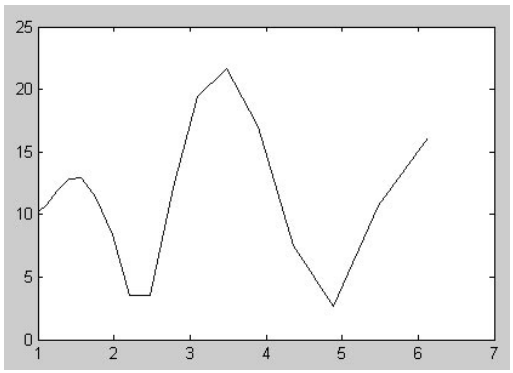
After detecting the features and their corresponding scales, the final thing to do is to obtain a local description of the region preferably invariant to rotation. This may be achieved by utilizing normalized derivatives at the

[2] The experiments ([25]) had been carried out on scaled versions of a given image. The selected scale for a point is said to be correct, if the ratio between the characteristic scales in corresponding points in the given images are equal to the scale factor between the images.

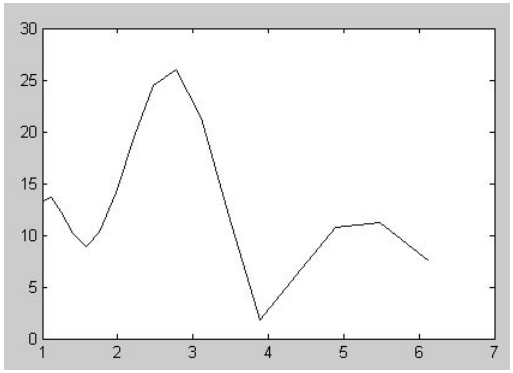
corresponding scales with the method described in Section 3.2.1.1. The descriptors mentioned in Sections 3.2.1.2 and 3.2.1.3 may also be rendered invariant to scale by a proper normalization.

(a)

(b)




(c)




(d)

Figure 3.10 Response of the scale-adapted Laplacian operator on **"**Haydarpaşa Tren İstasyonu" as a function of the scale parameter defined in (3.68). (a),(b) Haydarpaşa Tren İstasyonu in different scales, (c) response of the scale-adapted Laplacian operator applied at the point marked in red in (a), (d) response of the scale-adapted Laplacian operator applied at the point marked in red in (b). Note that the two plots are actually (roughly) rescaled versions each other. The scaling factor is expected to be equal to the scaling among the two images (where (3.68) is taken as the scale parameter).

### 3.2.3 Affine Invariants

As mentioned before, invariance of description under affine transformation of a patch may be useful while considering perspective deformations. Some practical schemes were proposed by Lindeberg and

Gårding ([18]), Baumberg ([4]), Mikolajczyk and Schmid ([26]), which actually exploit the same transformation property which will be explained. In fact, the Hessian also posseses the same property, and thinking that it is relatively easier to visualize the physical meaning of the Hessian, affine invariance in terms of normalizing the Hessian, which has not somehow drawn much attention, will be presented first.

#### 3.2.3.1 Normalizing the Hessian

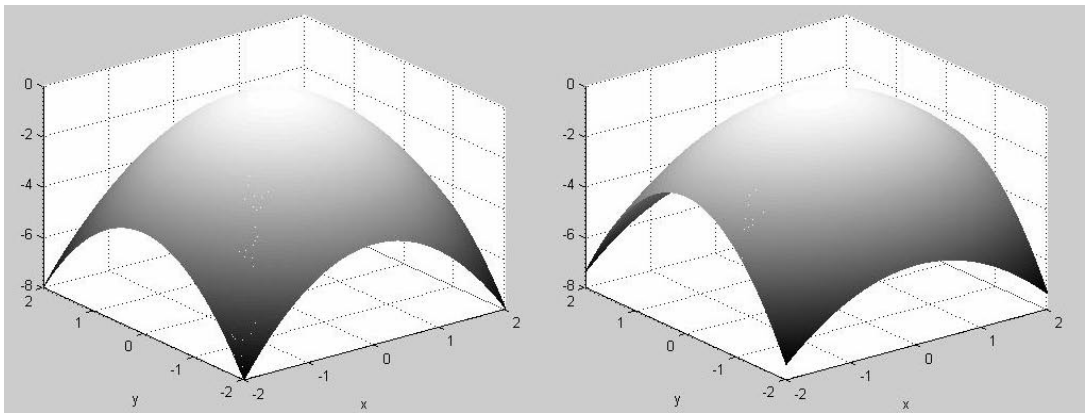
Consider the function

$$f(x,y)=\frac{1}{2}\left(x^2+y^2\right). \tag{3.76}$$

(see Figure 3.11). The second derivatives of this function in the $x$ and $y$ directions, evaluated at the origin, are found to be,

$$\partial_x^2 f(0,0)= f_{xx}=1 \tag{3.77}$$

$$\partial_y^2 f(0,0)= f_{yy}=1. \tag{3.78}$$




(a) (b)

Figure 3.11 Inverted graph of (a) $f(x,y)=x^2+y^2$, (b) $f(x,y)=(0.8x)^2+(1.1y)^2$

If the graph of the function is 'stretched' in the x-direction and 'squeezed' in the y-direction, one obtains

$$f'(x,y)=f(ax,by), \tag{3.79}$$

where $0<a<1$ and $1<b$.

Now, the second derivatives of this new function can be evaluated:

$$\partial_x^2 f' = f'_{xx} = a^2 f_{xx} = a^2, \tag{3.80}$$

$$\partial_y^2 f' = f'_{yy} = b^2 f_{yy} = b^2. \tag{3.81}$$

Under these circumstances, it can be observed that the knowledge on the second derivatives enables one to recover the original rotationally symmetric function as:

$$h(x,y) = f'\left(\frac{x}{\sqrt{f'_{xx}}}, \frac{y}{\sqrt{f'_{yy}}}\right) = f\left(\frac{ax}{a}, \frac{bx}{b}\right) = f(x,y). \tag{3.82}$$

A general affine normalization based on the Hessian matrix follows from a similar idea. For this purpose, let $f(X)$ , where $X=(x,y)$, be a function with the Hessian at the origin denoted by $f_H$ , as:

$$f_H = I . \tag{3.83}$$

and, let

$$h(X) = f(BX) \tag{3.84}$$

where $B$ is a non-singular 2x2 matrix.

It can be verified by an application of the chain rule that:

$$h_H(X) = B^T f_H(BX) B = M . \tag{3.85}$$

Due to (3.85), $M$ is a positive definite (PD) matrix. It is noted that what is at hand is the PD matrix $M$ and the desired matrix to be obtained is $B$. In fact, exactly recovering $B$ from $M$ is not possible, since for an orthonormal matrix $R$,

$$B^T R^T R B = B^T B = M . \tag{3.86}$$

In other words, $M$ determines $B$ up to a rotation. This is practically realized by utilizing *Cholesky Decomposition* ([5]). The decomposition algorithm, for a PD input $D$, returns a non-singular matrix $L$, such that,

$$D = L^T L . \tag{3.87}$$

Applying Cholesky decomposition to $M$,

$$M = A^T A \tag{3.88}$$

where $A = RB$, (the square root matrix) and $R$ representing an arbitrary rotation matrix.

Now, backtransforming $h$ using $A$, one obtains

$$k(X) = h(A^{-1}X) = f(BA^{-1}X) = f(R^{-1}X) \tag{3.89}$$

Considering another function,

$$h'(X) = f(EX) \tag{3.90}$$

where $E$ is another non-singular matrix, carrying out the same procedure, one obtains

$$k'(X) = f(R_2^{-1}X) = k(RR_2^{-1}X) = k(R_3X). \tag{3.91}$$

Thus, since $k(X)$ and $k'(X)$ are rotated versions of each other, it can be concluded that matching between these two arbitrarily affine transformed patches may be realized by utilizing rotational invariants introduced in the previous sections.

If it were possible to obtain the Hessian at a point, this method could be used to obtain an affine invariant description of any 'interest region'. However, as outlined in the previous sections, for differentiation, it is necessary to use a smoothing function, which is taken to be the Gaussian for this case. It will be shown that only under special conditions, the transformation property (3.85), which gives a chance to obtain affine invariance, holds, when the differentiation is carried out using derivatives of Gaussians.

Since differentiation of a function using the derivative of a Gaussian, is equivalent to smoothing the derivative of the function, by (3.20), the smoothed Hessian, denoted by $f_{g,H}$, calculated at a point by the use of the Gaussian derivatives of second order is given by:

$$f_{g_I,H}(X) = \int f_H(A) \frac{1}{2\pi t} e^{-\frac{(X-A)^T(X-A)}{2t}} dA. \tag{3.92}$$

For an affine transformed function, $h(X) = f(BX)$, the smoothed Hessian with respect to $X$, using (3.85) may be calculated writing $f_H$ in terms of $h_H$:

$$f_{g_I,H} = \int (B^{-1})^T h_H(B^{-1}A) B^{-1} \underbrace{\frac{1}{2\pi t} e^{-\frac{(B(B^{-1}X - B^{-1}A))^T (B(B^{-1}X - B^{-1}A))}{2t}}}_{T(X)} dA. \tag{3.93}$$

Since,

$$(\det B^{-1}) dA = dB^{-1}A, \tag{3.94}$$

and a generalized Gaussian may be written as:

$$g(X;\Sigma)=\frac{1}{2\pi\sqrt{\det\Sigma}}e^{-\frac{(X)^T\Sigma^{-1}(X)}{2}}=\frac{T(X+A)}{\left(\sqrt{\det\frac{\Sigma}{t}}\right)}=\frac{T(X+A)}{(\det B)}. \tag{3.95}$$

Thus,

$$T(X)=\sqrt{\det\left(\frac{\Sigma}{t}\right)}g\left(B^{-1}X-B^{-1}A;\Sigma\right)=\left(\det\left(B^{-1}\right)\right)g\left(B^{-1}X-B^{-1}A;\Sigma\right), \tag{3.96}$$

where $\Sigma=tB^{-1}\left(B^{-1}\right)^T$.

Substituting these relations in (3.93) gives:

$$\begin{aligned} f_{g_I,H}(X)&=\left(B^{-1}\right)^T\left(\int h_H\left(B^{-1}A\right)g\left(B^{-1}X-B^{-1}A;\Sigma\right)dB^{-1}A\right)B^{-1}\\ &=\left(B^{-1}\right)^T h_{g_\Sigma,H}\left(B^{-1}X\right)B^{-1}\end{aligned}. \tag{3.97}$$

So,

$$h_{g_\Sigma,H}(X)=\left(B^T\right)f_{g_I,H}(BX)B. \tag{3.98}$$

In words, for the transformation property to hold, it is necessary that the Gaussian used for smoothing (and differentiation) must be adapted to the affine transformation applied to the coordinate axes. However, for this adaptation, it is required to know the transformation (i.e. the matrix *B*) applied to the coordinates. It is noted that this is exactly what one would like to estimate through the observation of the Hessian calculated with a generalized Gaussian and thus, is not at hand. This difficulty may be overcome by the use of an iterative procedure. In fact, what should be achieved is to transform the image so that the Hessian calculated using the derivatives of an isotropic (or rotationally symmetric) Gaussian is equal to *cI*, where *c* is an arbitrary constant. Thus, the following algorithm, which is adapted from Lindeberg and Gårding ([18]), and Baumberg's ([4]) works, may be proposed:

Algorithm 3.2:

1. Take a point where the Hessian $H_0$, calculated using derivatives of an isotropic Gaussian (of the salient scale detected by the algorithm outlined in 3.3.2), is PD and write $H_0 = B_0^T B_0$ using Cholesky decomposition (and let *n=0*).

2. For a local patch at the origin of which lies the interest point chosen in step 1, obtain the affine transformed new patch

$$f_{n+1}(X) = f_n\left(\left(\frac{1}{\sqrt{\det B_n}} B_n\right)^{-1} X\right), \qquad (3.99)$$

where $H_n = B_n^T B_n$.

3. Calculate the Hessian $H_{n+1}$ for $f_{n+1}(X)$ using derivatives of the isotropic Gaussian utilised in step-1. If $H_{n+1}$ is not sufficiently close to $\sqrt{\det H_{n+1}}\, I$, increase $n$, go to step-2.

4. Obtain the local characterization of the image patch $f_{n+1}(X)$ using rotational invariants.

A few remarks should be made regarding the algorithm above:

Firstly, for a point, considering the applicability of the algorithm, it is necessary and sufficient, for the Hessian to be PD at the point. The Hessian, being PD ensures that Cholesky decomposition, may be applied to obtain the square root matrix and that the properties (3.84), (3.85) hold (sufficiency). Also, if the Hessian is not PD, there does not exist a non-singular square root matrix and the algorithm should not be applied at all (necessity).

Secondly, it is assumed throughout that the salient scale is not changed, if the transformations are carried out using a non-singular matrix with determinant equal to unity ($\det\left(\frac{1}{\sqrt{\det B}} B\right) = 1$). In fact, this need not be so, and theoretically, a local scale adaptation ([26]) should be made after each iteration of the algorithm. However, it is experimentally observed that the algorithm without this local scale adaptation also gives good results.

Thirdly, it is noted that the same algorithm is applicable in the case of Negative Definite Hessian possessing points as well, after a proper arrangement of the signs and square roots of the matrices.

### 3.2.3.2 Normalizing the 'Harris Matrix'

Observing that the transformation property (3.98) allowing a normalization, is actually the essence of the affine invariance obtained by the algorithm explained in the previous section, it is noted that other structures possessing similar characteristics may be used for this purpose. The Harris matrix (3.74) is an appropriate example. It will be shown in this section that this matrix behaves just like the Hessian under affine transformations, with similar restrictions on the Gaussians used.

Consider two functions related by an affine transformation

$$h(X)=f(BX) \tag{3.100}$$

and let the gradient vector of f(X) be denoted as,

$$f_X(X)=\begin{bmatrix} f_x(X) \\ f_y(X) \end{bmatrix} \tag{3.101}$$

then differentiating $h(X)$ applying the Chain rule:

$$h_X(X)=B^T f_X(BX). \tag{3.102}$$

Consider now, with the same reasoning as the last section (necessity of smoothing in differentiation), the smoothed derivative of $f$ is written:

$$f_{g_{tI},X}(X)=\int f_X(A)\frac{1}{2\pi t}e^{-\frac{(X-A)^T(X-A)}{2t}}dA \tag{3.103}$$

Carrying out similar a change of variables as in the last section,

$$f_{g_{tI},X}(X)=\int\left(B^T\right)^{-1}h_X\left(B^{-1}A\right)g\left(B^{-1}X-B^{-1}A,\Sigma\right)dB^{-1}A \tag{3.104}$$

where $\Sigma=tB^{-1}\left(B^{-1}\right)^T$.

Thus,

$$h_{g_\Sigma,X}(X)=B^T f_{g_{tI},X}(BX). \tag{3.105}$$

It's observed that for the transformation property (3.102) for the Gaussian derivatives (3.103) to hold, it is necessary to adapt the Gaussian.

Now, defining

$$f_{g_{tI},C}(X)=f_{g_{tI},X}(X)\cdot\left(f_{g_{tI},X}(X)\right)^T, \tag{3.106}$$

the Harris matrix for $f$ can be expressed as:

$$C_f(X,sI,tI)=f_{g_{tI},C}(X)*g(X;sI) \tag{3.107}$$

Moreover, the following (compare with (3.85)) can be deduced:

$$h_{g_\Sigma,C}(X) = B^T f_{g_{tI},C}(BX)B. \tag{3.108}$$

Explicitly writing (3.107),

$$C_f(X, sI, tI) = \int f_{g_{tI},C}(A)g(X - A; sI)dA \tag{3.109}$$

and once again changing variables,

$$\begin{aligned} C_f(X, sI, tI) &= \left(B^T\right)^{-1}\left(\int h_{g_\Sigma,C}(X)g\left(B^{-1}X - B^{-1}A; \frac{s}{t}\Sigma\right)dA\right)B^{-1} \\ &= \left(B^T\right)^{-1} C_h\left(B^{-1}X, \frac{s}{t}\Sigma, \Sigma\right)B^{-1} \end{aligned} \tag{3.110}$$

So,

$$C_h\left(X, \frac{s}{t}\Sigma, \Sigma\right) = B^T\left(C_f(BX, sI, tI)\right)B. \tag{3.111}$$

In words, the Harris matrix calculated using adapted Gaussians possesses the transformation property sufficient to obtain affine invariance. Thus, Algorithm 3.3.3.1 may be adapted for this case also. It is noted that the Harris matrix at the points extracted by (3.73) will necessarily be PD (because of (3.106), (3.107) and the fact that, at the interest point the eigenvalues are non-zero). The assumption that the salient scale and position of the interest point will not change may not hold and thus for higher precision (at the cost of higher computation time of course) the following algorithm is proposed by Mikolajczyk and Schmid:

Algorithm 3.3:

1. For an interest point extracted by using Algorithm 3.3.2 write the Harris matrix at that point as $C_0 = B_0^T B_0$ using Cholesky decomposition (and let $n=0$).

2. For a local patch at the origin of which lies the interest point, obtain the affine transformed new patch

$$f_{n+1}(X) = f_n\left(B_n^{-1}X\right), \tag{3.112}$$

where $C_n = B_n^T B_n$.

3. Update the scale and position of the interest point for this new patch. Calculate the $C_{n+1}$ at the updated position for $f_{n+1}(X)$ using derivatives of the isotropic Gaussian for differentiation(at the corresponding updated differentiation

scale) and an isotropic Gaussian for integration(at the corresponding updated integration scale). If $C_{n+1}$ is not sufficiently close to $\sqrt{\det C_{n+1}}I$, increase $n$, go to step-2.

4. Obtain the local characterization of the image patch $f_{n+1}(X)$ using rotational invariants.

# CHAPTER IV

# MATCHING

Establishing point correspondences in different views of the same scene is investigated in this chapter, using the results and methods developed in the previous chapters. In the following section, the geometry of the problem will be briefly explained, which is, in no ways meant to be exhaustive. This informal treatment will only be required for describing a model for the correspondence data to be fit. The existence of such a model is extremely useful for abandoning possible false matches, which will be evident in the following sections. After this step, different matching methods beginning with the simplest (in terms of robustness to the arbitrariness of the views) will be described. Following each method, the motivation for further refinement of the current algorithm will be stated, i.e. a case (possibly obvious) will be shown for which the current algorithm would fail. Thus, as in the previous chapter, as the sections proceed, the algorithms are intended to be more robust to the arbitrariness of the input images. In this regard, the presentation will start with the small-baseline case. Then, a neighboring constraint will be presented. After this, another method, utilizing the invariants, which are described in the last chapter, will be explained. The chapter will end with the use of affine invariants in order to handle severe perspective deformations.

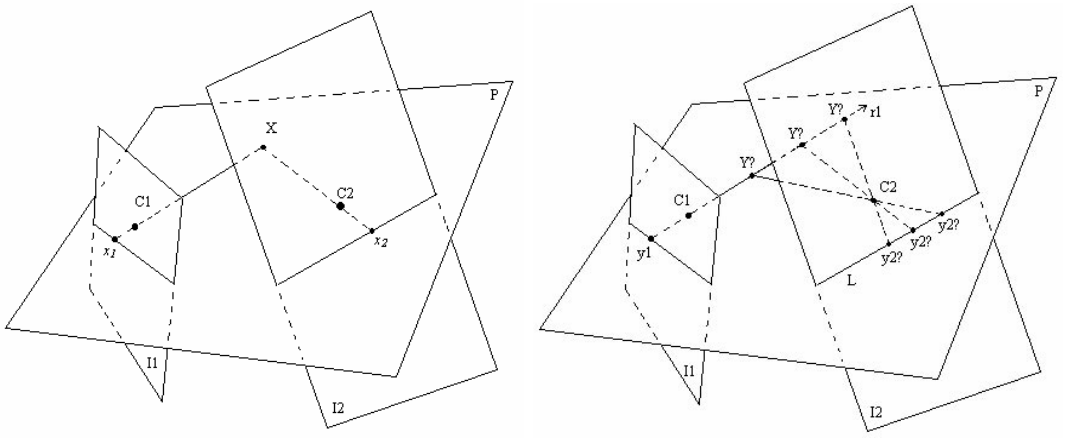
## 4.1 Epipolar Geometry

The geometric relation between the orientations of two cameras (or a single camera capturing the scene at two time instants from different

locations) is the most important constraint for the two sets of interest points on the image pair. In other words, given an interest point in one frame, the location of the corresponding interest point at the other frame is determined by this epipolar geometry. Such a relation simply should hold for all the correct correspondences on the image pair, hence it can be utilized as a consistency (or correctness) measure for each pair of interest points, once the relation is determined.

### 4.1.1 The Fundamental Matrix

Consider the setting in Figure 4.1(a), where a point X in 3-space is imaged in two views, at point $x_1$ in the first and at $x_2$ in the second. The image points $x_1$, $x_2$, the point in space X and the camera centers C1, C2 are observed to be coplanar. This plane, denoted by P, is determined by the




(a) (b)

Figure 4.1 Epipolar geometry. (a) A point X in 3-space is imaged using two cameras, (b) The corresponding pair for the image of a point is restricted to lie on a line in the other imageonce the epipolar geometry is known

line joining the camera centers, called the *baseline* and the ray back projected from $x_1$ (or equivalently the ray back projected from $x_2$). If the positions of C1 and C2, along with the orientations of the image planes I1 and I2 are known, then given the image (in the first image plane) at y1 of

some unknown space-point Y, this observation shall lead to a restriction on the position of the image in the second plane. For this, consider the ray r1 back projected from y (Figure 4.1(b)). Since y2 should lie in the plane P, determined by this ray and the baseline, it is restricted to lie on the intersection of this plane with the second image plane that is, the line L. Equivalently, this line may be thought of as the image of r1 in the second image plane, since the actual position of Y is not known and it may lie anywhere on this ray.

For point matching, the above observation leads to a reduction in the search space. Given a point x1 in the first image, if the camera positions and parameters are known, since the line in the second image on which the match x2 should lie will be known, the search space is restricted to a (1-D) line, instead of the whole (2-D) image. However, given two arbitrary images, without any information about the relative positions and parameters of the camera(s), it is not possible to directly make use of this relation.

A simple mathematical relation is required to be able utilize this important relation in general. For this, consider the homogeneous coordinates ([12]) of a point $\boldsymbol{x}$ in an image as:

$$\boldsymbol{x} = [x_{im},\ y_{im},\ 1]^T \tag{4.1}$$

where $x_{im}$ and $y_{im}$ are the coordinates of the point in the image. Utilizing homogeneous coordinates, a line in the plane given by $(ax + by + c = 0)$ may also be written $[a,\ b,\ c]\cdot x = 0$. Thus $[a,\ b,\ c]$ is taken to be the representation of the line. Whether a given point lies on the line or not is determined by the product of its homogeneous coordinates and the vector describing the line.

The epipolar geometry is expressed by the equation ([12])

$$x_1^T F x_2 = 0 \tag{4.2}$$

where $x_1$, $x_2$, represent the homogeneous coordinates in the first and the second images of a pair of matching points. $F$ is a 3x3, rank-2 matrix, called the *Fundamental matrix*, and is constructed, taking into account the relative positions and the internal parameters of the camera(s). It is noted that both $Fx_2$ and $x_1^T F$ represent a line in the sense described above. In

this regard, the equation $\left(x_2^T F^T\right)x_1 = 0$, states that $x_1$ lies on the line $x_2^T F^T$ in the first image and the equation $\left(x_1^T F\right)x_2 = 0$, states that $x_2$ lies on the line $x_1^T F$ in the second image.

### 4.1.2 Obtaining the Fundamental Matrix

Let $\boldsymbol{x} = [x, y, 1]^T$ and $\boldsymbol{x'} = [x', y', 1]^T$ be a pair of matching points. Then, the epipolar equation

$$\boldsymbol{x}'^T F \boldsymbol{x} = 0 \tag{4.3}$$

can be written as:

$$x'xF_{11} + x'yF_{12} + x'F_{13} + y'xF_{21} + y'yF_{22} + y'F_{23} + x'xF_{11} + xF_{31} + yF_{32} + F_{33} = 0 \tag{4.4}$$

where $F_{ij}$ denotes the entry of $F$ in the $i^{th}$ row, $j^{th}$ column. If further pairs of matching points $\boldsymbol{x_n} = [x_n, y_n, 1]^T$, $\boldsymbol{x_n'} = [x_n', y_n', 1]^T$ are obtained, these will also satisfy the epipolar equation (4.3) assembling these one obtains:

$$\underbrace{\begin{bmatrix} x_1'x_1 & x_1'y_1 & x_1' & y_1'x_1 & y_1'y_1 & y_1' & x_1 & y_1 & 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ x_n'x_n & x_n'y_n & x_1' & y_1'x_1 & y_1'y_1 & y_1' & x_1 & y_1 & 1 \end{bmatrix}}_{D} f = 0 \tag{4.5}$$

where $f = \begin{bmatrix} F_{11} & F_{12} & F_{13} & F_{21} & F_{22} & F_{23} & F_{31} & F_{32} & F_{33} \end{bmatrix}^T$.

Given enough number of corresponding pairs (minimum seven pairs ([12])), the fundamental matrix, thus the epipolar geometry of the images, may be obtained. If the data matrix $D$ in (4.5) is exact (i.e. no noise on the positions of the corresponding point pairs), $F$ can be determined up to scale by calculating the right null-space of $D$. However, when the number of pairs is greater than 8, due to the presence of noise, $D$ may be of rank 9 and for this case (4.5) does not hold. In this case, one should be interested in the least squares solution which may be obtained by Singular Value Decomposition (see Appendix-B) of $D$. Writing $D$ via SVD gives:

$$D = U \cdot S \cdot V^T, \tag{4.6}$$

$f$ is given by the last column of $V$, corresponding to the smallest singular value of $D$. It is noted ([12]) that the fundamental matrix, which is obtained in this way, may not have rank 2, hence this constraint should be

enforced. A practical and convenient way to do this is to again use SVD, this time on $F$:

$$F = T \cdot B \cdot Y, \tag{4.7}$$

where $B = diag(r, s, t)$, and $r \geq s \geq t$. $F$ is then replaced by the rank 2 matrix $F'$, given by:

$$F' = T \cdot B' \cdot Y, \tag{4.8}$$

where $B' = diag(r, s, 0)$. This matrix clearly has rank 2, and it is the unique matrix that minimizes the Euclidian distance between $f$ and $f'$.

The simple method described above is noted to be noise-prone. A small amount of noise may result in an estimate of $F$, far from the original. In order to make the method more robust, a simple normalization on the input data is suggested ([11]). The normalization consists of a translation and scaling of each image separately so that the centroid of the corresponding points in each image is at the origin of the coordinates and the RMS distance of the points from the origin is equal to $\sqrt{2}$. The algorithm may be stated as follows:

Algorithm 4.1:

Given n ≥ 8 corresponding point pairs, $\boldsymbol{x_1}$, ..., $\boldsymbol{x_n}$, $\boldsymbol{x_1'}$, ..., $\boldsymbol{x_n'}$,

1. Transform the image coordinates of the first image so that the centroid of $\hat{x}_i = Tx_i$, i=1,..,n is equal to [0,0,1] and the RMS distance of $\hat{x}_i$'s from the origin is equal to $\sqrt{2}$. Transform the second image so that $\hat{x}_i' = T'x_i'$, i=1,..,n possess similar characteristics.

2. Determine $\hat{F}$ from the singular vector corresponding to the smallest eigenvalue of $\hat{D}$, where $\hat{D}$ is constructed as in (4.5), using $\hat{x}_i$'s and $\hat{x}_i'$'s as the input data.

3. Replace $\hat{F}$ by $\hat{F}'$ as in (4.7) and (4.8) so that $\det \hat{F}' = 0$.

4. Obtain the fundamental matrix F corresponding to the original image coordinates by $F = T'^T \hat{F}' T$.

Regarding the estimation of the fundamental matrix and utilization of this knowledge for the purpose of matching, a problem, resembling the "chicken-egg problem", may be recognized. To be able to perform matching automatically, one should use fundamental matrix to reduce the search space, thus reducing, in effect, the number of possible false matches. However, to obtain the fundamental matrix, one needs to have at hand a number of preferably precise, corresponding pairs of points. A solution might be to obtain first, a number of *putative matches*, i.e. matches that are thought to be correct, but may certainly contain false pairs as well, and then try to fit the scene model outlined for a subset of these putative matches.

### 4.1.3 Error Measures

The fundamental matrix defines a variety (4.2) on the coordinates of the corresponding pairs. However, even for pairs where a small amount of noise is present (concerning the positions), (4.2) is not satisfied. To be able to distinguish these pairs from false matches, a plausible error measure is required.

For a corresponding pair ($m_{1i}$, $m_{2i}$) (where $i$ represents the index of the pair), a reasonable error measure is given by:

$$e(m_{1i}, m_{2i}) = d(m_{2i}, Fm_{1i})^2 + d(m_{1i}, F^T m_{2i})^2 \tag{4.9}$$

$$= \left( \frac{1}{(Fm_{1i})_1^2 + (Fm_{1i})_2^2} + \frac{1}{(F^T m_{2i})_1^2 + (F^T m_{2i})_2^2} \right) (m_{2i}{}^T F m_{1i}) \tag{4.10}$$

where $d(m_{2i}, Fm_{1i})$ represents the Euclidian distance of $m_{2i}$ to the epipolar line in the second image, defined by $m_{1i}$ and $F$, $d(m_{1i}, F^T m_{2i})$ represents the distance of $m_{1i}$ to the epipolar line in the first image, defined by $m_{2i}$ and $F$, $(v)_t$ representing the $t^{th}$ entry for a column vector $v$. The measure is known in the literature as "symmetric epipolar distance" ([41], [12]). Although this algebraic measure is widely used and intuitively reasonable at first sight, what is more interesting would be to calculate the minimum amount of replacement of both points in the pair, so that (4.2) would be satisfied. This value is called the *geometric error*, which may not be obtained in a

closed form in the general case ([12]). Fortunately, the first order approximation for this error, which is called *Sampson distance*, may be derived as follows.

For a given fundamental matrix $F$, a corresponding pair represented as $M = [\, m_{1i,1},\ m_{1i,2},\ m_{2i,1},\ m_{2i,2}\,]^T$, where $m_{ji,k}$ represents the $k^{th}$ coordinate ($k$=1,2) of the point in the $j^{th}$ image ($j$=1,2) from the $i^{th}$ pair, the variety (4.2) may be written as :

$$V_F(M) = 0. \tag{4.11}$$

In case the equality is not satisfied, one should be interested in the smallest (in norm) vector $\delta_M$ so that

$$V_F(M+\delta_M) = 0. \tag{4.12}$$

Assuming that the first order approximation of $V_F$ is sufficiently accurate, it may be written

$$V_F(M+\delta_M) \approx \underbrace{V_F(M)}_{\varepsilon} + \underbrace{\frac{\partial V_F(M)}{\partial M}}_{J}\delta_M \tag{4.13}$$

where

$$\begin{aligned} J &= \left[\frac{\partial V_F}{\partial m_{1i,1}} \quad \frac{\partial V_F}{\partial m_{1i,2}} \quad \frac{\partial V_F}{\partial m_{2i,1}} \quad \frac{\partial V_F}{\partial m_{2i,2}}\right] \\ &= \left[\left(F^T m_2\right)_1 \quad \left(F^T m_2\right)_2 \quad \left(F m_1\right)_1 \quad \left(F m_1\right)_2\right] \end{aligned} \tag{4.14}$$

Investigating (4.12) and (4.13), an equation that $\delta_M$ should satisfy is obtained:

$$J\delta_M = -V_F(M). \tag{4.15}$$

Utilizing Lagrange multipliers method, instead of minimizing $\delta_M$ with this constraint, the critical points of the following expression searched:

$$\delta_M{}^T\delta_M - 2\lambda(J\delta_M + \varepsilon). \tag{4.16}$$

Taking derivative w.r.t. $\delta_M$ and equating to zero one obtains,

$$\delta_M = \lambda J^T. \tag{4.17}$$

Also, equating the derivative w.r.t. $\lambda$ to zero yields the original constraint, i.e.,

$$J\delta_M = -\varepsilon. \tag{4.18}$$

Substituting (4.17) in (4.18),

$$\lambda J J^T = -\varepsilon. \tag{4.19}$$

Solving for λ and substituting the result in (4.17),

$$\delta_M = -J^T(JJ^T)^{-1}\varepsilon. \tag{4.20}$$

Then, the squared norm of $\delta_M$ is found as:

$$\left\|\delta_M\right\|^2 = \frac{\varepsilon^2}{\left(JJ^T\right)} \tag{4.21}$$

Expressed in terms of $F$ and $M$, this is written as:

$$\left\|\delta_M\right\|^2 = \frac{\left(m_{2i}Fm_{1i}\right)^2}{\left(Fm_{1i}\right)_1^2 + \left(Fm_{1i}\right)_2^2 + \left(F^T m_{2i}\right)_1^2 + \left(F^T m_{2i}\right)_2^2} \tag{4.22}$$

Either (4.21) or its square root may be utilized as a first order approximation to the geometric error squared or itself. It is claimed that Sampson distance is slightly superior to the symmetric epipolar distance ([12]).

### 4.1.4 Outlier Rejection Methods

Utilizing the error measures explained in the preceding sections, two popular methods for rejecting the outliers in a set of putative matches will be presented.

#### 4.1.4.1 Random Sample Consensus (RANSAC)

Given $K$>>8 putative matches, consider the following algorithm ([12]) for rejection of the outliers (and possibly a rough estimate of the fundamental matrix) :

Algorithm 4.2 :
Repeat for M times,

1. Select a random sample of 8 correspondences and compute the fundamental matrix $F$, as described in Algorithm 4.1 .
2. Calculate the error $e_i$ by (4.10) or (4.22) for each putative match for the fundamental matrix obtained e.
3. Compute the number of inliers consistent with $F$, that is the number of corresponding pairs for which $e_i < t$, where $t$ is a fixed threshold.

4. Choose $F$ with the largest number of inliers, and reject those pairs which yield $e_i > t$ for this particular $F$.

The number $M$ in Algorithm 4.2 is usually taken so that for a particular execution of the algorithm, the probability that there exists a sample (of eight pairs) consisting of correct matches is satisfactorily high. For instance, assuming that the ratio of the false pairs to the total are $b$, the probability of drawing eight correct matches is given by $(1-b)^8$ . Then, the probability that at least one of the pairs in the sample is false equals $1-(1-b)^8$. If it is desired that the chance that the algorithm selects a totally correct sample is 0.99, the number of samples $M$ required is given by (assuming that there exist an infinite number of pairs) : $M = \frac{\ln(0.01)}{\ln\left(1-(1-b)^8\right)}$.

#### 4.1.4.2 Least Median Squares (LMedS)

Given $K>>8$ putative matches, a similar (in spirit) algorithm may be proposed ([41]) as:

Algorithm 4.3 :

Repeat for M times,

1. Select a random sample of 8 correspondences and compute the fundamental matrix $F$, as described in Algorithm 4.1 .
2. Calculate the error $e_i$ by (4.10) or (4.22) for each putative match for the fundamental matrix obtained previously.
3. Choose $F$ for which the median of the squared residuals, denoted by $M_F$, with respect to the whole set of point correspondences is the minimum.
4. Reject those pairs which yield $e_i > t(M_F)$ for this particular $F$, where $t(M_F)$ is an adaptive threshold determined by $M_F$ ([41]).

It should be noted that the ratio of inliers to the whole set should be greater than 0.5 for this algorithm to succeed, since the median of the errors is taken as a measure in order to select the inliers and it is desired that this value belong to that of a correct match. Also, it is argued that

([41]) the samples chosen should belong to pairs, which are separated significantly in the views and the random selection should be omitted due to this reason. Instead, the authors divide the image into non-overlapping fields and propose a two-stage selection procedure. In the first stage, the field, where the selection will be made, is decided randomly, whereas in the second stage, a point is selected randomly from this field. It is assured that no two selections come from the same field. The number of runs of the algorithm, *M*, may be determined by the same consideration as in RANSAC.

Thus, it is seen that given a number of putative matches, it is possible to reject the outliers. Now, the remaining question is: "How, then, will the putative matches be obtained?". Due to the lack of knowledge of the scene geometry, the search space for a particular interest point in, say the first image, should be the whole second image. In order to restrict the search space, however, it will be first assumed that the difference between the given images is very small. In other words, the images are taken from almost the same position, which may be called as the case of *small-baseline stereo*, which constitutes the simplest case for matching.

## 4.2 Matching by Correlation:

Consider the images in Figure 4.2. It can be easily deduced by looking at the images that the camera displacement between the

(a) (b)

Figure 4.2 Example of a small-baseline image pair.

images is relatively small, thus the term small-baseline stereo. For an interest point (detected using any feature detector), it is expected that the corresponding point in the other image is in a neighborhood of the given interest point. Moreover, the regions surrounding the corresponding pairs are more or less the same, so simply looking at the squared sum of the differences gives a good measure for the purpose of matching. In order to be able to handle affine illumination changes (i.e. changes of the form $x' = ax + b$, where $x$ and $x'$ denote the brightness values for the same point in different images), the brightness values for a region are modified so that the average is zero and variance is equal to one. After this operation, instead of looking at the squared sum of differences, one can consider the correlation of the patches, as given below:

$$C(m_1,m_2)=\frac{\sum_{i=-n}^{n}\sum_{j=-m}^{m}\left[I_1(u_1+i,v_1+j)-\overline{I_1(u_1,v_1)}\right]\times\left[I_2(u_2+i,v_2+j)-\overline{I_2(u_2,v_2)}\right]}{(2n+1)(2m+1)\sqrt{\sigma^2(I_1)\times\sigma^2(I_2)}} \qquad (4.23)$$

where $\overline{I_k(u,v)}$ is the average of the function around $(u,v)$ and $\sigma(I_k)$ is the standard deviation and ( $u_i$ , $v_i$ ) denote the position of the interest point in the $i^{th}$ image.

Thus an algorithm utilizing this measure may be formed:

Algorithm 4.4

1.Extract the interest points in each image using Harris (or any other) feature detector.

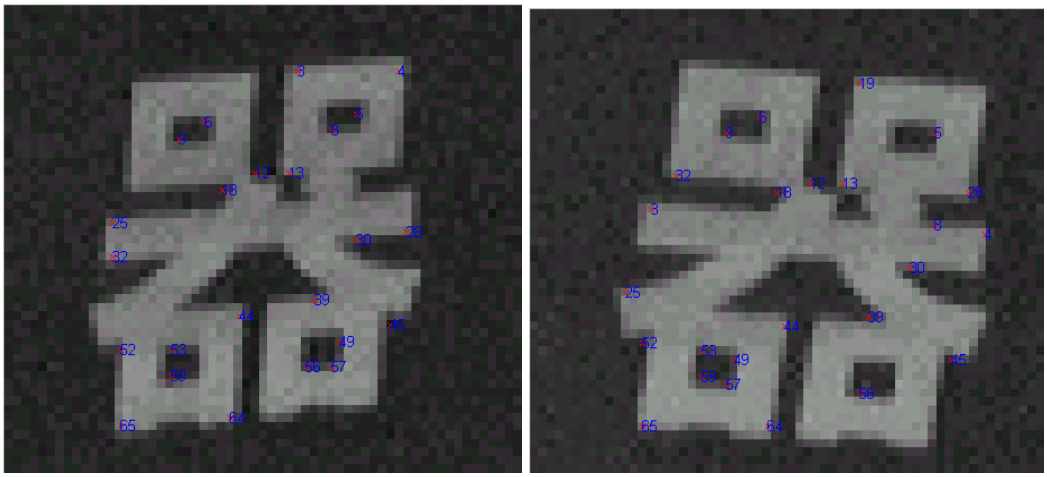
2. For each interest point in each image calculate (4.23) for the nearby interest points in the other image.

3. Let, for an interest point $m_1$ in the first image, $m_2$ be the interest point in the second image, which yields the highest correlation measure (4.23). If $m_1$ is the point, which yields the highest correlation measure for $m_2$, too, the pair is taken to be a putative match (the points then are said to be best matches).


(a) (b)

Figure 4.3 Two small-baseline images, where each interest point in one may be matched to several interest points in the other image, if the correlation score is the sole measure.

The algorithm is very simple but still effective, when the baseline, thus the neighborhood in which the corresponding pair to be sought, is very small, and regions surrounding the interest points are rather distinct. However, slightly widening the baseline, and considering nearby interest points, which are surrounded by patches that look almost the same, the number of matches and in addition, the ratio of the correct matches output by the algorithm significantly decreases. The case is illustrated in Figure 4.3. This behavior of the algorithm is inevitable, since the only



(a) (b)

Figure 4.4 Details of the matching results of the images in Figure 4.3, using Algorithm 4.4. Out of 24 matches found, 16 of them are correct.

measure for obtaining the matches is easily 'fooled' by nearby interest points which all look alike. Some measure taking into account the structure of the collection of points for a neighborhood should be of great use, and actually is, as will be explained in the next section.

## 4.3 Disambiguating Matches

As mentioned above, a point in the first image may be paired to several points in the second image, still yielding high correlation values for all. These are called *candidate matches*. For resolving these ambiguities, a number of techniques exist ([9], [13], [21], [41]). The one that will be explained uses a *neighboring constraint* as follows ([41]) :

"Consider a candidate match $(m_{1i}, m_{2j})$ where $m_{1i}$ is a point in the first and $m_{2j}$ in the second image. Let $N(m_{1i})$ and $N(m_{2j})$ be, respectively, the neighbors of $m_{1i}$ and $m_{2j}$ within a disc of radius *R*. If $(m_{1i}, m_{2j})$ is a good match, many other matches $(n_{1k}, n_{2l})$ exist, where $n_{1k} \in N(m_{1i})$ and $n_{2l} \in N(m_{2j})$, such that the position of $n_{1k}$ relative to $m_{1i}$ is similar to that of $n_{2l}$ relative to $m_{2j}$. On the other hand, if $(m_{1i}, m_{2j})$ is a bad match, it is expected to see only a few matches or even not any at all, in their neighborhood."

The idea is realized by defining a 'strength' for each match, and then choosing strong candidate matches in this respect. The strength for a matching pair $(m_{1i}, m_{2j})$ is given by:

$$S(m_{1i}, m_{2j}) = c_{ij} \sum_{n_{1k} \in N(m_{1i})} \left[ \max_{n_{2l} \in N(m_{2j})} \frac{c_{kl}\, \delta(m_{1i}, m_{2j}; n_{1k}, n_{2l})}{1 + \left(d(m_{1i}, n_{1k}) + d(m_{2j}, n_{2l})\right)/2} \right], \qquad (4.24)$$

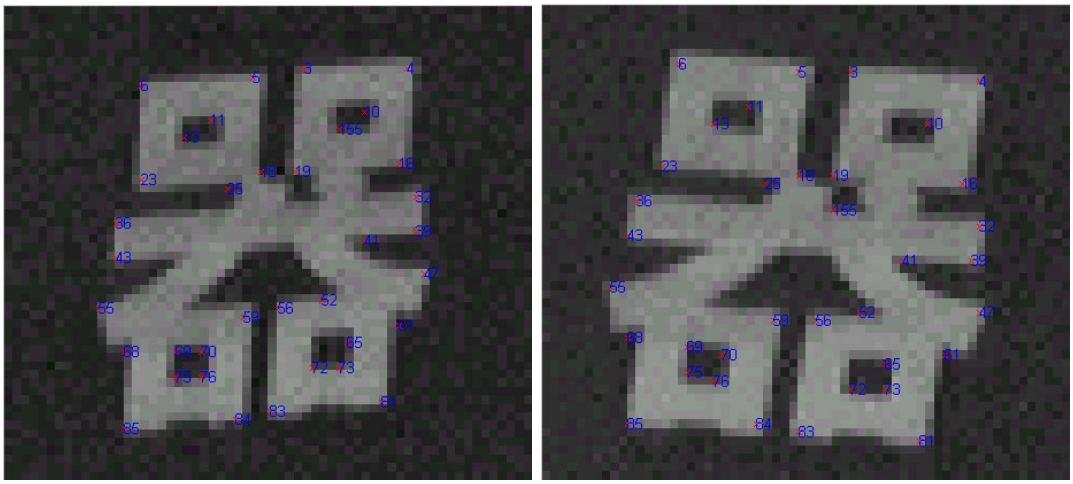
$d(m,n)$ denoting the Euclidian distance between the points $m$ and $n$, and

$$\delta(m_{1i}, m_{2j}; n_{1k}, n_{2l}) = \begin{cases} e^{-r/\varepsilon_r} & \text{if } (n_{1k}, n_{2l}) \text{ is a candidate match and } r < \varepsilon_r \\ 0 & \text{otherwise} \end{cases} \qquad (4.25)$$

where $r$, the relative distance, is defined as:

$$r = \frac{\left| d\left(m_{1i}, n_{1k}\right) - d\left(m_{2j}, n_{2l}\right) \right|}{\left( d\left(m_{1i}, n_{1k}\right) + d\left(m_{2j}, n_{2l}\right) \right) / 2}, \tag{4.26}$$

and $\varepsilon_r$ is a threshold on the relative distance difference. $c_{kl}$ and $c_{ij}$ denote the goodness of the candidate matches $\left(n_{1k}, n_{2l}\right)$ and $\left(m_{1i}, m_{2j}\right)$, respectively, which may be taken as the correlation scores.

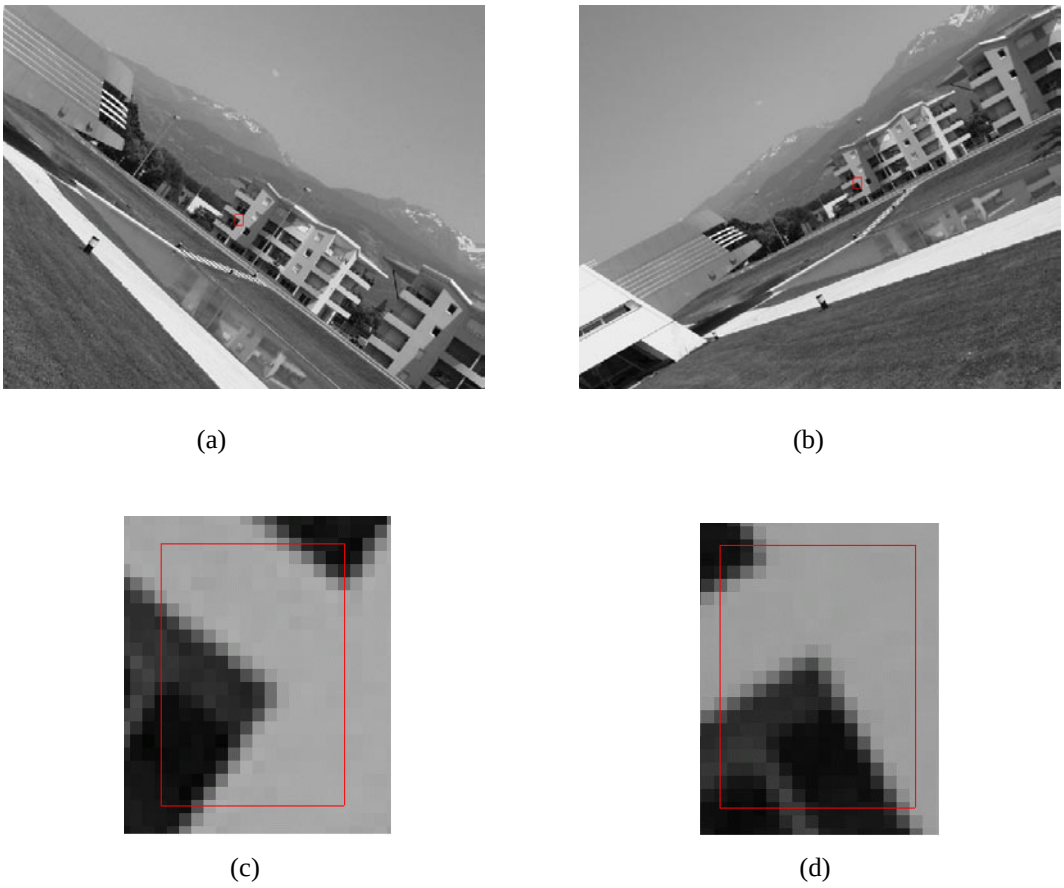


(a) (b)

Figure 4.5 Results of matching while disambiguating high correlation yielding pairs. Out of the 36 matches found in between two images, 35 are correct.

After calculating the strength for each candidate match (obtained by thresholding the correlation score) for both images, the putative matches are obtained by a relaxation procedure, which may be named as "winner-take-all". Following the calculation of the strengths for each candidate match, if for a candidate match $\left(m_{1i}, m_{2j}\right)$, the strongest match for $m_{1i}$ (in the first image) is $m_{2j}$ (in the second image), and the strongest match for $m_{2j}$ (in the second image) is $m_{1i}$ (in the first image), the pair is chosen as a correct match. Since a correct match will be unique, all other matches associated with these two points are eliminated and are no more taken into consideration. After all such pairs are found, the procedure is iterated, i.e., the strength values for the remaining candidate matches are calculated

once again using this updated candidate match list, and new putative matches are sought satisfying the mentioned criterion. The procedure is iterated until no more changes occur. In fact, this strategy is claimed to be "too" fast that it gets stuck at a local minimum (where the cost function is defined to be the sum of the strengths of all the candidate matches) ([41]). Other alternatives are also possible, but this scheme yields satisfactory results.



(a) (b)

(c) (d)

Figure 4.6 Example of a rotated image pair. (a),(b) Images of the same scene through rotated cameras, (c), (d) details of the windows in (a) and (b) respectively. Any measure based on correlation should simply be abandoned in this case.

Although the mentioned scheme was proposed to be used in the wide-baseline case, a problem is faced, when there exists a significant amount of rotation among the given images (e.g. Figure 4.6). In this case,

the use of normalized correlation is misleading and should be abandoned ([20]).

## 4.4 Rotation Invariant Matching

If there exists a rotation between the two views, a comparison method is required, which would be indifferent to this rotation. This is actually achieved by utilizing the rotational invariants, which are explained in Section 3.2.1. It is also necessary to extract the interest points with a rotationally invariant detector. The extraction process being invariant means that, the same points will be obtained for both the original image and the rotated counterpart. This is obviously necessary for matching, since matching is performed between the interest points of each image. In this regard, Harris feature detector is a promising rotationally invariant detector ([32]). After this stage, the local region, surrounding each interest point in the two images, is described by using rotational invariants and a comparison is made between these vectors, according to the Mahalanobis distance explained in Section 3.2.1. The decision that a correspondence is detected is taken when the vectors are sufficiently similar. Thus, the following algorithm may be proposed:

Algorithm 4.5:
1. Extract the interest points in both images using a rotationally invariant detector(e.g. Harris).
2. For each interest point, obtain the local characterization of the point using one type of the rotational invariants, introduced in Section 3.2.1 .
3. Compare the vectors obtained in one image with those of the other image (utilising Mahalanobis distance, if necessary, which is the case, if orthonormal filters are not used).
4. Obtain the best matches (refer to Algorithm 4.4 for the meaning of ‘best match’).
5. Eliminate the false matches using RANSAC or LMedS.

It is noted that steps 2 and 3 may be replaced by those of the method described in Section 4.3 . Lastly, the vector depending on the derivatives steered in the direction of the Gaussian are also given as:

$$\begin{bmatrix} f_{xx}\left(\bar{f}_x^2\right)+2f_{xy}\left(\bar{f}_x\bar{f}_y\right)+f_{yy}\left(\bar{f}_y^2\right) \\ -f_{xx}\left(\bar{f}_x\bar{f}_y\right)+f_{xy}\left(\bar{f}_x^2-\bar{f}_y^2\right)+f_{yy}\left(\bar{f}_x\bar{f}_y\right) \\ f_{xx}\left(\bar{f}_y^2\right)-2f_{xy}\left(\bar{f}_x\bar{f}_y\right)+f_{yy}\left(\bar{f}_x^2\right) \\ f_{xxx}\left(\bar{f}_x^3\right)+3f_{xxy}\left(\bar{f}_x^2\bar{f}_y\right)+3f_{xyy}\left(\bar{f}_x\bar{f}_y^2\right)+f_{yyy}\left(\bar{f}_y^3\right) \\ -f_{xxx}\left(\bar{f}_x^2\bar{f}_y\right)+f_{xxy}\left(\bar{f}_x^3-2\bar{f}_x\bar{f}_y^2\right)+f_{xyy}\left(2\bar{f}_x^2\bar{f}_y-\bar{f}_y^3\right)+f_{yyy}\left(\bar{f}_x\bar{f}_y^2\right) \\ f_{xxx}\left(\bar{f}_x\bar{f}_y^2\right)+f_{xxy}\left(\bar{f}_y^3-2\bar{f}_x^2\bar{f}_y\right)+f_{xyy}\left(\bar{f}_x^3-2\bar{f}_x\bar{f}_y^2\right)+f_{yyy}\left(\bar{f}_x^2\bar{f}_y\right) \\ -f_{xxx}\left(\bar{f}_y^3\right)+3f_{xxy}\left(\bar{f}_x\bar{f}_y^2\right)-3f_{xyy}\left(\bar{f}_x^2\bar{f}_y\right)+f_{yyy}\left(\bar{f}_x^3\right) \\ f_{xxxx}\left(\bar{f}_x^4\right)+4f_{xxxy}\left(\bar{f}_x^3\bar{f}_y\right)+6f_{xxyy}\left(\bar{f}_x^2\bar{f}_y^2\right)+4f_{xyyy}\left(\bar{f}_x\bar{f}_y^3\right)+f_{yyyy}\left(\bar{f}_y^4\right) \\ -f_{xxxx}\left(\bar{f}_x^3\bar{f}_y\right)+f_{xxxy}\left(\bar{f}_x^4-3\bar{f}_x^2\bar{f}_y^2\right)+3f_{xxyy}\left(\bar{f}_x^3\bar{f}_y-\bar{f}_x\bar{f}_y^3\right)+f_{xyyy}\left(3\bar{f}_x^2\bar{f}_y^2-\bar{f}_y^4\right)+f_{yyyy}\left(\bar{f}_x\bar{f}_y^3\right) \\ f_{xxxx}\left(\bar{f}_x^2\bar{f}_y^2\right)+2f_{xxxy}\left(\bar{f}_x\bar{f}_y^3-\bar{f}_x^3\bar{f}_y\right)+f_{xxyy}\left(\bar{f}_x^4-4\bar{f}_x^2\bar{f}_y^2+\bar{f}_y^4\right)+2f_{xyyy}\left(\bar{f}_x^3\bar{f}_y-\bar{f}_x\bar{f}_y^3\right)+f_{yyyy}\left(\bar{f}_x^2\bar{f}_y^2\right) \\ -f_{xxxx}\left(\bar{f}_x\bar{f}_y^3\right)+f_{xxxy}\left(3\bar{f}_x^2\bar{f}_y^2-\bar{f}_y^4\right)+3f_{xxyy}\left(\bar{f}_x\bar{f}_y^3-\bar{f}_x^3\bar{f}_y\right)+f_{xyyy}\left(\bar{f}_x^4-3\bar{f}_x^2\bar{f}_y^2\right)+f_{yyyy}\left(\bar{f}_x^3\bar{f}_y\right) \\ f_{xxxx}\left(\bar{f}_y^4\right)-4f_{xxxy}\left(\bar{f}_x\bar{f}_y^3\right)+6f_{xxyy}\left(\bar{f}_x^2\bar{f}_y^2\right)-4f_{xyyy}\left(\bar{f}_x^3\bar{f}_y\right)+f_{yyyy}\left(\bar{f}_x^4\right) \end{bmatrix} \quad (4.13)$$

where $\begin{bmatrix}\bar{f}_x & \bar{f}_y\end{bmatrix}$ denotes the unit vector in the direction of the gradient.

The derivatives above are actually obtained by using the derivatives of Gaussians. If Gaussians of different scales are used, a rough covering of the frequency plane may be achieved (see Figure 3.6), which also enables reconstruction from these vectors ([15]). Thus, it is reasonable to utilize derivatives of Gaussians of varying scale to characterize the local region around the interest point, which makes a clearer distinction possible, improving the results of the comparison. This in turn increases the size of the vector to be used for description (multiplying the original vector size with $n$ where $n$ is the number of distinct scales).

## 4.5 Scale Invariant Matching

The need for scale invariant matching arises when cameras of different focal length are used or when the distance to the scene is significantly different, as is the case in Figure 4.7. The problem is handled by utilizing a scale invariant interest point detection scheme, as outlined in Section 3.3.2 . Such an approach, combined with the rotational invariant scheme results in a scale and rotation invariant matching method. The rotational invariants must be properly normalized.

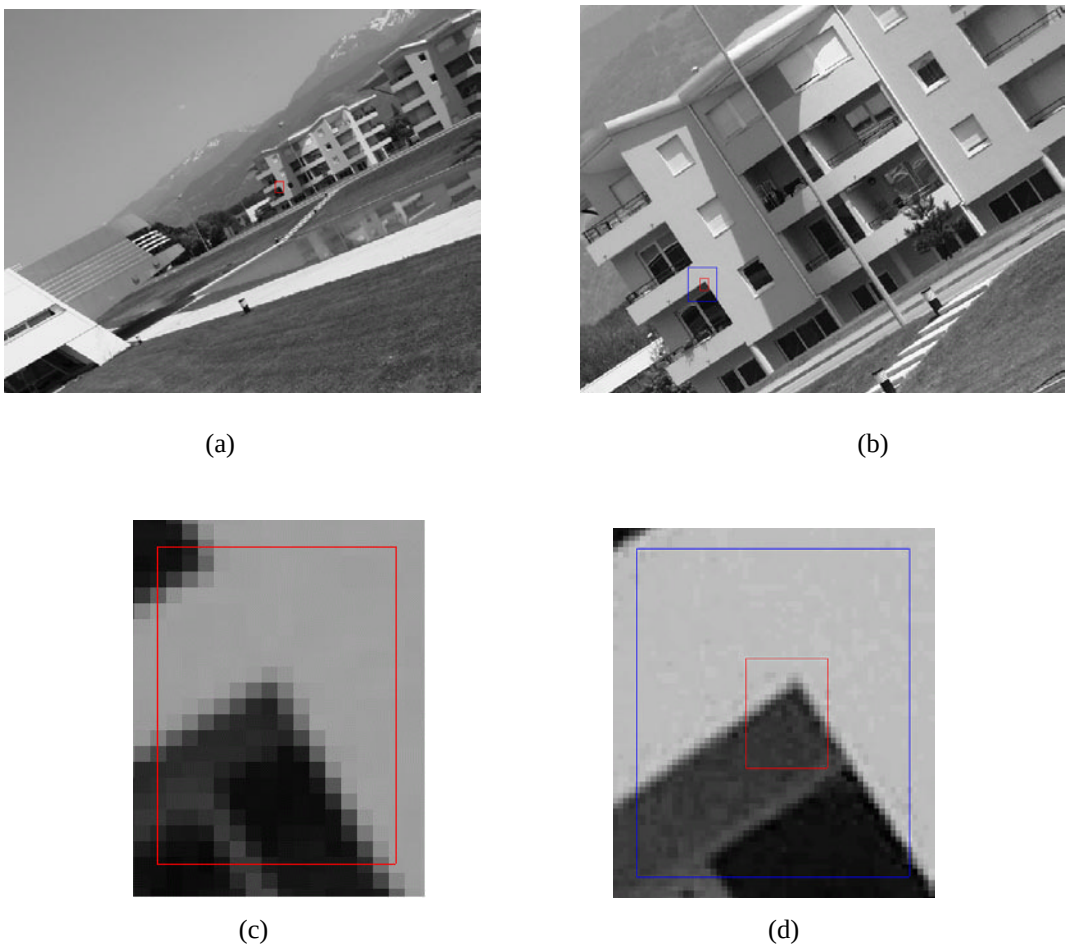


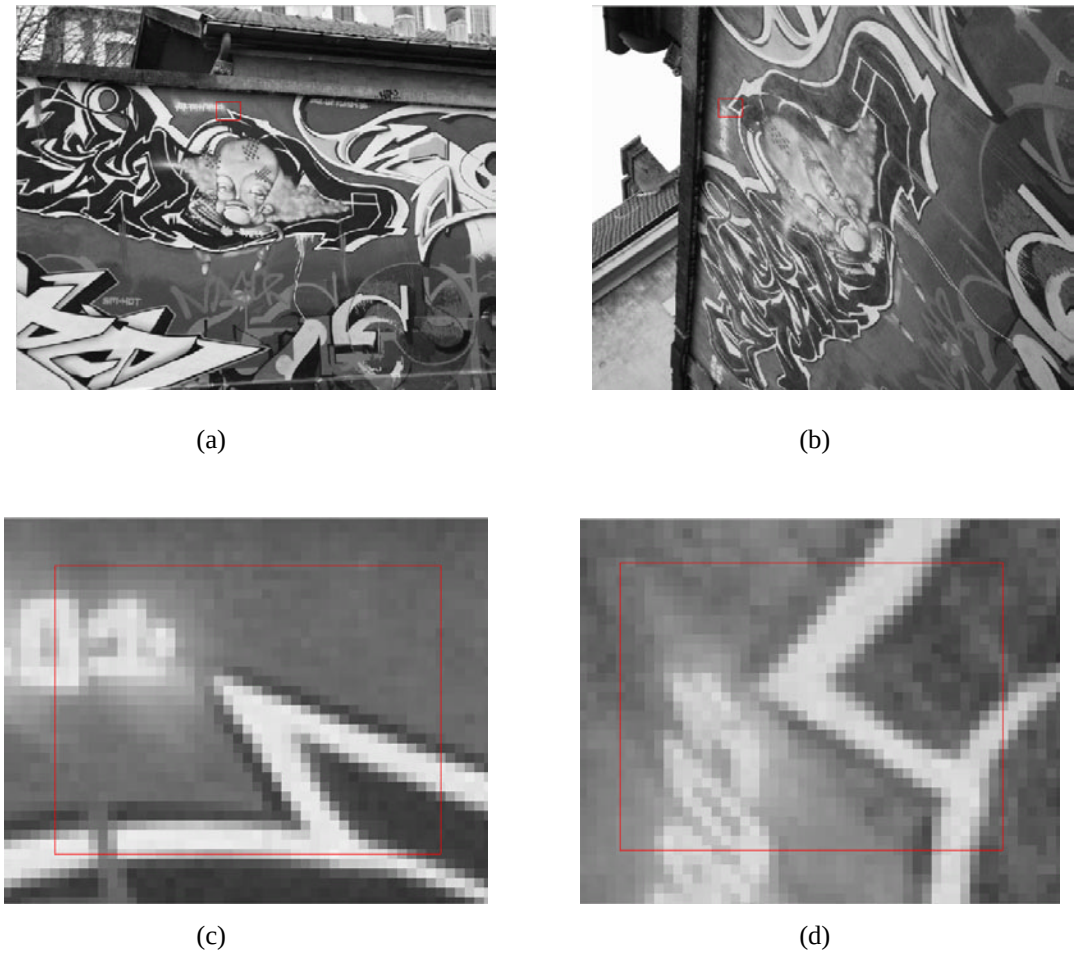
Figure 4.7 Example of images at different scales. (a),(b) Images, (c),(d) details of the windows in (a) and (b), respectively. The sizes of the red windows are equal in pixels, however, comparison should really be made between the red window in (c) and the blue window in (d).

For the Gaussian derivatives, this is achieved by using normalized derivatives as outlined in Section 3.1.4, which fit well into the scale-space framework and are preferred for this reason. However, it is also possible to render the other rotational invariants, invariant to scale by a proper normalization ([4]).

## 4.6 Affine Invariant Matching

For planar surfaces, the most general distortion, namely *perspective distortion*, due to possible changes of the viewpoint, is approximated locally by an affine transformation. This necessitates invariance to possible skew and stretch effects, which the rotational invariants may not handle (Figure

4.8). Thus, the requirement for the use of an affine invariant scheme, as outlined in Section 3.2.3. The scheme may be implemented, as is explained in Algorithm 3.3. It may be preferable to utilize RANSAC or LMedS to eliminate possible false matches.



(a) (b)

(c) (d)

Figure 4.8 Example of a pair of images from significantly different points of view.. (a),(b) Images, (c),(d) details of the windows in (a) and (b) respectively. A simple rotation is not sufficient to transform (d) into (c).

Apart from this method, different schemes have also been proposed in the literature ([24], [28], [37]). In [28], the authors look for corresponding paralellograms and estimate local homographies relating the corresponding structures. Once this is done, this knowledge enables one to obtain better affinity measures for nearby interest regions. The drawback

is noted to be the need for suitable structures to be present in the views in order to obtain the homographies ([4]).

Similar in spirit, Tuytelaars et al. ([37]) search for closed contours on planar surfaces which would be affine transformed if imaged from a different perspective. After finding these regions, ellipses are fit to these contours and transforming these ellipses to circles, and using rotational invariants, matching is made possible under perspective deformations. The method for extraction of the ellipses is noted to find points not on the edges and thus is claimed to be better suited for finding planar points.

Matas et al. ([24]) look for regions closed under contionuous perspective deformations. Once these are located, the regions are described using affine invariants and matching is performed for different views. Both [24] and [37] are noted for their method of extraction of regions which are scale invariant.

A unifying approach would be to probably use all of these invariants in one setting ([29]). In addition, the consistency of the corresponding features may be further checked by backtransforming each region to some unit structure and thresholding the cross-correlation ([37], [29]).

## 4.7 Discussion

In this chapter, as the sections proceeded, the need for higher degrees of invariance is presented. Despite their usefulness in such settings, as shown in the various figures in the chapter, these invariants all introduce a lack of describing the local patch that they are trying to describe with increasing degree of invariance. This insufficiency often shows itself when vectors corresponding to radically different patches are seen to be similar (in terms of the distance used). This problem is mostly tried to be overcome by using more descriptors. This means that for a scale-space derivative based description scheme, derivatives from a few scales should be utilized. Following this observation, the preferred approach for the highest performance, should be to put also the knowledge about the images into consideration, when choosing the method to be utilized. In other words, if it is known that a certain degree of invariance is

sufficient to handle the images at hand, one should not be tempted to use the most sophisticated invariants.

# CHAPTER V

# RESULTS

In this chapter, the results of the algorithms for obtaining corners, interest points, as well as matching these points, will be presented along with the values of the parameters that each particular algorithm possesses. After presenting the results, a conclusion will be drawn, based on the observations made on these results and an algorithm will be outlined for use in the most arbitrary case of interest point extraction and matching. This proposition will probably be recognised to be a compilation of the algorithms explained in the previous chapters that might suit the needs of such an arbitrary case.

In this respect, the first few sections will contain the results of the corner or interest point extraction methods. Following these results, the performances of the matching algorithms are presented. Lastly, the chapter is concluded with an algorithm intended to be able to extract and match interest points on images with a wide baseline.

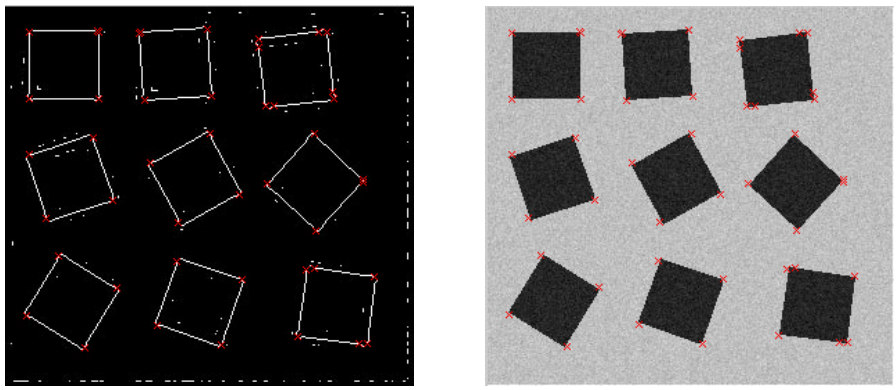
## 5.1 Simulation Results for Interest Point Detectors

In the following subsections, the performances for the corner and interest point detectors, which are explained in Chapter II and Chapter III, on a set of images are presented.

### 5.1.1 Corners as Local Maxima of Curvature Along Edges [Section 2.4]

In order to calculate the curvature, $k$-curvatures up to $k = 4$ are utilized. Each particular $k$-curvature is given the same weight (i.e. k-

curvatures with lower $k$ values are not emphasized, which might have been done otherwise) and added to compute the curvature at an edge point. The threshold for eliminating low curvature points is set equal to half of the maximum of the curvature values obtained in this way.



(a) (b)

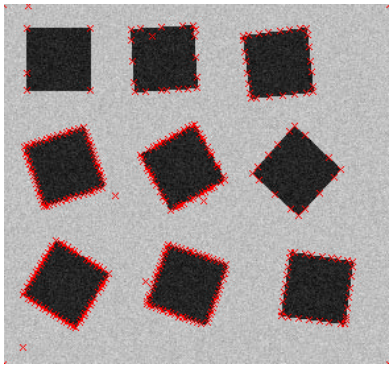
Figure 5.1 Results for high-curvature point detection scheme on 'Kareler'. (a)Points on the edge map obtained by LoG edge detector, (b) high curvature points imposed on the original 'kareler' image.

(a) (b)

Figure 5.2 Results for high-curvature point detection scheme on 'Goldhill' (a) Points on the edge map obtained by LoG edge detector, (b) high curvature points imposed on the original 'Goldhill' image.

### 5.1.2 Kitchen-Rosenfeld Cornerness Measure [Section 2.5]

The magnitude of the gradients is calculated first and the cornerness measure is computed at points, where the gradient magnitude is among the highest %15. According to this measure, only the points in the highest %0.5 are taken to be corners.



(a)

(b)

Figure 5.3 Results for Kitchen-Rosenfeld measure. (a) 'Kareler', (b)'Goldhill'.

### 5.1.3 Zuniga-Haralick Method [Section 2.6]

For each pixel in the image, the surrounding 7x7 region is approximated by a bicubic polynomial. The pixels for which the magnitude of the gradient estimated by $\sqrt{k_2^2 + k_3^2}$ lies in the highest %10 are further considered as edge candidates and on these points, the criterion for being an edge point is checked. A pixel is taken as an edge point, if the third directional derivative in the direction of the gradient, given by,

$$6\left(k_7 \cos^3\alpha + k_8 \cos^2\alpha \sin\alpha + k_9 \cos\alpha \sin^2\alpha + k_{10} \sin^3\alpha\right) \tag{5.1}$$

is negative, the second derivative taken in the direction of the gradient, given by,

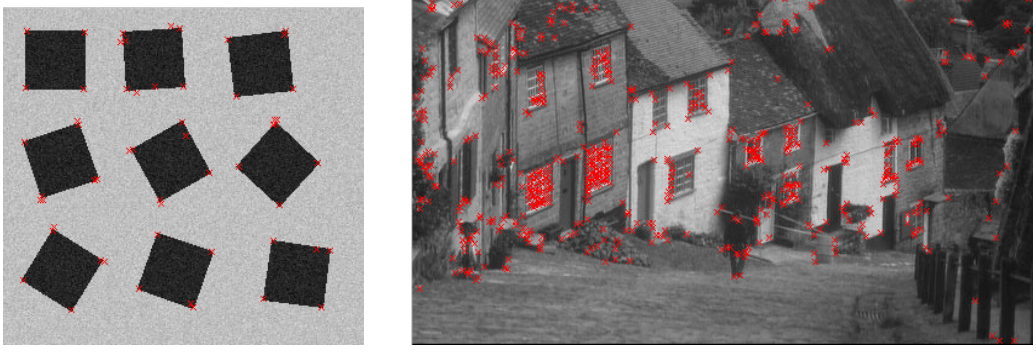
$$\begin{aligned} &6\left(k_7 \cos^3\alpha + k_8 \cos^2\alpha \sin\alpha + k_9 \cos\alpha \sin^2\alpha + k_{10} \sin^3\alpha\right)h + \\ &\left(k_4 \cos^2\alpha + k_5 \sin\alpha \cos\alpha + k_6 \sin^2\alpha\right) \end{aligned} \tag{5.2}$$

is zero for some $h$ such that $|h| < ½$ (the pixel is taken as circular with radius equal to ½) and the first directional derivative at this 'zero second derivative yielding value' of $h$, given by,

$$\begin{aligned} &\left(k_2 \cos\alpha + k_3 \sin\alpha\right) + 2\left(k_4 \cos^2\alpha + k_5 \sin\alpha \cos\alpha + k_6 \sin^2\alpha\right)h + \\ &3\left(k_7 \cos^3\alpha + k_8 \cos^2\alpha \sin\alpha + k_9 \cos\alpha \sin^2\alpha + k_{10} \sin^3\alpha\right)h^2 \end{aligned} \tag{5.3}$$

is non-zero.

Then for the edge pixels, the measure, given by (2.20) is calculated and the pixels for which the absolute value of this measure is greater than 16/180 are decided to be corner points. The outcomes are further refined by suppressing non-maxima.

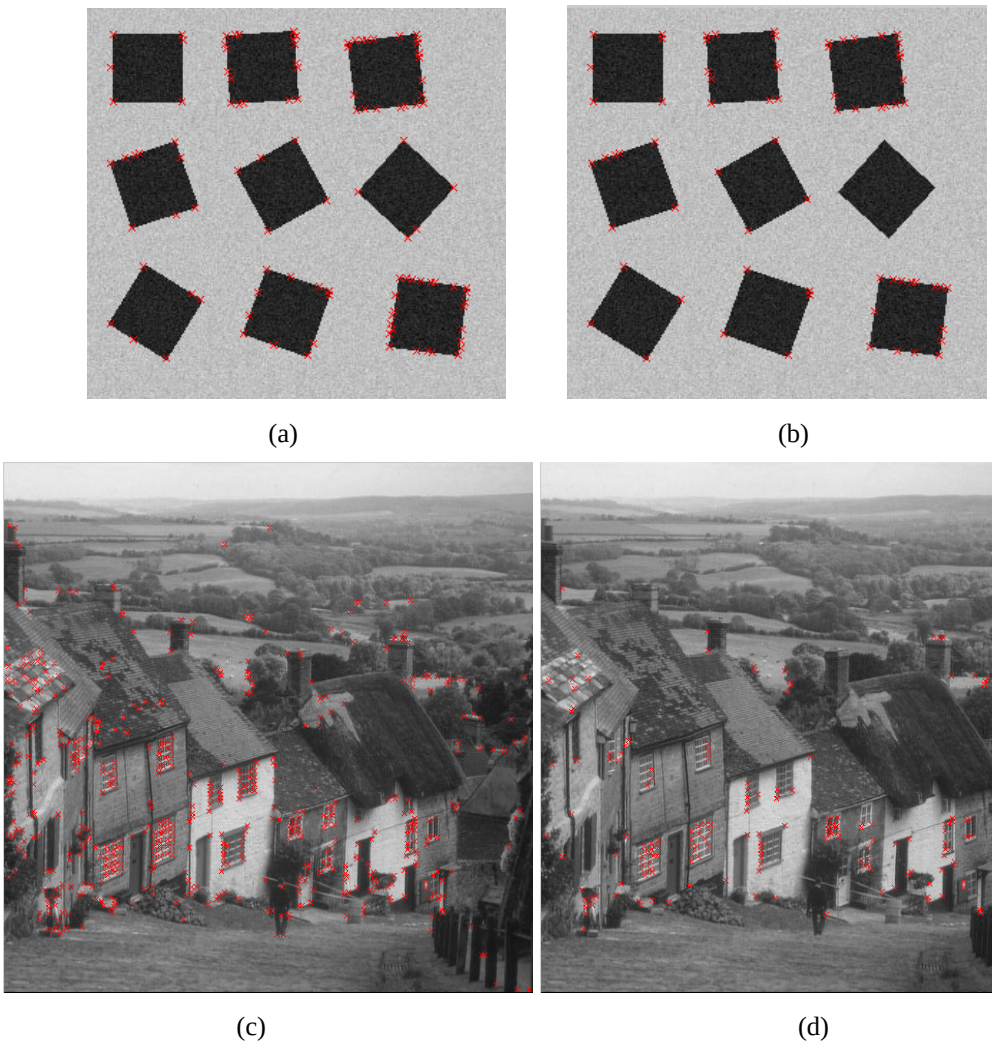


(a) (b)

Figure 5.4 Results for Zuniga-Haralick method. (a) 'Kareler', (b)'Goldhill'.

### 5.1.4 Harris-Stephens Corner Detector [Section 2.7]

In order to compute the required derivatives in the $x$ and $y$ directions, Sobel operator is used. The variance of the Gaussian function, which is used to sum up squared gradients and the threshold of the cornerness measure for selecting the corners are the parameters to be adjusted. It is observed that the performance of the detector relies heavily on these two parameters and by manual adjustment of these variables, significantly different results can be obtained.



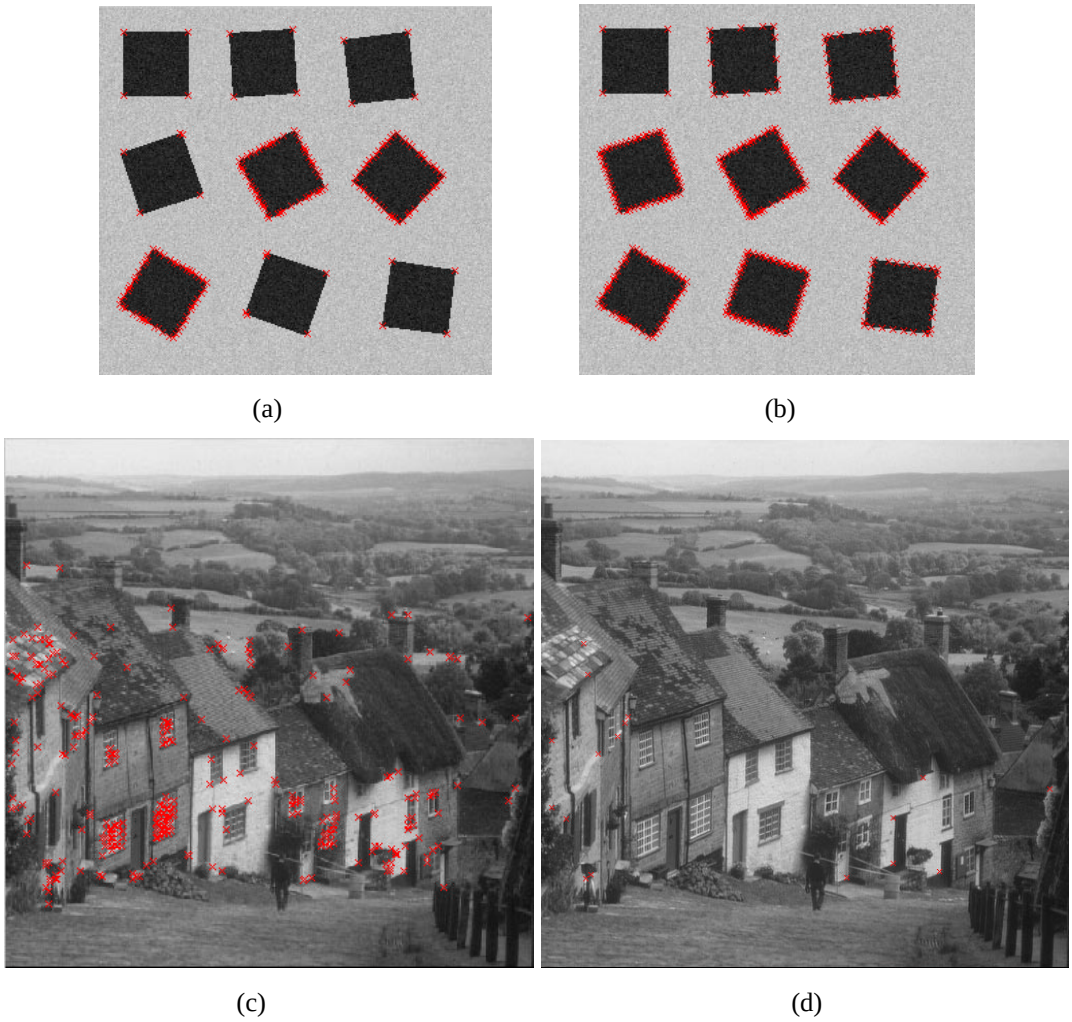
(a) (b)

(c) (d)

Figure 5.5 Corners extracted using Harris-Stephens corner detector (a) variance of the Gaussian = 0.9, threshold=5e7, (b)variance of the Gaussian = 1, threshold =5e7, (c) variance of the Gaussian = 1, threshold =1e7, (d) variance of the Gaussian = 1, threshold =5e7.

### 5.1.5 Fast Corner Detection [Section 2.8]

In this method, first of all, the image is filtered using a Gaussian filter, then the corner response function (CRF) (taking into account the horizontal and vertical directions) is computed at each point. If the CRF is



(a) (b)

(c) (d)

Figure 5.6 Corners detected with the Fast Corner Detection scheme. (a),(d) variance of the Gaussian filter = 1, r1=0.4, r2=0.2, (b),(c) variance of the Gaussian filter = 1, r1=0.1, r2=0.1 .

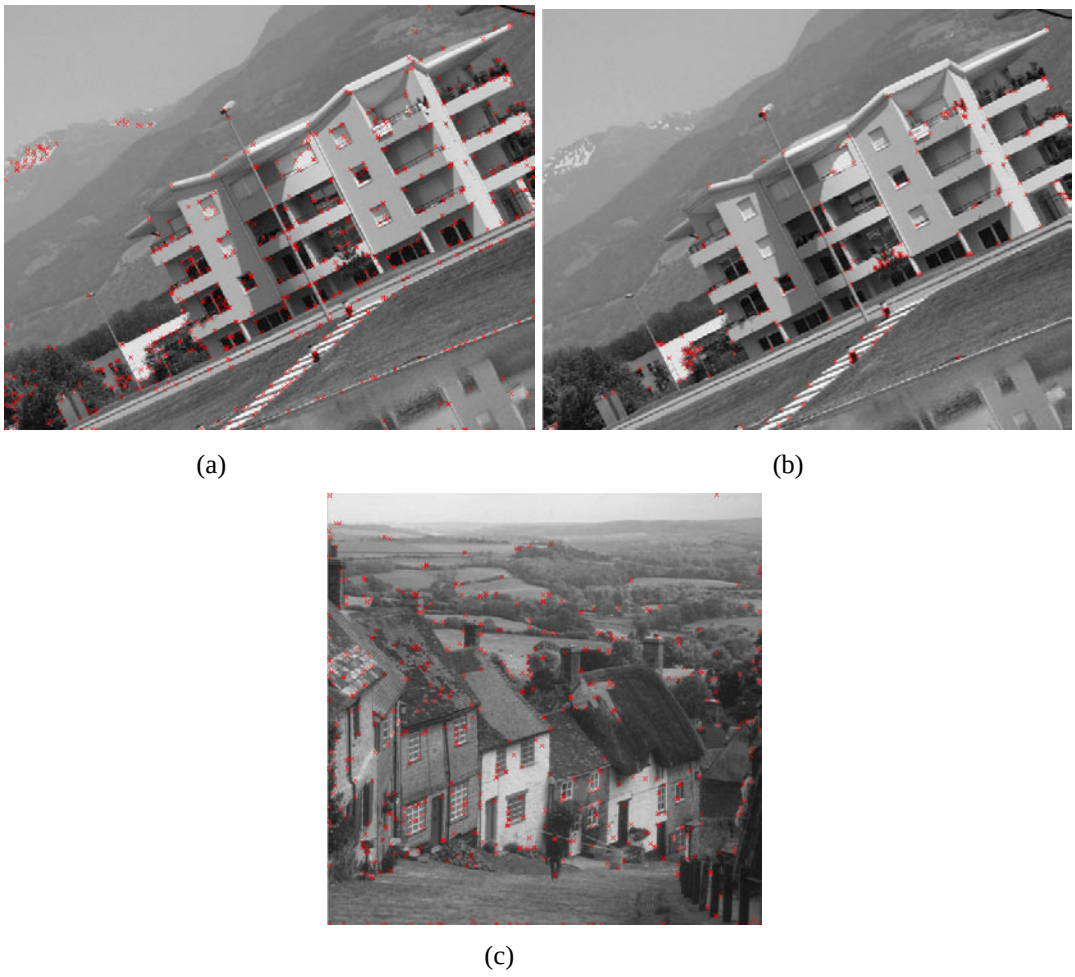
greater than a ratio (e.g. r1) of the maximum CRF obtained, the 3x3 neighborhood of these points are considered as candidate corners. This step is necessary to reduce the number of false corners due to the

presence of noise. Then, CRF (taking into account the horizontal and vertical directions) on the original image is computed at these candidate corners and is again thresholded by some ratio (e.g. r2) of the maximum CRF computed on the original image. The sizes of the windows used are 3x3, i.e., the smallest possible. Following this step, at points, where CRF exceeds the threshold, CRF is updated, taking into account any direction by the help of linear interpixel approximation. The points, at which this new CRF is above the previous threshold, are selected as the corner points, if CRF at the point is a local maximum. The performance of this detector is also dependent on the values of the three parameters, namely the Gaussian filter variance and the two ratios r1 and r2.

### 5.1.6 Scale Invariant Interest Point Detector [Algorithm 3.1]

As it is observed in the presented results under different values for the specific parameters that each detector possesses, it is concluded that the performance of each detector relies heavily on the choice of the parameters. However, the requirement to adjust the parameters for different images indicates that there is no "optimal" choice for these parameters. What should be done instead may be to set these parameters in an adaptive way, but then comes the problem of finding a suitable error function so that any adaptation may take place. A trivial proposal might be related to the number of interest points. This approach requires that the approximate number of interest points are known beforehand. Then, the threshold may be adapted by driving the number of the interest points near this approximate number. An error function, which might be constructed in this sense, may succeed, when the scene is more or less known. However, in a general case, as the image may be that of a bright sky or of a brick wall, the number of interest points may vary significantly. Thus, the use of such a scheme may not help. The concept of scale space offers an adaptive way to handle at least one of these parameters, that is of scale, as outlined in the introduction of Chapter III.

For the scale invariant detector, 17 different scales (ref. (3.68)) are considered and the integration scale at the $n^{th}$ scale, $s_{i,n}$ is set to $1.5\times1.2^n$ where $n$



(a) (b)

(c)

Figure 5.7 Interest points detected by (a), (c) scale invariant detector, (b) Harris-Stephens corner detector. Note in (a) the diffuse features detected in the reflection.

ranges from 0 to 16. The derivation scale $s_{d,n}$ is given by $s_{d,n} = 0.65\times s_{i,n}$ . The threshold for the Harris response at each scale is set to 1500.

## 5.2 Matching Results

In the subsections that follow, the performances of the different matching schemes outlined in Chapter IV are presented.

### 5.2.1 Matching by Correlation [Section 4.2]

A 7x7 window is used for correlation and the threshold is set to 0.8 . For a point in an image, 20x20 neighborhood of the point (i.e. a 41x41 area, at the center of which lies the point) is sought in the other image.

Figure 5.8 Results of matching by correlation. Optical flow vectors are superimposed on the first image. There are 161 matches, numerous false.

### 5.2.2 Matching with Disambiguating Matches [Section 4.3]

The neighborhood of a point is a square with size equal to 0.125 the size of the image and $\varepsilon_r$ is taken as 0.3 . Two points are considered candidate matches (i.e. matches for which strength will be calculated), if their correlation score (obtained using 3x3 windows) is above 0.8 . As in the previous algorithm, for a point in an image, 20x20 neighborhood of the point is sought in the other image.

Figure 5.9 Result of matching with disambiguating matches. There are 175 matches and the number of false matches are decreased significantly.

### 5.2.3 Matching Using Rotational Invariants [Section 4.4]

The interest points are detected using Harris-Stephens detector as explained in Section 5.1.4 . Experiments are carried out using the rotational invariants based on transforms (Section 3.2.1.2) and Gaussian derivatives (Section 3.2.1.1). The results using the second type will be presented in Section 5.2.4 along with scale invariance. For the invariants based on transforms, the variance of the Gaussian is taken to be 7, $m$ = 0, 1, ... 5 ; $n$ = 0, 1, 2, 3 . Invariants are obtained by first dividing the group for which m = constant by the magnitude of the element belonging to the group for which n=0; and then taking the real part of these complex numbers. After this process, since the elements with n=0 will be equal to 1 (in case they were not equal to 0 in the beginning), these elements are not further considered, thus 18 invariants are obtained for each region containing the interest point.

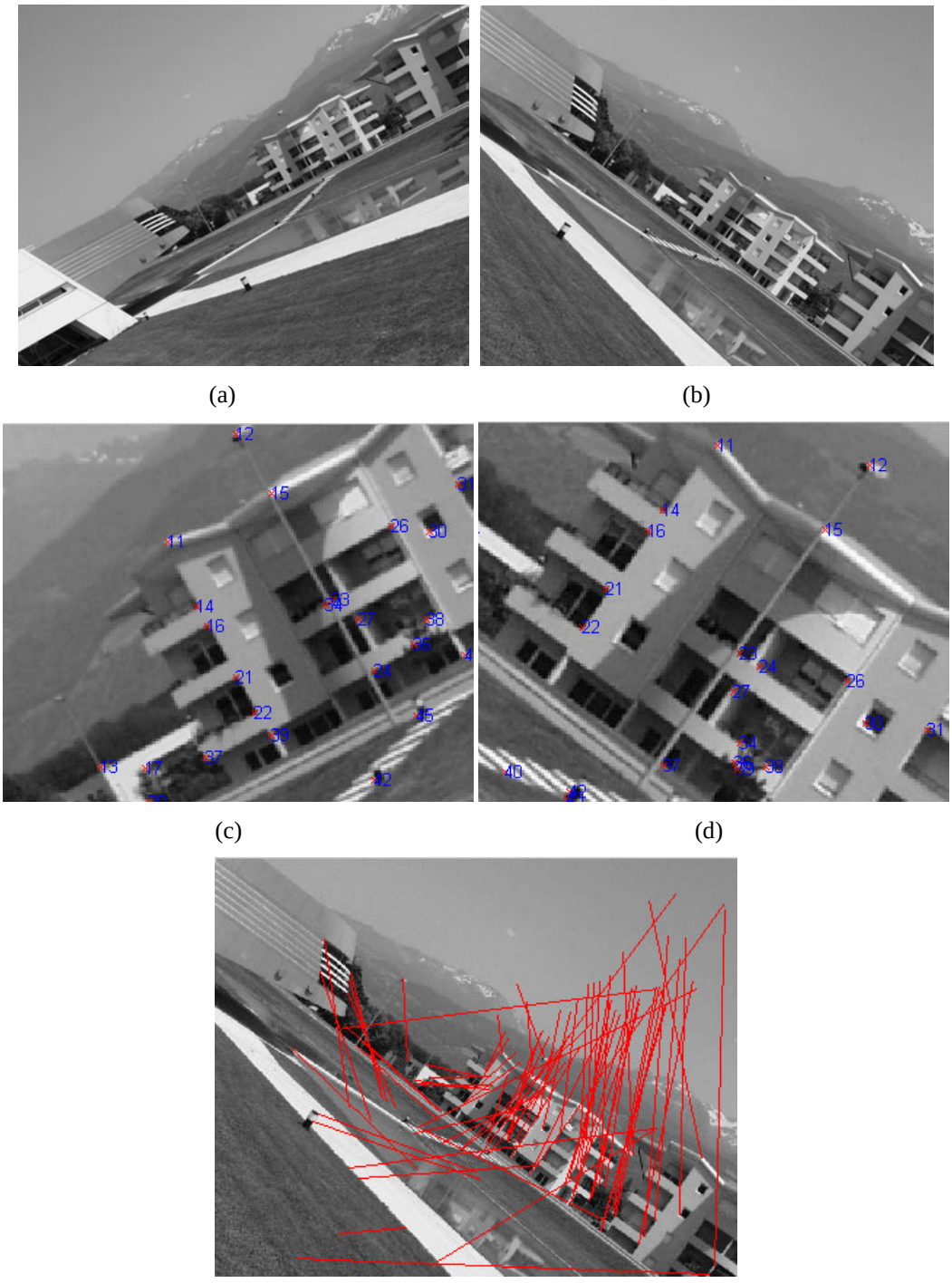



Figure 5.10 Matching using rotational invariants. (a), (b) The images to perform matching on, (c), (d) a detailed view of the matching result, (e) optical flow vectors superimposed on the first image.

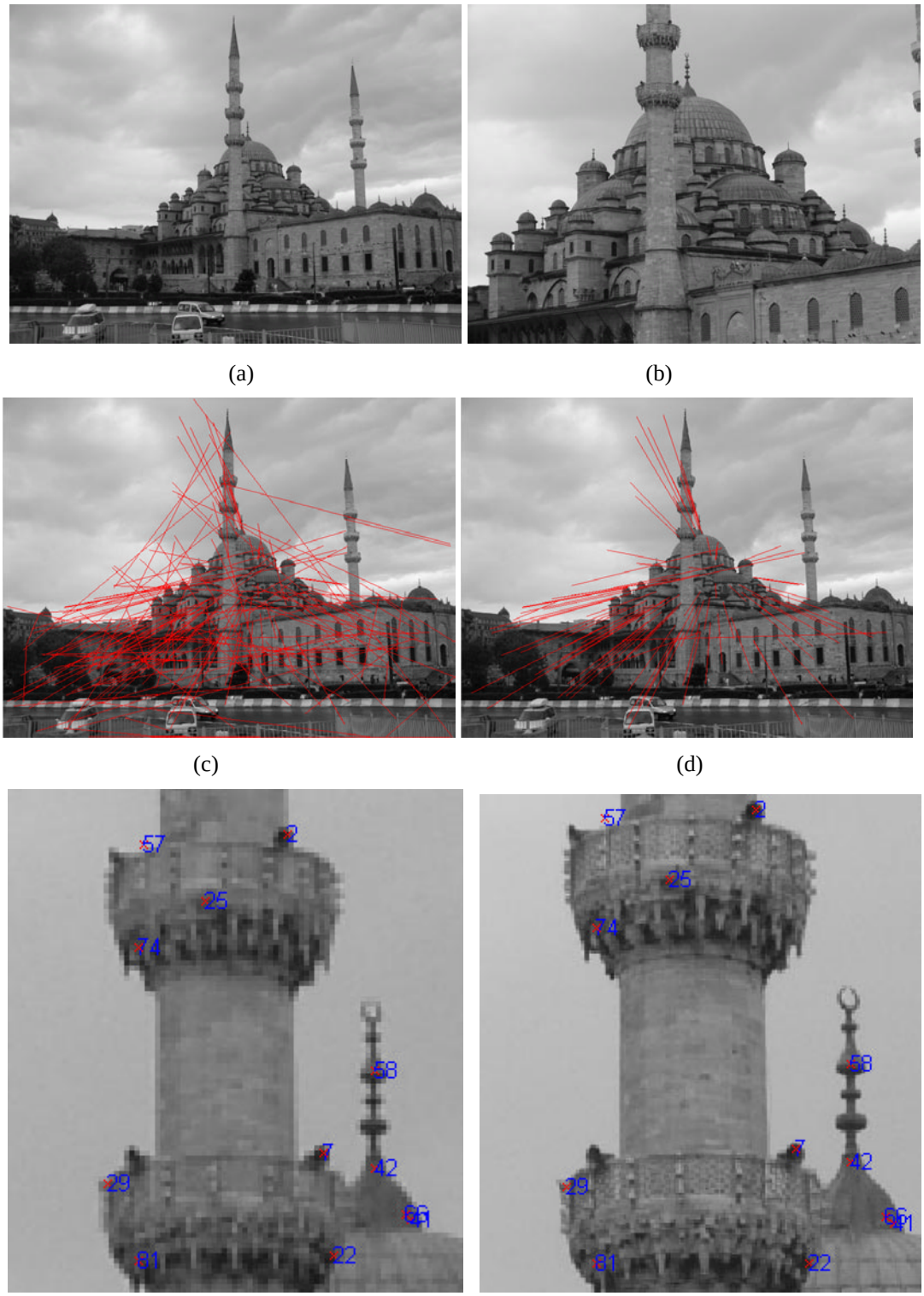
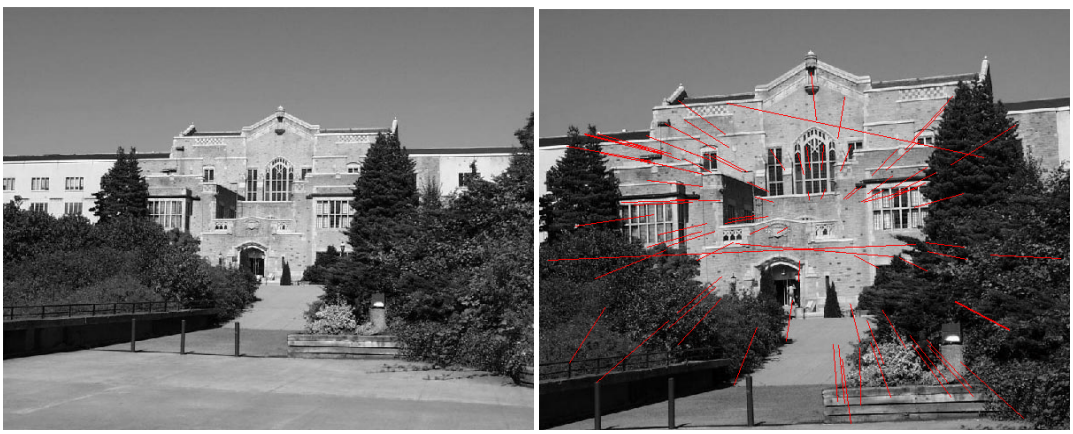



Figure 5.11 Matching using scale invariants (a), (b) Images of 'Yeni Cami' differing in scale only. (c) Optical flow vectors superimposed on the first image before RANSAC. There are 181 matches, many of which are seen to be false. (c) Optical flow vectors after RANSAC. There are 94 matches and a few false matches. (e), (f) Details of a region in the first image(e) and the second image (f).

Following the extraction of these descriptor vectors, the covariance matrix for the vectors is obtained, and the best matches with respect to the Mahalanobis distance are taken to be putative matches.

### 5.2.4 Scale Invariant Matching [Section 4.5]

The interest points are detected by using the scale invariant interest point detector as explained in Section 3.2.2. After this stage, for an interest point detected at a differentiation scale of $s_d$, the region surrounding this point is characterised using rotatioanal invariants based on the Gaussian derivatives (Section 3.2.1.1) of standard deviation equal to $s_d$, $1.5 \times s_d$, and $s_d/1.5$, adding to a total of 36 descriptors per interest point. Following the extraction of the interest points, the covariance matrix for the ensemble of these vectors are estimated and the matching is performed with the best match criterion, where Mahalanobis distance is used as the measure of similarity. Among the matches obtained this way, there exist many false ones. In order to eliminate these false matches, RANSAC is applied afterwards. Sampson distance squared (Section 4.1.3, Eqn. (22)) is used as the error measure and the threshold is set to 18.



(a) (b)

Figure 5.12 More scale invariant results (a) An image of UBC, (b) Another image of UBC slightly rotated and at a different scale, optical flow vectors superimposed. There are 87 matches, only a few matches are false.

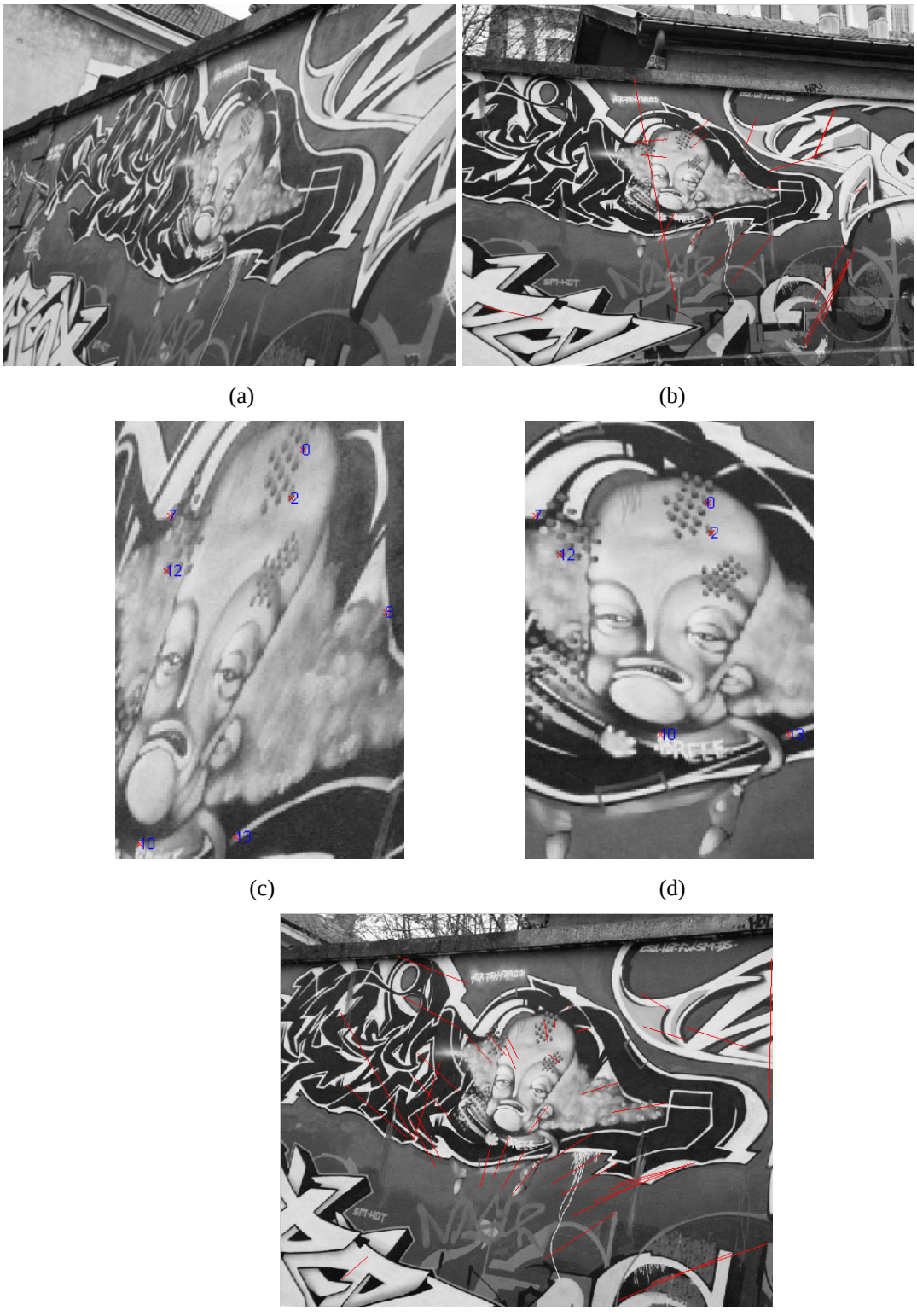



Figure 5.13 Matching using affine invariants. (a) The first image of the pair, (b) optical flow vectors superimposed on the second image. There are 21 matches, a few false. (c), (d) Details of a region in the first image(c) and the second image(d). (e) If the second image to be matched is taken to be the image shown(a slightly ‘easier’ case), 40 matches are found, a few false.

### 5.2.5 Affine Invariant Matching [Section 4.6]

First, the scale invariant interest point detector is utilised. Then, for each interest point, the normalized region surrounding the interest point is

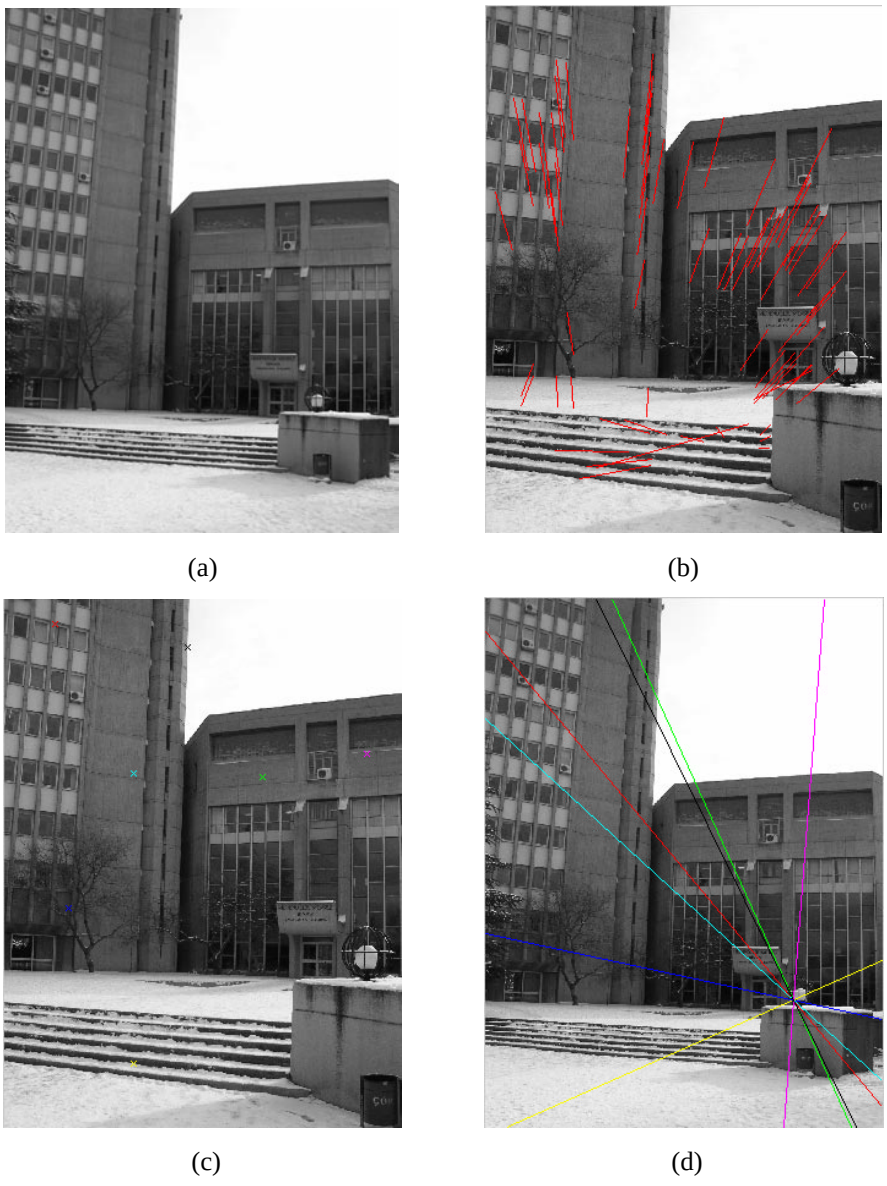


Figure 5.14 Affine invariants on relatively close images. (a) An image of 'MM Binası', (b)Another image of 'MM Binası' with optical flow vectors superimposed. There are 74 matches, 1 false. (c)Arbitrary points marked with different colors, (d) corresponding epipolar lines, drawn using the Fundamental matrix estimated by RANSAC.

obtained using the Algorithm 3.3 . For the normalization of a region, at most 5 iterations are performed, in case the distance of the Harris matrix to the properly scaled identity matrix is greater than 4.5e-5 . After normalizing each region the rotational invariants based on Gaussian derivatives are calculated as in Section 5.2.4 . The rest of the process is as explained in Section 5.2.4.

The scheme also yields satisfactory results, when it is applied in more 'realistic' pairs (Figure 5.14) . This leads to a conclusion that this scheme, despite its complexity in contrast to the small-baseline case, should be the one to be utilised for an arbitrary pair of views where no information about the pair is available beforehand.

## 5.3 A Quantitative Comparison

In the preceding sections, different matching schemes have been presented due to the different necessities of the image pairs to be matched. However, a quantitative comparison taking into account the performance of each matching scheme for the same image pair has not been made. In order to better understand how each scheme behaves under similar conditions, the methods are applied to groups of images, where each group represents a certain situation. 6 groups of 5 image pairs, adding upto 30 pairs are used. The groups are given in Table 5.1.

Table 5.1 The descriptions of different groups used to test the matching schemes.

| Label | Name | Description | Typical Pair (1) | Typical Pair (2) |
|---|---|---|---|---|
| G1 | Video | Extracted from video sequences |  |  |
| G2 | Rotation | Pairs which include only rotations; slight or significant |  |  |
| G3 | Wide Baseline | Pairs from different views |  |  |
| G4 | Extreme | Pairs where a great number of occlusions or discontinuities exist |  |  |
| G5 | Scale & Rotation | Pairs differing in scale, some also rotated |  |  |
| G6 | Affine | A planar surface is imaged from severely different points |  |  |

On each group, the methods, the results for which were mentioned in Section in 5.2, are applied. These are numbered as follows :

Table 5.2 Schemes tested and their labels.

| Label | Method |
|---|---|
| M1 | Matching by correlation |
| M2 | Matching by disambiguating matches |
| M3 | Matching using rotational invariants |
| M4 | Scale invariant matching |
| M5 | Affine invariant matching |

Following the application of each method, RANSAC is applied with the prementioned parameters. The results are shown in Table 5.3, Table 5.4, Table 5.5, Table 5.6 and Table 5.7. The number of inliers indicate a measure of the success of each method. However, if the number of inliers is low (<20) for a particular entry, the reliability of RANSAC in eliminating false matches decreases significantly. Due to this fact, care must be taken in interpreting the results. One should not be tempted to compare two entries possessing low values in this sense. On the other hand, when the number of inliers is significantly high, it turns out that there exists only a negligible amount of false matches among the inliers.

It can be inferred by a first glance on these tables that methods based on cross correlation are suitable and moreover may be preferred where the difference among the two views is not significant(see the results for G4(video)). However, when changes in scale or orientation occur, other methods with increasing degrees of invariance in description should be utilised, which is in agreement with the expectations.

Investigating the results for particular pairs also gives insight on where each method should be utilised or abandoned. Considering a slight rotation as in Figure 5.15, it is noted that cross-correlation of raw brightness values still works as a measure of similarity. However, rotational invariants are much more succesful for matching purposes in such a case. There is no explicit need for affine or scale invariants and the results for the methods utilizing these are poor compared to rotation invariant matching. It is also of interest to note that repeating patterns are handled more successfully by disambiguating matches (except for rotational invariant matching).

When there exists a significant scale change among the pairs as in Figure 5.16, the need for scale invariant schemes arises. Affine invariance is not necessary in this case, and in effect scale invariants perform slightly better compared to affine invariants.

In Figure 5.17, it is once again observed that if there exists no significant scale and view point change, but some rotation, utilising unnecessarily high degree invariants yields relatively poor results.

Figure 5.18 is an example where affine invariants perform clearly much better than the other methods. Use of cross-correlation is simply out of question in this case, and it is observed that using solely rotational invariants does not help. Scale and rotation invariants, on the other hand yield about half the number of correct matches that affine invariants do.

Table 5.3 Results of the experiments for M1, M2 and M3. G : Group number, # : Pair number in group, I : Number of inliers, O : Number of outliers, int. pt. : Number of interest points, I/O, I/int pt.: Corresponding ratios. The result for a particular image in a certain group is given in the row corresponding to the number of the image in the group. The highest in each row(considering all the methods) is shown in bold. Bold numbered image pairs' results are also given in the figures.

| | | | | M1 | | | | | | M2 | | | | | M3 | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| G | # | I | O | I/O | int. pt. | %(I/int. pt.) | I | O | I/O | int. pt. | %(I/int. pt.) | I | O | I/O | int. pt. | %(I/int. pt.) |
| | 1 | 186 | 84 | 2,214 | **1027** | 18,111 | 261 | 89 | 2,933 | **1027** | 25,414 | **306** | **118** | 2,593 | **1027** | **29,796** |
| | 2 | 68 | 4 | 17,000 | **132** | 51,515 | **93** | 4 | 23,250 | **132** | **70,455** | 91 | 3 | **30,333** | **132** | 68,939 |
| G1 | 3 | 332 | 40 | 8,300 | **864** | 38,426 | 438 | **60** | 7,300 | **864** | 50,694 | **596** | 12 | **49,667** | **864** | **68,981** |
| | 4 | 304 | 34 | 8,941 | **847** | 35,891 | 356 | 30 | 11,867 | **847** | 42,031 | **549** | 18 | **30,500** | **847** | **64,817** |
| | 5 | 198 | 24 | 8,250 | **377** | 52,520 | 265 | **25** | 10,600 | **377** | 70,292 | **280** | 2 | **140,000** | **377** | **74,271** |
| | | | | | | | | | | | | | | | | |
| | 1 | 10 | 8 | 1,250 | 153 | 6,536 | 8 | 5 | **1,600** | 153 | 5,229 | 13 | 18 | 0,722 | 153 | 8,497 |
| | 2 | 82 | **188** | 0,436 | **1635** | 5,015 | 196 | 110 | 1,782 | **1635** | 11,988 | **493** | 136 | **3,625** | **1635** | **30,153** |
| G2 | **3** | 104 | **72** | 1,444 | **595** | 17,479 | 202 | 21 | 9,619 | **595** | 33,950 | **311** | 23 | **13,522** | **595** | **52,269** |
| | 4 | 12 | 25 | 0,480 | **713** | 1,683 | 12 | 23 | **0,522** | **713** | 1,683 | 17 | **109** | 0,156 | **713** | 2,384 |
| | 5 | 11 | 15 | 0,733 | 451 | 2,439 | 11 | 19 | 0,579 | 451 | 2,439 | 38 | 43 | 0,884 | 451 | 8,426 |
| | | | | | | | | | | | | | | | | |
| | 1 | 50 | 194 | 0,258 | **1289** | 3,879 | 60 | **500** | 0,120 | **1289** | 4,655 | **81** | 227 | **0,357** | **1289** | **6,284** |
| | 2 | 57 | 42 | 1,357 | 368 | 15,489 | 83 | 56 | 1,482 | 368 | 22,554 | **94** | 28 | **3,357** | 368 | **25,543** |
| G3 | 3 | 102 | 215 | 0,474 | **1288** | 7,919 | **174** | **426** | 0,408 | **1288** | **13,509** | 159 | 199 | **0,799** | **1288** | 12,345 |
| | 4 | 140 | 130 | 1,077 | **1063** | 13,170 | **253** | **189** | 1,339 | **1063** | **23,801** | 163 | 118 | **1,381** | **1063** | 15,334 |
| | 5 | 69 | 85 | **0,812** | **1025** | 6,732 | 87 | 154 | 0,565 | **1025** | 8,488 | **121** | **158** | 0,766 | **1025** | **11,805** |
| | | | | | | | | | | | | | | | | |
| | **1** | 29 | 21 | 1,381 | 292 | 9,932 | 36 | 18 | **2,000** | 292 | 12,329 | **73** | **39** | 1,872 | 292 | **25,000** |
| | 2 | **44** | 106 | **0,415** | **760** | **5,789** | **44** | **199** | 0,221 | **760** | **5,789** | 23 | 149 | 0,154 | **760** | 3,026 |
| G4 | **3** | 22 | 255 | 0,086 | **1957** | 1,124 | **36** | **476** | 0,076 | **1957** | **1,840** | 19 | 324 | 0,059 | **1957** | 0,971 |
| | 4 | 16 | 79 | **0,203** | **687** | 2,329 | **21** | **143** | 0,147 | **687** | **3,057** | 15 | 111 | 0,135 | **687** | 2,183 |
| | 5 | 16 | 109 | 0,147 | **806** | 1,985 | **36** | **245** | 0,147 | **806** | **4,467** | 16 | 92 | 0,174 | **806** | 1,985 |
| | | | | | | | | | | | | | | | | |
| | 1 | 24 | **98** | 0,245 | 438 | 5,479 | 23 | 56 | 0,411 | 438 | 5,251 | 14 | 58 | 0,241 | 438 | 3,196 |
| | 2 | 9 | 7 | 1,286 | 128 | 7,031 | 8 | 3 | **2,667** | 128 | 6,250 | 10 | 9 | 1,111 | 128 | **7,813** |
| G5 | 3 | 21 | 284 | 0,074 | **1698** | 1,237 | **47** | **608** | 0,077 | **1698** | **2,768** | 15 | 245 | 0,061 | **1698** | 0,883 |
| | **4** | 13 | 51 | 0,255 | **382** | 3,403 | **19** | **90** | 0,211 | **382** | 4,974 | 13 | 56 | 0,232 | **382** | 3,403 |
| | 5 | 14 | 63 | 0,222 | 514 | 2,724 | 17 | 112 | 0,152 | 514 | 3,307 | 11 | 61 | 0,180 | 514 | 2,140 |
| | | | | | | | | | | | | | | | | |
| | 1 | 18 | 274 | 0,065 | **2313** | 0,778 | **34** | **568** | 0,060 | **2313** | **1,470** | 19 | 404 | 0,047 | **2313** | 0,821 |
| | 2 | 79 | 190 | 0,415 | 1025 | 7,707 | 133 | **350** | 0,380 | 1025 | 12,976 | **314** | 95 | **3,305** | **1025** | **30,634** |
| G6 | 3 | 15 | 114 | 0,131 | 617 | 2,431 | **24** | **176** | 0,136 | 617 | **3,890** | 13 | 102 | 0,127 | 617 | 2,107 |
| | 4 | 16 | 132 | 0,121 | 1235 | 1,296 | 23 | 199 | 0,116 | 1235 | 1,862 | **33** | **216** | **0,153** | 1235 | **2,672** |
| | **5** | 11 | 27 | 0,407 | 465 | 2,366 | 14 | 47 | 0,298 | 465 | 3,011 | 14 | 46 | 0,304 | 465 | 3,011 |

Table 5.4 Results of the experiments for M4 and M5. G : Group number, # : Pair number in group, I : Number of inliers, O : Number of outliers, int. pt. : Number of interest points, I/O, I/int pt.: Corresponding ratios. The result for a particular image in a certain group is given in the row corresponding to the number of the image in the group. The highest in each row(considering all the methods) is shown in bold. Bold numbered image pairs' results are also given in the figures.

| | | | | M4 | | | | | | M5 | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| G | # | I | O | I/O | int. pt. | %(I/int. pt.) | | I | O | I/O | int. pt. | %(I/int. pt.) |
| | 1 | 46 | 15 | **3,067** | 334 | 13,772 | | 33 | 25 | 1,320 | 334 | 9,880 |
| | 2 | 51 | 6 | 8,500 | 126 | 40,476 | | 35 | **8** | 4,375 | 126 | 27,778 |
| G1 | 3 | 98 | 22 | 4,455 | 368 | 26,630 | | 92 | 31 | 2,968 | 368 | 25,000 |
| | 4 | 93 | **37** | 2,514 | 436 | 21,330 | | 104 | 34 | 3,059 | 436 | 23,853 |
| | 5 | 88 | 7 | 12,571 | 225 | 39,111 | | 79 | 9 | 8,778 | 225 | 35,111 |
| | | | | | | | | | | | | |
| | 1 | 31 | 79 | 0,392 | **722** | 4,294 | | **76** | **93** | 0,817 | **722** | **10,526** |
| | 2 | 231 | 107 | 2,159 | 1293 | 17,865 | | 194 | 118 | 1,644 | 1293 | 15,004 |
| G2 | **3** | 118 | 29 | 4,069 | 418 | 28,230 | | 99 | 40 | 2,475 | 418 | 23,684 |
| | 4 | **29** | 58 | 0,500 | 712 | **4,073** | | 22 | 103 | 0,214 | 712 | 3,090 |
| | 5 | 149 | **100** | 1,490 | 904 | 16,482 | | **165** | 96 | **1,719** | **904** | **18,252** |
| | | | | | | | | | | | | |
| | 1 | 25 | 127 | 0,197 | 1017 | 2,458 | | 23 | 149 | 0,154 | 1017 | 2,262 |
| | 2 | 50 | 43 | 1,163 | **428** | 11,682 | | 41 | **58** | 0,707 | **428** | 9,579 |
| G3 | 3 | 59 | 126 | 0,468 | 819 | 7,204 | | 50 | 111 | 0,450 | 819 | 6,105 |
| | 4 | 84 | 139 | 0,604 | 967 | 8,687 | | 79 | 130 | 0,608 | 967 | 8,170 |
| | 5 | 20 | 64 | 0,313 | 624 | 3,205 | | 14 | 84 | 0,167 | 624 | 2,244 |
| | | | | | | | | | | | | |
| | **1** | 54 | 32 | 1,688 | **353** | 15,297 | | 53 | 38 | 1,395 | **353** | 15,014 |
| | 2 | 16 | 69 | 0,232 | 672 | 2,381 | | 13 | 79 | 0,165 | 672 | 1,935 |
| G4 | **3** | 13 | 124 | 0,105 | 1121 | 1,160 | | 16 | 131 | **0,122** | 1121 | 1,427 |
| | 4 | 11 | 64 | 0,172 | 609 | 1,806 | | 12 | 70 | 0,171 | 609 | 1,970 |
| | 5 | 11 | 62 | 0,177 | 505 | 2,178 | | 12 | 58 | **0,207** | 505 | 2,376 |
| | | | | | | | | | | | | |
| | 1 | 22 | 52 | 0,423 | **460** | 4,783 | | **28** | 36 | **0,778** | **460** | **6,087** |
| | 2 | 13 | 72 | 0,181 | **614** | 2,117 | | **29** | **76** | 0,382 | **614** | 4,723 |
| G5 | 3 | 16 | 105 | **0,152** | 960 | 1,667 | | 14 | 128 | 0,109 | 960 | 1,458 |
| | **4** | 15 | 29 | 0,517 | 251 | **5,976** | | 14 | 26 | **0,538** | 251 | 5,578 |
| | 5 | **43** | 88 | **0,489** | **825** | **5,212** | | 23 | **127** | 0,181 | **825** | 2,788 |
| | | | | | | | | | | | | |
| | 1 | 21 | 160 | **0,131** | 1704 | 1,232 | | 21 | 227 | 0,093 | 1704 | 1,232 |
| | 2 | 116 | 127 | 0,913 | 999 | 11,612 | | 118 | 114 | 1,035 | 999 | 11,812 |
| G6 | 3 | 20 | 99 | **0,202** | **992** | 2,016 | | 14 | 157 | 0,089 | **992** | 1,411 |
| | 4 | 12 | 124 | 0,097 | 1385 | 0,866 | | 13 | 206 | 0,063 | **1385** | 0,939 |
| | **5** | 26 | 56 | 0,464 | **696** | 3,736 | | **48** | **84** | **0,571** | **696** | **6,897** |

Table 5.5 Average and standard deviations of the terms in Table 5.3 and Table 5.4 for M1 and M2 on each group. . G : Group number, I : Number of inliers, O : Number of outliers, int. pt. : Number of interest points, I/O, I/int pt.: Corresponding ratios. The highest in each row (considering all the methods) is shown in bold.

| | | | | M1 | | | | | | M2 | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | I | O | I/O | int. pt. | %(I/int. pt.) | | I | O | I/O | int. pt. | %(I/int. pt.) |
| G1 | Average | 217,600 | 37,200 | 8,941 | **649,400** | 39,293 | | 282,600 | **41,600** | 11,190 | **649,400** | 51,777 |
| | std. dev. | 94,144 | 26,400 | 4,713 | **337,516** | 12,533 | | 115,118 | 29,696 | 6,776 | **337,516** | **17,221** |
| | | | | | | | | | | | | |
| G2 | Average | 43,800 | 61,600 | 0,869 | 709,400 | 6,630 | | 85,800 | 35,600 | 2,820 | 709,400 | 11,058 |
| | std. dev. | 37,222 | **61,195** | 0,373 | **455,739** | 5,202 | | 84,401 | 34,446 | 3,138 | **455,739** | 10,962 |
| | | | | | | | | | | | | |
| G3 | average | 83,600 | 133,200 | 0,796 | **1006,600** | 9,438 | | 131,400 | **265,000** | 0,783 | **1006,600** | 14,601 |
| | std. dev. | **33,374** | 64,867 | 0,397 | **337,717** | 4,268 | | 72,129 | **169,070** | 0,534 | **337,717** | **7,555** |
| | | | | | | | | | | | | |
| G4 | Average | 25,400 | 114,000 | 0,446 | **900,400** | 4,232 | | **34,600** | **216,200** | **0,518** | **900,400** | 5,496 |
| | std. dev. | 10,461 | 77,258 | 0,480 | **558,698** | 3,263 | | 7,473 | 150,508 | **0,742** | **558,698** | 3,665 |
| | | | | | | | | | | | | |
| G5 | Average | 16,200 | 100,600 | 0,416 | **632,000** | 3,975 | | **22,800** | **173,800** | **0,704** | **632,000** | **4,510** |
| | std. dev. | 5,492 | 96,205 | 0,440 | **548,531** | 2,049 | | **13,060** | **220,191** | **0,988** | **548,531** | 1,286 |
| | | | | | | | | | | | | |
| G6 | average | 27,800 | 147,400 | 0,228 | 1131,000 | 2,916 | | 45,600 | **268,000** | 0,198 | 1131,000 | 4,642 |
| | std. dev. | 25,701 | 82,087 | 0,151 | **652,216** | 2,478 | | 44,157 | **178,208** | 0,121 | **652,216** | 4,254 |

Table 5.6 Average and standard deviations of the terms in Table 5.3 and Table 5.4 for M3 and M4 on each group. The highest in each row (considering all the methods) is shown in bold.

| | | | | M3 | | | | | | M4 | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| G | | I | O | I/O | int. pt. | %(I/int. pt.) | | I | O | I/O | int. pt. | %(I/int. pt.) |
| G1 | average | **364,400** | 30,600 | **50,619** | **649,400** | 61,361 | | 75,200 | 17,400 | 6,221 | 297,800 | 28,264 |
| | Std. dev. | **186,009** | **44,098** | **47,144** | **337,516** | 16,065 | | 22,085 | 11,395 | 3,804 | 109,684 | 10,272 |
| | | | | | | | | | | | | |
| G2 | average | **174,400** | 65,800 | **3,782** | 709,400 | 20,346 | | 111,600 | 74,600 | 1,722 | **809,800** | 14,189 |
| | Std. dev. | **177,762** | 43,648 | **4,579** | **455,739** | 16,933 | | 69,540 | 26,042 | 1,226 | 262,405 | 8,328 |
| | | | | | | | | | | | | |
| G3 | average | **123,600** | 146,000 | **1,332** | **1006,600** | 14,262 | | 47,600 | 99,800 | 0,549 | 771,000 | 6,647 |
| | Std. dev. | 33,176 | 69,602 | **1,064** | **337,717** | 6,353 | | 23,380 | 38,654 | 0,337 | 219,314 | 3,441 |
| | | | | | | | | | | | | |
| G4 | average | 29,200 | 143,000 | 0,479 | **900,400** | 6,633 | | 21,000 | 70,200 | 0,475 | 652,000 | 4,565 |
| | Std. dev. | **22,076** | **97,199** | 0,698 | **558,698** | 9,207 | | 16,601 | 29,869 | 0,608 | 258,155 | 5,383 |
| | | | | | | | | | | | | |
| G5 | average | 12,600 | 85,800 | 0,365 | **632,000** | 3,487 | | 21,800 | 69,200 | 0,352 | 622,000 | 3,951 |
| | Std. dev. | 1,855 | 81,876 | 0,378 | **548,531** | 2,341 | | 11,016 | 26,664 | 0,155 | 252,746 | 1,730 |
| | | | | | | | | | | | | |
| G6 | average | **78,600** | 172,600 | **0,787** | 1131,000 | 7,849 | | 39,000 | 113,200 | 0,362 | **1155,200** | 3,892 |
| | Std. dev. | **117,916** | 128,424 | **1,262** | **652,216** | 11,417 | | 38,761 | 34,557 | 0,305 | 350,966 | 3,984 |

Table 5.7 Average and standard deviations of the terms in Table 5.3 and Table 5.4 for M5 on each group. The highest in each row(considering all the methods) is shown in bold.

| | | | | M5 | | |
|---|---|---|---|---|---|---|
| G | | I | O | I/O | int. pt. | %(I/int. pt.) |
| G1 | average | 68,600 | 21,400 | 4,100 | 297,800 | 24,324 |
| | std. dev. | 29,343 | 10,929 | 2,532 | 109,684 | 8,217 |
| G2 | average | 111,200 | **90,000** | 1,374 | **809,800** | 14,111 |
| | std. dev. | 56,413 | 24,145 | 0,714 | 262,405 | 6,376 |
| G3 | average | 41,400 | 106,400 | 0,417 | 771,000 | 5,672 |
| | std. dev. | 22,703 | 32,364 | 0,225 | 219,314 | 3,003 |
| G4 | average | 21,200 | 75,200 | 0,412 | 652,000 | 4,545 |
| | std. dev. | 15,967 | 31,096 | 0,492 | 258,155 | 5,243 |
| G5 | average | 21,600 | 78,600 | 0,398 | 622,000 | 4,127 |
| | std. dev. | 6,530 | 43,292 | 0,243 | 252,746 | 1,744 |
| G6 | average | 42,800 | 157,600 | 0,370 | 1155,200 | 4,458 |
| | std. dev. | 39,686 | 53,809 | 0,383 | 350,966 | 4,292 |

(a) (b)

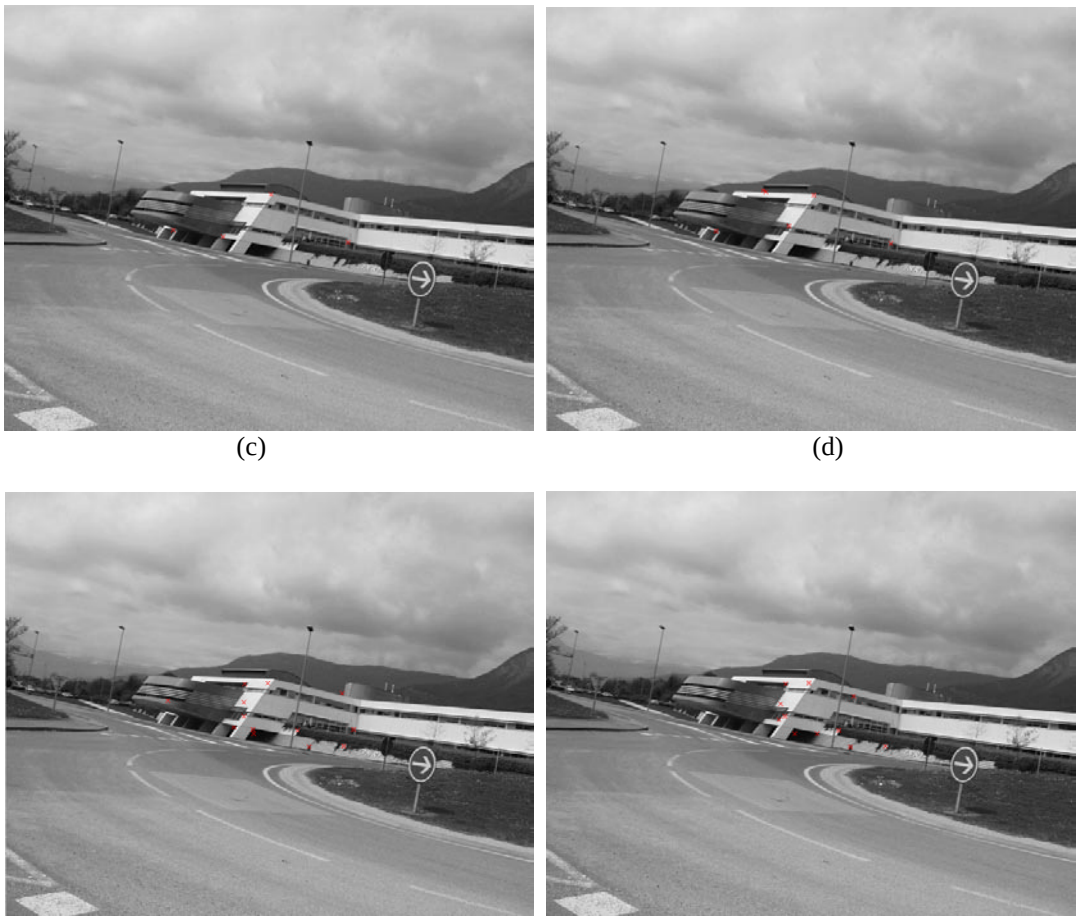
(c) (d)

(e) (f)

(g)

Figure 5.15 Matches determined correctly by each method for the 3rd image of G2(rotation). (a), (b) Image pair to be matched. Correct matches for (c)M1; 86 matches, (d) M2, 186 matches, (e)M3, 292 matches, (f) M4, 107 matches, (g) M5, 91 matches.

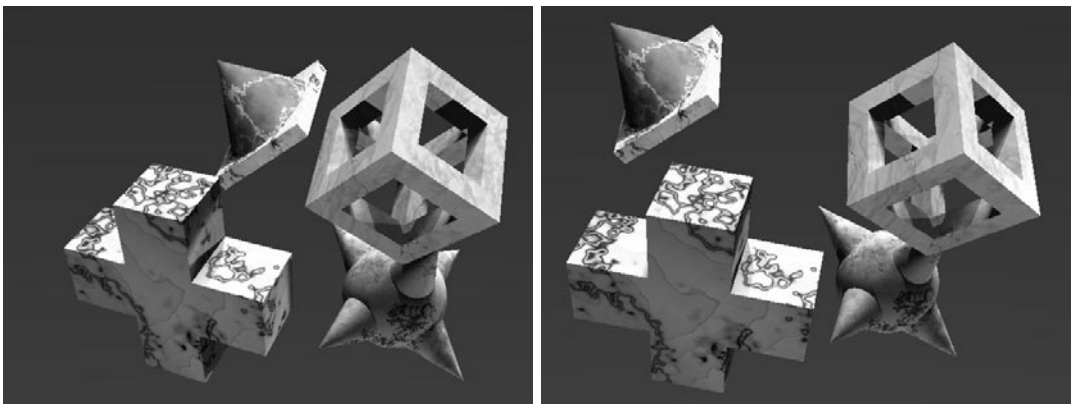
(a) (b)



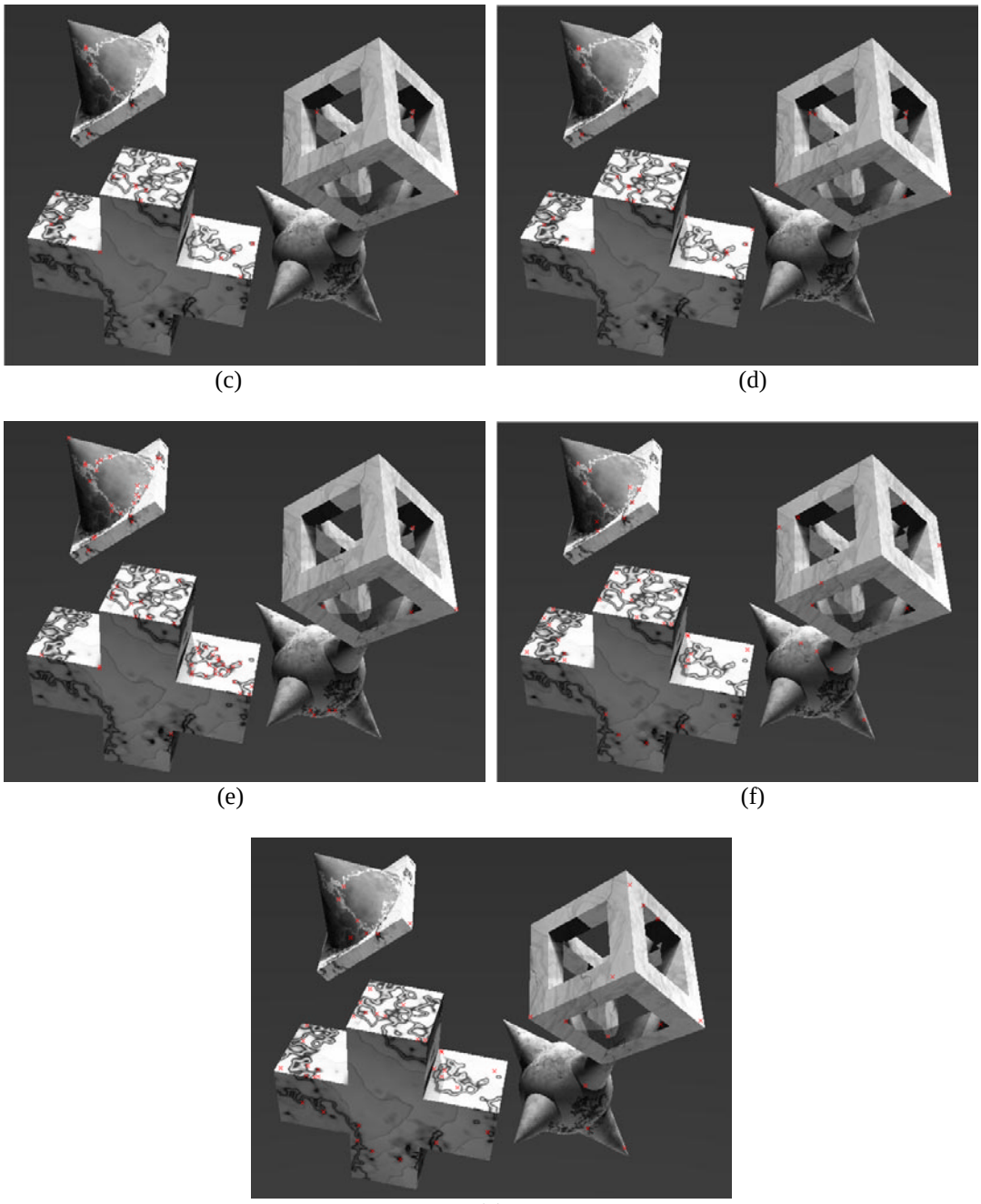
(c) (d)

(e) (f)

Figure 5.16 Matches determined correctly by each method for the 4th image of G5(scale & rotation). (a), (b) Image pair to be matched. Correct matches for (c)M1; 4 matches, (d) M2, 6 matches, (e)M4, 13 matches, (f) M5, 11 matches. M3 yielded no correct matches.



(a) (b)



(c) (d)

(e) (f)

(g)

Figure 5.17 Matches determined correctly by each method for the 1st image of G4(extreme). (a), (b) Image pair to be matched. Correct matches for (c)M1; 28 matches, (d) M2, 35 matches, (e)M3, 70 matches, (f) M4, 52 matches, (g) M5, 51 matches.


(a) (b)


(c)


(d)

Figure 5.18 Matches determined for the 5th image of G6(affine). (a), (b) Image pair to be matched. Correct matches for (c)M4; 23 matches, (d) M5, 44 matches. Other methods fail on this pair.

(a) (b)

Figure 5.19 3rd pair of G4(extreme), where all of the methods fail.

Lastly, in the absence of planar or smooth surfaces to perfom matching on (see Figure 5.19), the methods simply fail due to the failure of description of the regions surrounding the interest points. Even for a small change of viewpoint, the regions around the interest points are altered severely and it is not possible to obtain an invariant description.

# CHAPTER VI

# CONCLUSION

In this thesis, the correspondence problem and possible solutions for different settings are investigated. For this purpose, candidate solutions are divided into stages and each stage is examined seperately. In this respect, a solution may be said to be composed of two main stages, namely the 'interest point detection' and 'interest point matching'. (Note that the latter may further be analysed in substages, as in Figure 1.1)

For the 'interest point detection' step, major but highly distinct approaches are considered. The main problem with most of the detectors in Chapter II is observed to be their heavy dependence on the values of parameters and thresholds. These parameters are used inescapably in order to distinguish a region from its surroundings, possibly showing similar characteristics, but less significant in some respect. In terms of avoiding this problem, it is observed that scale invariant interest point detection is the best scheme among the others presented in the thesis, despite the fact that it still also cannot fully abandon thresholds. Moreover, the algorithm intended to apply this scheme is much more expensive in computational cost (but not in complexity!). This cost mainly stems from the need to convolve the image with high variance Gaussian filters. In fact, the cost could be reduced, if the discrete-analogue of the Gaussian is used [16], or if recursive implementations for filtering with Gaussians and its derivatives are preferred. Another alternative is to down-sample the filtered image at each level, in order to build a pyramid [19], but such a solution introduces the need to compute the actual position in the original-sized image for a feature detected in some later level of down-sampling.

The ‘interest point matching’ stage also has several candidates for employment. The performance of these methods basically depend on the configuration of the cameras. As the cameras become seperated, the need for more advanced schemes, utilizing higher orders of invariance to the changes in the images, arises. The most advanced scheme in this regard can handle local affine transformations (including scale, rotation, skewing, stretching, etc. of the feature). The experiments show that this scheme also succeeds on images even where a simple correlation measure would also suffice. However, if a certain degree of invariance is sufficient to handle the image pair at hand, higher degrees of invariants usually perform worse. Thus, if there is some knowledge about the image pair prior to the matching step, this knowledge can be put to use for better results. An adaptive algorithm may also be devised, starting form the lowest level of invariants (simple cross correlation), and determining on the way the required degree of invariance. In such a case however, the selection of the cost function is not clear.

Even though the final schemes for matching are noted to be ‘tiresome’ for the hardware which is excecuting the related algorithms, they are far from being brute-force methods (even though, given the constraints, how something is done would not mean much in a customer-based-engineering point of view; the question is : is there another point of view?). Moreover, the algorithm can also be realised ‘parallel-wise’, if some dedicated hardware is used, since the most time consuming stage, namely the scale invariant interest point detection stage consists mainly of filtering the image with Gaussians of different variance, thus practically size. Given the interest points, the second significantly costly process is obtaining the invariant description of the patch containing the interest points, and it is noted that for each point this process is an independent process, which would allow a parallel implementation.

The methods explained in this thesis may be combined with other related methods [24], [37] as in [29], to strengthen the matching stage. The next step, following this, would probably be to take into account the

problem of calibrating cameras as a whole starting from the interest point matching stage.

# APPENDIX-A

# MASKS FOR POLYNOMIAL APPROXIMATION

Given an image patch $f(x,y)$, practically of size 5x5 or 7x7, the goal is to approximate the patch by a bi-cubic polynomial, i.e.,

$$f(x,y) \approx k_1 + k_2 x + k_3 y + k_4 x^2 + k_5 xy + k_6 y^2 + k_7 x^3 + k_8 x^2 y + k_9 xy^2 + k_{10} y^3 \quad \text{(A.1)}$$

For this purpose, let $f_s = \begin{bmatrix} f(s,n) & f(s,n-1) & \dots & f(s,1) \end{bmatrix}$ for an $n$x$n$ patch, i.e., $f_s$ is a row matrix that contains the values that the function assumes on the $s^{th}$ row (naming the rows starting from $n$ and ending at 1). Then form $F = \begin{bmatrix} f_n & f_{n-1} & \dots & f_1 \end{bmatrix}^T$. In a similar fashion, form the column matrices $P_m$ containing the values that each monomial on the r.h.s of (A.1) assumes, to be stuffed into the matrix $A = \begin{bmatrix} P_1 & P_2 & \dots & P_{10} \end{bmatrix}$. Letting $K = \begin{bmatrix} k_1 & k_2 & \dots & k_{10} \end{bmatrix}$, the r.h.s of (A.1) is represented as $Ak$, and the desired relation among this collection of matrices is,

$$AK = F \quad \text{(A.2)}$$

If $F$ is not in the column space of $A$, which may certainly be (and usually is) the case for interesting patches, the equality in (A.2) does not hold. What should be sought in such a setting is the least squares solution of (A.2). This can be obtained by taking the pseudo-inverse of $A$, that is $(A^TA)^{-1}A^T$ and multiplying both sides of (A.2) by this matrix.

$$K = \left(A^T A\right)^{-1} A^T F \quad \text{(A.3)}$$

It should be noted that the pseudo-inverse matrix needs to be calculated only once and the same matrix is used for any patch of the same size.

Alternatively, one can form masks of the size of the patch to calculate the coefficients. For each coefficient in (A.1), a mask can be created so that the inner product of the mask with the image patch is equal to the coefficient ([10]). This is achieved by first forming an orthogonal polynomial basis and then exploiting the linearity of the process.

## Discrete Orthogonal Polynomials:

For a symmetric interval of given length, say $2n+1$ (ranging from $-n$ to $n$), a monomial of the form $x^k$ is represented by a vector composed of the values that it assumes in the interval, i.e. $M_{x^k} = \left[(-n)^k \quad (-n+1)^k \quad \ldots \quad 0 \quad \ldots \quad n^k\right]$. Thus, a set of vectors is obtained where each vector represents a different monomial. Applying the well-known Gram-Schmidt orthogonalization method, the set may be rendered orthonormal. Then an orthonormal set $Pj\big|_{j=1}^{t}$ is obtained, such that,

$$P_j = \frac{\hat{P}_j}{\sqrt{\left\langle \hat{P}_j, \hat{P}_j \right\rangle}} \tag{A.4}$$

where,

$\hat{P}_0 = M_{x^0}$ , and

$$\hat{P}_j = M_{x^j} - \sum_{i=0}^{j-1} \left\langle M_{x^i}, P_i \right\rangle P_i \tag{A.5}$$

After the process, each element $P_j$ of the orthonormal set, represents a polynomial of the form, $x^j + a_{j-1}\, x^{j-1} + a_{j-2}\, x^{j-2} + \ldots + a_1\, x + a_0$ .

Using these vectors, a 2-D set can be formed, with elements $D_{k,l} = P_k^T \cdot P_l$ . The inner product of two elements in this set can be obtained as,

$$\left\langle D_{k,l}, D_{c,v} \right\rangle = \left(P_k^T \cdot P_c\right) \cdot \left(P_l^T \cdot P_v\right) \tag{A.6}$$

It is noted that if $k \neq c$ or $l \neq v$, (A.6) equals 0 and if $k = c$ and $l = v$, it equals 1. In other words, the 2-D set obtained is orthonormal. This new set represents 2-D polynomials of the form $\sum_{i=0}^{n}\sum_{j=0}^{n} a_{ij} x^i y^j$ .

A given image patch can be approximated using this orthonormal set as, $f(x,y) \approx \sum_{i=0}^{n}\sum_{j=0}^{n} c_{i,j} D_{i,j}$ , where each coefficient $c_{i,j}$ is obtained by the result of the inner product of $f(x,y)$ and $D_{i,j}$. Once the $c_{i,j}$'s are obtained, the coefficients of the monomials in (A.1) may be calculated as linear combinations of these.

However, this procedure can be reversed and instead of first finding $c_{i,j}$'s and then getting the linear combinations, the linear combinations of the kernels for ($D_{i,j}$) may first be constructed and $k_i$'s may be calculated directly using these kernels as promised. For instance, suppose that a monomial, $x^n y^m$, is contained in each $D_{ij}$ with a weight of $w_{ij,nm}$. Then the kernel for estimating the coefficient corresponding to the monomial is obtained by adding the kernels for $D_{ij}$, weighted by $w_{ij,nm}$.

# APPENDIX-B

# SINGULAR VALUE DECOMPOSITION

SVD states that any $m \times n$ matrix $A$ can be factored as ([34]):

$$A = \underbrace{Q_1}_{m \times m} \underbrace{\Sigma}_{m \times n} \underbrace{Q_2^T}_{n \times n} \tag{B.1}$$

The columns of the orthonormal matrix $Q_1$ are the eigenvectors of $AA^T$ and the columns of the orthonormal matrix $Q_2$ are the eigenvectors of $A^TA$. The $r$ ($\leq \min(m,n)$) singular values on the diagonal matrix $\Sigma$ are the square roots of the non-zero eigenvalues of $AA^T$ and $A^TA$.

The proof that such a decomposition exists is given actually by constructing it. Consider $A^TA$ ($n \times n$) which has a complete set of orthonormal eigenvectors $x_j$;

$$A^T A x_j = \lambda_j x_j \text{ for j = 1,.., n and } x_i^T x_j = \begin{cases} 0 & \text{if } i \neq j \\ 1 & \text{if } i = j \end{cases}. \tag{B.2}$$

$Q_2$ is constructed by placing these $n$ eigenvectors in its columns.

Taking the inner product with $x_j$,

$$x_j^T A^T A x_j = \lambda_j x_j^T x_j = \lambda_j \text{ or } x_j^T A^T A x_j = \left\| A x_j \right\|^2 = \lambda_j . \tag{B.3}$$

Thus, $\lambda_j$ cannot be negative.

Suppose that $r$ ($\leq n$) of the eigenvalues are positive. Then for these positive eigenvalues, setting $q_j = \dfrac{Ax_j}{\sqrt{\lambda_j}}$, it's seen that $q_j$'s have unit norm and are orthogonal, since

$$q_i^T q_j = \frac{x_i^T A^T A x_j}{\sqrt{\lambda_i \lambda_j}} = \frac{\lambda_j x_i^T x_j}{\sqrt{\lambda_i \lambda_j}} = \begin{cases} 0 & if\ i \neq j \\ 1 & if\ i = j \end{cases}. \tag{B.4}$$

Since for each non-zero eigenvalue, a corresponding $q$ vector can be found, it is satisfied that $r \leq m$ also. Moreover, that $q_j$'s are eigenvectors of $AA^T$ is seen by:

$$AA^T q_j = \frac{AA^T A x_j}{\sqrt{\lambda_j}} = \frac{\lambda_j A x_j}{\sqrt{\lambda_j}} = \lambda_j q_j \tag{B.5}$$

Looking back, the $r$ orthonormal $q_j$'s can be extended to an orthonormal basis by Gram-Schmidt orthogonalization procedure. These form the columns of $Q_1$. Now considering,

$$q_i^T A x_j = 0 \quad if\ j > r \text{ (since } Ax_j=0), \tag{B.6}$$

and otherwise if $j \leq r$,

$$q_i^T A x_j = q_i^T \sqrt{\lambda_j} q_j = \begin{cases} 0 & if\ i \neq j \\ \sqrt{\lambda_j} & if\ i = j \end{cases}. \tag{B.7}$$

Thus, the only non-zero elements of the product $Q_1^T A Q_2 = \Sigma$ are the first $r$ diagonal entries and these are equal to the square roots of the eigenvalues of $AA^T$. Since $Q_1$ and $Q_2$ are orthonormal, it can be concluded that

$$A = Q_1 \Sigma Q_2^T, \tag{B.8}$$

as required.

An application of SVD that has found its use in Chapter IV is the following:

"Minimize $\|Ay\|^2$ where $y$ is a unit vector of size $n$ and $A$ is an $m \times n$ matrix."

Since the columns of $Q_2$ form an orthonormal basis, $y$ can be written as:

$$y = \sum_{j=1}^{n} \alpha_j x_j, \tag{B.9}$$

where

$$\sum_{j=1}^{n} \alpha_j^2 = 1 \tag{B.10}$$

Then,

$$\begin{aligned}\|Ay\|^2 &= y^T A^T Ay = y^T \left[ A^T A \left( \sum_{j=1}^{n} \alpha_j x_j \right) \right] \\ &= y^T \sum_{j=1}^{n} \alpha_j \lambda_j x_j = \left( \sum_{k=1}^{n} \alpha_k x_k \right) \left( \sum_{j=1}^{n} \alpha_j \lambda_j x_j \right) \\ &= \sum_{l=1}^{n} \alpha_l^2 \lambda_l \geq \sum_{l=1}^{n} \alpha_l^2 \min_j (\lambda_j) = \min_j (\lambda_j) = \|Ax_{\min}\|\end{aligned} \tag{B.11}$$

where $x_{min}$ is the eigenvector corresponding to the least eigenvalue.

In words, (B.11) tells that choosing $y$ as the eigenvector of $A^TA$ corresponding to the least eigenvalue minimizes $\|Ay\|^2$. Note that for the least eigenvalue, if there exists more than one eigenvector, it suffices to choose $y$ from their span, with the unit length constraint in mind.

# APPENDIX-C

## FOURIER TRANSFORM OF THE GAUSSIAN FUNCTION

Consider the normalized Gaussian function with variance $\sigma^2$,

$$g\left(x;\sigma^2\right)=\frac{1}{\sqrt{2\pi\sigma^2}}e^{-\frac{x^2}{2\sigma^2}} \tag{C.1}$$

The Fourier Transform of this function ([39]) is found by evaluating :

$$G\left(\omega;\sigma^2\right)=F\left\{g\left(x;\sigma^2\right)\right\}=\int_{-\infty}^{\infty}\frac{1}{\sqrt{2\pi\sigma^2}}e^{-\frac{x^2}{2\sigma^2}}e^{-j\omega x}dx \tag{C.2}$$

Differentiating (C.2) with respect to ω,

$$\begin{aligned}\frac{dG}{d\omega}&=\int_{-\infty}^{\infty}\frac{1}{\sqrt{2\pi\sigma^2}}e^{-\frac{x^2}{2\sigma^2}}\left(-jx\right)e^{-j\omega x}dx=j\sigma^2F\left\{-\frac{x}{\sigma^2\sqrt{2\pi\sigma^2}}e^{-\frac{x^2}{2\sigma^2}}\right\}\\&=j\sigma^2F\left\{\frac{d}{dx}g\left(x;\sigma^2\right)\right\}\end{aligned} \tag{C.3}$$

For any function h(x) with F{h(x)}=H(ω), it can be written,

$$h(x)=\frac{1}{2\pi}\int_{-\infty}^{\infty}H(\omega)e^{j\omega x}d\omega \tag{C.4}$$

Differentiating both sides of (C.4) with respect to x,

$$\frac{dh}{dx}=\frac{1}{2\pi}\int_{-\infty}^{\infty}H(\omega)(j\omega)e^{j\omega x}d\omega \tag{C.5}$$

i.e.,

$$F\left\{\frac{dh}{dx}\right\}=j\omega H(\omega) \tag{C.6}$$

Using this result in (C.3), it's concluded that

$$\frac{dG}{d\omega} = j\sigma^2 \, j\omega G = -\omega\sigma^2 G \tag{C.7}$$

Rearranging (C.7),

$$\frac{\left(\frac{dG}{d\omega}\right)}{G} = -\omega\sigma^2 . \tag{C.8}$$

Integrating both sides,

$$\int_0^a \left( \frac{\left(\frac{dG}{d\omega}\right)}{G} \right) d\omega = \int_0^a -\omega\sigma^2 d\omega \tag{C.9}$$

$$\ln\left(G\left(a;\sigma^2\right)\right) - \ln\left(G\left(0;\sigma^2\right)\right) = -\frac{\omega^2\sigma^2}{2} \tag{C.10}$$

Since $\ln\left(G\left(0;\sigma^2\right)\right) = \ln\left( \int_{-\infty}^{\infty} \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{x^2}{2\sigma^2}} dx \right) = \ln(1) = 0$, as a result of the fact that $g\left(x;\sigma^2\right)$ is normalized,

$$G\left(\omega;\sigma^2\right) = e^{-\frac{\omega^2\sigma^2}{2}} . \tag{C.11}$$

Stating in words, the Fourier Transform of a Gaussian with variance $\sigma^2$ is a Gaussian with variance $1/\sigma^2$.

# APPENDIX-D

# CHOLESKY DECOMPOSITION

The Cholesky decomposition algorithm develops a method to write a positive definite(PD) matrix $P$ as $P = LL^T$, where L is a square matrix and is said to be the square root of P. The algorithm is applicable to $n$x$n$ PD matrices[5], but attention will be restricted to the much simpler case of 2x2 symmetric PD matrices, which are just enough for the purposes mentioned in the thesis. For this, consider the quadratic form:

$$\underbrace{\begin{bmatrix} x_1 & x_2 \end{bmatrix}}_{X^T} \underbrace{\begin{bmatrix} a & b \\ b & c \end{bmatrix}}_{P} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = ax_1^2 + 2bx_1x_2 + cx_2^2 . \tag{D.1}$$

Since $P$ is PD, (D.1) is greater than zero for any choice of $x_1$, $x_2$ by the definition of positive definiteness. Moerover, $a > 0$, $c > 0$, otherwise choosing respectively for the two cases, $x_1 = -1$, $x_2 = 0$ and $x_1 = 0$, $x_2 = -1$, (D.1) could be made non-positive. Also, $ac - b^2 > 0$ (or simply det $P > 0$ ), or else for $x_1 = b$, $x_2 = -a$, (D.1) would again be non-positive. Rearranging (D.1), it is written as:

$$\begin{aligned} X^T PX &= \left( \sqrt{a}x_1 + \frac{b}{\sqrt{a}} x_2 \right)^2 - \left( \frac{b}{\sqrt{a}} x_2 \right)^2 + cx_2^2 \\ &= \left( \sqrt{a}x_1 + \frac{b}{\sqrt{a}} x_2 \right)^2 + \left( \sqrt{c - \frac{b^2}{a}} x_2 \right)^2 \\ &= \begin{bmatrix} y_1 & y_2 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \end{bmatrix} \end{aligned} \tag{D.2}$$

where

$$\begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = \underbrace{\begin{bmatrix} \sqrt{a} & \dfrac{b}{\sqrt{a}} \\ 0 & \sqrt{c - \dfrac{b^2}{a}} \end{bmatrix}}_{L^T} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}. \tag{D.3}$$

Note that all the square root terms are real, a fact ensured by the preceding discussions. Following this change of variables, it is observed that $P = LL^T$ as desired.

The last point to note is that both the Hessian and Harris matrices are PD and symmetric at the interest points. Actually, this was one of the criteria of being an interest point, where the other is that of being a local maximum (according to a measure considering the similarity of the eigenvalues). If one determines the interest point using any other detector, he/she should ensure the positive definiteness of these matrices before attempting to calculate the square root matrix.

Also, as mentioned in section 3.3.3, points where the Hessian is negative definite ($P$ is said to be negative definite if (D.1) is always negative) may also be considered as interest points. In this case, the square root of the negative of the Hessian matrix may be utilized.

# REFERENCES


[1] Freeman W. T., Adelson E. H.: The design and use of steerable filters. IEEE Transactions on Pattern Analysis and Machine Intelligence, 13[9] : pp. 891-906, 1991.

[2] Asada H., Brady M. : The curvature primal sketch. IEEE Transactions on Pattern Analysis and Machine Intelligence, 8[1] : pp.2-14, 1986.

[3] Babaud J., Witkin A. P., Baudin M., Duda R. O. : Uniqueness of the Gaussian kernel for scale-space filtering. IEEE Transactions on Pattern Analysis and Machine Intelligence, 8[1] : pp. 26-33, 1986.

[4] Baumberg A. : Reliable feature matching across widely-separated views. In proceedings of the conference on Computer Vision and Pattern Recognition, Hilton Head Island, South Carolina, USA, pp. 774-781, 2000.

[5] Bierman G. J. : Factorization Methods for Discrete Sequential Estimation. Academic Press, Inc., 1977.

[6] Burt P. J., Adelson E. H. : The Laplacian pyramid as a compact image code. IEEE Transactions on Communications, 9[4] : pp. 532-540, 1983.

[7] Carmo, M. P. D. : Differential Geometry of Curves and Surfaces. Englewood Cliffs, NJ : Prentice Hall, 1976.

[8] Deriche R., Giraudon G. : A computational approach for corner and vertex detection. Internatioanal Journal of Compter Vision, 10[2] : pp. 101-124, 1993.

[9] Dufournaud Y., Schmid C., Horaud R. : Image matching with scale adjustment. Research Report, no. 4428, 2002

[10] Haralick R. M., Shapiro L. G. : Computer and Robot Vision. Reading, Mass. : Addison-Wesley Pub. Co., 1992.

[11] Hartley R. : In defence of the eight-point algorithm. IEEE Transactions on Pattern Analysis and Machine Intelligence, 19[6] : pp. 580-593, 1997.

[12] Hartley R., Zisssermann A.: Multiple View Geometry in Computer Vision. Cambridge University Press, 2001.

[13] Horaud R:, Skordas T. : Stereo Correspondence Thorugh Feature Grouping and Maximal Cliques. IEEE Transactions on Pattern Analysis and Machine Intelligence, 11[12] : pp. 1168-1180, 1989.

[14] Jain R., Kasturi R., Schunk B. G. : Machine Vision. New York : McGraw Hill, 1995.

[15] Jones D. G., Malik J. : A computational framework for determining stereo correspondence from a set of linear spatial filters. In ECCV'92, pp. 395-410, 1992.

[16] Lindeberg T. : Scale-Space Theory in Computer Vision. Kluwer Publishers, 1994.

[17] Lindeberg T. : Feature detection with automatic scale selection. International Journal of Computer Vision, 30[2] : pp. 79-116, 1998.

[18] Lindeberg T., Gårding J. : Shape-Adapted Smoothing in estimation of 3-D shape cues from affine deformations of local 2-D brightness structure. Image and Vision Computing, 15[6] : pp. 415-434, 1997.

[19] Lowe D. G. : Object recognition from local scale invariant features. In International Conference on Computer Vision, pp. 1150-1157, 1999.

[20] Kanatani K., Kanazawa Y. : Automatic thresholding for correspondence detection. Internatioanl Journal of Image and Graphics, 4[1] : pp.21-33, 2004

[21] Kanazawa Y., Kanatani K. : Robust image matching under a large disparity. In proceedings of Workshop on Science of Computer Vision, Okayama, Japan, pp. 46-52, September 2002.

[22] Manmatha R. : Measuring the Affine Transform Using Gaussian Filters. In proceedings of the European Conference on Computer Vision,1994.

[23] Manmatha R., Oliensis J. : Measuring the affine transform -i: Scale and rotation. Technical report CMPSCI TR 92-74, University of Massachusetts at Amherst, MA, 1992.

[24] Matas J., Chum O., Urban M., Pajdla T. : Robust Wide Baseline Stereo from Maximally Stable Extremal Regions. In The British Machine Vision Conference, 2002.

[25] Mikolajczyk K., Schmid C. : Indexing based on scale invariant interest points. In proceedings of the 8th International Conference on Computer Vision, Vancouver, Canada, pp. 525-531, 2001.

[26] Mikolajczyk K., Schmid C. : An affine invariant interest point detector. In proceedings of the 7th European Conference on Computer Vision, Copenhagen, Denmark, pp. 0-7, May 2002.

[27] Mohanna F., Mokhtarian F. : Performance Evaluation of Corner Detection Algorithms under Similarity and Affine Transforms. In British Machine Vision Conference, pp. 353-362, 2001.

[28] Pritchett P., Zissermann A. : Wide-baseline stereo matching. In proceedings of the 6th International Conference on Computer Vision, Bombay, India, pages 754-760, 1998.

[29] Schaffalitzky F., Zisserman A. : Multi-view matching for unordered image sets, or "How do I organize my holiday snaps?". In proceedings of the European Conference on Computer Vision, pp. 414-431, 2002.

[30] Schalkoff R. : Digital image processing and computer vision. In Wiley, 1989.

[31] Schmid C., Mohr R. : Matching by local invariants. Research report, no. 2644, INRIA, 1995.

[32] Schmid C., Mohr R., Bauckhage C. : Evaluation of interest point detectors. International Journal of Computer Vision, 37[2] : pp. 151-172, 2000.

[33] Smith S. M., Brady J. M. : SUSAN - a new approach for low level image processing. International Journal of Computer Vision, 23[1] : pp. 45-78, May 1997.

[34] Strang G. : Linear Algebra and its Applications. International Thomson Publishing, 1988.

[35] Therrien C. W. : Discrete Random Signals and Stochastic Signal Processing, In Prentice Hall, 1992.

[36] Trajkovic M., Hedley M. : Fast corner detection. Image and Vision Computing, 16, pp. 75-87, 1998.

[37] Tuytelaars T., Van Gool L. : Wide baseline stereo based on local, affinely invariant regions. In The Eleventh British Machine Vision Conference, pp. 412-425, 2000.

[38] Ullman S. : The Interpretation of Visual Motion. Cambridge, Mass.: MIT Press, 1979.

[39] Van Vliet L. J., Rieger B., Verbeek P. W. : Fourier Transform of a Gaussian, June2004
URL:http://www.ph.tn.tudelft.nl/~lucas/education/tn254/2002/Fourier%20transform%20of%20a%20Gaussian.pdf

[40] Witkin A. P. : Scale-Space Filtering. In proceedings of the 7th Internatioanal Joint Conference on Artificial Intelligence, Kalsrühe, West Germany, pp. 1019-1021, 1983.

[41] Zhang Z., Deriche R., Faugeras O., Luang Q.T. : A robust technique for matching two uncalibrated images through the recovery of the epipolar geometry. Research report, no. 2273, INRIA, 1994.